



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



Instituto de Ingeniería del  
Agua y Medio Ambiente

# Influencia del Régimen Hídrico en la Zonificación de la Vegetación de Ribera

Autor: Joaquín Real Llanderal

Director: Prof. Félix Francés García



# RESUMEN:

Las riberas son una parte esencial de los ecosistemas fluviales. Representan una zona de transición entre el medio acuático, de caudales circulantes, y el medio terrestre adyacente al río, recibiendo la influencia hidrológica de ambos, al constituir un espacio compartido en el ciclo del agua, de los sedimentos y de los nutrientes.

La situación actual de una buena parte de las riberas en España es mala (Ibero, 1996). Son numerosos los factores que han provocado esta situación (agricultura, transportes, urbanización, contaminación, canalizaciones...) dando lugar a que las riberas pierdan o experimenten un deterioro tanto en su estructura como en su funcionalidad. La limitación del caudal circulante así como la modificación de la frecuencia e intensidad de las avenidas se han constituido como algunos de los principales factores que impiden la regeneración natural de los bosques riparios.

Es por ello que los planes de restauración de riberas deben pasar necesariamente por dotar al río de libertad y de suministrarle los caudales adecuados para favorecer su correcto funcionamiento, la conexión con su llanura de inundación y la continuidad de los procesos ecológicos que en ella se desarrollan.

El siguiente trabajo de investigación tiene como objetivo principal el desarrollo de un modelo matemático que permita simular la vegetación riparia para un determinado tramo de río. Para conseguir este objetivo se ha adaptado el modelo de humedad del suelo "Ribav" a las condiciones de la España mediterránea. Posteriormente se ha desarrollado el software de Ribav-1D con el lenguaje de programación Visual Basic.net, con el fin de poder utilizar el modelo en puntos de simulación individuales. Con este software se ha realizado un análisis de sensibilidad manual del modelo "Ribav". A continuación se ha calibrado y validado el modelo "Ribav" para un tramos de estudio natural y 1 tramo alterado de la demarcación hidrográfica del Júcar.

Para poder utilizar mapas grid con este modelo se ha creado también el software Ribav 2D que permite utilizar el modelo bidimensionalmente con capas de formato ASCII-Raster. Como Ribav es un modelo que no contempla algunos procesos que ocurren en las riberas (efecto del esfuerzo cortante, sucesión vegetal, reclutamiento de especies...) se ha optado por unir Ribav 2D con el software del modelo de vegetación dinámica "Ripflow v.2", utilizando la tecnología COM (*Component Object Model*). Para comparar los resultados de las simulaciones con los datos obtenidos en campo del modelo unido "Ribav2D-Ripflow" se ha diseñado una aplicación visual con librerías de MapWindow GIS que permite determinar el grado de aciertos en la simulación del tipo de vegetación mediante el coeficiente Kappa de Cohen.

# ABSTRACT:

Riverbanks are an essential part of the riparian ecosystems. They represent a transition zone between the aquatic medium of the river flows and the terrestrial medium adjacent to the river, receiving the influence of both. They compose a shared space for the water cycle, the sediments and the nutrients.

The current situation of most of the riverbeds in Spain is bad. There are a lot of factors which have caused this situation (agriculture, transport, urbanization, pollution, canalizations...) making the river shores deteriorate both in their structure and in their functionality. The limitation in river flows and the modification of the frequencies and intensities of the floods are some of the main causes which prevent the correct regeneration of the riparian forests.

Therefore the riverbank restoration plans should begin by supplying the river the correct flows to favour their proper functioning, the connection with their floodplain and the continuity of the ecological processes which take place.

The general goal of the following research project is to develop a mathematical model which simulates the riparian vegetation of a given river stretch. To obtain this goal the soil moisture model "Ribav" has been upgraded and adapted to the conditions of the Mediterranean regions of Spain. Subsequently the Ribav1D software has been developed with the Visual Basic.net programming language. The aim of this program is to use the model for individual simulation points. With the help of this software a manual sensitivity analysis of the Ribav software has been carried out. Additionally, the Ribav model has been calibrated and validated for 1 non-altered and 1 altered river stretches of Jucar basin district.

The Ribav2D software has been created to use grid maps, with ASCII-Raster format. Even so Ribav is a model that does not take into account certain riparian processes (shear stress effect, plant succession, recruitment...). For that reason the Ribav2D model has been merged with a dynamical vegetation model called "Ripflow v.2", using COM (*Component Object Model*) technology. Additionally a visual application has been designed to compare the results of the simulations of the "Ribav2D-Ripflow" merged model (by using MapWindow GIS libraries). This application determines the accurateness of the simulation by using the Cohen Kappa Coefficient.

# RESUM:

Les riberes són una part essencial dels ecosistemes fluvials, aunque la situació actual d'una bona part de les riberes a Espanya és roïna. Són nombrosos els factors que han provocat esta situació (agricultura, transports, urbanització, contaminació, canalizaciones...) donant lloc a què les riberes perden o experimenten un deteriorament tant en la seua estructura com en la seua funcionalitat És per això que els plans de restauració de riberes han de passar necessàriament per dotar al riu de llibertat i de subministrar-li els cabals adequats per a afavorir el seu funcionament correcte, la connexió amb la seua plana d'inundació i la continuïtat dels processos ecològics que en ella es desenrotllen.

El següent treball d'investigació té com a objectiu principal el desenrotllament d'un model matemàtic que permeta simular la vegetació riparia per a un determinat tram de riu. Per a aconseguir este objectiu s'ha adaptat el model d'humitat del sòl 'Ribav' a les condicions de l'Espanya mediterrània. Posteriorment s'ha desenrotllat el programa de Ribav-1D amb el llenguatge de programació Visual Basic.net, a fi de poder utilitzar el model en punts de simulació individuals. Amb este programa s'ha realitzat una anàlisi de sensibilitat manual del model 'Ribav'. A continuació s'ha calibrat i validat el model 'Ribav'.

Per a poder utilitzar mapes grid amb este model s'ha creat també el programa Ribav 2D que permet utilitzar el model bidimensionalment amb capes de format ASCII-Raster. Com Ribav és un model que no contempla alguns processos que ocorren en les riberes s'ha optat per unir Ribav 2D amb el programa del model de vegetació dinàmica 'Ripflow v.2', uilizando la tecnologia COM (Component Object Model). Per a comparar els resultats de les simulacions amb les dades obtinguts en camp del model unit 'Ribav2D-Ripflow', s'ha dissenyat una aplicació visual amb llibreries de MapWindow GIS, que permet determinar el grau d'encerts en la simulació del tipus de vegetació per mitjà del coeficient Kappa de Cohen.

## TABLA DE CONTENIDOS

<b>1.-</b>	<b>INTRODUCCIÓN .....</b>	<b>6</b>
1.1.-	OBJETIVOS Y METODOLOGÍA DEL TRABAJO.....	7
1.2.-	ANTECEDENTES .....	10
1.3.-	ESTADO DEL ARTE.....	12
1.3.1.-	<i>Balance Hídrico Conceptual: Movimientos de Agua en la Ribera .....</i>	<i>12</i>
1.3.2.-	<i>La Evapotranspiración de la Vegetación de Ribera .....</i>	<i>14</i>
<b>2.-</b>	<b>DESCRIPCION DE LOS MODELOS MATEMATICOS UTILIZADOS.....</b>	<b>18</b>
2.1.-	MODELO DE HUMEDAD DEL SUELO RIBAV .....	19
2.1.1.-	<i>Esquema general.....</i>	<i>19</i>
2.1.2.-	<i>Curva de humedad del suelo .....</i>	<i>21</i>
2.1.3.-	<i>Modelización de los flujos.....</i>	<i>22</i>
2.1.3.1.-	<i>Ascenso Hidráulico Radicular .....</i>	<i>22</i>
2.1.3.2.-	<i>Ascenso Hidráulico Capilar .....</i>	<i>22</i>
2.1.3.3.-	<i>Excedente hídrico del suelo .....</i>	<i>22</i>
2.1.3.4.-	<i>Evapotranspiración .....</i>	<i>23</i>
2.1.4.-	<i>El índice de ET .....</i>	<i>26</i>
2.1.5.-	<i>Resumen de variables y parámetros .....</i>	<i>26</i>
2.1.5.1.-	<i>Parámetros básicos de suelo del modelo .....</i>	<i>26</i>
2.1.5.2.-	<i>Parámetros básicos de vegetación del modelo.....</i>	<i>27</i>
2.1.5.3.-	<i>Datos de entrada hidro-meteorológicos .....</i>	<i>27</i>

2.1.5.4.-	Parámetros generales del modelo .....	27
2.1.5.5.-	Variables internas del modelo .....	28
2.1.5.6.-	Variable de salida del modelo .....	28
2.2.-	MODELO DE VEGETACION DINÁMICA RIPFLOW .....	29
2.2.1.-	<i>Introducción</i> .....	29
2.2.2.-	<i>Zonificación Espacial</i> .....	30
2.2.3.-	<i>Estructura del Modelo</i> .....	31
2.2.3.1.-	Primer Componente del Modelo: Condición Inicial Estática .....	31
2.2.3.2.-	Segunda Componente del Modelo: Llanura de Inundación Dinámica.....	33
1.1.1.1.-	Tercera Componente del Modelo: Vistas Output .....	39
2.3.-	COEFICIENTE KAPPA DE COHEN .....	40
<b>3.-</b>	<b>DISEÑO DEL SOFTWARE</b> .....	<b>43</b>
3.1.-	MODELO DE HUMEDAD DE SUELO “RIBAV” .....	44
3.1.1.-	<i>Plataforma y características generales</i> .....	44
3.1.2.-	<i>Estructura del programa</i> .....	44
3.1.3.-	<i>Interfaz gráfica de usuario</i> .....	46
3.1.4.-	<i>Utilización del Software de Ribav 1D</i> .....	48
3.1.4.1.-	Instalación del software y operaciones previas .....	48
3.1.4.2.-	Carpetas y archivos del programa .....	49
1.1.1.2.-	Preparación previa de los archivos de entrada (inputs) .....	50
1.1.1.3.-	Organización de Ventanas de Ribav1D .....	55

1.1.1.4.-	Ejecución del Software en Ribav1D .....	56
3.1.4.3.-	Resultados obtenidos mediante la simulación del software de Ribav1D.....	61
3.2.-	MODELO “RIBAV2D” Y SU INTEGRACIÓN CON RIPFLOW .....	68
3.2.1.-	<i>Diferencias entre Ribav 1D y 2D.....</i>	68
3.2.2.-	<i>Integración de Ribav 2D en Ripflow.....</i>	69
3.2.3.-	<i>Utilización de la tecnología Component Object Model (COM) .....</i>	72
3.2.4.-	<i>Utilización del Modulo Ribav2D .....</i>	76
3.3.-	PROGRAMA PARA EL CALCULO DEL COEFICIENTE KAPPA DE COHEN ENTRE LOS MAPAS DE VEGETACIÓN OBSERVADA Y VEGETACIÓN SIMULADA DE RIPFLOW V.3....	77
<b>4.-</b>	<b>CASOS DE ESTUDIO .....</b>	<b>78</b>
4.1.-	LOCALIZACIÓN DE LOS TRAMOS DE ESTUDIO .....	79
4.1.1.-	<i>Río Serpis en Lorcha .....</i>	80
4.1.2.-	<i>Río Mijares en el Terde.....</i>	81
4.2.-	ESTIMACIÓN DE PARÁMETROS.....	84
4.2.1.-	<i>Parámetros de Suelo .....</i>	84
	<i>Datos Hidrometeorológicos.....</i>	87
	<i>Parámetros de Vegetación.....</i>	88
1.1.1.5.-	Clasificación utilizada para los de tipos funcionales de vegetación .....	89
4.3.-	ANÁLISIS DE SENSIBILIDAD .....	93
4.3.1.-	<i>Planteamiento del análisis de sensibilidad.....</i>	93
4.3.1.1.-	<i>Parámetros de Vegetación.....</i>	93
4.3.1.2.-	<i>Variable de Salida Principal del Modelo.....</i>	94



4.3.1.3.-	Escenarios Utilizados durante el Análisis de Sensibilidad .....	95
4.3.1.3.1.-	Escenarios de suelos .....	96
4.3.1.3.1.1	Escenarios con distintos regímenes de caudal .....	96
4.3.1.3.2.-	Escenarios para distintos puntos con distinta cota.....	97
4.3.2.-	<i>Modificaciones de los valores de los parámetros .....</i>	98
4.3.3.-	<i>Resultados del análisis de sensibilidad.....</i>	101
4.3.4.-	<i>Conclusiones del análisis de sensibilidad.....</i>	102
4.4.-	CALIBRACIÓN DEL MODELO .....	103
4.4.1.-	<i>Matriz de confusión.....</i>	104
4.4.2.-	<i>Proceso de Calibración en Lorcha del Río Serpis (Régimen alterado).....</i>	105
4.4.2.1.-	Simulación 1 de Lorcha.....	105
4.4.2.2.-	Simulación 2 de Lorcha.....	106
4.4.2.3.-	Simulación 3 de Lorcha.....	108
1.1.1.6.-	Simulación 4 de Lorcha.....	109
4.4.2.4.-	Parámetros de calibración en Lorcha.....	111
4.4.3.-	<i>Proceso de Calibración en Terde del Río Mijares (Régimen Natural).....</i>	112
1.1.1.7.-	Simulación 1 de Terde .....	113
4.4.3.1.-	Simulación 2 de Terde .....	114
4.4.3.2.-	Simulación 3 de Terde .....	116
4.5.-	VALIDACIÓN DEL MODELO.....	117
5.-	<b>CONCLUSIONES Y POSIBLES LÍNEAS DE ACTUACIÓN FUTURA.....</b>	<b>118</b>

5.1.-	CONCLUSIONES.....	119
5.2.-	LÍNEAS DE ACTUACIÓN CON RESPECTO AL MODELO.....	120
5.3.-	LÍNEAS DE ACTUACIÓN CON RESPECTO AL SOFTWARE.....	123
6.-	REFERENCIAS .....	127
	<b>ANEXO I TUTORIAL DE UTILIZACIÓN DEL MODULO RIBAV 2D (INGLÉS) .....</b>	<b>130</b>
	<b>ANEXO IV TUTORIAL DE UTILIZACIÓN DE LA APLICACIÓN PARA EL CÁLCULO DEL COEFICIENTE DE COHEN PARA MAPAS DE VEGETACIÓN (INGLÉS).....</b>	<b>145</b>
	<b>ANEXO VII CÓDIGO FUENTE DEL PROGRAMA RIBAV-1D .....</b>	<b>152</b>
	<b>ANEXO VIII: CÓDIGO FUENTE DEL PROGRAMA RIBAV-2 .....</b>	<b>213</b>
	<b>ANEXO IX: CÓDIGO FUENTE PROGRAMA PARA EL CALCULO DEL COEFICIENTE KAPPA DE COHEN ENTRE LOS MAPAS DE VEGETACIÓN OBSERVADA Y VEGETACIÓN SIMULADA.....</b>	<b>243</b>
	<b>ANEXO X: TABLAS DE RESULTADOS DEL ANÁLISIS DE SENSIBILIDAD.....</b>	<b>262</b>
	<b>ANEXO XI: TABLAS DE PARÁMETROS DE SUELO.....</b>	<b>271</b>

## **1.- INTRODUCCIÓN**

## 1.1.- OBJETIVOS Y METODOLOGÍA DEL TRABAJO

Los ecosistemas de ribera son ecosistemas muy dinámicos que dependen de determinados regímenes de caudales para su persistencia, debido a que muchas especies vegetales necesitan quedar inundadas o cubiertas por nuevos sedimentos para poder regenerarse. En la actualidad quedan muy pocos bosques de ribera en Europa y se considera que ha desaparecido el 90 % de su extensión original debido a causas antrópicas. La directiva europea de hábitats considera que los bosques riparios son el ecosistema natural más amenazado.

Los bosques riparios albergan una gran diversidad de especies vegetales y proporcionan también hábitat o refugio a una gran cantidad de animales. Además tienen otras funciones de gran importancia: proporcionan corredores a la fauna a lo largo del territorio, almacenan agua y recargan los acuíferos durante los periodos de avenida, proporcionan recursos silvícolas y controlan o reducen la contaminación al reciclar el exceso de nutrientes producidos en las zonas agrícolas.

A pesar de la gran importancia de estos ecosistemas y de su conservación y restauración, en algunos casos existen conflictos en los usos fluviales. En el pasado las mayores amenazas a los procesos riparios han sido su deforestación para uso agrícolas, la modificación del recorrido de los cauces y la construcción de presas y azudes. Todas estas actuaciones han tenido como efecto, el reducir las conexiones entre los ríos y sus llanuras de inundación, reduciendo al mismo tiempo la habilidad de los bosques riparios para regenerarse de forma natural. Hoy en día todas estas alteraciones continúan existiendo, aunque existe una mayor concienciación por parte de la sociedad sobre la necesidad de restaurar los ecosistemas amenazados y de la “reconectar” los ríos con sus llanuras de inundación adyacentes.

Los bosques riparios son muy dependientes de las series de caudales y de la renovación de sedimentos. Necesitan una gran variedad de flujos en el río, zonas con elevada humedad y gran cantidad de sedimentos para la regeneración, semillas y material vegetativo para propagarse y niveles freáticos adecuados para el desarrollo de los árboles.

Específicamente los bosques de ribera necesitan caudales de intensidad de media a baja que recarguen y rellenen las capas freáticas. Estos flujos permiten el crecimiento de los árboles recién establecidos. Además también necesitan flujos altos de forma periódica que causen modificaciones en el recorrido del cauce y que produzcan la deposición de sedimentos, lo que causa que se creen lugares aptos para la regeneración vegetal. La variación de caudales y niveles del río durante el periodo de crecimiento también es positiva porque favorece la distribución de semillas y el establecimiento de plántulas. También es necesario que después de un periodo de crecidas el flujo disminuya pero de forma gradual para que el nivel freático también disminuya de la misma forma para que las plántulas se desarrollen de forma adecuada.

Las especies arbóreas pioneras de los bosques riparios necesitan lugares abiertos porque no toleran muy bien la competencia. Como las especies pioneras no tienen un sistema radicular muy desarrollado necesitan que durante el primer periodo de crecimiento el nivel freático no esté a demasiada profundidad. Los lugares cercanos a la orilla del río suelen ser los más adecuados para su establecimiento por tener más humedad, más sedimentos acumulados y más materia orgánica.

Los bosques riparios se pueden propagar de diversas formas. Una de ellas es el transporte de semillas producido por el mismo río y que quedan depositadas durante las inundaciones. Otra forma es mediante la propagación de semillas por la acción de los vientos dominantes. Una tercera forma

sería mediante el material vegetativo (tallos, bulbos, raíces....) que es transportado durante las crecidas o se crea de forma local como resultado de éstas.

Con este trabajo de investigación se pretende contribuir a la modelización de la mayoría de los procesos antes mencionados que tienen lugar en los ecosistemas vegetales riparios a escala local. Se pretende que las herramientas desarrolladas en este trabajo, contribuyan a facilitar la comprensión sobre el funcionamiento de estos complejos ecosistemas. Al mismo tiempo se pretende facilitar la toma de decisiones por parte de los organismos que se dedican a la gestión y a la restauración de los espacios fluviales. Por tanto el objetivo principal de este trabajo es el desarrollo de un modelo matemático que permita simular la distribución de los distintos tipos de vegetación riparia para un determinado tramo de río. Para conseguir este objetivo se han llevado a cabo las siguientes actividades:

- Perfeccionamiento y adaptación del modelo de humedad del suelo “Ribav” a las condiciones de la España mediterránea. El modelo Ribav había sido utilizado con éxito en tramos de estudio de Guatemala y ha sido adaptado para ser utilizado en los ríos de la España mediterránea.
- Desarrollo del software de Ribav-1D con Visual Basic.net 2008, con el fin de poder utilizar el modelo en puntos de simulación individuales. Este software ha sido utilizado durante el desarrollo y adecuación del modelo y podrá ser utilizado también en el futuro en otros trabajos de investigación o también por los usuarios finales de modelo.
- Realización de un análisis de sensibilidad manual del modelo “Ribav”. Este análisis ha servido para determinar el grado de relevancia de los parámetros del modelo que no habían podido ser estimados previamente. Adicionalmente ha servido para facilitar la comprensión de los procesos que intervienen en el modelo.
- Calibración y validación del modelo “Ribav” para 1 tramo de estudio natural y 1 tramo alterados de la demarcación hidrográfica del Júcar. Con los parámetros más sensibles se ha llevado a cabo la calibración del modelo. Para la validación se ha utilizado un tramo natural y otro alterado para probar la adecuación del modelo.
- Desarrollo del software de Ribav-2D, que permite utilizar el modelo bidimensionalmente con capas grid de formato ASCII-Raster. Esta nueva versión ha sido realizada porque en la actualidad las plataformas SIG permiten que los datos grid sean más fáciles de procesar y los resultados más fáciles de interpretar, a pesar de que las simulaciones tengan un coste computacional más alto.
- Unión del programa Ribav-2D con el software del modelo de vegetación dinámica “Ripflow v.2”. Tras los procesos de calibración/validación se ha observado que el modelo funciona satisfactoriamente en los tramos de estudio de la zona mediterránea española. Aún así dicho modelo no tiene en cuenta ciertos procesos destacados que ocurren en las zonas riparias (esfuerzo cortante, reclutamiento, sucesión vegetal...). Con el fin de mejorar las simulaciones, el modelo Ribav2D ha sido unido y complementado con el modelo Ripflow que tiene en cuenta más procesos.
- Diseño de una aplicación visual con librerías de “MapWindow GIS”, que permite determinar el grado de aciertos en las simulaciones mediante el coeficiente Kappa de Cohen. Aunque los resultados de una simulación 2D son más fáciles de interpretar, estos han exigido el

desarrollo de esta aplicación con el fin de automatizar el tratamiento de la gran cantidad de datos producidos.

## 1.2.- ANTECEDENTES

Las riberas son una parte esencial de los ecosistemas fluviales. Representan una zona de transición entre el medio acuático, de caudales circulantes, y el medio terrestre de las inmediaciones del río, recibiendo la influencia hidrológica de ambos, al constituir un espacio compartido en el ciclo del agua, de los sedimentos y de los nutrientes (González del Tánago 1998; Weltya et al., 2002).

La modelación de procesos a escala de ribera, puede partir tanto de una clasificación de su vegetación basada en la filogénesis (Ferreira et al, 2005; Rodríguez-González et al, 2004; Tabacchi y Planty-Tabacchi, 2003) como en otros sistemas no filogenéticos e incluso de descripción genérica basada en teledetección (Lite y Stromberg, 2005; Webb y Leake, 2006). En climas semiáridos destaca la aplicación de la clasificación por grupos funcionales de vegetación (Baird y Maddock 2005; Horton et al, 2001) como marco propicio de análisis de la interacción de factores bióticos y abióticos.

Es claro que la supervivencia de las plantas no se debe exclusivamente a la disponibilidad de agua, si no también a la presencia de nutrientes, sales minerales, dióxido de carbono, etc. (Terradas, 2001). Sin embargo, en climas áridos y semiáridos, estas últimas juegan un papel prácticamente secundario, siendo la frecuencia de ocurrencia del límite de la disponibilidad de agua en el suelo, la responsable de que la vegetación entre en estado de estrés hídrico (Goodwin y Tarboton, 2001; Rodríguez-Iturbe y Porporato, 2004).

A escala de ribera, el ciclo hidrológico debe estimarse con la suficiente precisión como para ser útil en la evaluación de recursos hídricos y de frecuencia de las crecidas (Altier et al, 2002; Richards et al, 1996).

En la mayor parte de los modelos de vegetación, la ecuación que determina su crecimiento real en función del crecimiento potencial, selecciona el máximo estrés entre los de humedad, nutrientes y temperatura, de tal forma que si uno domina, el resto no juega ningún papel (Altier et al, 2002). En el caso de las riberas de los cauces, el contenido de humedad y la posición del nivel freático están marcados por el régimen hidrológico del río (Richards et al, 1996), que depende tanto de la hidrología de la cuenca vertiente como de las modificaciones introducidas por el hombre a través especialmente de la construcción de embalses. Además, es crucial en el desarrollo de la vegetación de ribera la frecuencia y magnitud de las inundaciones (Tabacchi et al, 1998), ya que éstas generan rupturas bruscas en el crecimiento por arrastre de parte o de toda la masa vegetal.

La situación actual de una buena parte de las riberas en España es mala (Ibero, 1996). Son numerosos los factores que han provocado esta situación (agricultura, transportes, urbanización, contaminación, canalizaciones...) dando lugar a que las riberas pierdan o experimenten un deterioro tanto en su estructura como en su funcionalidad. La limitación del caudal circulante así como la modificación de la frecuencia e intensidad de las avenidas se han constituido como algunos de los principales factores que impiden la regeneración natural de los bosques riparios.

Es por ello que los planes de restauración de riberas deben pasar necesariamente por dotar al río de libertad y de suministrarle los caudales adecuados para favorecer su correcto funcionamiento, la conexión con su llanura de inundación y la continuidad de los procesos ecológicos que en ella se desarrollan.

Estas ideas ya se reflejaron en el Plan Hidrológico del Júcar (1999). Según el artículo 18 se considera como caudal ecológico o medioambiental *la disponibilidad de caudales que permitan el mantenimiento y la recuperación de los ecosistemas propios de cada tramo de río*. En el mismo artículo, apartados d y e, se considera prioritario el mantenimiento del caudal medioambiental en los siguientes casos:

- Espacios naturales valiosos determinados por la Administración Medio Ambiental, con especial atención a casos singulares de bosque de ribera.
- Afección a zonas húmedas ó a tramos fluviales de interés ambiental.

Ya más recientemente, la Ley de Aguas, después de su modificación en 2005, establece que a efectos de la *“la asignación y reserva de recursos para usos y demandas actuales y futuros, así como para la conservación y recuperación del medio natural”*... *“ se determinarán los caudales ecológicos, entendiendo como tales los que mantiene como mínimo la vida piscícola que, de manera natural, habitaría o pudiera habitar en el río, así como su vegetación de ribera.”*



## 1.3.- ESTADO DEL ARTE

### 1.3.1.- *Balance Hídrico Conceptual: Movimientos de Agua en la Ribera*

Se han hecho importantes avances para demostrar la relevancia del ascenso hidráulico radicular y de la redistribución hidráulica de agua en el suelo y en la disponibilidad de agua de las plantas. Así se ha consolidado la necesidad de modelar la redistribución del agua del suelo propiciada por la conectividad del sistema radicular con zonas más húmedas del perfil, y recientemente se han realizado unos pocos y valiosos trabajos de modelación del fenómeno.

Durante los períodos secos del año (de especial interés para climas semiáridos o estación del año con ausencia de lluvia) la parte del fenómeno de redistribución hidráulica que tiene mayor relevancia es el ascenso hidráulico radicular. Por ello, reconociendo que modelar el fenómeno dentro del contexto de la hidrología y ecología de las zonas de ribera es una innovación trascendental, en el modelo RibAV se propone la modelación mediante una aproximación simple y enfatizando el flujo ascendente de agua en la raíz (ascenso hidráulico o, en inglés “hydraulic lift”). Con base en el trabajo de Ryel et al (2002), el Ascenso hidráulico radicular ( $R_{wu}$ ) hacia una capa de suelo (i) desde una capa (j) puede expresarse a través de la ecuación siguiente:

$$R_{wu} = CRT \sum (P_{mat(j)} - P_{mat(i)}) \cdot MAX(c_i, c_j) \cdot \frac{R_i - R_j}{1 - R_x}$$

Por tanto el Ascenso Hidráulico Radicular ( $R_{wu}$ ) depende de:

- ( $P_{mat}$ ): Es el potencial matricial de cada capa de suelo.
- (CRT): Es la máxima conductividad radial raíz-suelo del sistema radicular activo tendiendo las siguientes unidades [ $cm MPa^{-1} h^{-1}$ ],
- ( $R_i$ ): Es la fracción de raíces activas en la capa (i),
- ( $R_j$ ): Es la fracción de raíces activas en la capa (j),
- ( $D_{tran}$ ): Es un factor que reduce el movimiento del agua entre capas debido a las raíces cuando la planta está transpirando. Se supone que  $D_{tran}=1$  durante la noche cuando la transpiración es mínima y  $D_{tran}=0$  durante el día.
- ( $R_x$ ): Puede tener dos valores: [ $R_x=R_i$  cuando  $H_i>H_j$ ] ó [ $R_x=R_j$  cuando  $H_j>H_i$ ] siendo H la humedad de cada una de las capas de suelo.

- Los factores (ci) y (cj) se obtienen mediante una relación empírica de van Genuchten a partir de la siguiente expresión, siendo: ( $P_{m50}$ ) el potencial hídrico del suelo al que se reduce la conductancia suelo-raíz en un 50%. y b una constante empírica.

$$C_i = \frac{1}{1 + \left( \frac{P_{mat}}{P_{m50}} \right)^b}$$

Para el Ascenso Capilar desde la zona saturada  $C_{wu}(t)$  se hizo una adaptación de la propuesta de Altier et al (2002) utilizada para riberas de ríos en el modelo REMM. Dicha propuesta considera que en presencia de un nivel freático poco profundo (entre 1 y 3 metros debajo de la capa de suelo) existe un flujo estacionario ascendente desde la zona saturada. En RibAV no se restringe el flujo mediante la definición de una profundidad por debajo de la capa del suelo, pues mediante simulaciones se ha observado que el flujo está suficientemente controlado por las variables de estado que le rigen.

De acuerdo con Altier et al (2002) el ascenso capilar  $C_{wu}$  entre dos puntos del suelo se puede describir utilizando la ecuación de Richards (1931). En dicha ecuación la tasa del movimiento de agua es función de: la conductividad hidráulica no saturada ( $K_h$ ), la diferencia de cota entre dos puntos de suelo ( $\partial z$ ) y por la diferencia de potencial mátrico ( $\partial P_{mat}$ ) entre dicho puntos:

$$C_{wu} = -K_h \cdot \left( \frac{\partial P_{mat}}{\partial z} - 1 \right)$$

Con base en la curva característica de Campbell(1974) es posible obtener las variables  $K_h$  y  $P_{mat}$ . La efectividad de predicción de esta curva se ha demostrado ya para riberas de río. Para esta curva la presión matricial ( $P_{mat}$ ) depende de: la humedad de suelo (H), la porosidad ( $P_{st}$ ), de la presión de burbujeo ( $P_b$ ) y del índice de poros del suelo ( $I_p$ ) tal como se puede observar en la siguiente ecuación:

$$\frac{H}{P_{st}} = \left( \frac{P_b}{P_{mat}} \right)^{I_p}$$

Por otro lado la conductividad hidráulica no saturada ( $K_h$ ) de la curva de Campbell(1974) depende: de la conductividad hidráulica en suelo saturado ( $K_s$ ), de la humedad de suelo (H), de la porosidad ( $P_{st}$ ), del índice de poros del suelo ( $I_p$ ), y de la presión matricial ( $P_{mat}$ ) calculada en la ecuación anterior.

$$\frac{K_h}{K_s} = \left( \frac{H}{P_{st}} \right)^{3 + \frac{2}{\lambda}}$$

### 1.3.2.- La Evapotranspiración de la Vegetación de Ribera

La evapotranspiración (ET) es un término colectivo para todos los procesos en los que el agua en fase líquida o sólida, cercana a la superficie terrestre, se convierte en vapor de agua atmosférico. Incluye por tanto la evaporación del agua líquida de ríos, lagos, suelo descubierto, y superficies vegetadas; la evaporación de dentro de las hojas de las plantas (transpiración); y la sublimación de las superficies con hielo o nieve (Dingman, 2002). Sin embargo, es frecuente el uso del término evapotranspiración para referirse a la evaporación total, y así enfatizar el proceso combinado de evaporación del suelo y las superficies con agua más la transpiración de las plantas (Ward y Robinson, 2000) considerando que en la naturaleza ambos procesos ocurren simultáneamente y no es fácil distinguirlos (Rivas, 2004). En este documento se hará énfasis en los trabajos de estimación de la ET para la ribera, por ello el resumen no incluye otros conceptos y métodos que se revisan en el documento completo.

La evapotranspiración potencial (ETP) es la tasa a la que la evapotranspiración podría ocurrir para un área grande que está cubierta de forma completa y uniforme por una vegetación creciente que tiene acceso a un abastecimiento ilimitado de agua del suelo, y sin considerar los efectos de advección o almacenamiento de calor (Dingman, 2002). Debido a que el proceso de evapotranspiración producido en una superficie natural no siempre se encuentra en las condiciones hipotéticas planteadas, se ha considerado la evapotranspiración real (ET<sub>r</sub>) que se define como la cantidad de agua verdaderamente perdida por el suelo a través del sistema suelo-planta hacia la atmósfera. En general, la ET<sub>r</sub> es menor, o en el caso límite, igual a la ETP (Rivas, 2004). Uno de los métodos más extensamente usados para estimar la ET<sub>r</sub> hace uso de datos meteorológicos para estimar la ETP y el uso de la siguiente ecuación:

$$ET_r = f(\theta_{rel}) \cdot ETP$$

Donde  $f(\theta_{rel})$  es el contenido relativo de agua en el suelo, definido como:

$$\theta_{rel} = \frac{\theta - \theta_{wp}}{\theta_{fc} - \theta_{wp}}$$

Donde,  $\theta$  es el contenido de agua actual,  $\theta_{fc}$  es la capacidad de campo y  $\theta_{wp}$  el punto de marchitez permanente en la zona radicular del suelo (Dingman, 1992).

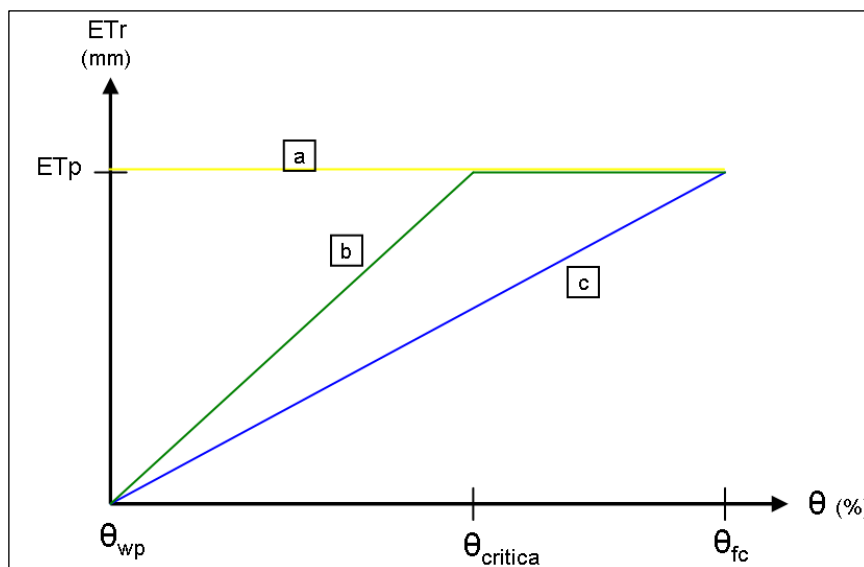
La relación entre ETP y ET<sub>r</sub> depende del contenido de humedad del suelo (Shaw, 1994). Según Dingman (2002), la relación entre la tasa ET<sub>r</sub>/ETP y  $\theta_{rel}$  usualmente presenta una forma en la que ET<sub>r</sub>/ETP se incrementa de forma cuasi-lineal a medida que  $\theta_{rel}$  se incrementa, y alcanza el valor de 1 en algún contenido de agua  $\theta_{crit}$ . Típicamente:  $(0.5 \cdot \theta_{cc}) \leq \theta_{crit} \leq (0.8 \cdot \theta_{cc})$ .

De acuerdo con Shaw (1994), a capacidad de campo se puede asumir que ET<sub>r</sub> = ETP; por lo que la ET ocurre a la tasa máxima posible determinada por las condiciones meteorológicas. Si no existen recargas del abastecimiento de agua, la humedad del suelo disminuirá gradualmente por las demandas de la vegetación para producir un déficit de humedad del suelo (DHS). Mientras el DHS

incrementa, la ETr se va haciendo incrementalmente menor que la ETP. Los valores de DHS y ETr varían con el tipo de suelo y vegetación.

Se han explorado correlaciones entre el potencial hídrico del suelo y la evaporación mediante curvas de secado que muestran tasas entre la transpiración y la transpiración sin estrés para vegetación en condiciones de poca disponibilidad de agua. Estas curvas de secado representan un modelo de caja negra que sustituye el entendimiento detallado del complejo agua-suelo-planta. Sin embargo, han permitido avances significativos en el entendimiento, medida y modelación de la evaporación de las coberturas vegetales y han constituido el medio para estimar las pérdidas de evapotranspiración real (Ward y Robinson, 2000).

Wild (1992) señala que el concepto clásico de disponibilidad de agua conduce a diversas hipótesis en cuanto a los efectos del agotamiento del agua sobre el crecimiento de las plantas. Las dos hipótesis clásicas extremas son: línea "a", igual disponibilidad desde CC a PMP; línea "c", la disponibilidad decrece gradualmente al reducirse el contenido de humedad del suelo. Entre esos extremos se han propuesto distintos patrones o hipótesis, y una de las más clásicas (línea "b") supone que hay una máxima intensidad de la actividad de la planta a una igual disponibilidad de agua desde la capacidad de campo hasta un punto de inflexión, en el que se alcanza un contenido de humedad crítica por debajo de la cual la disponibilidad disminuye.



**Figura 1.1.** Tres hipótesis clásicas donde se relaciona la evapotranspiración real de las plantas con el porcentaje de humedad del suelo.

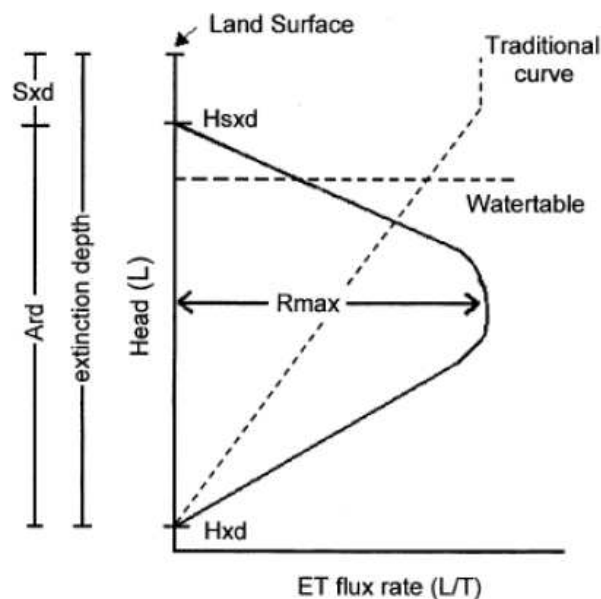
Una pista de ese punto de inflexión de la curva hipotética intermedia lo puede dar la figura de variación típica de la función de extracción de humedad  $f(\theta)$  (Fig. 25) presentada anteriormente de acuerdo con (Shuttleworth, 1993).

Actualmente una de las propuestas más avanzadas en materia de actividad de la planta frente a la dinámica de la humedad del suelo, con énfasis en el estrés hídrico, se presenta a través de un modelo complejo que considera tanto estrés estático como dinámico -dependiente de la escala

temporal- (Laio et al, 2001; Porporato et al, 2001; Rodríguez-Iturbe y Porporato, 2004). El vínculo entre el estrés hídrico de la planta y las dinámicas de humedad del suelo es un problema complicado, debido tanto a la naturaleza estocástica de las dinámicas como a la complejidad de las respuestas de las plantas al estrés hídrico (Porporato et al, 2001).

El modelo RIP-ET (Baird et al, 2003; Baird & Maddock, 2005; Maddock III y Baird, 2003) presenta curvas de evapotranspiración para condiciones semiáridas y para cada grupo funcional de vegetación, representado para condiciones climáticas medias, que también permiten considerar el tamaño de la planta (pequeña, mediana o grande) y/o la densidad de la población (baja, mediana o alta). En la modelación pueden tenerse desde una curva anual promedio de ET, hasta cuatro o más curvas que son función de las estaciones del año. Considerando la versatilidad de dichas curvas dentro del modelo, se aconseja al modelador el crear curvas para cada tipo de condiciones climáticas o estaciones.

En la figura mostrada a continuación se observa una curva genérica de tasa de flujo ET. El punto  $H_{xd}$  representa la profundidad de extinción por elevación (o la elevación del nivel freático debajo de la cual las raíces no pueden obtener agua y por tanto la ET es cero).



**Figura 1.2. Curva que relaciona la Evapotranspiración Real de las raíces con respecto a la profundidad de suelo.**

Conforme el nivel freático va subiendo, el agua está disponible para las raíces y las tasas de flujo ET crecen hasta llegar al máximo ( $R_{max}$ ), que corresponde a la tasa máxima promedio diaria de flujo de ET, medida o estimada. La elevación del nivel freático asociada con la muerte de la planta es la profundidad de extinción por elevación saturada ( $H_{sxd}$ ), que en este punto ET es cero también. Con fines de modelación, las relaciones entre las tasas de flujo y los niveles freáticos necesitan definirse como profundidades (L) o distancias relativas al nivel del suelo. La curva

“tradicional” de la figura, se usa exclusivamente para modelar la evaporación del suelo descubierto o del cuerpo de agua.

Las elevaciones de extinción pueden aproximarse en función de la profundidad de raíces de las especies dentro de un determinado grupo funcional; dichos valores pueden obtenerse mediante trabajo de campo, revisión de literatura o ambos. Aunque esas elevaciones pueden no ser precisas, sí que proveen de un buen punto de inicio para representar los valores máximo y mínimo de tolerancia de agua para las especies en cuestión (Baird et al, 2003; Baird & Maddock, 2005; Maddock III y Baird, 2003).

Para la modelación de los flujos ET se necesita su estimación mediante medidas de campo o bien factores de escala que suelen consistir en el área basal total que se basa en relaciones entre el diámetro a la altura del pecho del árbol y el área específica del tejido conductor de la savia. Asimismo es necesario escalar áreas, mediante la determinación del área cubierta por cada subgrupo funcional en una celda del modelo ( $fSG(i)$ ) y posteriormente el área con cobertura vegetal dentro de cada área del sub grupo funcional ( $fPC(i)$ ). Así, el área del  $i$  sub-grupo funcional vegetal que contribuye a la ET, se denomina cobertura fraccional ( $fCov(i)$ ) que equivale a:

$$fCov_{(i)} = fSG_{(i)} \cdot fPC_{(i)}$$

Considerando un sub-grupo funcional de vegetación presente en una celda. Si  $(h(k), R(k))$  y  $(h(k+1), R(k+1))$  son las coordenadas de vértices consecutivos que definen el segmento  $k$ -ésimo para ese subgrupo funcional de vegetación, y  $h$  es la altura en la celda, entonces la tasa de flujo ET para el subgrupo,  $R(h)$  está dado por:

$$R(h) = R(k) + \frac{h - h(k)}{h(k+1) - h(k)} (R(k+1) - R(k))$$

## **2.- DESCRIPCION DE LOS MODELOS MATEMATICOS UTILIZADOS**

## 2.1.- MODELO DE HUMEDAD DEL SUELO RIBAV

### 2.1.1.- *Esquema general*

Para este modelo se representa la porción de suelo estudiada como un tanque relleno de un material poroso. Dicho tanque contendrá agua, que en nuestro modelo llamaremos "Humedad del Suelo (H)". Como solo tendremos en cuenta la dimensión vertical, la cantidad de humedad solo será medida mediante su cota en milímetros. La humedad vertical tendrá una variación temporal de escala diaria por lo que nuestro modelo calculará dicha humedad para cada uno de los días de nuestro periodo de estudio.

Los límites verticales del tanque conceptual están definidos por las siguientes cantidades de agua:

- (Hfc): Humedad del suelo equivalente a capacidad de campo (mm): Es la cantidad máxima que puede retener un suelo sin dañar a las plantas por anaerobiosis.
- (Hwp): Humedad del suelo equivalente en el Punto de Marchitez Permanente (mm): Es la cantidad de agua en el suelo por la cual las plantas no pueden extraer más agua y pierden su turgencia.

En el modelo habrá que tener en cuenta las siguientes cotas (o profundidades absolutas):

- (Wte): Cota del Nivel Freático Calculada (m). Se considera la cota donde esta el límite entre la zona no saturada y la zona saturada del suelo. Para zonas de ribera o próximas a ella se considera equivalente al nivel del río, debido a que se considera el nivel freático como horizontal. El cálculo del nivel del río se calcula por la interpolación lineal entre la serie diaria de caudales y la curva de aforo de cada sección transversal (transecto) del tramo del río.
- (Czr): Cota de Profundidad Máxima de Raíces (m). Es un parámetro específico para cada tipo funcional de vegetación e indica la cota máxima a la que las raíces pueden acceder al agua del nivel freático.
- (Cze): Cota de Profundidad Efectiva de Raíces (m). Es la cota máxima a la que se encuentran las raíces que captan agua y evapotranspiran de la zona no saturada de suelo, que queda por encima del nivel freático. También es un parámetro específico para cada tipo funcional de vegetación y además es posiblemente uno de los parámetros más sensibles del modelo.
- (Cpsat): Cota para la Extinción por Saturación (m). Es la cota máxima a la cual las plantas pueden tolerar tener el suelo saturado de agua sin ser perjudicadas debido a la asfixia



radicular. Aunque las plantas necesitan tomar agua del suelo, también necesitan estar en contacto con gases atmosféricos mediante las raíces y/u otros órganos. En algunos casos esta cota puede ser superior a la cota de la superficie del suelo. Esto ocurre en las plantas acuáticas, que toleran bastante el anegamiento parcial de su parte aérea durante grandes periodos de tiempo. Cabe destacar que en este modelo se considera a la Capacidad de Campo como punto de almacenamiento máximo del tanque, a pesar de que el punto de almacenamiento máximo de los suelos es el de saturación que es equivalente a la porosidad (debido a que todos los poros representativos del suelo están repletos de agua). Esto es debido a que cuando la cantidad de agua de un suelo excede la capacidad de campo se produce la asfixia radicular, anulando la Evapotranspiración Real (ETR) de la planta.

Los flujos de entrada y salida en el tanque son los siguientes (todos en mm/día):

- (PP): La Precipitación Diaria
- (Rwu): Ascenso Hidráulico Radicular
- (Cwu): El Ascenso Hidráulico Capilar
- (Exc): El excedente hídrico del suelo
- (ETRtot): Evapotranspiración Real Total

En la siguiente figura se muestran los flujos y procesos que son simulados por el modelo:

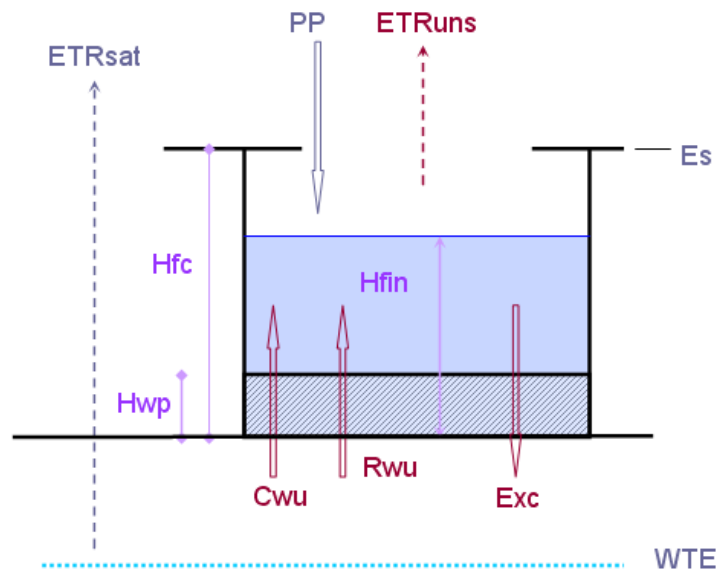


Figura 2.3. Esquema conceptual del modelo RibAv

### 2.1.2.- Curva de humedad del suelo

En cada intervalo diario habrá una serie de flujos de entrada y de salida, que harán que varíe el nivel de humedad del tanque. Para calcular algunos de dichos flujos será necesario calcular la Presión Matricial diaria (P<sub>mat</sub>). Dicha presión, medida en Kilopascales (Kpa), se calcula mediante la ecuación correspondiente de la curva característica de Campbell (1974):

$$P_{mat} = \frac{P_b}{\left( \frac{H_{fin(t-1)}}{P_{st}} \right)^{1/l_p}}$$

donde:

- H<sub>fin</sub>(t-1): Es la humedad del tanque del día anterior (mm)
- P<sub>st</sub>: Es la Porosidad o ratio de huecos del suelo ( ).
- l<sub>p</sub>: Es el Índice de Poros del suelo ( ).
- P<sub>b</sub>: Es la Presión de Burbujeo del suelo (Kpa).

Existe otra curva de Campbell (1974) que sirve para calcular diariamente la conductividad hidráulica no saturada (K<sub>h</sub>) a partir de algunos de los parámetros y variables antes descritos y de la Conductividad Hidráulica Saturada (K<sub>s</sub>). La ecuación derivada de dicha curva es la siguiente:

$$K_h = K_s \cdot \left( \frac{P_b}{P_{mat}} \right)^{3 \cdot lp + 2}$$

## 2.1.3.- Modelización de los flujos

### 2.1.3.1.- Ascenso Hidráulico Radicular

Es la transferencia vertical nocturna de agua de zonas más húmedas a zonas más secas producida por las raíces de las plantas. Depende de algunos parámetros y variables mencionados anteriormente y de otros como: la Presión para el Punto de Capacidad de Campo ( $P_{fc}$ ); del Factor de Transpiración en Zona No Saturada ( $R_j$ );  $P_{m50}$  es potencial hídrico del suelo en Kpa al que se reduce la conductancia suelo-raíz en un 50%, y CRT es la máxima conductividad radial raíz-suelo del sistema radicular activo (suele tomarse normalmente como  $0.97 \text{ mm MPa}^{-1} \text{ h}^{-1}$ ). La ecuación está basada en la de los modelos REMM y queda de la siguiente forma:

$$Rwu = MAX \left( \frac{-Crt}{1000} \cdot (P_{fc} - P_{mat}) \cdot \frac{1}{1 + \left( \frac{P_{mat}}{P_{m50}} \right)^{3.32}} \cdot R_j \right)$$

### 2.1.3.2.- Ascenso Hidráulico Capilar

Es producido por las fuerzas de capilaridad que ejercen los poros de suelo sobre las moléculas de agua. Dicha atracción capilar es producida por la tensión superficial de los líquidos que resulta representativa en los conductos de reducido diámetro como son los poros de la esponjas o del suelo. Así mismo la tensión superficial presente en los líquidos es producida por la descompensación que producen las superficies sólidas sobre las fuerzas de atracción de las moléculas de agua (puentes de hidrógeno). Su ecuación está basada en la de Richards (1931) y queda de la siguiente forma

$$Cwu = MIN \left( \frac{H_{fc} - H_{fin(t-1)} - Rwu}{\left( \frac{0.102 \cdot P_{mat}}{WTE - Cze} - 1 \right) \cdot K_h \cdot 24} \right)$$

### 2.1.3.3.- Excedente hídrico del suelo

El excedente (Exc) será el principal flujo de salida de agua del tanque y representará el agua que se pierde por escorrentía o por percolación, debido a la capacidad limitada de almacenamiento de la

porción de suelo modelizada. Su ecuación incluirá el sumatorio de los flujos de entrada antes mencionados (PP, Cwu y Rwu), más la Humedad de Suelo del Día Anterior ( $H_{fin(t-1)}$ ), menos la Humedad del Suelo a Capacidad de Campo ( $H_{fc}$ ).

$$Exc = MAX \left( \begin{array}{c} 0 \\ PP + Cwu + Rwu - H_{fc} + H_{fin(t-1)} \end{array} \right)$$

#### 2.1.3.4.- Evapotranspiración

Antes de describir los procesos de evapotranspiración cabría describir las siguientes variables y parámetros:

- (Win): Entrada de agua en suelo (mm). Es la diferencia entre los flujos de entrada y el de salida anteriormente descritos.

$$Win = PP + Cwu + Rwu - Exc$$

- (Hini): Humedad del suelo antes de considerar la Evapotranspiración Real (mm). Esta variable no es la humedad del suelo al final del día pero sirve para el cálculo de ésta.

$$H_{ini} = Win + H_{fin(t-1)}$$

- (Hcrit): Agua Equivalente a humedad crítica (mm). Para dicha humedad normalmente se considera la presión crítica de 95 Kpa.
- (Hrel): Humedad Relativa (mm). Esta variable representa la relación lineal existente entre el aumento de Humedad del Suelo y la Evapotranspiración Real en el tramo que va entre el punto de marchitez permanente y el punto crítico.

$$H_{rel} = MIN \left( \begin{array}{c} 1 \\ \frac{H_{ini} - H_{wp}}{H_{crit} - H_{wp}} \end{array} \right)$$

- (Cov): Fracción de Cobertura de las plantas ( ). Es un parámetro de vegetación que actúa como factor reductor de la Evapotranspiración Real de una forma similar al Coeficiente de Cultivo de las plantas agrícolas.
- (Ri): Factor de Transpiración de la Zona No Saturada ( ): Es una factor multiplicador que influye sobre los procesos de evapotranspiración que tienen lugar en la zona no saturada del suelo.

Por último solo queda describir las variables de salida del modelo que tienen que ver con los procesos de evapotranspiración. Primeramente la evapotranspiración es el proceso, que forma parte del ciclo hidrológico donde el agua es devuelta a la atmósfera en forma de vapor. Dicho proceso está compuesto por dos mecanismos: el de evaporación, donde el agua es captada desde las superficies; y el de transpiración, que es realizado por los vegetales. Sin embargo, aunque los dos mecanismos son diferentes y se realizan independientemente resulta difícil separarlos, pues ocurren por lo general de manera simultánea; de este hecho deriva la utilización del concepto más amplio de evapotranspiración que los engloba. En relación al proceso de Evapotranspiración existen las siguientes variables:

- (ETP): Evapotranspiración Potencial (mm). Es la máxima cantidad de agua que puede evaporarse desde un suelo completamente cubierto de vegetación, que se desarrolla en óptimas condiciones, y en el supuesto caso de no existir limitaciones en la disponibilidad de agua. Por tanto, la magnitud de la ETP está regulada solamente por condiciones meteorológicas e indica la cantidad máxima de agua que puede evapotranspirar una planta en condiciones ambientales óptimas.
- (ETRtot): Evapotranspiración Real Total (mm). Es el agua que utiliza una planta bajo unas determinadas condiciones y en un determinado estado vegetativo. Es la variable principal de salida del modelo porque indica el grado de adecuación de una planta (o grupo funcional) a unas condiciones ambientales. Según la Cota diaria del Nivel Freático (Wte), se subdivide en dos variables:
- (ETRuns): Evapotranspiración Real de la Zona No Saturada. Es la ETR que tienen lugar en la porción del suelo, cuyos poros no se encuentran saturados de agua en ese momento. Se considera que cuando la Cota diaria del Nivel Freático (WTE) está por encima de la Cota de Extinción por Saturación (Cpsat), el valor de ETRuns es igual a 0. Si la condición anterior no se cumple y la WTE es mayor o igual a la Cota de Raíces Efectivas (Cze) entonces la ecuación sería la siguiente:

$$ETRuns = Cov \cdot ETP \cdot Ri \cdot \left( 1 - \left( \frac{WTE - Czr}{Cpsat - CZr} \right) \right)$$

En el caso opuesto, cuando  $WTE < Cze$  la ecuación de ETRuns sería la siguiente:

$$ETRuns = MIN \left( \begin{array}{c} H_{ini} - H_{wp} \\ Ri \cdot \frac{H_{ini} - H_{wp}}{H_{crit} - H_{wp}} \cdot Cov \cdot ETP \end{array} \right)$$

- (ETRsat): Evapotranspiración Real de la Zona Saturada. Es la ETR que tienen lugar en la porción de suelo cuyos poros se encuentran saturados de agua en ese momento. Se

considera que cuando la Cota diaria del Nivel Freático (Wte desplazamiento) está por encima de Cota para la Extinción por Saturación (Cpsat) o por debajo de la Profundidad Máxima de Raíces el valor de ETRsat es igual a 0. Si las condiciones anteriores no se cumplen entonces la ecuación sería la siguiente:

$$ETRs_{at} = \text{MIN} \left( \begin{array}{l} Cov \cdot ETP - ETR_{uns} \\ Cov \cdot ETP \cdot R_j \cdot \frac{WTE - CZr}{CZe - CZr} \end{array} \right)$$

En la siguiente gráfica se muestra la evolución de la Evapotranspiración Real (ETR) según va variando la humedad de suelo y los poros van llenándose de agua. Se observa dicha evolución para distintas cantidades de humedad determinadas por el parámetro de suelo de Capacidad de Campo (Pfc) y por los parámetros de vegetación: Punto Crítico (Pcrit) y Punto de Marchitez Permanente (Pwp). La gráfica comienza por con la ETR siendo mínima hasta que se llega al Punto de Marchitez Permanente (Pwp) que empieza a ascender. Dicho ascenso continúa hasta que se llega al Punto Crítico donde la ETR se estabiliza en una recta horizontal. El valor vertical de esta recta sería el de la Evapotranspiración Potencial (ETP), que es la evapotranspiración máxima que permiten las circunstancias climáticas de ese momento. Esta situación prosigue hasta que se sobrepasa a la Capacidad de Campo del suelo, donde la ETR empieza a descender hasta que la humedad del suelo llega a ser equivalente a la porosidad. Este descenso en la ETR es producido por la asfixia radicular de la planta. La relación entre ETR/ETP es un indicador muy importante porque muestra el estado fisiológico de la planta en ese momento. Este es óptimo cuando ETR=ETP ó lo que es lo mismo: ETR/ETP=1

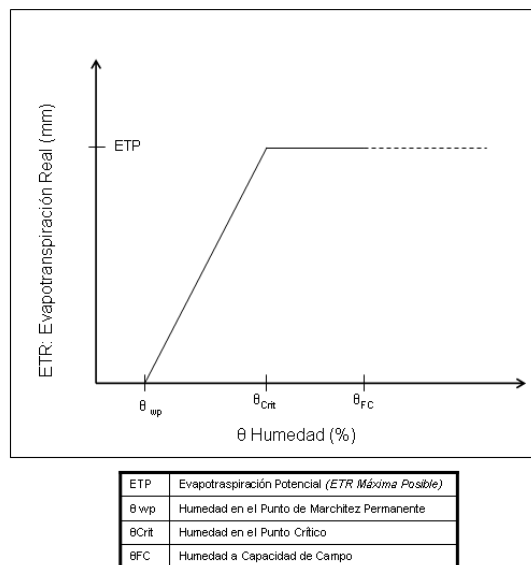


Figura 2.4. **Gráfica que relaciona la ETR con la Humedad del Suelo**

## 2.1.4.- El índice de ET

Para cada intervalo diario se calcula la evapotranspiración real total (ETR<sub>tot</sub>) para cada tipo funcional de vegetación. La suma de ETR<sub>tot</sub> se divide entre la suma de Evapotranspiración Potencial (ETP) diaria, lo que dará el Índice de Evapotranspiración (ETindex) de cada tipo funcional de vegetación. Si este índice es más grande y más cercano a 1 significará que las plantas que forman parte del grupo funcional serán más aptas para las condiciones ambientales simuladas. En la tabla siguiente se puede observar que para cada punto de simulación se considera como tipo funcional simulado al que tenga un índice de evapotranspiración más alto, el cual se representará con el valor "1".

Punto de Simulación	Índice de Evapotranspiración (ETind) Simulado para cada Tipo Funcional				Tipo Funcional con el ETind más Elevado (Representado por "1")			
	ETind_RH	ETind_RJ	ETind_RA	ETind_TV	RH	RJ	RA	TV
1 1 54	0.3623	0.4072	0.4072	0.5559				1
1 2 54	0.3806	0.4279	0.6228	0.6345				1
1 3 54	0.4174	0.4816	0.7802	0.9970				1
1 4 21	0.5680	0.8871	0.9726	0.6740			1	
1 5 21	0.8683	0.9960	0.9991	0.5030			1	
1 6 21	0.9578	0.9988	0.9986	0.4072		1		
1 7 18	0.9655	0.9968	0.9949	0.2291		1		
1 8 19	0.9992	0.9284	0.8129	0.0320	1			
1 9 8	0.9984	0.4861	0.5294	0.0002	1			
1 10 9	0.9990	0.8870	0.7488	0.0141	1			
1 11 10	0.9866	0.9985	0.9982	0.3532		1		
1 12 13	0.8033	0.9698	0.9993	0.5429			1	
1 13 14	0.8000	0.9426	0.9993	0.6593			1	
1 14 55	0.4335	0.5107	0.6843	0.9969				1
1 15 55	0.3950	0.4462	0.5771	0.9864				1
2 1 54	0.3802	0.4274	0.6187	0.6305				1
2 2 54	0.3913	0.4424	0.7056	0.7965				1

Figura 2.5. Salida de datos del programa.

## 2.1.5.- Resumen de variables y parámetros

A continuación se va a mostrar una serie de tablas donde aparecen los distintos tipos de variables y parámetros del modelo.

### 2.1.5.1.- Parámetros básicos de suelo del modelo

En esta tabla se mostrarán los parámetros de suelo que intervienen en el modelo y cuya obtención y tratamiento se explica en el apartado de estimación de parámetros:

Tabla 2.1 Parámetros básicos de suelo

Siglas	Nombre	Unidades
Mfc	Humedad a capacidad de Campo	[ ]
Pst	Porosidad	[ ]
Ks	Conductividad Hidráulica Saturada del Suelo	[mm/hr]
Pb	Presión de Burbujeo	[Kpa]
Ip	Índice de Porosidad	[ ]

Es	Cota de la Superficie del Suelo	[m]
Zcmin	Profundidad a considerar en el Ascenso Hidráulico Capilar	[m]

### 2.1.5.2.- *Parámetros básicos de vegetación del modelo*

En esta tabla se mostrarán los parámetros de suelo que intervienen en el modelo y cuyo significado se explica en el apartado de estimación de parámetros. Algunos de estos parámetros son muy sensibles para el modelo y por ello son utilizados en el apartado de calibración.

**Tabla 2.2 Parámetros básicos de vegetación**

<b>Siglas</b>	<b>Nombre</b>	<b>Unidades</b>
Pcrit	Presión para el Punto de Humedad Crítica	[Kpa]
Pwp	Presión para el Punto de Marchitez Permanente	[Kpa]
Zr	Profundidad Máxima de Raíces	[m]
Ze	Profundidad Efectiva de Raíces	[m]
Zsat	Prof. de Extinción por Saturación	[m]
Rj	Factor de Transpiración de la Zona Saturada	[ ]
Cov	Fracción de Cobertura de las Plantas	[ ]
Crt	Conductividad Máxima de Agua Raíz-Suelo	[mmMpa-1h-1]
Ri	Factor de Transpiración de la Zona No Saturada	[ ]
Pm50	Presión para el Punto Medio de Saturación	[Kpa]

### 2.1.5.3.- *Datos de entrada hidro-meteorológicos*

En la siguiente tabla se mostrarán los datos de entrada relativos tanto a procesos hidrológicos como meteorológicos.

**Tabla 2.3 Datos de entrada hidro-meteorológicos**

<b>Siglas</b>	<b>Nombre</b>	<b>Unidades</b>
PP(t)	Precipitación Diaria	[mm]
ETp(t)	Evapotranspiración Potencial Diaria	[mm]
Qcurve(j)	Caudal de Río de la Curva de Aforo	[m3/s]
Hcurve(j)	Nivel de Río de la Curva de Aforo	[m]
Qdaily(t)	Caudal Diario Observado en el Río	[m3/s]
WTE(t)	Cota del Nivel Freático Calculada	[m]

### 2.1.5.4.- *Parámetros generales del modelo*

En la tabla mostrada a continuación se muestran los parámetros generales del modelo, que son obtenidos a partir de los parámetros de las tablas anteriores, y cuyos valores son iguales para todos los intervalos diarios dentro de un mismo punto de simulación.

**Tabla 2.4 Parámetros generales del modelo**

<b>Siglas</b>	<b>Nombre</b>	<b>Unidades</b>
Mcrit	Humedad en el Punto Crítico	[ ]
Mwp	Humedad en el Punto de Marchitez Permanente	[ ]
CZr	Cota de la Profundidad Máxima de Raíces	[m]
Cze	Cota de la Profundidad Efectiva de Raíces	[m]
Cpsat	Cota para la Extinción por Saturación	[m]
Hfc	Agua Equivalente para la Capacidad de Campo	[m]
Hwp	Agua Equivalente para el Punto de Marchitez Permanente	[mm]



Hcrit	Agua Equivalente para el Punto de Humedad Crítica	[mm]
Pfc	Presión para el Punto de Capacidad de Campo	[Kpa]

#### 2.1.5.5.- Variables internas del modelo

En la tabla siguiente se muestran las variables internas del modelo, obtenidas a partir de las variables y parámetros descritos en las tablas anteriores, y cuyos valores son variables para cada intervalo diario.

**Tabla 2.5 Variables internas del modelo**

<b>Siglas</b>	<b>Nombre</b>	<b>Unidades</b>
Pmat	Potencial Mátrico	[Kpa]
Kh	Conductividad Hidráulica No Saturada	[mm/hr]
Rwu	Ascenso Hidráulico Radicular	[mm]
Cwu	Ascenso Hidráulico Capilar	[mm]
Exc	Excedente de Agua	[mm]
Win	Entrada de Agua en el Suelo	[mm]
Hini	Humedad antes de la ETR	[mm]
Hrel	Humedad Relativa	[ ]
ETRuns	Evapotranspiración Real de la Zona No Saturada	[mm]
ETRsats	Evapotranspiración Real de la Zona Saturada	[mm]
ETRtot	Evapotranspiración Real Total	[mm]
Hfin	Humedad Final del Día	[mm]
Bal	Error de Balance Hídrico Diario	[mm]

#### 2.1.5.6.- Variable de salida del modelo

En esta tabla se muestra la variable de salida principal del modelo. Se calcula esta variable para cada tipo funcional de cada punto de simulación:

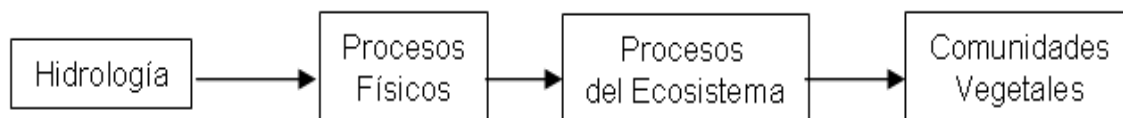
**Tabla 2.6 Variable de salida del modelo**

<b>Siglas</b>	<b>Nombre</b>	<b>Unidades</b>
ETindex	Índice de Evapotranspiración	[ ]

## 2.2.- Modelo de vegetación dinámica ripflow

### 2.2.1.- Introducción

El concepto fundamental para el desarrollo de un Modelo Dinámico de Vegetación de Ribera (Ripflow) es la relación funcional entre: la hidrología, los procesos físicos, los ecosistemas riparios y las distintas comunidades vegetales (Benjankar 2006).



**Figura 2.6.** Concepto fundamental del modelo dinámico de vegetación Riparia (Benjankar 2006).

Ripflow es un modelo de vegetación con reglas dinámicas que se basan en los parámetros físicos simulados, en los datos observados y en las reglas de expertos. Los componentes del modelo quedan representados en forma de malla raster que simulan los procesos de sucesión y regresión para un intervalo de tiempo anual. El objetivo principal del modelo es el de simular el efecto de los procesos físicos significativos sobre los ecosistemas riparios. Dichos procesos son: la resistencia al esfuerzo cortante, el estrés fisiológico debido al anegamiento y la influencia del nivel freático sobre el desarrollo vegetal. El dato de salida del modelo es la Vegetación Potencial Natural, que es independiente de las influencias antrópicas tales como ganadería, agricultura, selvicultura, etc. Al ser Ripflow un modelo de vegetación dinámico trabaja con sucesiones vegetales que comienzan a partir de una comunidad muy simple llamada "pionera" hasta terminar en otra muy compleja llamada "clímax". En los ecosistemas simulados las comunidades vegetales se ven afectadas por sucesiones y regresiones debido a las diferentes alteraciones del medio a las que se ven sometidas.

Los procesos físicos son simulados con un modelo hidrodinámico auxiliar. Inicialmente el modelo hidrodinámico se desarrolla para simular los parámetros físicos para el modelo de vegetación como pueden ser el esfuerzo cortante, la duración de inundación y las distintas cotas de inundación. Para simular los procesos físicos todos los hidrogramas anuales de caudal utilizados en el estudio se clasifican en diferentes clases basadas en condiciones climáticas, como por ejemplo: años húmedos, medios y secos. Se considera que todos los procesos simulados en los hidrogramas pertenecientes a la misma clase tienen el mismo impacto en la dinámica de vegetación. Por ello no hay ningún enlace dinámico entre el modelo hidráulico y de vegetación, por lo que los parámetros y los datos de entrada deben ser almacenados en un lugar y con un formato específico para que el modelo de vegetación pueda acceder a dichos datos de entrada.

## 2.2.2.- Zonificación Espacial

El área de estudio se divide en 3 zonas: Acuática, de Ribera y de Llanura de Inundación. Esta clasificación se en la magnitud y presencia de las inundaciones, que rigen la ausencia y/o presencia de especies vegetales riparias:



**Figura 2.7.** Zonas contempladas en el modelo

- Zona Acuática (ZA): Se considera parte del río principal que se inunda debido al caudal promedio del río.
- Zona de Ribera (ZR): Es el área inmediatamente adjunta a la zona acuática, estando sometida a un elevado grado de perturbaciones. La vegetación normalmente tiene un ciclo de vida corto debido a las perturbaciones constantes.
- Zona de Llanura de Inundación (ZL): La llanura de inundación es una menos perturbada que la zona de Ribera. Se puede considerar que la perturbación es casi insignificante debido a las grandes distancias y cotas con respecto al cauce principal. Por ello los tipos de vegetación tienden a ser más robustos y mejor desarrollados.

Los límites entre las distintas zonas quedan implementados de la siguiente forma:

**Tabla 2.7 Límites entre las distintas zonas de modelación**

Borde	Caudales	Valores
ZA/ZR	Cota de la lamina de Agua Media de Río	Qmedio
ZR/ZL	Ribera Totalmente Inundada	*HQ1-HQ1,5
ZL	Históricos	*HQ100

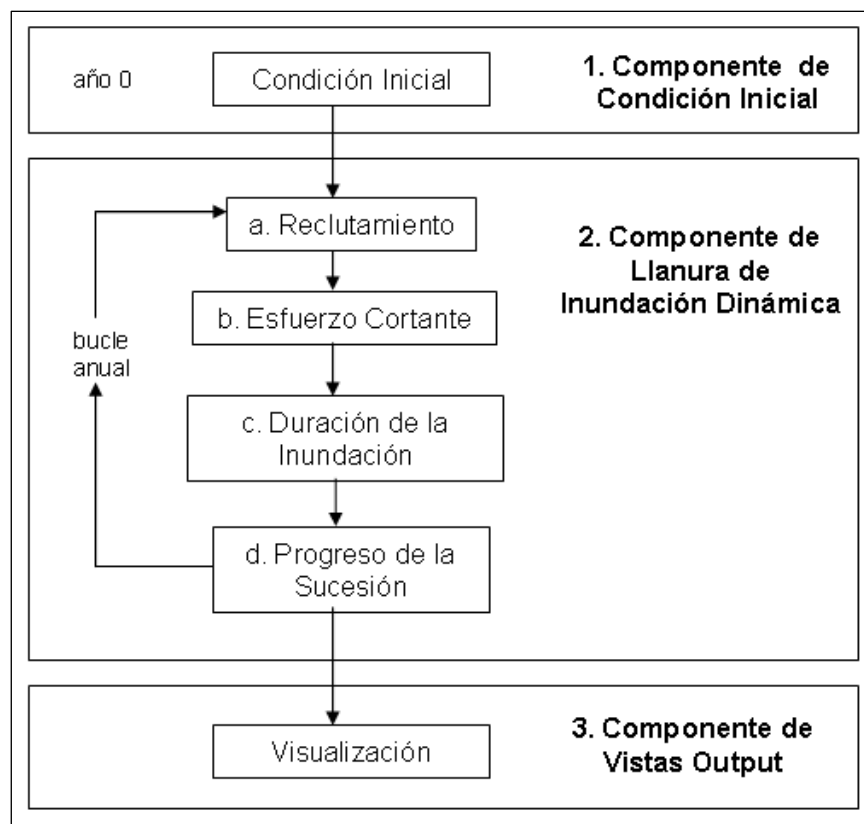
\*Siendo HQx la cota de la lamina de agua para un caudal con un periodo de retorno de x años.

### 2.2.3.- Estructura del Modelo

Los procesos de modelación se clasifican en 3 fases que corresponden con los 3 diferentes componentes que son:

- Condición Inicial Estática.
- Llanura de Inundación Dinámica, que asimismo se subdivide en 4 sub-componentes.
- Vistas Output

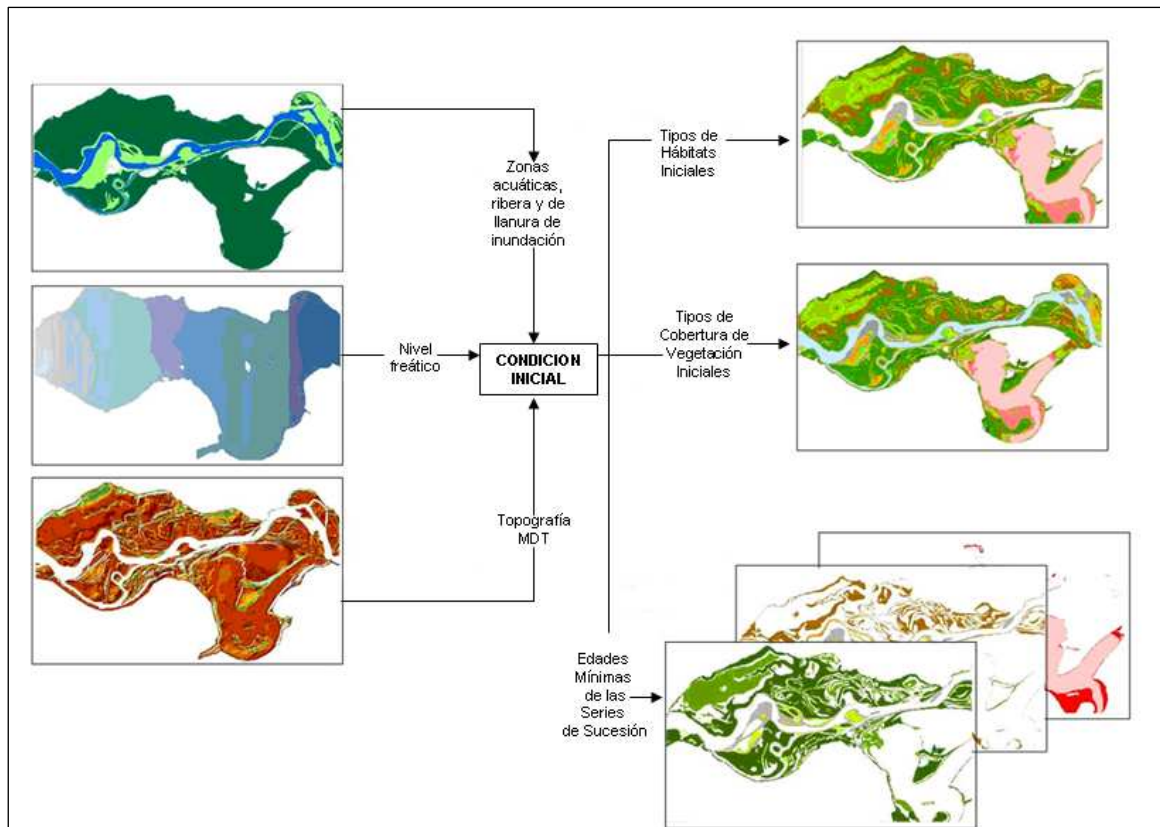
En la gráfica siguiente se observa que las flechas muestran que el output de cada proceso es el input del siguiente proceso. Los datos hidráulicos y topográficos son proporcionados como inputs externos, por lo que deben ser obtenidos mediante otros procesos de modelación.



**Figura 2.8.** Estructura del modelo

#### 2.2.3.1.- Primer Componente del Modelo: Condición Inicial Estática

La condición inicial además de ser estática, predice los diferentes hábitats basándose en la profundidad del nivel freático (*cota del nivel freático menos la cota de la superficie del suelo*) y en la distinta zonificación por caudales de retorno (*Acuática, Riparia y de Llanura de Inundación*). El objetivo principal de este componente es el de predecir la vegetación potencial natural y utilizarla como condiciones iniciales para el modulo siguiente. Los mapas que intervienen en esta componente del modelo son los siguientes:



**Figura 2.9.** Mapas input y output que intervienen en la componente inicial estática.

Restando los valores de cada celda de los mapas input de topografía (MDT) y el de cota de nivel freático se calcula la profundidad a la que está el nivel Freático. Combinando los valores discretizados de esta profundidad con el mapa de las distintas zonas de inundación (Acuática, Ribera y Llanura de Inundación) que relacionan la frecuencia de inundación, se obtienen: los tipos de hábitats iniciales, la serie de vegetación a la que pertenecen y la edad (mínima) de la vegetación de dicho hábitat. Dicha relación se observa en la tabla siguiente:

**Tabla 2.8** *Ejemplo de tabla que relaciona las alturas sobre el nivel freático y las zonas de inundación con el tipo de cobertura de vegetación y su edad. Estos valores deben ajustarse durante el proceso de calibración.*

Zona según Frecuencia de Inundación	Alturas sobre el Nivel Freático	Sub-Zonas	Tipo de Cobertura (Español)	Tipo de Cobertura (Inglés)	Rango de Años
<b>Zona Acuática</b>	-15/-1	Zona Acuática Profunda	Río Profundo	Deep River	-
	-1/0	Zona Acuática Somera	Río Somero	Shallow River	-
<b>Zona de Ribera</b>	0-0.2	Zona de Ribera Baja	Banco de Grava y Arena	Gravel and Sand Bar	0-1
	0.2-1	Zona de Ribera Media	Vegetación Pionera	Pioneer Vegetation	2-3
	1-1.6	Zona de Ribera Alta	Cañaveral y Herbáceas	Reed and Forbs	4-25
	1.6-20		Sauces y Chopos Arbustivos	Willow and Cottonwood Shrub	4-15
<b>Zona de Llanura de Inundación</b>	-5/1	Humedal Bajo	Marisma Profunda	Deep Marsh	0-3
	-1/0	Humedal Medio	Marisma Somera y Prado Húmedo	Shallow Marsh and Wet Meadow	4-25
	0-0.5	Humedal Alto	Herbáceas y Arbustivos de Humedal	Wet Forbs and Shrubs	26-110
	0.5-0.9	Cañaveral Bajo en Llanura de Inundación	Cañaveral y Herbáceas de Humedal	Reed Forbs and Shrubs	26-110
	0.9-2	Bosque Ripario Bajo de Llanura de Inundación	Bosque de Chopos Jóvenes	Young Cottonwood Forest	16-55
	2-4	Bosque Ripario Medio de Llanura de Inundación	Bosque de Chopos Adultos	Old Cottonwood Forest	56-110
	4-20	Bosque Ripario Alto de Llanura de Inundación	Bosque Mixto Maduro	Mature Mixed Hardwood Forest	110-300

### 2.2.3.2.- Segunda Componente del Modelo: Llanura de Inundación Dinámica

Este componente evalúa el crecimiento (sucesión) y la distribución espacial de la vegetación de la llanura de inundación basándose en cinco parámetros físicos diferentes (topografía, profundidad sobre el nivel freático, zonificación por caudales de retorno, esfuerzo cortante y duración de la inundación), Hay 3 tipos diferentes de series de sucesión (bosque ripario, cañaveral, y pradera riparia). Cada serie de sucesión tiene etapas diferentes, cuyo establecimiento se basa en un rango definido de edades.

Esta componente del modelo dará como output los mapas anuales de tipo de cobertura de vegetación. Tendrá como inputs los mapas de: edades mínimas de las series de sucesión, tipos de hábitats iniciales, nivel freático, topografía (MDT), láminas de agua del río, esfuerzo cortante y duración de la inundación.

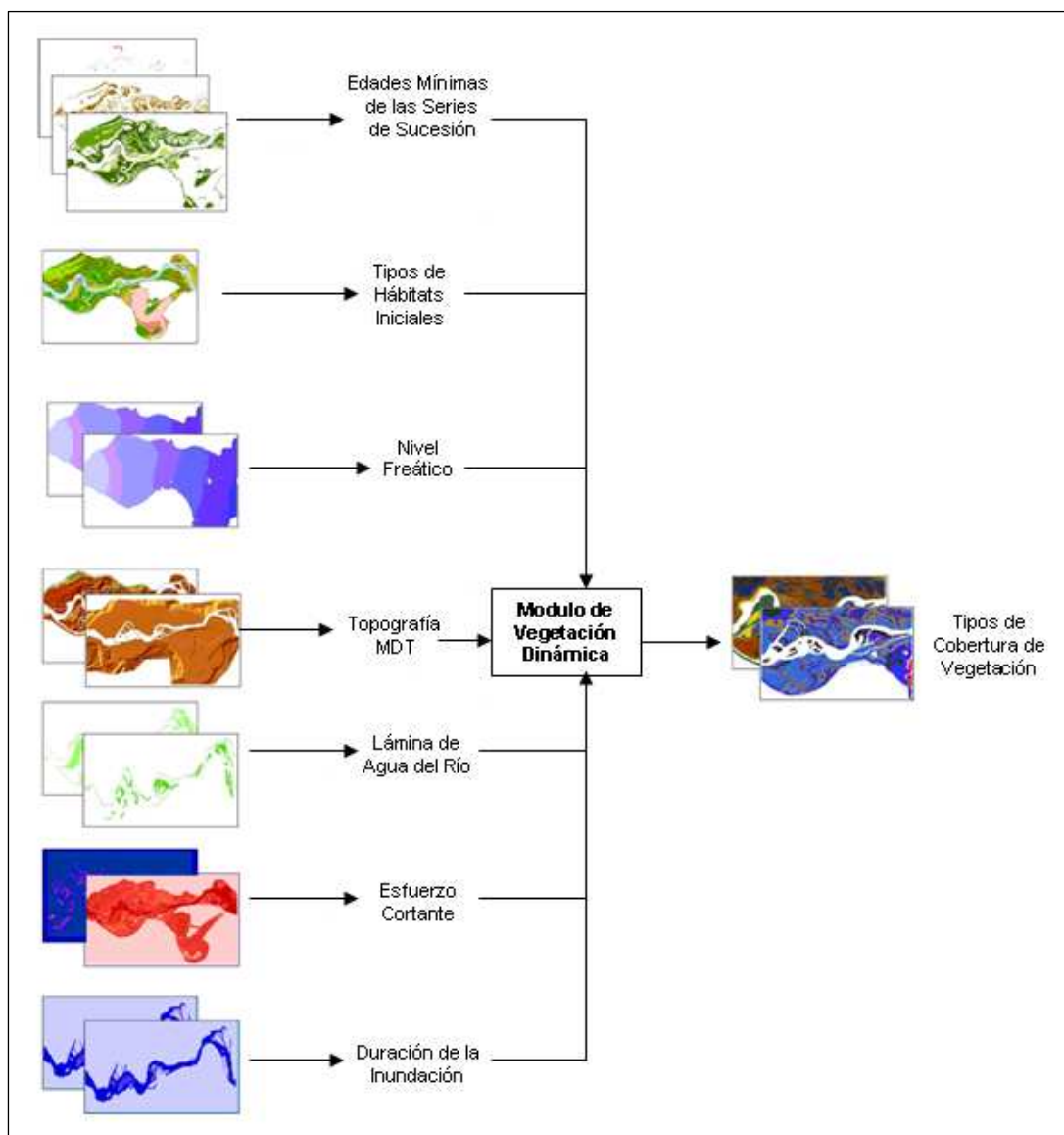
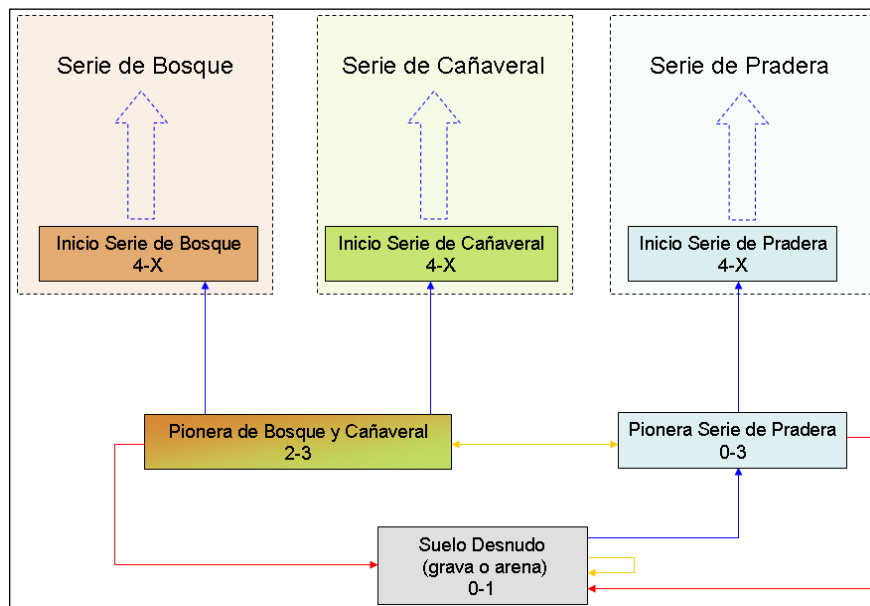


Figura 2.10. Mapas input y output de la componente de vegetación dinámica del modelo.

El componente dinámico se subdivide en cuatro módulos que se basan en sus funciones: reclutamiento, esfuerzo cortante, duración de la inundación y sucesión. Este componente también almacena la cantidad de píxeles (celdas) para cada comunidad vegetal en cada iteración de intervalo anual del modelo.

a. Módulo de Reclutamiento

Este módulo define el área de reclutamiento potencial y las áreas donde ocurre una perturbación debido a la influencia de factores hidráulicos y topográficos. En este modulo se establece una banda por encima del escenario del río que facilita el establecimiento de las semillas del bosque de ribera. La fase inicial solo es el área donde el reclutamiento tiene lugar y sigue una sucesión a las fases pioneras que tienen lugar al año siguiente. La vegetación pionera existente sigue una sucesión hacia la fase de matorral tanto de bosque o de cañaveral después de llegar a una cierta edad. La sucesión del bosque de ribera solo se lleva a cabo en una banda específica de alturas sobre el nivel de río (Ej. 0.6-2m). Por debajo de esta banda las plantas pioneras son destruidas y se vuelve a la fase inicial, mientras que por encima las plantas no se pueden desarrollar porque sus raíces en los estados iniciales no son lo suficiente largas para alcanzar el nivel freático. En otras circunstancias, si la vegetación existente tiene más edad que la edad de los pioneros, sobrevivirá a la perturbación debido a que se asume que dichas comunidades son lo suficientemente resistentes para resistir dichas perturbaciones. Las etapas iniciales de las series de sucesión que son afectadas por este proceso se muestran a continuación:



**Figura 2.11.** Diagrama con las etapas iniciales de sucesión que son afectadas por los procesos de perturbación por socavación y reclutamiento.

En la tabla siguiente se muestra las regresiones o sucesiones que tendrán lugar durante las etapas iniciales de sucesión debido al efecto del nivel del río (reclutamiento y socavación) sobre las zonas de ribera y de llanura de inundación.

**Tabla 2.9 Tabla resumen del efecto del nivel del río sobre las etapas iniciales de sucesión debido a los procesos de reclutamiento y socavación**

Zona	Intervalo de Alturas	Grava	Ultimo año pionero	Bosque ripario (Brp) y Cañaveral Pionero (Cp)	Pradera riparia Pionera (Prp)
BZ	BZ1	Permanece Grava	----->	Retrogresión a Grava	Retrogresión a Grava
	BZ2	Reclutado por (BrP y CP)	SI	Se convierte en Brp	Se convierte en Brp
			NO	Permanece igual	Reclutado por (Brp y Cp)
	NO /BZ1 Y NO /BZ2	Reclutado por (BrP y CP)	SI	Se convierte en Prp	Se convierte en Brp
NO			Permanece igual	Reclutado por (Brp y Cp)	
FZ	NO /FZ3	Reclutado por (BrP y CP)	SI	Se convierte en Prp	Se convierte en Brp
			NO	Permanece igual	Reclutado por (Brp y Cp)
	FZ3	Reclutado por (BrP y CP)	----->	Se convierte en Prp	Continua la Sucesion

BZ: Zona de Ribera  
 FZ: Zona de Llanura de Inundación  
 BZ1: Altura sobre el nivel del río en la cual ocurre una perturbación por socavación  
 BZ2: Altura sobre el nivel del río apta para el establecimiento del bosque de ribera  
 FZ3: Altura sobre el nivel del río apta para el reclutamiento de la pradera riparia  
 Brp: Bosque ripario pionero  
 Cp: Cañaveral pionero  
 Prp: Pradera riparia pionera

Además de los efectos hidráulicos, el módulo de reclutamiento considera cambios en la topografía del río. Cuando la topografía cambia parte de la zona de ribera (o de la llanura de inundación) puede pasar a ser zona acuática y viceversa. Los mismos principios son aplicables entre el resto de zonas adyacentes. Como se puede observar en la tabla siguiente, cuando la zona acuática (ZA) pasa a una de las zonas terrestres (BZ y FZ), se convierte en grava. Si por el contrario las zonas terrestres (BZ y FZ) se convierten en zona acuática (AZ), la vegetación se destruye tal como se puede ver en la tabla siguiente:

**Tabla 2.10 Tabla resumen del efecto del cambio de topografía sobre las series de sucesión.**

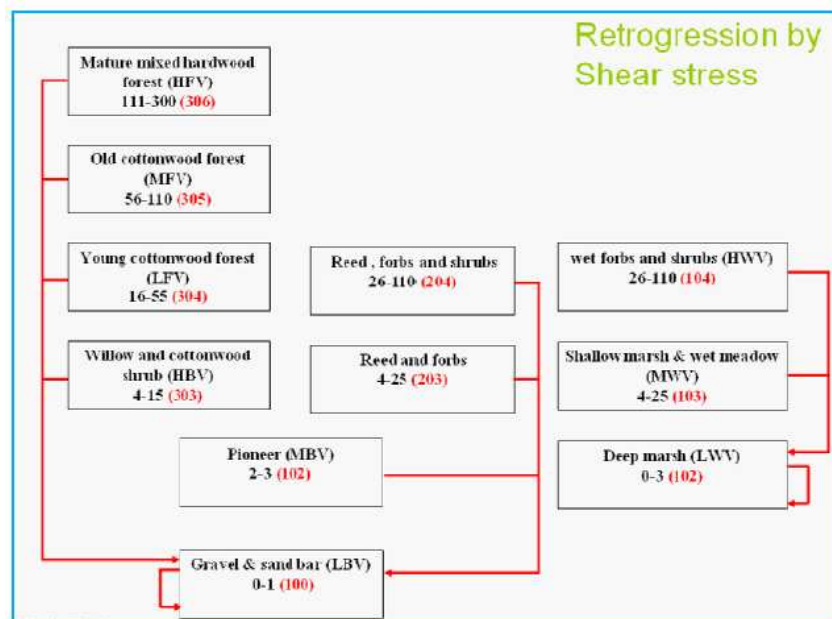
Vieja Topografía	Nueva Topografía (Zonas)		
	AZ	BZ	FZ
	AZ	Se mantiene AZ	Se convierte en Grava
BZ	Pierde su vegetación; se convierte en AZ	Mantiene su vegetación	Mantiene su vegetación
FZ	Pierde su vegetación; se convierte en AZ	Mantiene su vegetación	Mantiene su vegetación

AZ: Zona Acuática    BZ: Zona de Ribera    FZ: Zona de Llanura de Inundación



## b. Módulo de Esfuerzo Cortante

Se define al esfuerzo cortante como la fuerza por unidad de área aplicada por el agua a lo largo de las superficies en contacto. El esfuerzo cortante crítico puede ser utilizado como un indicador de la erosión de superficies. Es el máximo esfuerzo cortante que puede ser aplicado sobre una superficie sin causar erosión ni transporte de sedimentos y es dependiente del diámetro de las partículas. En la siguiente figura se puede observar que si se supera el esfuerzo crítico de una determinada etapa de sucesión, dicha vegetación retrocede directamente a la fase inicial de su serie.



**Figura 2.12.** Efecto del esfuerzo cortante sobre las series de vegetación del tramo de estudio del río Kootenai (EE.UU.)

Una combinación de la cota del agua y el esfuerzo cortante es una de las combinaciones más importantes de parámetros físicos que lleva a la erosión y deposición de sedimentos en las llanuras de inundación y por ello es responsable de sus procesos geomórficos.

Este módulo considera que si la vegetación existente es destruida por el esfuerzo cortante, se retrocede directamente a la fase inicial. La vegetación es destruida cuando el esfuerzo cortante es superior al esfuerzo cortante crítico de una determinada comunidad vegetal. El valor crítico del esfuerzo cortante de cada tipo de vegetación es además variable según la edad que marca la etapa de la sucesión.

### c. Módulo de Duración de la Inundación

La duración de la inundación es la cantidad de tiempo en la que una zona riparia queda anegada durante una determinada estación o año. El anegamiento mata a las plantas por el efecto anaeróbico. Los efectos anaeróbicos son más importantes en los ambientes en los que el anegamiento permanece en periodos prolongados de tiempo. El anegamiento a largo plazo causa también un estrés fisiológico a la planta que se asocia con el agotamiento de oxígeno de la zona radicular. Se considera que si la duración de la inundación es más larga que el valor crítico entonces la vegetación retrocede a estados anteriores de la serie. La intensidad de la alteración se clasifica en tres rangos: alto, medio y bajo. La clasificación depende en el número de días inundados durante el periodo vegetativo. El tipo de comunidad que surja tras una regresión depende de la intensidad de la alteración y del tipo de vegetación preexistente. Incluso las especies más tolerantes a la inundación necesitan por los menos estar un 55-60 por ciento del periodo vegetativo sin anegamiento. El número de días que corresponden a cada intervalo de intensidad de inundación se determina mediante reglas de experto. El siguiente menú es el que utiliza el software de Ripflow para determinar los días de inundación para cada intensidad (baja, media, alta):

The image shows three stacked windows for defining expert rules. Each window has a title bar and a table with two columns: 'Old values' and 'New values'. Below each table are several control buttons.

Old values	New values
0 - 89	NoData
89 - 119	1
119 - 366	NoData
NoData	NoData

Buttons: Classify..., Unique, Add Entry, Delete Entries, Load..., Save..., Reverse New Values, Precision...

Old values	New values
0 - 119	NoData
119 - 150	1
150 - 366	NoData
NoData	NoData

Buttons: Classify..., Unique, Add Entry, Delete Entries, Load..., Save..., Reverse New Values, Precision...

Old values	New values
0 - 149	NoData
149 - 366	1
NoData	NoData

Buttons: Classify..., Unique, Add Entry, Delete Entries, Load..., Save..., Reverse New Values, Precision...

**Figura 2.13.** Menú que utiliza Ripflow para introducir la relación entre el número de días de inundación y la intensidad de los consiguientes impactos (bajo, medio y alto).

El efecto de la inundación siempre una implica regresión en la serie, pero dependiendo de la intensidad dicha regresión hará retroceder más o menos la vegetación dentro de la serie. A continuación se muestra una figura donde se muestran los efectos causados por la inundación en una serie de vegetación de un tramo de estudio en el río Kootenai (EE.UU.).

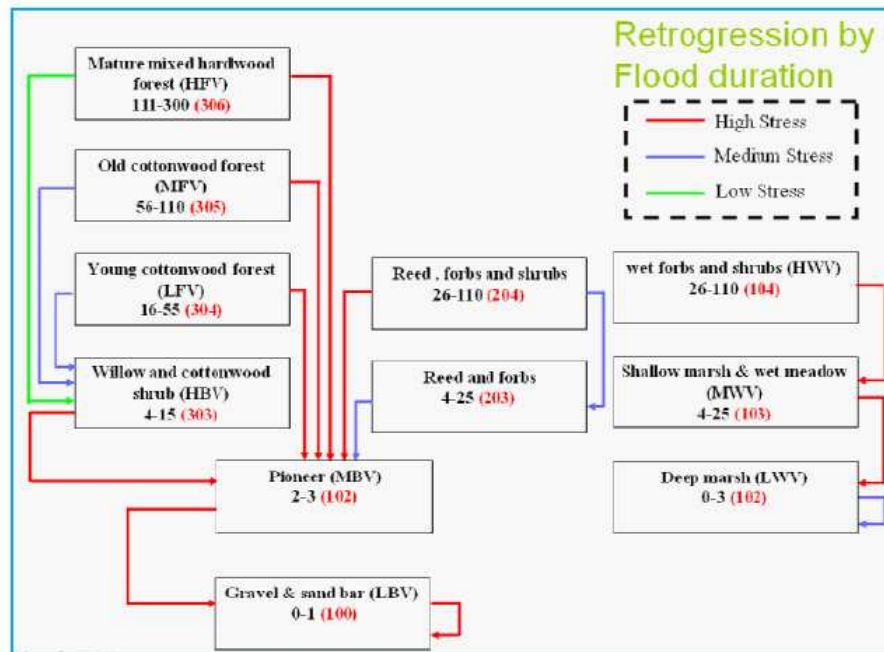
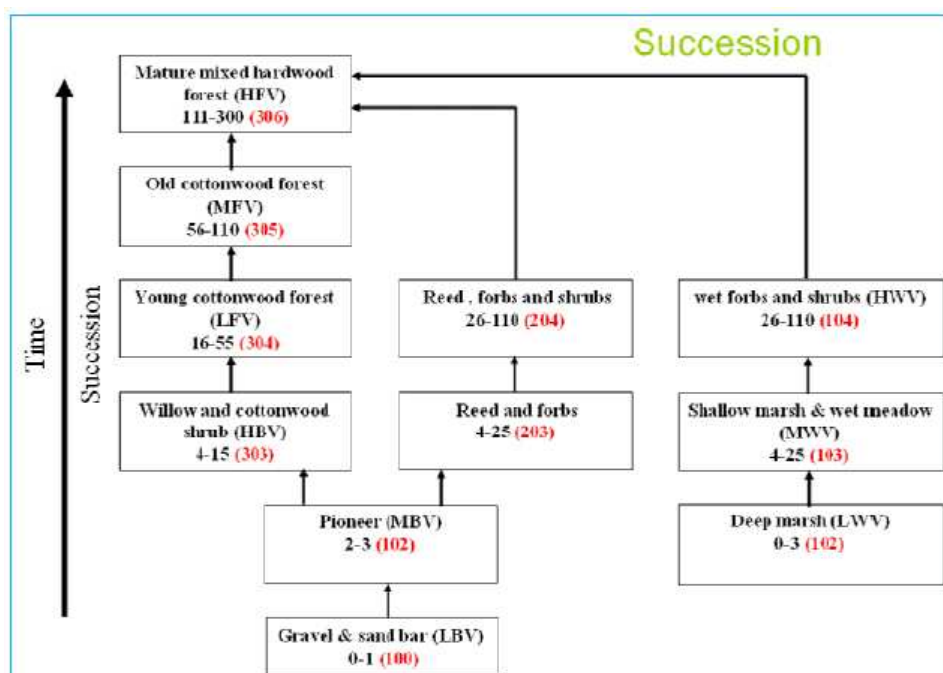


Figura 2.14. Efecto de la duración de la inundación sobre las series de vegetación del río Kootenai (EE.UU.)

d. Módulo del Progreso de la Sucesión

Este módulo asume que si en una determinada zona no ocurren las alteraciones por: socavación, condiciones de suelo seco (solo en Ripflow versión 3), duración de inundación considerable, o esfuerzo cortante superior al crítico; entonces se seguirá una pauta de sucesión hacia un estado maduro, basándose en el rango de edades de cada estado de sucesión. Se asume además que las series de pradera y de cañaveral evolucionan hacia un bosque mixto maduro después de una determinada cantidad de años, que es específica para cada tramo de estudio.

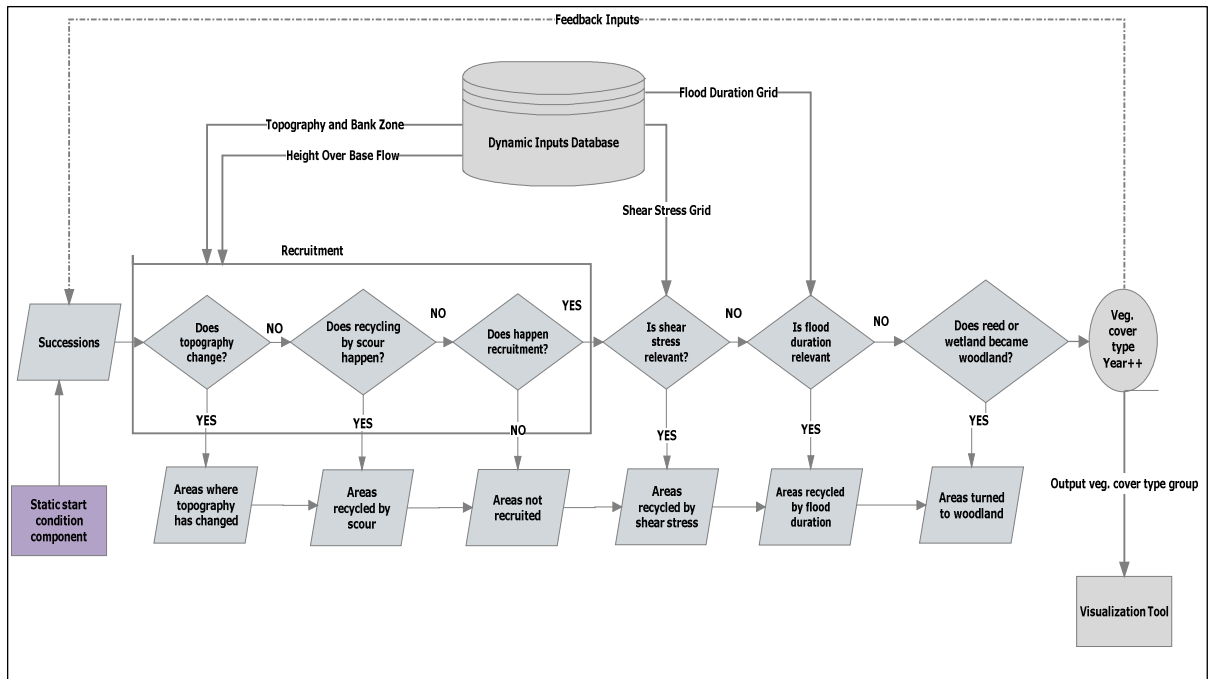


**Figura 2.15.** Se puede observar en esta figura que en el tramo de estudio del río Kootenai (EE.UU.), las 3 series de sucesión (*bosque, cañaveral y pradera*) evolucionan a la etapa *clímax* (*Bosque Maduro Mixto*) cuando tienen 110 años sin sufrir alteraciones significativas.

1.1.1.1.- Tercera Componente del Modelo: Vistas Output

Esta última componente no realiza ninguna tarea computacional, tan solo se utiliza para representar el output del modelo con una leyenda unificada por el usuario. El output principal es un Mapa de Vegetación Potencial para cada año de la simulación.

El procedimiento de cálculo de Ripflow comienza con la “condición estática inicial” que solo se ejecuta una vez por simulación en lo que podría considerarse el año 0. Posteriormente se entra en la “componente dinámica” del modelo donde aparece un bucle de intervalo anual donde el número de repeticiones es equivalente al número de años simulados. En cada simulación anual y para cada celda se pasará por cada uno de los módulos y submodelos, en caso de que los procesos simulados sean representativos. Cada uno de esos módulos requerirá para cada intervalo anual de determinados mapas que son agrupados en un archivo que se denomina “Dynamic Inputs Database” (*Base de Datos Dinámica de Inputs*). Al final de cada año de la simulación las coberturas de vegetación serán probablemente modificadas. Esta nueva distribución de coberturas retroalimentará a modo de “feedback” las condiciones iniciales de la simulación del año siguiente. Una vez que la simulación de todos los años haya sido realizada se pasará al tercer y último componente del modelo (Vistas Output) que se dedicará a reorganizar los mapas con leyendas y criterios comunes.



**Figura 2.16.** Diagrama general de flujos del modelo Ripflow.

### 2.3.- Coeficiente Kappa de Cohen

El coeficiente Kappa de Cohen evalúa el grado de aciertos de la simulación de un modelo. Se utiliza con elementos cualitativos y tiene en cuenta el efecto producido por el azar. Específicamente para este modelo se evalúa el grado de semejanza entre la vegetación observada y la simulada. Para llevar a cabo esta evaluación, el algoritmo construirá en primer lugar una matriz de confusión donde las columnas representan los elementos observados ( $m_1, m_2, m_3$ ) y las filas los elementos simulados ( $n_1, n_2, n_3$ ). Cada celda del tramo de estudio comparará el tipo de vegetación del mapa observado con el tipo de vegetación del mapa simulado, incrementando en 1 el valor de la celda de la tabla para la correspondiente fila y columna. El número de filas y el número de columnas será igual al número de elementos de vegetación. A modo de ejemplo se va a asumir que un cierto tramo de estudio tiene tipos de vegetación que producen la siguiente matriz de confusión:

**Tabla 2.11 Matriz de Confusión de 3 tipos de vegetación utilizada para evaluar el coeficiente de Cohen.**

		Observed			
		$m_1$	$m_2$	$m_3$	
Simulated	$n_1$	$V_{11}$	$V_{21}$	$V_{31}$	$\sum V_{m,1}$
	$n_2$	$V_{12}$	$V_{22}$	$V_{32}$	$\sum V_{m,2}$
	$n_3$	$V_{13}$	$V_{23}$	$V_{33}$	$\sum V_{m,3}$
		$\sum V_{1,n}$	$\sum V_{2,n}$	$\sum V_{3,n}$	$\sum V_{m,n}$

Después de construir la matriz, se calculan los siguientes sumatorios:

- Suma de los valores de cada **fila** (e.j.  $\sum V_{m,1} = V_{11} + V_{21} + V_{31}$ )
- Suma de los valores de cada **columna** (e.j.  $\sum V_{1,n} = V_{11} + V_{12} + V_{13}$ )
- Suma de los valores de la **diagonal principal** ( $\sum V_{m,n} = V_{11} + V_{22} + V_{33}$ ), que coincide con el número de aciertos. Para evitar confusiones con la notación de los sumatorios llamaremos a esta variable: **(fo)**.
- Suma de todos los valores de la matriz ( $\sum V_{m,n} = V_{m,1} + V_{m,2} + V_{m,3}$ ) ó ( $\sum V_{m,n} = V_{1,n} + V_{2,n} + V_{3,n}$ ). Para evitar confusiones con la notación de los sumatorios llamaremos a esta variable: **(N)**.

Posteriormente se calculan las frecuencias esperadas  $fe_{(i)}$  de cada tipo de vegetación. Esto se calcula al multiplicar: [la suma de los valores de cada **fila** que coincide con el tipo de vegetación] por [la suma de los valores de cada **columna** que coincide con el tipo de vegetación] y dividiendo entre [la suma de los valores de la matriz **(N)**].

- para la vegetación 1:

$$fe_{(2)} = \frac{\sum V_{m,2} \cdot \sum V_{2,n}}{N}$$

- para la vegetación 2:

- para la vegetación 3:

$$fe_{(3)} = \frac{\sum V_{m,3} \cdot \sum V_{3,n}}{N}$$

A continuación se realiza también la suma de todas las frecuencias esperadas:

$$fe = \sum fe_{(i)} = fe_{(1)} + fe_{(2)} + fe_{(3)}$$

Por consiguiente ya tenemos los valores de todas las variables ( $f_o$ ,  $f_e$ ,  $N$ ) requeridos para obtener el coeficiente Kappa de Cohen ( $k$ ), que se obtiene a través de la ecuación siguiente:

$$k = \frac{f_o - f_e}{N - f_e}$$

Si el número total de valores ( $N$ ) de la matriz es alto, entonces se puede asumir que pertenece a una distribución normal:

- Error Estándar ( $\sigma_x$ ): 
$$\sigma_x = \frac{\sqrt{f_o \cdot \left(1 - \frac{f_o}{n}\right)}}{N - f_e}$$

- Intervalo de confianza del 95%:  $k \pm 1,96 \cdot \sigma_x$

- Intervalo de confianza del 99%:  $k \pm 2,58 \cdot \sigma_x$

El valor máximo del coeficiente Kappa ( $k$ ) es 1, y el valor mínimo es 0. Si  $k=1$  entonces todos los casos tienen lugar en la diagonal principal y el resto de las celdas de la matriz de confusión quedarán vacías. Si  $k=0$  significa que no hay ningún grado de aciertos no aleatorios entre los elementos observados y simulados. El rango de valores del coeficiente Kappa se clasifican de la siguiente manera.

- $0.80 < k < 1$  : Excelente
- $0.60 < k < 0.80$  : Bueno
- $0.40 < k < 0.60$  : Aceptable

- $0.20 < k < 0.40$  : Inadecuado
- $0 < k < 0.20$  : Inaceptable

### **3.- DISEÑO DEL SOFTWARE**



### **3.1.- MODELO DE HUMEDAD DE SUELO “RIBAV”**

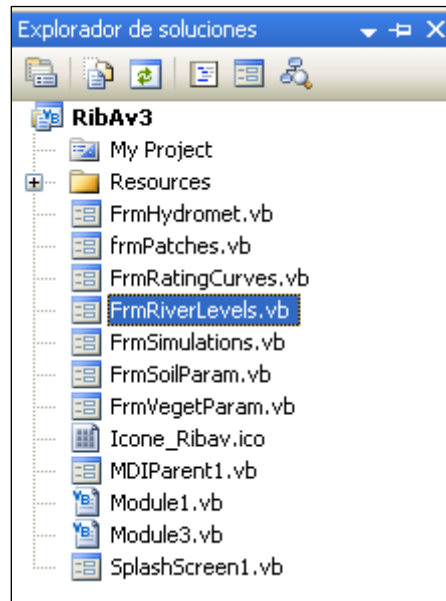
#### **3.1.1.- Plataforma y características generales**

El modelo Ribav se ha implementado en la plataforma Visual Studio 2008 con el lenguaje de programación Visual Basic.net. Se ha escogido esta herramienta porque permite desarrollar con facilidad interfaces graficas de usuario que facilitan bastante el manejo del software por parte del usuario. Adicionalmente la plataforma Visual Studio permite la utilización de varios lenguajes de programación dentro de un mismo proyecto o solución. Esto es bastante útil en los casos en los que se desean combinar porciones de código que han sido desarrollados por personas diferentes con lenguajes de programación diferentes pero que sí son compatibles con la plataforma .net. Algunos de los lenguajes compatibles con .Net y por tanto con Visual Basic.net son: C#, C++, P# (implementación de Prolog), Ironpython, Visual Fortran, etc.

La ventaja de utilizar en concreto el lenguaje Visual Basic.net es que su sintaxis es bastante intuitiva, pareciéndose mucho a la del inglés, por lo que su código puede ser leído, revisado y modificado por personas que no tengan muchos conocimiento de programación. La primera versión del lenguaje BASIC fue desarrollada a principio de la década de los 60 por Thomas Kurtz y John Kemeny con la finalidad de que profesionales sin base previa en desarrollo de software pudieran programar, por lo que de ello viene su nombre: *Beginners All-purpose Symbolic Instruction Code*. Otra ventaja de utilizar una versión moderna de Visual Basic (como es la 2008), es que está enteramente orientada a objetos. La programación orientada a objetos trae ciertas ventajas como facilitar el control en el desarrollo de aplicaciones grandes, las cuales normalmente están desarrolladas por varias personas. En nuestro caso, además nos permitirá unir este software con el programa Ripflow que ha sido desarrollado con el language Python 2.4 dentro de la plataforma “Model Builder” de Esri ArcGis 9.2.

#### **3.1.2.- Estructura del programa**

El código fuente del programa ha sido estructurado en formularios o módulos. Estos formularios y módulos son los distintos archivos en los que está almacenado el código. Un formulario (form en inglés) se diferencia de un modulo en que el formulario además de almacenar los algoritmos de cálculo (lógica del programa) también almacenará el código que determina la representación gráfica de las distintas ventanas del programa (interfaz gráfica de usuario).



**Figura 3.17.** Explorador de soluciones de Visual Studio 2008, que muestra los distintos formularios y módulos del programa.

Visual Basic además de ser un lenguaje orientado a objetos también es un lenguaje estructurado por lo que el código de dentro de los distintos módulos y formularios estará también agrupado en métodos (funciones y subrutinas). Esta segunda agrupación facilita la organización y comprensión del código facilitando tanto el diseño del programa como su posterior búsqueda/corrección de errores y el desarrollo de nuevas versiones y ampliaciones.

```

        'print all column values for a row
        For indexCols As Integer = 0 To numCols - 1
            tw.Write(tableRiverLevels.Rows(indexRows).Cells(indexCols).Value)
            If (indexCols <> numCols) Then
                tw.Write(";")
            End If
        Next
        tw.WriteLine()
    Next
End Using
End Sub

Private Sub btLoad_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btLoad.Click
    Dim reply As DialogResult = MessageBox.Show("Do you want to save the current data?", "", _
        MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question, MessageBoxDefaultButton.
        Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call LoadValues()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSave.PerformClick()
        'And now after Saving the data we Clear the Textboxes
        Call LoadValues()
    End If
End Sub

Private Sub LoadValues()

    Dim RiverLevelsFileName As String = ""
    OpenFileDialog1.InitialDirectory = (System.Environment.CurrentDirectory + "\data\")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files (*.*)|*.*"
    ' OpenFileDialog1.Filter = "Text Files (*.txt)|*.txt|All Files (*.*)|*.*"

    'OpenFileDialog1.FilterIndex = 2
    OpenFileDialog1.RestoreDirectory = True

```

**Figura 3.18.** Porción del código del programa. Nótese como las subrutinas quedan definidas dentro de instrucciones “Private Sub ...” y “End Sub”, en donde Visual Studio automáticamente traza una recta horizontal.

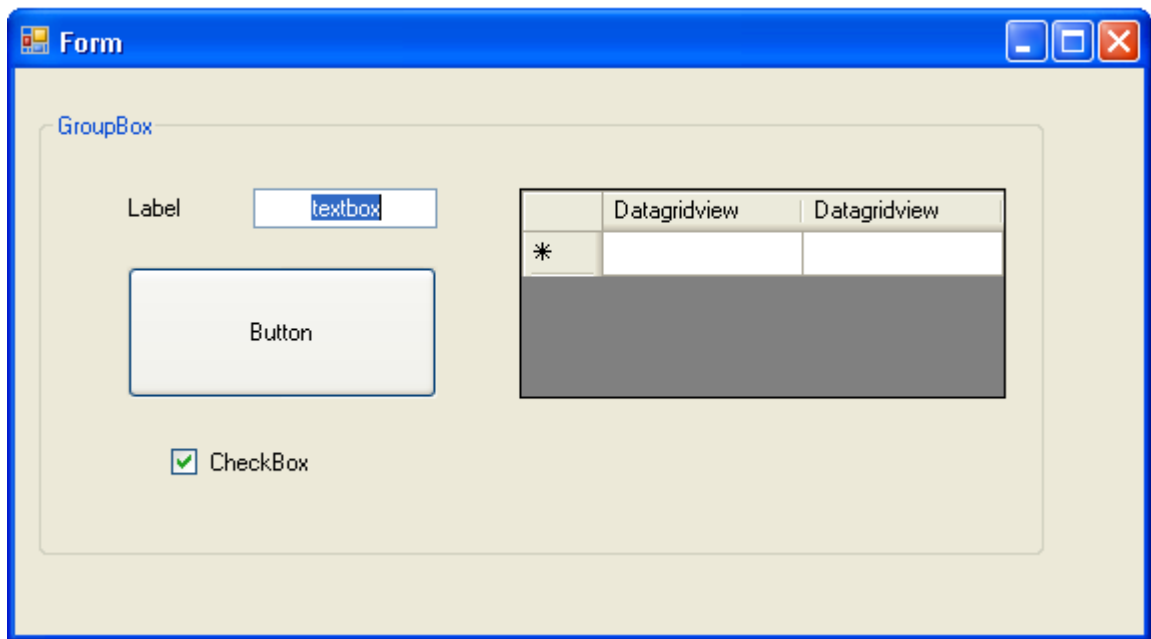
Los métodos y variables que intervienen en el programa podrán ser también públicas o privadas. Los métodos y variables públicas podrán ser utilizados desde cualquier parte del programa siendo definidos en Visual Basic por la sintaxis “Public” o “Friend”. Por otro lado las privadas solo podrán ser utilizadas en determinadas partes del programa y serán definidos por “Dim” (para variables) o “Private” (para métodos). Esta determinación del alcance de las variables y métodos se desarrolla plenamente en la programación orientada a objetos mediante la “Encapsulación”. Esto favorece el control del código y previene la aparición de errores en programas de cierta envergadura.

### 3.1.3.- Interfaz gráfica de usuario

Se ha desarrollado una interfaz gráfica con el fin de que los usuarios del modelo puedan manejar el software de una forma fácil e intuitiva. Para facilitar el diseño de la interfaz gráfica de usuario, Visual Studio dispone de una “Barra de Herramientas” y de una “Barra de Propiedades”. La barra de herramientas permite la selección de los distintos controles que van a ser colocados sobre cada formulario. Una vez seleccionado el control, puede ser colocado en cualquier parte del formulario y

se le puede designar las dimensiones deseadas con relativa facilidad. Los controles más habituales utilizados en el programa aparte de los formularios son:

- Button (Botón): Es el control más habitual para llamar a las subrutinas de los eventos en tiempo de ejecución.
- Textboxes (Cajas de Texto): Estos controles se utilizan para introducir pequeñas cantidades de datos que darán valores a las variables. Si por el contrario tienen la propiedad "ReadOnly" con valor = "True", entonces tan solo se utilizarán para visualizar resultados.
- Label (Etiquetas): Normalmente no intervienen en la lógica del programa, teniendo solo carácter informativo y estético. Aun así en algunos casos podrían ser utilizados para visualizar resultados al igual que los Textboxes.
- Checkboxes (Cajas de Selección): Son unos iconos de forma cuadrada que suelen ser activados o desactivados por el usuario. Son utilizadas para gestionar estructuras de control condicional con respuesta booleanas (de lógica binaria).
- Groupboxes (Cajas de Agrupación): Tienen carácter estético principalmente agrupando controles relacionados entre sí.
- DataGridView (Tablas de Datos): Son controles utilizados para la introducción/visualización de gran cantidad de datos. Estas tablas tienen el mismo formato que muchas tablas de base de datos, donde las columnas representan los campos y las filas los registros.



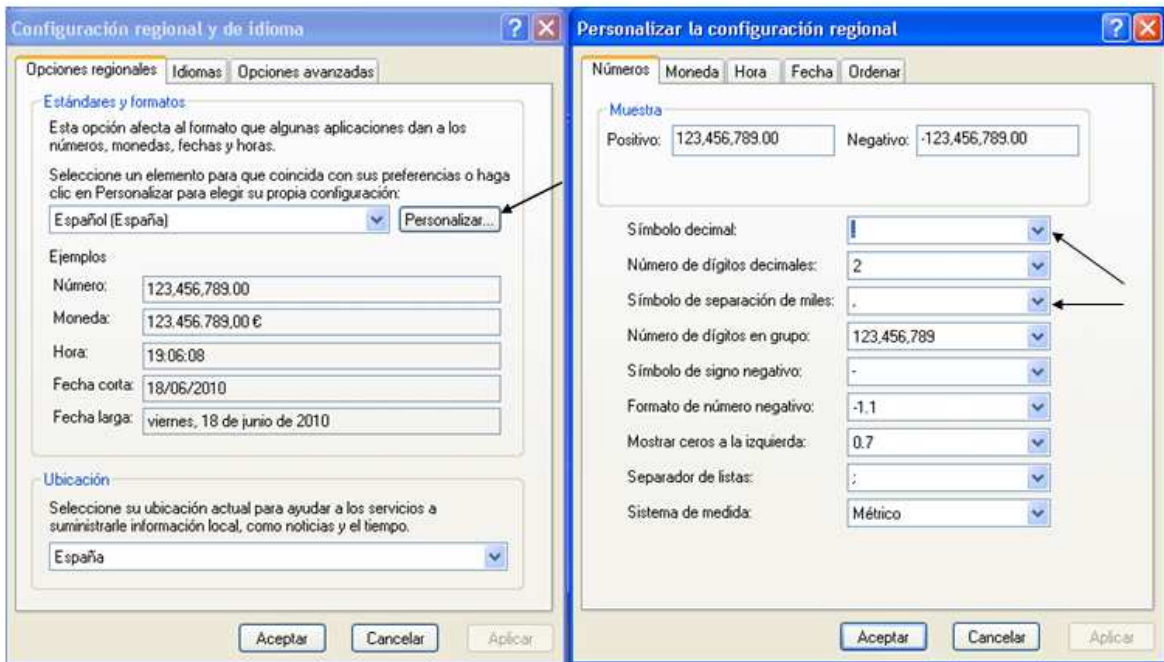
**Figura 3.19.** Ejemplo de controles utilizados más comúnmente en la interfaz gráfica de usuario del Software de Ribav 1D.

### **3.1.4.- Utilización del Software de Ribav 1D**

#### **3.1.4.1.- Instalación del software y operaciones previas**

Los archivos que componen el programa son situados en una carpeta comprimida. Para utilizar el programa habría primeramente que descomprimir dichos archivos para obtener el directorio del programa. El programa utiliza rutas de archivos relativas en vez de absolutas. Por lo tanto podrá colocarse el directorio del programa en cualquier parte de cualquier unidad de disco duro o de almacenamiento donde este habilitada la escritura. Aun así hay que tener en cuenta que el sistema operativo Windows XP limita la longitud de las rutas de archivo a 256 caracteres.

Antes de ejecutar el programa hay que cambiar el separador decimal del sistema a operativo a punto (.) en vez de coma (,). Esto se realiza accediendo a la aplicación de Windows XP de: "Inicio->Configuración->Panel de Control->Configuración Regional y de Idioma". Una vez abierta dicha aplicaron se pulsa el botón "Personalizar" de la pestaña "Opciones Regionales" que aparece por defecto. Al pulsar ese botón aparecerá un nuevo formulario donde se deberá poner "." en la caja de separador decimal y "," en la caja de separador de miles, antes de pulsar el botón "Aplicar"



**Figura 3.20.** Opciones a las cuales hay que acceder dentro de la aplicación “Configuración regional y de idioma” dentro del “Panel de Control” de Windows XP para introducir el punto (.) como separador decimal en caso de no estarlo con anterioridad.

### 3.1.4.2.- Carpetas y archivos del programa

Al abrir el directorio del programa se encuentran 3 archivos y un subdirectorio. De los 3 archivos uno es el archivo ejecutable (Ribav3.exe), otro es una base de datos de depuración (Ribav3.pdb) y el tercero es un archivo XML (Lenguaje de Marcas Extensibles) llamado (Ribav3.xml)



**Figura 3.21.** Componentes del directorio principal del software de Ribav 1D.

Dentro del subdirectorio “Data” se encuentran otras dos subcarpetas: la carpeta “Default” y la

carpeta "Proyectos". La carpeta "Proyectos" almacena los archivos tanto de entrada como de salida, producidos en cada simulación del programa dentro de cada proyecto. Por otro lado la carpeta "Default" almacena los archivos que el programa cargará por defecto cada vez que éste se ejecute. Estos archivos pueden ser muy útiles cuando se desea utilizar el programa de forma continuada, realizando pocos cambios tras una determinada ejecución. Un ejemplo de este uso sería el de un proceso de calibración manual. Los archivos que son cargados por defecto en dicha carpeta son los siguientes:

- "Hydromet.csv": Datos hidro-meteorológicos
- "RatingCurves.csv": Curvas de gasto
- "DefaultSimulationPoints.csv": Datos relativos a los punto de simulación
- "VegetationParameters.csv": Parámetros de vegetación
- "SoilParameters.csv": Parámetros de suelo

El formato que deberán tener dichos archivos se explica en el apartado siguiente.



**Figura 3.22.** Archivos de entrada (inputs) que se cargan por defecto al iniciar el programa y que están situados en la subcarpeta "/Data/Default".

#### 1.1.1.2.- Preparación previa de los archivos de entrada (inputs)

Antes de ejecutar el programa hay que preparar los ficheros de entrada o inputs del modelo. En todos los archivos inputs no se deben incluir cabeceras. Estos incluirán el punto "." como separador decimal y el punto y coma ";" como separador de columnas o campos. Para facilitar la visualización y/o manejo de estos ficheros se recomienda utilizar además de editores de texto como "WordPad" y también hojas de cálculo como "Oppenoffice Calc" o "MS Excel". Estos archivos deberán ser guardados en formato CSV "Comma Separated Values" (*delimitados por comas o punto y coma*).

a) Fichero de datos Hidro-meteorológicos (“Hydromet.csv”)

En este fichero se ubican de izquierda derecha los siguientes campos (columnas):

- Fecha (dd/mm/aaaa)
- Precipitación Diaria (mm)
- Evapotranspiración Potencial (mm)
- Caudal Diario del tramo de estudio (m3/s)

C. DAILY HYDROMETEOROLOGICAL INPUTS			
Date	Daily Precipitation [mm]	Potential Evapotranspiration [mm]	Daily Flow [m3/s]
	c.1	c.2	c.3
	PP	ETP	Q
01/01/1999	1.700	1.174	0.791
02/01/1999	0.717	1.17	0.783
03/01/1999	0	1.192	0.718
04/01/1999	0	1.172	0.672
05/01/1999	0	1.198	0.728
06/01/1999	0	1.18	0.865
07/01/1999	0	1.182	1.145
d	.....	.....	.....

01/01/1999;1.7004;1.174;0.586
02/01/1999;0.7167;1.17;0.578
03/01/1999;0;1.192;0.573
04/01/1999;0;1.172;0.569
05/01/1999;0;1.198;0.564
06/01/1999;0;1.18;0.573
07/01/1999;0;1.182;0.558
08/01/1999;0;1.178;0.558
09/01/1999;0;1.181;0.565
10/01/1999;0;1.155;0.554
11/01/1999;0;1.12;0.558
12/01/1999;0;0.721;0.559
13/01/1999;0;1.401;0.564
14/01/1999;0.4962;1.057;0.554
15/01/1999;0.9924;1.165;0.555
16/01/1999;0;1.382;0.556
17/01/1999;2.3352;1.107;0.551
18/01/1999;0;1.014;0.549
19/01/1999;0;1.131;0.546
20/01/1999;0;1.107;0.555
21/01/1999;0;1.029;0.551

**Figura 3.23.** Archivos de datos hidro-meteorológicos con cabeceras en una hoja de cálculo (izquierda) y en formato CSV de Ribav1D (derecha).

b) Fichero de Parámetros de Vegetación (“VegetationParameters.csv”)

En este fichero se ubican de izquierda derecha los siguientes campos (columnas):

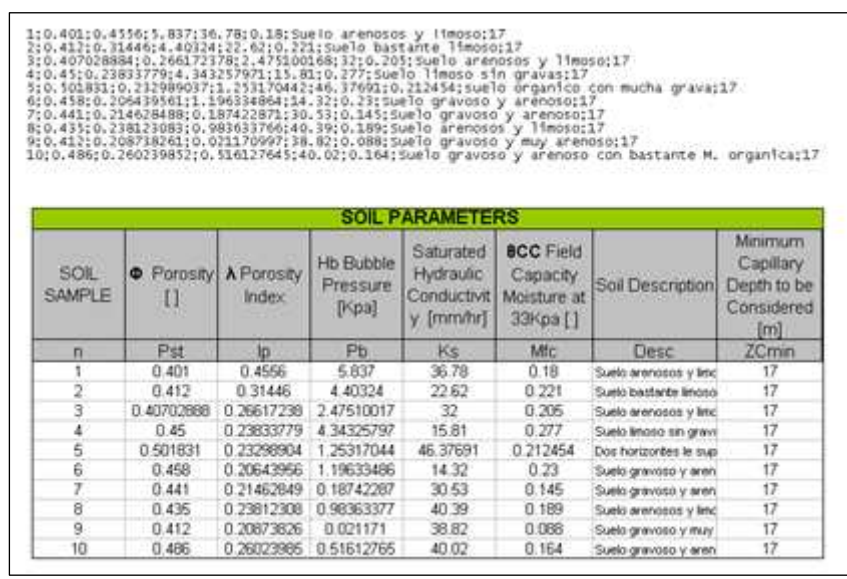
- n: Clave numérica de la tabla que indica el numero de fila (número entero)
- Veg: Código alfanumérico que indica el tipo de vegetación a parametrizar no interviniendo en los cálculos (cadena de caracteres)
- Zr: Profundidad Máxima de Raíces (m)
- Ze: Profundidad Efectiva de Raíces (m)
- Zsat: Profundidad de Extinción por Saturación (m)
- Ri: Factor de Transpiración de la Zona No Saturada ( )
- Rj: Factor de Transpiración de la Zona Saturada ( )



- Crt: Conductividad Máxima de Agua Raíz-Suelo (mmMpa-1h-1)
  - Descr: Descripción de la vegetación a parametrizar no interviniendo en los cálculos (cadena de caracteres)
  - Cov1: Fracción de Cobertura de las Plantas en Enero ( )
  - Cov2: Fracción de Cobertura de las Plantas en Febrero ( )
  - Cov3: Fracción de Cobertura de las Plantas en Marzo ( )
  - Cov4: Fracción de Cobertura de las Plantas en Abril ( )
  - Cov5: Fracción de Cobertura de las Plantas en Mayo ( )
  - Cov6: Fracción de Cobertura de las Plantas en Junio ( )
  - Cov7: Fracción de Cobertura de las Plantas en Julio ( )
  - Cov8: Fracción de Cobertura de las Plantas en Agosto ( )
  - Cov9: Fracción de Cobertura de las Plantas en Septiembre ( )
  - Cov10: Fracción de Cobertura de las Plantas en Octubre ( )
  - Cov11: Fracción de Cobertura de las Plantas en Noviembre ( )
  - Cov12: Fracción de Cobertura de las Plantas en Diciembre ( )
- c) Fichero de Parámetros de Suelo ("SoilParameters.csv")

En este fichero se ubican de izquierda derecha los siguientes campos (columnas):

- n: Clave numérica de la tabla que indica el numero de fila (número entero)
- Pst: Porosidad ( )
- Ip: Índice de Porosidad ( )
- Pb: Presión de Burbujeo (Kpa)
- Ks: Conductividad Hidráulica Saturada del Suelo (mm/hr)
- Mfc: Humedad a Capacidad de Campo ( )
- Descr: Descripción de la vegetación a parametrizar no interviniendo en los cálculos (cadena de caracteres)
- Zcmin: Profundidad a considerar en el Ascenso Hidráulico Capilar (m)



**Figura 3.24.** Archivos de parámetros de suelo en formato CSV de Ribav1D (*arriba*) y con cabeceras en una hoja de cálculo (*abajo*).

d) Fichero de Curvas de Gasto (“RatingCurves.csv”)

Se considera que el nivel freático es horizontal al nivel del río. En este fichero se ubica en los campos (columnas) impares los caudales y en los pares los niveles freáticos o de río y todo ello para cada uno de los transectos que forman parte los tramos de estudio. A modo de ejemplo el orden de las columnas de dicho fichero quedaría de la siguiente forma:

- Caudal del Transecto 1 (m<sup>3</sup>/s)
- Nivel de agua del Transecto 1 (m)
- Caudal del Transecto 2 (m<sup>3</sup>/s)
- Nivel de agua del Transecto 2 (m)
- Caudal del Transecto N (m<sup>3</sup>/s)
- Nivel de agua del Transecto N (m)

Transecto 1		Transecto 2		Transecto 3	
Q1(m3/s)	H1(m)	Q2 (m3/s)	H2 (m)	Q3 (m3/s)	H3 (m)
0	228.62	0	228.59	0	228.71
0.001	228.65	0.001	228.62	0.001	228.74
0.003	228.68	0.002	228.65	0.003	228.77
0.005	228.71	0.003	228.68	0.005	228.8
0.009	228.74	0.005	228.71	0.009	228.83
0.014	228.77	0.008	228.74	0.014	228.86
0.021	228.8	0.013	228.77	0.021	228.89

```

0; 228.62; 0; 228.59; 0; 228.71; 0; 228.68; 0;
0.001; 228.65; 0.001; 228.62; 0.001; 228.74
0.003; 228.68; 0.002; 228.65; 0.003; 228.77
0.005; 228.71; 0.003; 228.68; 0.005; 228.8;
0.009; 228.74; 0.005; 228.71; 0.009; 228.83
0.014; 228.77; 0.008; 228.74; 0.014; 228.86
0.021; 228.8; 0.013; 228.77; 0.021; 228.89;
0.031; 228.83; 0.019; 228.8; 0.031; 228.92;
0.044; 228.86; 0.027; 228.83; 0.044; 228.95
0.061; 228.89; 0.037; 228.86; 0.061; 228.98
0.082; 228.92; 0.051; 228.89; 0.082; 229.01
0.107; 228.95; 0.068; 228.92; 0.108; 229.04
0.139; 228.98; 0.088; 228.95; 0.141; 229.07

```

**Figura 3.25.** Archivos de curvas de gasto con cabeceras en una hoja de cálculo (*arriba*) y en formato CSV de Ribav1D (*abajo*).

e) Fichero de Puntos de Simulación ("RatingCurves.csv")

En este fichero se ubican de izquierda derecha los siguientes campos (columnas):

- Código punto de simulación.
- Cota absoluta en metros sobre el nivel del mar del punto de simulación (m).
- Tipo de suelo (clave numérica de la tabla de suelos)
- Transecto al que pertenece el punto de simulación y que lo relacionará con la correspondiente curva de gasto.
- Distancia horizontal del punto de simulación con respecto al Talweg del cauce (*solo tiene carácter informativo y no participa en los cálculos*).
- Código del parche de vegetación al que pertenece el punto (*solo tiene carácter informativo y no participa en los cálculos*).
- Acrónimo del tipo de vegetación.
- Descripción del tipo de vegetación observado (*solo tiene carácter informativo y no participa en los cálculos*).

- Especies observadas en el parche (*solo tiene carácter informativo y no participa en los cálculos*).

Código Punto Simulación	Cota Punto (m)	Tipo de Suelo	Transecto	Distancia Horizontal Punto (m)	Parche al que pertenece el Punto	Acrónimo de Vegetación Observada en el Parche	Descripción de Vegetación Observada	Especies mas Representativas del Parche
1_1_54	236.133	4	1	-33.197	54	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)
1_2_54	232.898	4	1	-28.197	54	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)
1_3_54	231.45	4	1	-23.197	54	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)
1_4_21	230.722	4	1	-18.197	21	RA+TV	Riparian Trees or Big Shrut	Parche evolucionado, entrada de pino de ladera
1_5_21	230.431	4	1	-13.197	21	RA+TV	Riparian Trees or Big Shrut	Parche evolucionado, entrada de pino de ladera
1_6_21	230.268	4	1	-8.197	21	RA+TV	Riparian Trees or Big Shrut	Parche evolucionado, entrada de pino de ladera
1_7_18	229.964	8	1	-3.197	18	RH+RA	Riparian Herbs + Riparian T	Herbazal de Brachypodium con arbolado (populi
1_8_19	229.589	7	1	-0.4	19	RH	Riparian Herbs	Herbazal húmedo de Apium nodiflorum. impact
1_9_8	229.31	10	1	12.363	8	RH+RJ	Riparian Herbs + Riparian	Carrizal de phragmites. hay regeneracion de ss
1_10_9	229.516	9	1	15.952	9	RH	Riparian Herbs	Herbazal humedo mas franja seca de mayor col
1_11_10	230.176	2	1	19.541	10	RJ	Riparian Vegetation or Sm	Salix. restos acumulados en troncos de salix pr
1_12_13	230.499	2	1	25.523	13	RA	Riparian Trees or Big Shrut	Adelfas y majuelos. Impacto de avenida. Contac
1_13_14	230.697	1	1	29.91	14	TV	Terrestrial Vegetation	Adelfas, lentiscos y pinos. Aquí se recogió suel
1_14_55	231.447	1	1	33.84	55	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)
1_15_55	232.197	1	1	36.607	55	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)
2_1_54	232.969	4	2	-27.87857	54	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)
2_2_54	232.219	4	2	-26.1924665	54	TV	Terrestrial Vegetation	Ecosistema forestal (outside of riparian corridor)

Figura 3.26. Archivos de puntos de simulación visualizado mediante una hoja de cálculo.

### 1.1.1.3.- Organización de Ventanas de Ribav1D

Con el fin de lograr un manejo lo más intuitivo posible del programa, se ha dividido el software en varias ventanas (forms) que aparecen de forma sucesiva. El programa de Ribav estará gestionado por una Interfaz de Múltiples Documentos (MDI). La interfaz MDI consiste en una “ventana madre” que en este caso será la ventana del proyecto que englobará a todas las demás “ventanas hijas” que irán apareciendo de forma sucesiva.

De los formularios 1º hasta el 7º se podrá avanzar y retroceder mediante los botones “next” y “back” respectivamente. Se podrá acceder a los formularios 2º, 3º y 4º pulsando sobre los iconos que aparecen en la parte superior izquierda de la ventana MDI. La mayoría de las ventanas contienen los botones:

- Load: Sirve para cargar los datos de un archivo CSV distinto al archivo que se carga por defecto de la carpeta “Default”.
- Default: Sirve para cargar los datos por defecto del archivo correspondiente de la carpeta “Default”.
- Erase: Sirve para eliminar todos los datos de una determinada ventana.
- Save: Sirve para guardar los datos de una determinada ventana en un archivo tras haber realizado modificaciones.

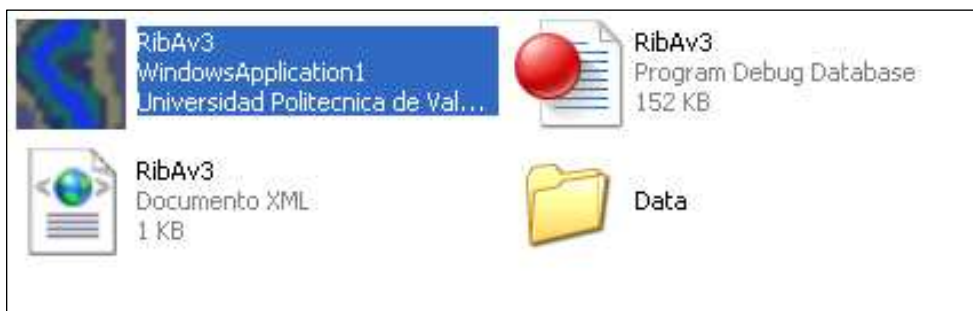
Las ventanas que aparecen dentro de Ribav1D en orden de aparición son las siguientes:

**Tabla 3.12 Ventanas (forms) del Software de Ribav1D**

Orden	Nombre de la Ventana	Descripción
1	Main Menu	Menú de Inicio
2	Soil Parameters	Parámetros de Suelo
3	Vegetation Parameters	Parámetros de Vegetación
4	Hydro-meteorological Input Data	Datos Hidro-meteorológicos de entrada
5	Rating Curves	Curvas de Gasto
6	Daily River Levels	Niveles de Río de cada Intervalo Diario
7	Simulation Point Data	Datos de los Puntos de Simulación
8	Simulations Results Form	Formulario de Resultados de la Simulación

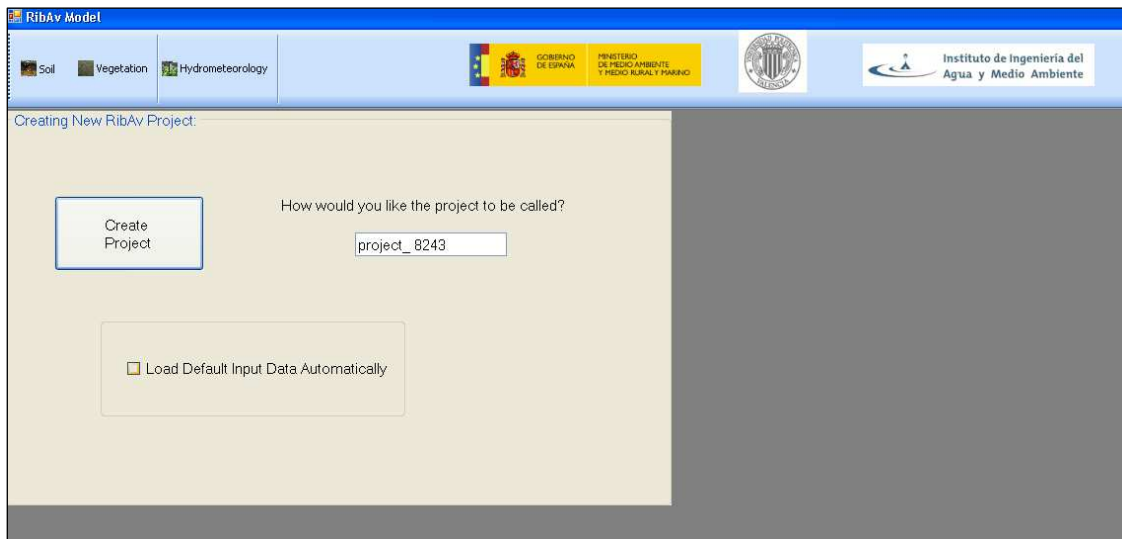
#### 1.1.1.4.- Ejecución del Software en Ribav1D

- a. En primer lugar se carga el programa pulsando sobre el archivo: “Ribav3.exe” dentro del directorio principal del programa (que tiene el icono de un río).



**Figura 3.27.** Archivos contenidos en el directorio principal de Ribav1D

- b. A continuación se cargará el programa y tras pasar la pantalla de bienvenida aparecerá el menú de inicio. En él se da la posibilidad de darle un nombre al proyecto, diferente al que sale por defecto de forma aleatoria. Adicionalmente se da la posibilidad de cargar los datos por defecto de la carpeta “Default” de forma automática, marcando sobre una “checkbox” (caja de selección). Para pasar al siguiente paso habrá que pulsar sobre el botón “Create Project”.



**Figura 3.28.** Menú de Inicio de Ribav1D

- c. La siguiente ventana es la de parámetros de suelo. Al pulsar sobre los números de la tabla de la derecha, aparecerán sobre las cajas de texto los valores de cada tipo de suelo. En la parte derecha de la ventana hay una barra de desplazamiento relacionada con los valores de humedad del suelo de una caja de texto. Estos valores de la humedad de suelo no participan en los cálculos posteriores de la simulación del modelo aunque pueden utilizarse para tener una idea general del funcionamiento de la curva de retención de humedad del suelo a partir de los parámetros de suelo, al visualizar los resultados de las cajas de texto de: “Matrix Potential” e “Hydraulic Conductivity”. Para pasar a la siguiente ventana bastará con pulsar el botón “Next”.

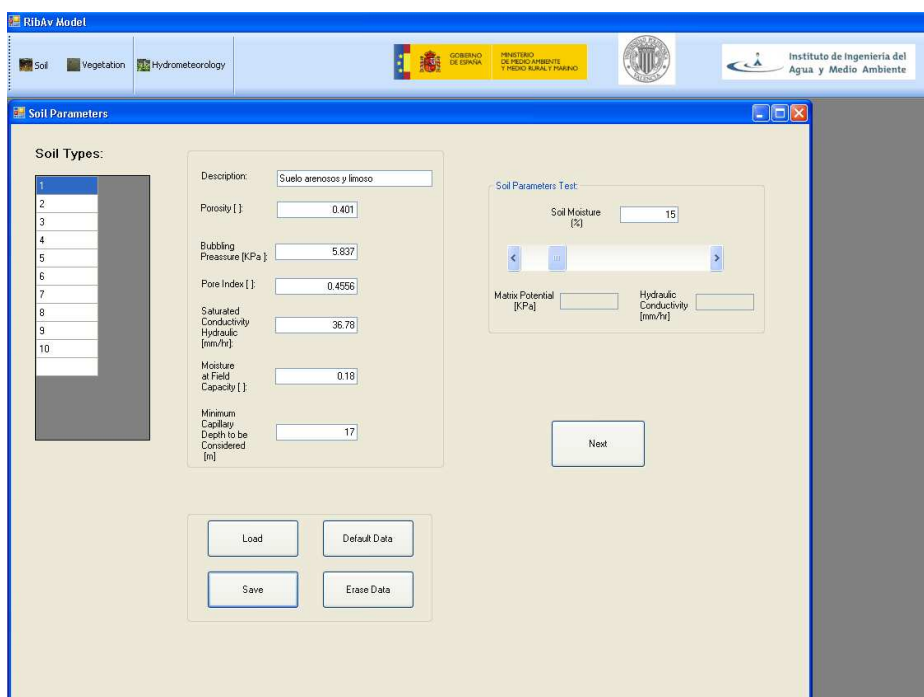


Figura 3.29. Ventana de Parámetros de Suelo de Ribav1D

- d. A continuación se muestra la ventana de parámetros de vegetación, donde para cada tipo funcional aparecen sus parámetros al pulsar sobre el número correspondiente. En la parte inferior de la ventana hay una caja de agrupación que contiene 12 cajas de texto, donde se puede introducir el factor de cobertura para cada mes del año.

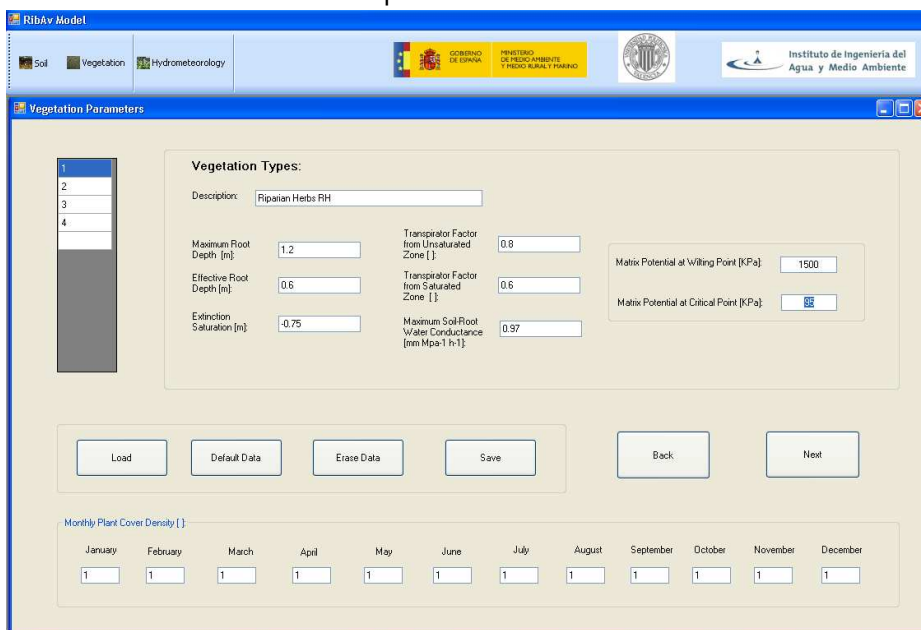


Figura 3.30. Ventana de Parámetros de Vegetación de Ribav1D

- e. La siguiente ventana es la de datos hidro-meteorológicos donde aparece una tabla con los registros diarios de fecha, precipitación, evapotranspiración potencial y caudal del río, en el mismo orden que en el archivo CSV.



**Figura 3.31.** Ventana de Datos Hidro-meteorológicos de Ribav1D

- f. La ventana siguiente es la de las curvas de gasto de todos los transectos del tramo de estudio. Al igual que en el archivo CSV, las columnas impares serán la de los caudales y las pares la de los niveles de río. Para pasar a la siguiente ventana habrá que pulsar el botón “Calculate Daily Flow (Next)”. Entonces se calcularán los niveles de río diarios para cada transecto.

Flow 1 Q (m3/s)	River Level1 H (m)	Flow 2 Q (m3/s)	River Level2 H (m)	Flow 3 Q (m3/s)	River Level3 H (m)	Flow 4 Q (m3/s)	River Level4 H (m)	Flow 5 Q (m3/s)	River Level5 H (m)	F C
0	228.62	0	228.59	0	228.71	0	228.68	0	228.828	0
0.001	228.65	0.001	228.62	0.001	228.74	0.001	228.71	0.001	228.856	0
0.003	228.68	0.002	228.65	0.003	228.77	0.002	228.74	0.002	228.884	0
0.005	228.71	0.003	228.68	0.005	228.8	0.003	228.77	0.004	228.912	0
0.009	228.74	0.005	228.71	0.009	228.83	0.006	228.8	0.007	228.94	0
0.014	228.77	0.008	228.74	0.014	228.86	0.01	228.83	0.012	228.968	0
0.021	228.8	0.013	228.77	0.021	228.89	0.016	228.86	0.018	228.996	0
0.031	228.83	0.019	228.8	0.031	228.92	0.023	228.89	0.026	229.024	0
0.044	228.86	0.027	228.83	0.044	228.95	0.034	228.92	0.037	229.052	0
0.061	228.89	0.037	228.86	0.061	228.98	0.047	228.95	0.051	229.08	0
0.082	228.92	0.051	228.89	0.082	229.01	0.065	228.98	0.068	229.108	0
0.107	228.95	0.068	228.92	0.108	229.04	0.086	229.01	0.089	229.136	0
0.139	228.98	0.088	228.95	0.141	229.07	0.112	229.04	0.116	229.164	0
0.176	229.01	0.113	228.98	0.179	229.1	0.143	229.07	0.148	229.192	0
0.22	229.04	0.143	229.01	0.225	229.13	0.18	229.1	0.186	229.22	0
0.272	229.07	0.179	229.04	0.28	229.16	0.224	229.13	0.23	229.248	0
0.332	229.1	0.221	229.07	0.344	229.19	0.275	229.16	0.283	229.276	0
0.402	229.13	0.271	229.1	0.418	229.22	0.334	229.19	0.344	229.304	0
0.482	229.16	0.328	229.13	0.503	229.25	0.402	229.22	0.415	229.332	0
0.574	229.19	0.394	229.16	0.601	229.28	0.48	229.25	0.496	229.36	0
0.677	229.22	0.47	229.19	0.713	229.31	0.568	229.28	0.588	229.388	0
0.794	229.25	0.556	229.22	0.838	229.34	0.667	229.31	0.693	229.416	0
0.925	229.28	0.654	229.25	0.981	229.37	0.779	229.34	0.811	229.444	1

**Figura 3.32.** Ventana de Curvas de Gasto de Ribav1D

- g. El resultado de los cálculos realizados se expone en la siguiente ventana. En ella se puede ver el valor del nivel del río en cotas absolutas para cada intervalo diario y para cada transecto.



RibAv Model

Soil Vegetation Hydrometeorology

GOBIERNO DE ESPAÑA MINISTERIO DE MEDIO AMBIENTE Y MEDIO RURAL Y PESQUERO

Instituto de Ingeniería del Agua y Medio Ambiente

River Levels for each Transect

Date	River Level(m)_1	River Level(m)_2	River Level(m)_3	River Level(m)_4	River Level(m)_5	River Level(m)_6	River Level(m)_7	River Level(m)_8	River Level(m)_9	River Level(m)_10
01/10/1998	229.249	229.286	229.329	229.343	229.439	229.487	229.554	229.635	229.678	229.7
02/10/1998	229.247	229.284	229.327	229.341	229.437	229.485	229.552	229.634	229.676	229.7
03/10/1998	229.231	229.267	229.311	229.324	229.422	229.471	229.538	229.619	229.662	229.7
04/10/1998	229.219	229.255	229.299	229.311	229.41	229.46	229.528	229.608	229.652	229.7
05/10/1998	229.233	229.27	229.314	229.326	229.424	229.474	229.541	229.622	229.665	229.7
06/10/1998	229.266	229.304	229.345	229.361	229.455	229.503	229.568	229.651	229.692	229.7
07/10/1998	229.324	229.363	229.401	229.42	229.508	229.554	229.617	229.701	229.74	229.8
08/10/1998	229.33	229.37	229.407	229.427	229.515	229.559	229.623	229.707	229.745	229.8
09/10/1998	229.33	229.369	229.406	229.426	229.515	229.559	229.622	229.706	229.744	229.8
10/10/1998	229.342	229.382	229.418	229.44	229.527	229.57	229.633	229.717	229.755	229.8
11/10/1998	229.346	229.385	229.422	229.443	229.53	229.573	229.636	229.721	229.758	229.8
12/10/1998	229.346	229.385	229.421	229.443	229.53	229.573	229.635	229.72	229.757	229.8
13/10/1998	229.342	229.381	229.418	229.439	229.526	229.57	229.633	229.717	229.755	229.8
14/10/1998	229.338	229.377	229.414	229.435	229.522	229.566	229.629	229.713	229.751	229.8
15/10/1998	229.333	229.372	229.41	229.43	229.518	229.562	229.625	229.709	229.747	229.8
16/10/1998	229.341	229.38	229.417	229.438	229.525	229.569	229.631	229.716	229.754	229.8
17/10/1998	229.332	229.371	229.408	229.429	229.516	229.56	229.624	229.708	229.746	229.8
18/10/1998	229.338	229.377	229.414	229.435	229.522	229.566	229.629	229.713	229.751	229.8
19/10/1998	229.319	229.358	229.396	229.415	229.505	229.549	229.613	229.697	229.735	229.8
20/10/1998	229.263	229.301	229.343	229.357	229.452	229.5	229.566	229.648	229.689	229.7

Load Save Back Next

Figura 3.33. Ventana de Datos de los Puntos de Simulación de Ribav1D

- h. La siguiente ventana es la última en la cual se introducen datos. En ella aparece una tabla que describe las características de cada punto de simulación.

RibAv Model

Soil Vegetation Hydrometeorology

GOBIERNO DE ESPAÑA MINISTERIO DE MEDIO AMBIENTE Y MEDIO RURAL Y PESQUERO

Instituto de Ingeniería del Agua y Medio Ambiente

Patches

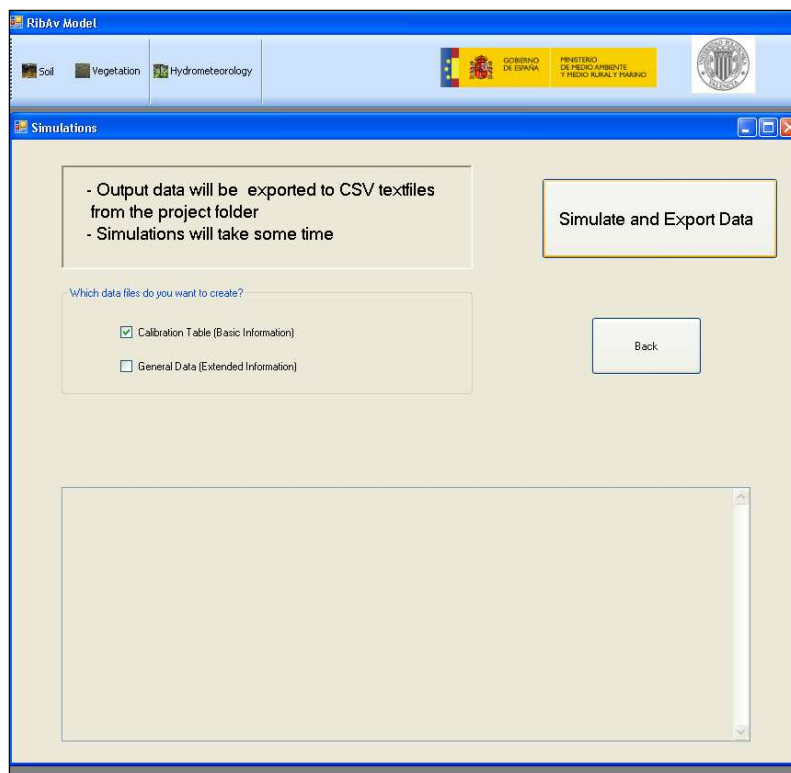
Simulation Point	Elevation (m)	Soil Type	Transect	Horizontal Distance (m)	Observed Veg. Code	Observed Veg. Acronym	Observed Vegetation Description
1_1_54	236.133	4	1	-33.197	54	TV	Terrestrial Vegetation
1_2_54	232.898	4	1	-28.197	54	TV	Terrestrial Vegetation
1_3_54	231.45	4	1	-23.197	54	TV	Terrestrial Vegetation
1_4_21	230.722	4	1	-18.197	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
1_5_21	230.431	4	1	-13.197	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
1_6_21	230.268	4	1	-8.197	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
1_7_18	229.954	8	1	-3.197	18	RH+RA	Riparian Herbs + Riparian Trees or Big Shrubs
1_8_19	229.589	7	1	-0.4	19	RH	Riparian Herbs
1_9_8	229.31	10	1	12.363	8	RH+RJ	Riparian Herbs + Riparian Vegetation or Small Shrubs
1_10_9	229.516	9	1	15.952	9	RH	Riparian Herbs
1_11_10	230.176	2	1	19.541	10	RJ	Riparian Vegetation or Small Shrubs
1_12_13	230.499	2	1	25.523	13	RA	Riparian Trees or Big Shrubs
1_13_14	230.697	1	1	29.91	14	TV	Terrestrial Vegetation
1_14_55	231.447	1	1	33.84	55	TV	Terrestrial Vegetation
1_15_55	232.197	1	1	36.607	55	TV	Terrestrial Vegetation
2_1_54	232.969	4	2	-27.87857002	54	TV	Terrestrial Vegetation
2_2_54	232.219	4	2	-26.19246649	54	TV	Terrestrial Vegetation
2_3_21	231.469	4	2	-23.3985	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
2_4_21	230.765	4	2	-18.3985	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
2_5_21	230.574	4	2	-13.3985	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
2_6_21	230.388	4	2	-8.3985	21	RA+TV	Riparian Trees or Big Shrubs + Terrestrial Vegetation
2_7_18	230.31	8	2	-3.3985	18	RH+RA	Riparian Herbs + Riparian Trees or Big Shrubs
2_8_19	230.31	7	2	-2.199	19	RH	Riparian Herbs

Load Erase Save Default Data

Back Next

**Figura 3.34.** Ventana de Datos de los Puntos de Simulación de Ribav1D

- i. En la ventana final se realizan el grueso de los cálculos de la simulación y se almacenan los resultados en ficheros CSV. Según la caja de selección que se pulse, se podrá acceder a 2 tipos diferentes de resultados:
  - “CalibrationTable (Basic Information)”: Es una tabla resumen de información básica donde se expone para cada uno de los puntos de simulación el resultado del Índice de Evapotranspiración (ETindex) de cada tipo de vegetación.
  - “General Data (Extended Information)”: Con esta opción se muestran los resultados diarios para todos los puntos de simulación y todos los tipos de vegetación, no solo de la variable de salida del Índice de Evapotranspiración (ETindex) sino de todas las variables internas del modelo. Al haber gran cantidad de datos almacenados, para esta opción se requerirá mayor tiempo de computación y un espacio bastante mayor de almacenamiento en el disco duro.



**Figura 3.35.** Ventana de Resultados de la Simulación de Ribav1D

#### 3.1.4.3.- Resultados obtenidos mediante la simulación del software de Ribav1D

Tras realizar la simulación se obtendrán los siguientes archivos y directorios de resultados. Se llamará "X" al nombre del proyecto, "Y" al nombre genérico del punto de simulación y "Z" al nombre genérico del tipo funcional de vegetación.

a) Calibration Matrix project\_X.csv

Es el único archivo que se obtendría si solo se pulsase la caja de selección de "CalibrationTable-Basic Information" Se llama así porque es el archivo utilizado principalmente para la calibración-validación del modelo. Es una tabla resumen donde se expone para cada uno de los puntos de simulación el resultado del Índice de Evapotranspiración (ETindex) de cada tipo de vegetación. En cuanto al orden de las columnas aparecería en primer lugar el número del punto de simulación (que sería el orden en el que aparece en la Ventana de Datos de los Puntos de Simulación). A la derecha aparecería la columna que indicaría la vegetación observada en cada uno de los puntos de simulación. Por último aparecerían las columnas con los valores de los ETindex para cada tipo funcional de vegetación en el orden en el que aparecerían dichos tipos en la ventana de "Parámetros de Vegetación".

*****					
CALIBRATION MATRIX Serpis					
*****					
Point	ObsVeg	ETindex_1	ETindex_2	ETindex_3	ETindex_4
1	TV	0.36629038	0.410969	0.42442954	0.55791244
2	TV	0.38060611	0.42791638	0.62971526	0.6344822
3	TV	0.41742141	0.48151927	0.78216459	0.99702037
4	RA+TV	0.5678561	0.88659049	0.97605433	0.67396301
5	RA+TV	0.86797555	0.9951305	0.97537079	0.50295375
6	RA+TV	0.95726532	0.99701423	0.97437576	0.40724392
7	RH+RA	0.96431799	0.9582163	0.94869097	0.22908959
8	RH	0.98646066	0.49213515	0.55963072	0.03204763
9	RH+RJ	0.80956568	0.10351015	0.18959274	0.00020442
10	RH	0.96277159	0.37979647	0.46141962	0.01412991
11	RJ	0.98593708	0.99557326	0.97322142	0.35324955
12	RA	0.8029952	0.98899417	0.9755736	0.54289857

Figura 3.36. Ejemplo de archivo de resultados generales en Ribav1D (Calibration Matrix project\_Serpis.csv)

b) Summary from project\_X.csv

Este archivo proporcionará también para cada punto de simulación los valores del ETIndex y de vegetación observada pero adicionalmente también indicará el tipo de suelo y el transecto al que pertenece dicho punto de simulación.

SUMMARY FROM Serpis	
PATCH 1_1_54	
VegType	ETRtot_Index
1	0.36629038
2	0.410969
3	0.42442954
4	0.55791244
Observed VegTV	
Patch Elevati	236.133
Soil Type	4
Transect	1
PATCH 1_2_54	
VegType	ETRtot_Index
1	0.38060611
2	0.42791638
3	0.62971526
4	0.6344822
Observed VegTV	
Patch Elevati	232.898
Soil Type	4
Transect	1
PATCH 1_3_54	
VegType	ETRtot_Index
1	0.41742141
2	0.48151927
3	0.78216459
4	0.99702037
Observed VegTV	
Patch Elevati	231.45
Soil Type	4
Transect	1

**Figura 3.37.** Ejemplo de archivo de resumen de resultados en Ribav1D (*Summary from Serpis.csv*)

c) Results from Patch\_Y.csv

Cada punto de simulación tendrá adjudicado un subdirectorío donde se almacenará archivos que darán información adicional sobre las simulaciones realizadas para dicho punto de simulación.



**Figura 3.38.** Ejemplo del contenido del subdirectorio de un determinado punto de simulación en Ribav1D (*Results from Patch\_1\_3\_54*)

d) Summary from Patch\_Y.csv

Será uno de los archivos contenidos en cada subdirectorio de "*Results from Patch\_Y.csv*". Este archivo proporcionará información de los valores de los parámetros de suelo, parámetros de vegetación y datos de la curva de gasto aplicables a dicho punto de simulación.

SUMMARY FROM PATCH 1_3_54										
Patch Elevation: 231.45 meters										
Soil Type: 4										
Pertaining Transect: 1										
Observed Vegetation: Terrestrial Vegetation										
-----										
Soil Parameters 4										
-----										
Soil Type: 4										
Porosity: 0.45 []										
Porosity Index: 0.23833779										
Bubble Pressure: 4.34325797Kpa										
Saturated Conductivity: 15.81 mm/hr										
Field Capacity Moisture: 0.277 []										
-----										
Evaporation Index Results and Vegetation Input Parameters										
-----										
Simulated Vegetation Type /VegType										
Simulated Evapotranspiration Index [ ] ETIndex										
Maximum Root Depth [m] Zr										
Effective Root Depth [m] Ze										
Saturation Extinction Depth [m] Zsat										
Transpiration Factor from Saturated Zone [ ] Ri										
Maximum Soil-Root Water Conductance [mmMpa-1h-1] CRT										
Transpiration Factor from Unsaturated Zone [ ] Ri										
Plant Cover Month 1 [ ] Cov1										
Plant Cover Month 2 [ ] Cov2										
Plant Cover Month 3 [ ] Cov3										
Plant Cover Month 4 [ ] Cov4										
Plant Cover Month 5 [ ] Cov5										
Plant Cover Month 6 [ ] Cov6										
Plant Cover Month 7 [ ] Cov7										
Plant Cover Month 8 [ ] Cov8										
Plant Cover Month 9 [ ] Cov9										
Plant Cover Month 10 [ ] Cov10										
Plant Cover Month 11 [ ] Cov11										
Plant Cover Month 12 [ ] Cov12										
VegType	ETInd	Zr	Ze	Zsat	Ri	Rj	CRT	Description	Cov1	Cov2
RH	0.417	1.2	0.6	-0.75	0.8	0.6	0.97	Riparian He	1	1
RJ	0.482	1.5	0.8	0	0.9	0.35	0.97	Riparian Juv	1	1
RA	0.782	4	0.8	-0.1	0.9	0.35	0.97	Riparian adu	0.9	0.9
TV	0.997	2.2	2	0.3	1	0	0.97	Terrestrial v	1	1
-----										
Rating Curve										
-----										
Flow [m3/s] Qcurve										
Water Level [m] Hcurve										
Qcurve	Hcurve									
0	228.62									
0.001	228.65									
0.003	228.68									
0.005	228.71									

Figura 3.39. Ejemplo de archivo de resumen de datos de un determinado punto de simulación en Ribav1D (Summary from Patch 1\_3\_54.csv)

e) Vegetation\_Z in Patch Y.csv

Habr  un archivo de este tipo para cada tipo funcional de vegetaci n de cada punto de simulaci n. En el aparecer n todos los valores de todas las variables del modelo para cada intervalo diario de la simulaci n. De izquierda a derecha aparecer n las columnas de las siguientes variables o datos de entrada:

- Fecha (dd/mm/aaaa)
- Precipitaci n (mm)
- Evapotranspiraci n Potencial (mm)
- Caudal diario (m<sup>3</sup>/s)
- Cobertura Mensual ( )
- Cota de nivel nivel fre tico (m)
- Humedad de suelo al final del d a anterior (mm)
- Presi n matricial (Kpa)
- Ascenso hidr ulico radicular (mm)
- Conductividad hidr ulica del suelo no saturado (mm/hr)
- Ascenso capilar (mm)
- Excedente de agua en la porci n de suelo (mm)
- Entrada de agua en la porci n de suelo (mm)
- Humedad de suelo inicial sin contar evapotranspiraci n (mm)
- Humedad relativa de suelo ( )
- Evapotranspiraci n real no saturada (mm)
- Evapotranspiraci n real saturada (mm)
- Evapotranspiraci n total (mm)
-  ndice de Evapotranspiraci n ( )

RESULTS FROM PATCH 1_3_54 OF VEGETATION_2																		
Date	PP	ETP	Qdaj	DailyCo	VTE	Hfin	Pmat	Rwu	Kh	Cwu	Exc	Win	Hini	Hrel	ETRuns	ETRsat	ETRtot	ETIndex
01/10/1998	0	2.215	0.731	1	229.249	0	0	0	0	0	0	0	0	0	0	0	0	0
02/10/1998	0	2.511	0.783	1	229.247	170.8207	95	0	0.00363969	0.516	0	0.516	173.0806	1	2.2599	0	2.2599	0.9
03/10/1998	0	2.598	0.718	1	229.231	168.3948	99.1362	0	0.003242	0.4767	0	0.4767	171.2973	0.9848	2.3026	0	2.3026	0.8863
04/10/1998	0.827	2.598	0.672	1	229.219	167.9884	103.7085	0	0.00286844	0.4401	0	1.2671	170.2618	0.9723	2.2735	0	2.2735	0.8751
05/10/1998	0.353	2.215	0.728	1	229.233	168.8668	106.3404	0	0.00267976	0.428	0	0.781	168.7694	0.9544	1.9025	0	1.9025	0.8589
06/10/1998	0.1103	2.392	0.865	1	229.266	165.3788	109.3716	0	0.00248289	0.4207	0	0.531	167.3979	0.9379	2.0191	0	2.0191	0.8441
07/10/1998	0	2.392	1.145	1	229.324	163.8175	113.5603	0	0.00224203	0.4162	0	0.4162	165.7951	0.9186	1.9776	0	1.9776	0.8267
08/10/1998	0	2.249	1.181	1	229.33	162.3895	118.1711	0	0.0020124	0.3927	0	0.3927	164.2102	0.8995	1.8208	0	1.8208	0.8096
09/10/1998	0	2.081	1.177	1	229.33	161.1077	122.5929	0	0.00182137	0.3704	0	0.3704	162.7598	0.8821	1.6521	0	1.6521	0.7939
10/10/1998	0	2.333	1.249	1	229.342	159.6431	126.7373	0	0.00166416	0.3548	0	0.3548	161.4625	0.8665	1.8194	0	1.8194	0.7739
11/10/1998	0	2.416	1.27	1	229.346	158.1326	131.6878	0	0.00149974	0.3348	0	0.3348	159.9779	0.8487	1.8453	0	1.8453	0.7638
12/10/1998	0	2.815	1.267	1	229.346	156.3431	137.0469	0	0.0013458	0.3139	0	0.3139	158.4465	0.8302	2.1034	0	2.1034	0.7472
13/10/1998	0	2.361	1.248	1	229.342	154.915	143.7497	0	0.00118217	0.2897	0	0.2897	156.6328	0.8084	1.7178	0	1.7178	0.7276
14/10/1998	0	2.394	1.222	1	229.338	153.4819	149.3925	0	0.00106483	0.2713	0	0.2713	155.1862	0.791	1.7044	0	1.7044	0.7119
15/10/1998	0	2.451	1.197	1	229.333	152.029	155.3331	0	0.00095785	0.2536	0	0.2536	153.7354	0.7736	1.7065	0	1.7065	0.6962
16/10/1998	0	2.424	1.24	1	229.341	150.619	161.6573	0	0.0008595	0.2392	0	0.2392	152.2682	0.7559	1.6492	0	1.6492	0.6804
17/10/1998	0	2.571	1.188	1	229.332	149.1323	168.1023	0	0.00077294	0.2228	0	0.2228	150.8418	0.7388	1.7095	0	1.7095	0.6649
18/10/1998	0	2.271	1.222	1	229.338	147.8683	175.2465	0	0.00069035	0.2092	0	0.2092	149.3414	0.7208	1.4732	0	1.4732	0.6487
19/10/1998	1.1826	1.992	1.12	1	229.319	147.9615	181.6182	0	0.00062656	0.1943	0	1.3769	149.2451	0.7196	1.2836	0	1.2836	0.6476
20/10/1998	0	2.051	0.851	1	229.283	146.8442	181.1385	0	0.00063107	0.1866	0	0.1866	148.1481	0.7064	1.304	0	1.304	0.6358
21/10/1998	0	1.982	0.848	1	229.282	145.7953	186.9822	0	0.00057887	0.177	0	0.177	147.0212	0.6929	1.2359	0	1.2359	0.6236
22/10/1998	0	2.222	0.958	1	229.287	144.597	192.7573	0	0.00053206	0.1718	0	0.1718	145.957	0.6801	1.36	0	1.36	0.6121
23/10/1998	0	2.249	0.912	1	229.277	143.4108	199.4911	0	0.00048561	0.1611	0	0.1611	144.7581	0.6656	1.3473	0	1.3473	0.5991
24/10/1998	0	2.392	1.151	1	229.325	142.1666	206.5068	0	0.00044211	0.1581	0	0.1581	143.5688	0.6513	1.4022	0	1.4022	0.5862
25/10/1998	0.2321	2.483	1.199	1	229.334	141.1205	214.1961	0	0.00040034	0.1499	0	0.382	142.5486	0.6391	1.4281	0	1.4281	0.5752
26/10/1998	0.5442	2.087	1.207	1	229.335	140.6236	220.9377	0	0.00036803	0.1425	0	0.6867	141.8072	0.6202	1.1836	0	1.1836	0.5671

**Figura 3.40.** Ejemplo de archivo de datos para un determinado punto de simulación y un determinado tipo funcional de vegetación en Ribav1D (*Vegetation\_2 in Patch 1\_3\_54.csv*)



### 3.2.- MODELO “RIBAV2D” Y SU INTEGRACIÓN CON RIPFLOW

#### 3.2.1.- Diferencias entre Ribav 1D y 2D

La modelización matemática de Ribav1D es idéntica a Ribav2D pero la versión 1D trata con puntos de simulación y la versión 2D con mapas grid. En la siguiente tabla se muestran las diferencias de archivos input y output de ambos programas:

Tabla 3.13 *Tipos de datos y archivos utilizados en las dos versiones de Ribav.*

Tipo de datos	Archivos utilizados por las distintas versiones del software de Riabav	
	Ribav-1D	Ribav-2D
Datos Espaciales	Datos relativos a los puntos de simulación	Mapa de tipos de suelo
		Mapa de modelo digital del terreno (MDT)
		Mapa de tipos de vegetación inicial
Parámetros Estimados	Parámetros de suelo	Parámetros de suelo
Parámetros Calibrados	Parámetros de vegetación	Parámetros de vegetación
Meteorología y Caudal del Río	Datos hidro-meteorológicos	Datos hidro-meteorológicos
Datos del Modelo Hidráulico	Archivo de curvas de gasto de todos los transectos	Archivos que relaciona el caudal del río con los distintos mapas de nivel freático
		Subdirectorío donde están contenidos los distintos mapas de nivel freático
Datos de Salida	Información de variables internas y de salida (índice ET) del modelo que permiten determinar la vegetación simulada por el modelo, al final del periodo de simulación	Directorío donde se almacenan los mapas con valores medios anuales del índice de ET para cada año
		Directorío donde se almacenan los mapas de tipo de vegetación al final de cada año

En la tabla se puede observar que Ribav1D gestiona la información espacial mediante puntos de simulación colocados en las zonas más representativas del tramo de estudio, mientras que Ribav2D utiliza mapas grid para gestionar la información variable espacialmente (suelos, vegetación, MDT). Respecto al modelo hidráulico, Ribav1D utiliza curvas de gasto de transectos situados perpendicularmente al tramo de estudio por lo que los puntos de simulación deben de ir situados en dichos transectos. Por otro lado Ribav2D utiliza mapas de lámina de agua y nivel freático que cubren toda la zona de estudio. Cada uno de esos mapas lleva relacionado un determinado caudal de río que produce esa altura de lámina de agua. Dicha relación entre mapas de lámina de agua y los caudales del río correspondientes queda asignado en un fichero de texto aparte. La versión 1D calcula los datos de salida como (el índice de evapotranspiración) una vez para cada simulación, mientras que la versión 2D lo calcula para cada año. Por último Ribav2D simula la vegetación de forma dinámica (en intervalos anuales), mientras que Ribav1D la simula de forma estática (considera que la vegetación no cambia en todo el periodo de simulación).

### **3.2.2.- Integración de Ribav 2D en Ripflow**

Una de las razones principales por la que se diseñó la versión 2D de Ribav fue para poder integrarse dentro del Ripflow. El modelo Ripflow exige la utilización de mapas grid y además calcula la distribución espacial de la vegetación de forma dinámica en intervalos anuales.

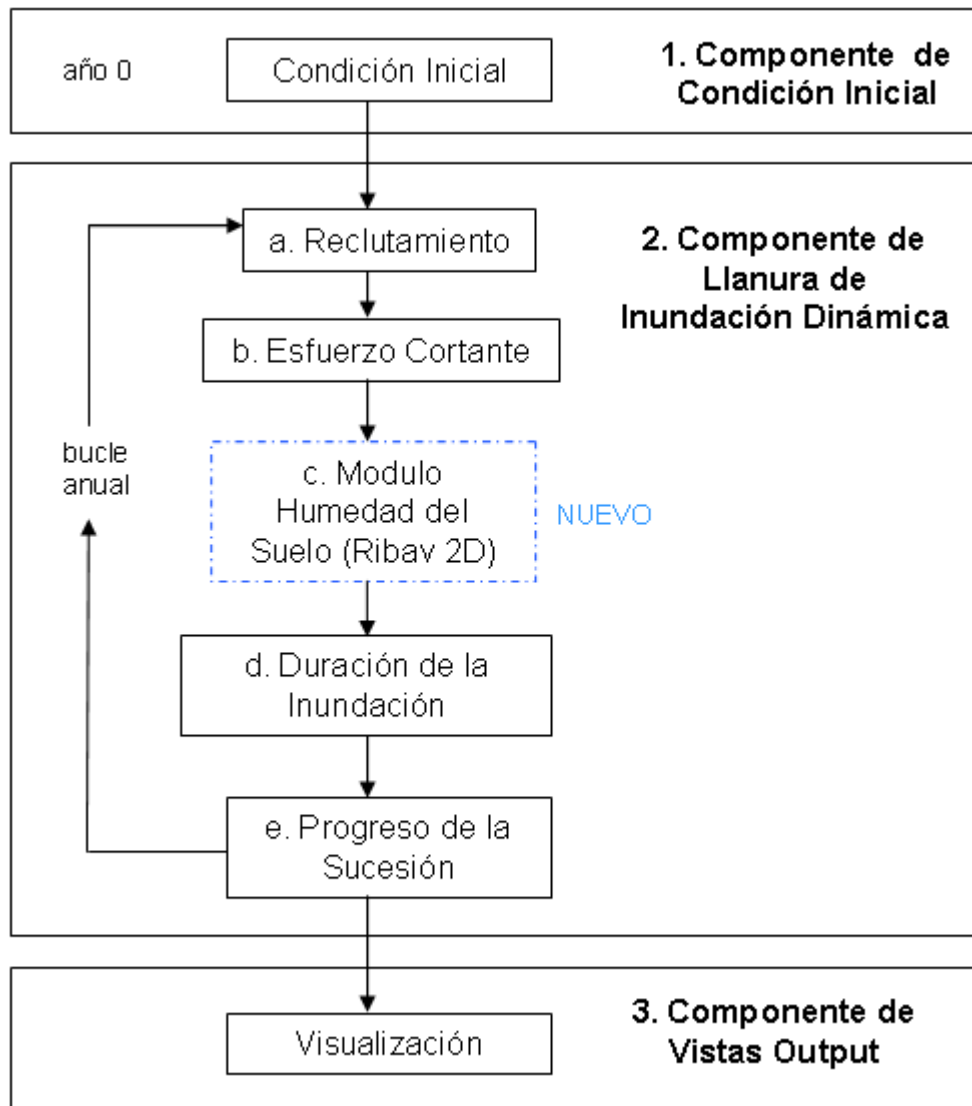
El módulo de "Humedad de Suelo-Ribav2D" irá situado dentro de la "Componente de Llanura de inundación dinámica", que tiene un bucle de repetición anual. Dentro de esta componente, irá ubicado entre los módulos de "Esfuerzo Cortante" y de "Duración de la inundación". Se situará detrás del módulo de esfuerzo cortante debido a que este módulo tiene para cada celda del grid dos posibles outputs:

- El esfuerzo cortante del año del bucle supera al esfuerzo cortante crítico del tipo de vegetación, por lo que la planta se destruye y se vuelve al estado de sucesión inicial de vegetación.
- El esfuerzo cortante del año del bucle no supera al esfuerzo cortante crítico del tipo de vegetación, por lo que dicha planta sobrevive y sobretodo añade un contador de año a su edad.

Si la planta no es destruida por el efecto cortante pasará al módulo de "Humedad del Suelo" (Ribav 2D). El módulo de Ribav2D devolverá un mapa grid con valores del Índice de Evapotranspiración (ETindex) de todas sus celdas. El valor del ETindex tiene un rango entre 0 y 1, siendo 0 el valor en el que la planta sufre mayor estrés hídrico y siendo 1 el valor en el que obtiene las condiciones más favorables para su desarrollo. Cada tipo de vegetación tiene dos umbrales de evapotranspiración (superior e inferior), que serán variables que intervendrán en la calibración. Dichos umbrales afectarán a la vegetación de la siguiente forma:

- Si el valor del ETindex está entre 0 y el valor del umbral inferior, entonces la planta se muere y la celda pasa al estado de sucesión vegetal inicial.
- Si el valor del ETindex está entre el valor del umbral inferior y el valor del umbral superior, entonces la planta no se muere pero pierde el contador de año que había ganado en el módulo de esfuerzo cortante.
- Si el valor del ETindex está entre el valor del umbral superior y 1, entonces la planta no solo no se muere sino además conserva el contador de año que había ganado en el módulo de esfuerzo cortante.

El componente de "Humedad del Suelo" (Ribav 2D) debe de estar situado detrás del modulo de esfuerzo cortante debido a la relación existente entre ambos que involucra al contador de año en los dos últimos casos de umbrales de ETindex descritos anteriormente. Como el modulo Ribav está dentro de la componente de Llanura de Inundación Dinámica, que depende del bucle anual, entonces será llamada una vez por cada año de simulación. En la figura siguiente se explica la estructura y orden de cálculo de los componentes de Ripflow.



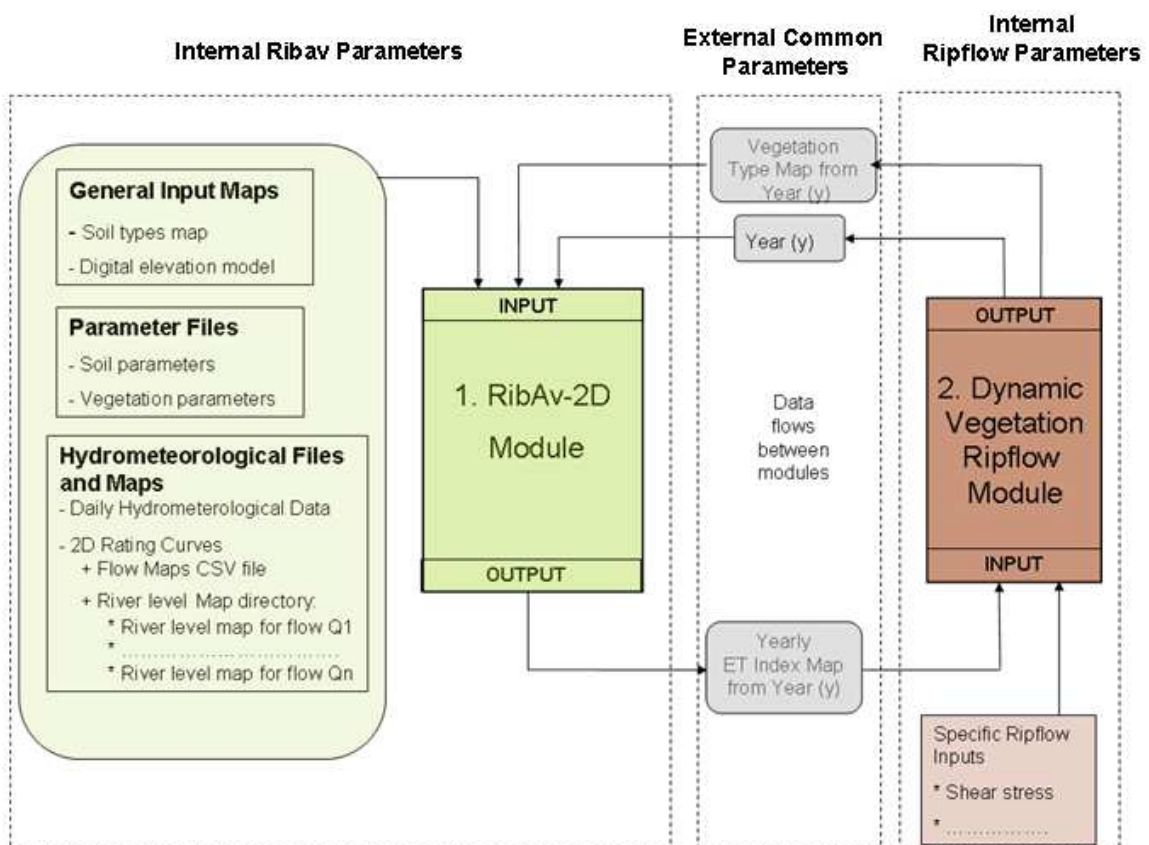
**Figura 3.41. Esquema que describe la integración del modulo Ribav2D dentro del modelo Ripflow**

En el apartado anterior se han descrito los parámetros y datos internos de Ribav2D (parámetros estimados de suelo, parámetros calibrados de vegetación, datos hidro-meteorológicos etc.) El resto de los módulos de Ripflow (*pertenecientes a la versión anterior de dicho programa*) también tendrá

sus propios datos de entrada (Mapas de Esfuerzo Cortante, Mapa de Zona de Ribera, Mapas de Días de Inundación etc.). Aun así habrá datos externos que serán comunes tanto para Ribav2D como del resto de Ripflow. Estos ficheros de datos externos y comunes serán los siguientes:

- Mapas de Tipos de Vegetación al principio del año (y). Es un dato de salida de Ripflow pero al mismo tiempo de entrada de Ribav 2D. Mediante un mapa grid en formato ASCII, describe mediante números enteros el tipo de vegetación inicial al principio del año en el que se esté realizando la simulación.
- Contador de año (y). También es un dato de salida de Ripflow y de entrada de Ribav 2D. Indica un valor ordinal entero que representa el año en el que se está realizando en esos momentos la simulación. Es un dato imprescindible porque lo necesita Ribav2D para extraer los datos hidro-meteorológicos del año correspondiente.
- Mapas de valores del Índice de Evapotranspiración (ETindex) al final del año (y). A diferencia de los anteriores, es un dato de entrada de Ripflow y de salida de Ribav 2D. Mediante un mapa grid en formato ASCII, describe mediante números decimales los valores de ETindex promedios del año en el que se esté realizando la simulación. Estos valores serán utilizados por Ripflow para realizar la correspondiente comparación con los umbrales de Evapotranspiración tal como se ha descrito anteriormente.

En la siguiente figura se representa el flujo de datos internos y externos de Ribav2D y del resto de Ripflow. En verde se representa a los datos o parámetros internos de Ribav, en rojo a los datos internos de Ripflow y en gris a los datos externos comunes a ambos:



**Figura 3.42. Diagrama de flujo de datos internos y externos de Ribav2D y del resto de Ripflow.**

### **3.2.3.- Utilización de la tecnología Component Object Model (COM)**

El modulo Ribav2D ha sido escrito en el lenguaje de programación Visual Basic.net, mientras que el resto de Ripflow ha sido escrito en Python utilizando la aplicación "Model Builder" de ArcGis. Para no tener que traducir el código de alguna de las partes de Ripflow a un lenguaje común se ha optado por utilizar la tecnología COM (*Component Object Model*). La tecnología COM es una plataforma para componentes de software creada por Microsoft en 1993, que permite la comunicación entre objetos diseñados con lenguajes de programación distintos. Para permitir esta comunicación se ha compilado el modulo de Ribav2D en una librería DLL que además es visible a la tecnología COM. Las librerías DLL son Librerías de Enlace Dinámico (*Dynamic Linked Libraries*), que son tipos de archivos que contienen subrutinas o variables que pueden ser llamadas desde cualquier aplicación de los sistemas operativos Windows. Con respecto al archivo DLL de Ribav2D, Ripflow incluirá instrucciones que realizarán las siguientes acciones:

- En primer lugar se ejecutará el archivo-librería DLL que contiene al módulo de Ribav2D como si de un archivo ejecutable (.exe) se tratase. Dicha librería estará situada en la subcarpeta de ensamblado de Windows. En Windows XP la subcarpeta de ensamblado se encuentra en la siguiente ruta: "C:\WINDOWS\assembly"
- Posteriormente se enviará información a las variables de entrada de dicha librería. Los datos enviados serán:
  - ⇒ una cadena de caracteres que representa la dirección relativa del archivo del mapa grid de vegetación inicial (ej. "\Data\INPUTS\Maps\VegYear2..asc")
  - ⇒ un número entero que define el año en el que transcurre dicha parte de la simulación. (ej. "3")
  - ⇒ otra cadena de caracteres que define el directorio de trabajo, que combinada con las direcciones relativas permite obtener las direcciones absolutas de los archivos (ej. "C:\Ripflow\Ribav")
- Finalmente se almacenará el valor devuelto por la librería de Ribav2D tras su ejecución. El valor devuelto es la dirección del archivo del mapa grid de los Índices de Evapotranspiración (ETindex), creado tras cada llamada al modulo Ribav2D del bucle año. (ej. "C:\Ripflow\Ribav\Data\OUTPUTS\ETindexMaps\ETindexYear3.asc")

Para que está interacción entre Ribav2D y Ripflow sea posible se crea en el código fuente de Ribav2D una nueva clase que es visible a la tecnología COM y actúa como nexo entre ambos programas. Las instrucciones más representativas de la clase COM son las siguientes:

```

1. Imports System.Runtime.InteropServices
2.
3.
4. <Guid("63E4EE68-637A-431f-AE69-D263C8A2A522"), InterfaceType(ComInterfaceType.InterfaceIsIDispatch)> _
5. Public Interface _Logic
6. <DispId(1)> Sub CallingSimulate(ByVal VegMap As String, ByVal Year As Integer?, ByVal ConfigFileDirectory As String)
7. <DispId(2)> ReadOnly Property ETIndexPath() As String
8. End Interface
9.
10.
11.
12. <Guid("1971C5EE-DADE-46cb-BCBF-2D49792A3FCE"), _
13. ClassInterface(ClassInterfaceType.None), _
14. ProgId("Ribav2DModule")> Public Class Logic
15. Implements _Logic
16.
17.
18. Public ReadOnly Property ETIndexPath() As String Implements _Logic.ETIndexPath
19. Get
20.     Return (OutputETIndexPath)
21. End Get
22. End Property
23.
24.
25. Public Sub CallingSimulate(ByVal VegMap As String, ByVal Year As Integer?, ByVal ConfigFileDirectory As String) _
26. Implements _Logic.CallingSimulate
27.     Call LoadDEFAULTFilePaths(ConfigFileDirectory)
28.     Call MainHydromet(ConfigFileDirectory)
29.     Call Simulate(VegMap, Year)
30.     'Mas Instrucciones y Llamadas a Subrutinas
31. End Sub
32.
33. End Class

```

**Figura 3.43. Ejemplo de Interfaz y Clase COM utilizada como nexo entre Ribav y el resto de Ripflow v.3**

- (Línea 1): *Imports System.Runtime.InteropServices*: Esta instrucción llama o importa la clase “InteropServices” que se encuentra dentro del espacio de nombres “System.Runtime”. Esta clase está incluida por defecto en Visual Studio 2008 (aunque tiene que ser importada previamente) y es la que permite crear clases visibles a la tecnología COM.
- (Líneas 4 y 5): *<Guid("63E4EE68-637A-431f-AE69-D263C8A2A522"), InterfaceType(ComInterfaceType.InterfaceIsIDispatch)> \_ Public Interface \_Logic* Está instrucción declara la interfaz “logic” que acaba en la línea 9. Una interfaz es similar a una clase porque es una agrupación de variables (*también llamadas propiedades*) y métodos (*que engloban a las subrutinas y funciones*). La diferencia principal entre una interfaz y una clase es que la clase es utilizada cuando se crea un objeto instanciado a dicha clase. Por el contrario una interfaz no puede ser instanciada, aunque sí que puede ser heredada o implementada por una clase que de esta forma puede utilizar sus propiedades y métodos.

Al comienzo de la línea 4 aparece el GUID (Globally Unique Identifier), que es una cadena de caracteres generada al azar con el fin de que sea única (cosa que es altamente probable al haber un número total de caracteres elevado). La plataforma COM utiliza las claves GUID para llamar a los archivos DLL sin tener que referirse al nombre de dicho archivo. Es preferible utilizar estas claves, en vez de nombres, para evitar que en la carpeta de ensamblado haya confusiones entre distintas librerías que tengan un mismo nombre o entre distintas versiones de una misma librería.

Al final de la línea 4 se define a la interfaz del tipo "InterfacelsIDispatch", lo que obliga a declarar el tipo de dato de las propiedades que aparezcan más adelante. Esta opción es recomendable para este caso, porque acelera el tiempo de ejecución y previene errores.

En la línea 5 se declara el nombre de la interfaz como "\_Logic" y se la declara "Public", es decir pública, para poder ser accedida externamente sin problemas.

- **(Línea 6):** `<Displd(1)> Sub CallingSimulate(ByVal VegMap As String, ByVal Year As Integer?, ByVal ConfigFileDirectory As String)`

En esta línea se declara la subrutina "CallingSimulate", que es la subrutina de Ribav2D a la cual se llama de forma externa desde Ripflow. Esta subrutina tiene 3 parámetros de entrada: VegMap (*dirección relativa del mapa de vegetación inicial anual*), Year (*el año de la simulación*) y ConfigFileDirectory (*directorío de trabajo o workspace, donde están situados los ficheros de entrada y salida de Ribav2D*).

- **(Línea 7):** `<Displd(2)> ReadOnly Property ETIndexPath() As String`

En esta línea se declara el segundo de los componentes de la interfaz que en este caso es una propiedad (*variable*) de tipo string (*cadena de caracteres*). Se declara como "ReadOnly" (*propiedad de solo lectura*), debido a que almacena datos de una variable de salida de Ribav2D cuya información (en este caso la ruta del mapa del índice de ET) va a ser exportada a Ripflow.

- **(Líneas 12, 13 y 14):** `<Guid("1971C5EE-DADE-46cb-BCBF-2D49792A3FCE"), _  
ClassInterface(ClassInterfaceType.None), _  
ProgId("Ribav2DModule")> Public Class Logic`

En estas líneas se va a definir la clase "Logic", que heredara la subrutina y la propiedad antes descritas de la interfaz "\_Logic". En primer lugar se utiliza otra clave GUID para identificar a esta clase. Posteriormente se declarará el tipo de la interfaz como: "ClassInterfaceType.None", debido a que Microsoft lo recomienda para que no se creen interfaces adicionales que puedan crear problemas de compatibilidad. En la línea 14 se declara no solo el nombre de la clase "Logic", si no también el nombre externo al que se llamará mediante la tecnología COM: ProgId("Ribav2DModule"). Este nombre queda como alternativa de llamada a la clave GUID.

- **(Línea 15):** `Implements _Logic`

Esta instrucción hace que la clase Logic herede las propiedades y métodos de la interfaz “\_Logic”.

- **(Líneas 18 a 22):**

```
Public ReadOnly Property ETIndexPath() As String _  
    Implements  
        _Logic.ETIndexPath  
    Get  
        Return (OutputETIndexPath)  
    End Get  
End Property
```

Anteriormente se había declarado la propiedad “ETIndexPath” dentro de la interfaz “\_Logic” (línea 7). Por ello en estas líneas se definirán las instrucciones internas de “ETIndexPath”. En primer lugar (línea 19), se enlazará esta propiedad con su declaración anterior mediante la sintaxis “Implements”. Como se había declarado anteriormente a esta propiedad como de “solo lectura” (ReadOnly) será necesario definir la variable interna que va a ser retornada (línea 20), que en este caso se llama: “OutputETIndexPath”.

- **(Líneas 25 a 31):**

```
Public Sub CallingSimulate(ByVal VegMap As String, ByVal Year As Integer?, ByVal ConfigFileDirectory As  
String) Implements _Logic.CallingSimulate  
    Call LoadDEFAULTFilePaths(ConfigFileDirectory)  
    Call MainHydromet(ConfigFileDirectory)  
    Call Simulate(VegMap, Year)  
'Mas Instrucciones y Llamadas a Subrutinas  
End Sub
```

En estas líneas aparecerán las instrucciones contenidas dentro de la subrutina “CallingSimulate”, que había sido anteriormente declarada en la interfaz “\_Logic”. Los parámetros de entrada de la subrutina son: “VegMap”, “Year” y “ConfigFileDirectory”; que ya han sido declarados anteriormente. Cabe destacar que la variable “Year” ha sido declarada como “As Integer?”, para que mediante el signo de interrogación pueda aceptar además de valores enteros, también valores nulos.

En concreto en la línea 26 se enlazará esta subrutina con su declaración anterior dentro de la interfaz “Logic.CallingSimulate”, mediante la sintaxis “Implements”. Se considera a “CallingSimulate” como la primera subrutina o “rutina principal” del módulo Ribav2D. Por ello contendrá (líneas 27 a 30) llamadas a subrutinas secundarias del programa, además de poder contener instrucciones propias.



### **3.2.4.- Utilización del Modulo Ribav2D**

Para describir la utilización de Ribav2D se ha escrito un tutorial, que al estar escrito en inglés debido a requerimientos del proyecto, ha sido incluido en este documento como "Anexo I".

### **3.3.- PROGRAMA PARA EL CALCULO DEL COEFICIENTE KAPPA DE COHEN ENTRE LOS MAPAS DE VEGETACIÓN OBSERVADA Y VEGETACIÓN SIMULADA DE RIPFLOW V.3**

Para facilitar el proceso de calibración/validación de Ripflow v.3 (*Ripflow v.2 combinado con Ribav2D*), se ha diseñado un programa que determina el número de aciertos mediante el coeficiente Kappa de Cohen. El coeficiente Kappa de Cohen calcula el número de aciertos pero sin tener en cuenta el efecto causado por el azar. En Ripflow v.3 se compararán los mapas de vegetación simulada por el modelo con los mapas de vegetación observada en campo. Para facilitar la visualización de los datos de entrada y de salida de este programa, incluyendo el número de aciertos de la simulación, se han utilizado las librerías de "MapWindow", que es un sistema de información geográfica de código abierto que ha sido desarrollado por la Universidad de Idaho.

Para describir la utilización de este programa se ha escrito un tutorial, que al estar escrito en inglés, debido a requerimientos del proyecto, ha sido incluido en este documento como "Anexo II".

## **4.- CASOS DE ESTUDIO**

#### 4.1.- LOCALIZACIÓN DE LOS TRAMOS DE ESTUDIO

Los tramos finalmente estudiados se recogen de forma resumida en la siguiente tabla y se describen en los siguientes apartados.

**Tabla 4.14.**Tabla de los tramos de estudio.

Tramo	Río	Régimen	Nº transectos	Longitud	X	Y
Lorcha	Serpis	Regulado	11	239	733362.20	4304164.78
Terde	Mijares	Natural	20	539	689463.54	4448872.32



**Figura 4.44.** Localización de los tramos de estudio dentro de la demarcación hidrográfica del río Júcar.

#### **4.1.1.- Río Serpis en Lorcha**

El tramo de estudio localizado en el río Serpis se encuentra situado bajo la presa de Beniarrés, en las proximidades del municipio de Lorcha, a la entrada del paraje conocido como “Barranc del Infern”. En él el río empieza a encajarse, tallando un espectacular cañón en forma de V, aprovechando las diferentes fallas y otras debilidades geológicas de los terrenos calizos de la zona.

El tramo está incluido dentro del LIC Serra de la Safor y de la masa de agua 21.06. Su altitud media es de unos 229 msnm. La accesibilidad es relativamente sencilla, gracias a la vía verde (antiguo trazado de la línea férrea Alcoy-Gandia) que recorre una de las orillas del río a lo largo del tramo. Es posible recorrer casi todo el cauce a pie debido a su poca profundidad y las orillas son fácilmente transitables, a excepción de algunas zonas con zarzal y cañaveral de alta densidad.

Durante los trabajos de campo (veranos de 2007 y 2008), la anchura media de la lámina de agua fue de 13.98 m, con una profundidad media de 0.41 m, con máximos de unos 0.60 m. La turbidez del tramo es alta. El lecho es bastante estable, en ocasiones completamente sólido, por los afloramientos de roca madre. La mayor parte del tramo tiene una corriente media-débil, que se hace fuerte cuando el cauce se estrecha y el sustrato se hace más grueso. Predominan los cantos y las gravas, con excepción del margen derecho, más elevado, donde las arenas son el sustrato predominante. La acumulación de gravas y gravilla en el centro y márgenes del cauce condiciona la formación de depósitos y bancos que se cubren con herbáceas anuales cada primavera y que desaparecen con las primeras lluvias del otoño.

En las zonas arboladas, el porcentaje de sombra sobre el cauce es elevado, en el resto predominan los grandes claros. La presencia de troncos y ramas sobre el agua, así como raíces expuestas aportan heterogeneidad de hábitats. Están presentes las 4 categorías de regímenes de velocidad/profundidad: lento-profundo, lento-somero, rápido-profundo y rápido-somero, aunque la frecuencia de rápidos es escasa, predominando las corrientes y tablas.

Geomorfológicamente se trata de una ribera con potencialidad intermedia para soportar una zona vegetada, ya que existen varias zonas con sustrato duro, no apto para el enraizamiento de la vegetación. No obstante, el sustrato predominante es grava y arena, con presencia ocasional de bloques y grandes bloques coluviales, procedentes de las laderas próximas. La zona riparia tiene un desnivel comprendido entre los 20 y 45°, en ocasiones escalonado, y una cobertura vegetal superior al 50 %.

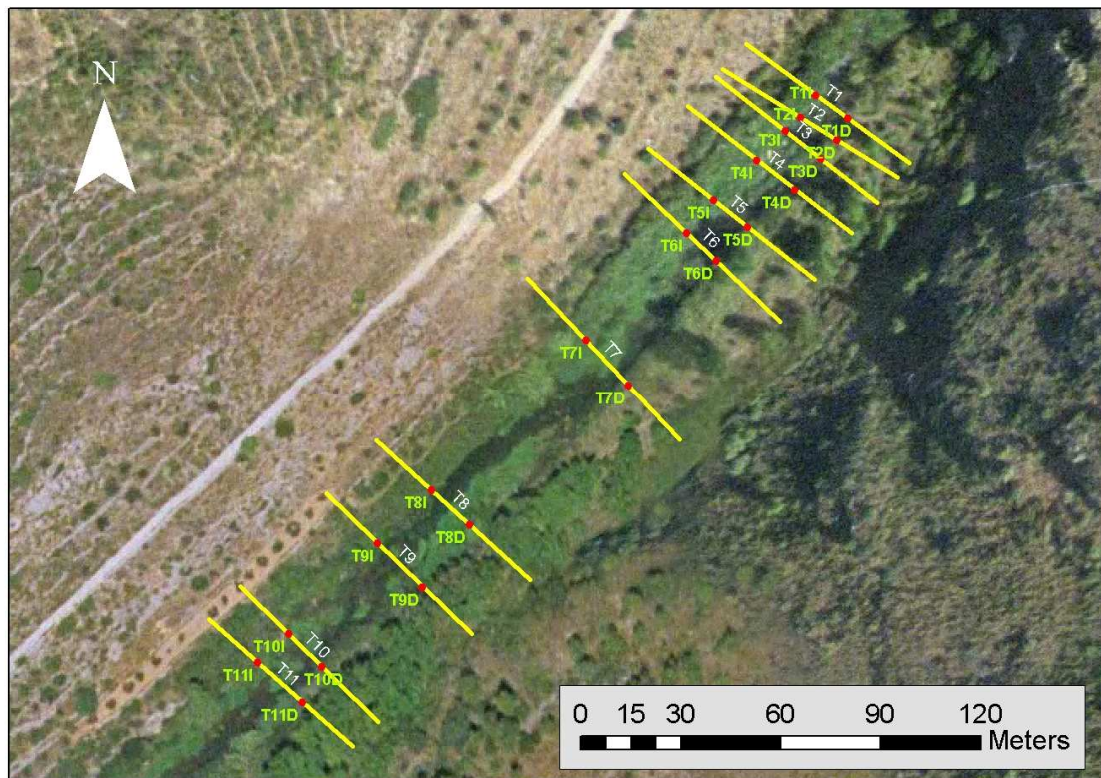


Figura 4.45. Foto aérea del tramo de estudio Lorcha (río Serpis).

#### 4.1.2.- Río Mijares en el Terde

Se encuentra en las inmediaciones del Molinete, antiguo molino transformado actualmente en casa de turismo rural. A estas instalaciones se accede desde la carretera que une Mora de Rubielos con la autovía A-23 desviándonos a la derecha pasados 10.5 kilómetros desde Mora de Rubielos. Ambos tramos del Mijares alto pertenecen a masa de agua 10.03. Tiene una longitud de unos 218.2 m. La llanura de inundación es relativamente estrecha en comparación con el tramo Terde 2, teniendo mayor anchura de ribera en la margen derecha. La accesibilidad al tramo es sencilla. Una vez en la casa rural El Molinillo es posible acceder al río, se desciende unos 500 m aguas abajo encontrando la cola del tramo de Terde.

Este tramo, como es propio de tramos no regulados, tiene una presencia mayoritaria de corrientes. El sustrato es bastante variado, predominando la grava, cantos y bloques. Los elementos de heterogeneidad en el cauce son diversos, destaca la presencia de troncos, ramas y diques naturales así como la presencia de cauces secundarios por los que circula habitualmente el agua.

Es un tramo corto (218.2 m.) con la presencia de una isla. Las laderas como se ha mencionado antes son escarpadas en la margen izquierda. En la margen derecha se aprecia la presencia de antiguas terrazas. Actualmente no se pueden observar restos de actividades agrícolas ni pies de especies agrícolas asilvestrados.

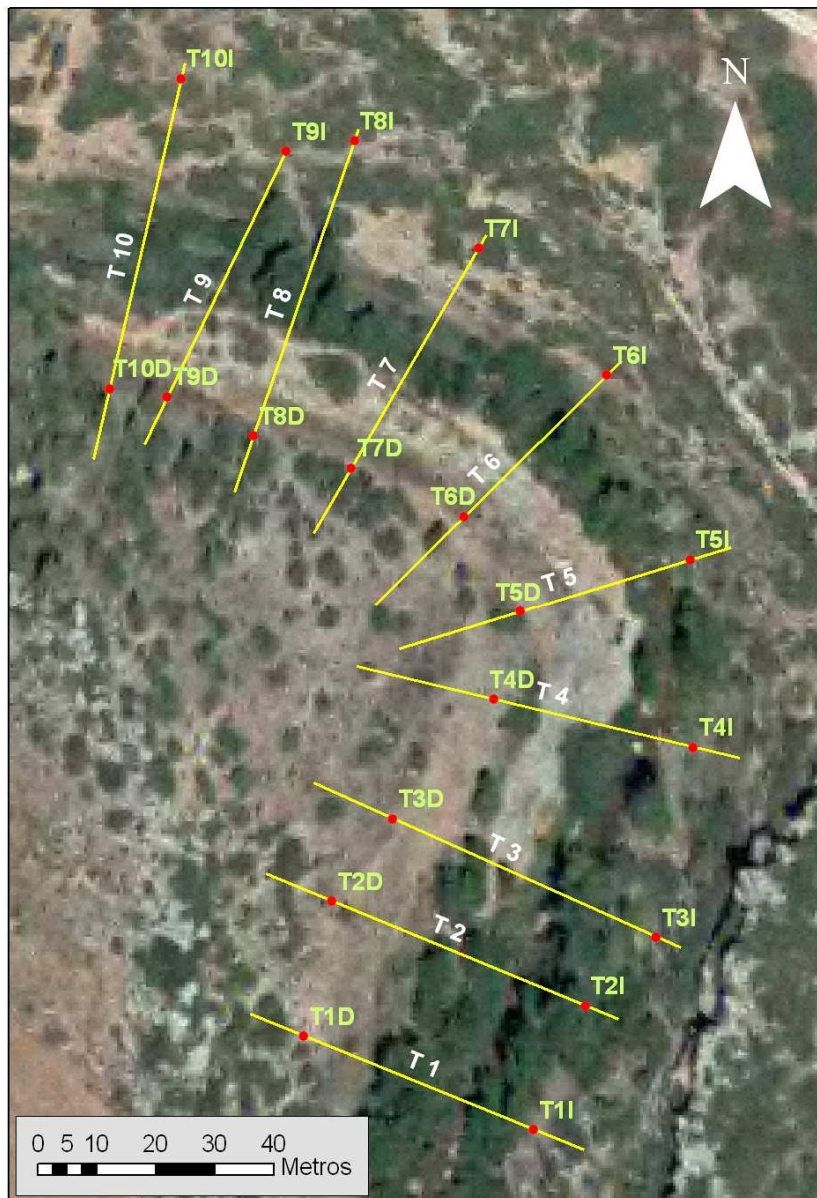


Figura 4.46. Foto aérea de la parte norte del tramo de estudio Terde (río Mijares).

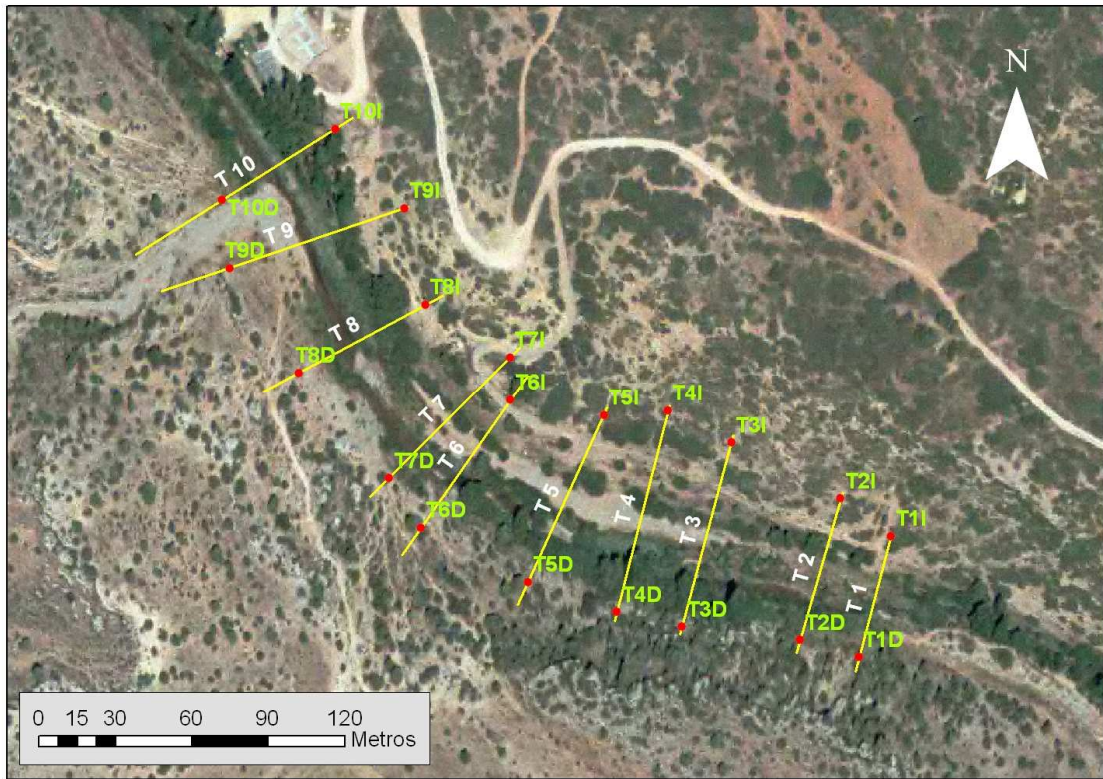


Figura 4.47. Foto aérea de la parte norte del tramo de estudio Terde (río Mijares).



## 4.2.- ESTIMACIÓN DE PARÁMETROS

### 4.2.1.- *Parámetros de Suelo*

Indican las características del suelo para cada punto de simulación. El tramo de estudio se ha dividido en zonas donde se agrupan los suelos con similares características. Se han recogido muestras de suelo para cada una de esas zonas. Las muestras se extraen mediante la excavación de una cata de al menos 50 cm de profundidad. Se extrae el suelo a esa profundidad porque se evita así las capas de suelo superficiales que tengan horizontes orgánicos o compactados, que no sean representativos con el suelo que está en contacto con las raíces, cuyas características si son de interés para el modelo.

El suelo contenido en el tanque descrito anteriormente podrá ser saturada o no saturado dependiendo de la cota del nivel freático en ese momento (WTE). La zona saturada es la que queda por encima del nivel freático será y en esta zona los poros de suelo quedan saturados de agua y las presiones serán positivas. La zona no saturada queda por encima del nivel freático y sus poros no están saturados de agua. A pesar de que las presiones en esta zona son negativas (al predominar las fuerzas de succión), en este modelo se toman los valores absolutos para estas presiones.

Los parámetros de suelo del modelo son obtenidos previamente a partir de los parámetros básicos de las muestras de suelo. Los parámetros básicos de suelo son:

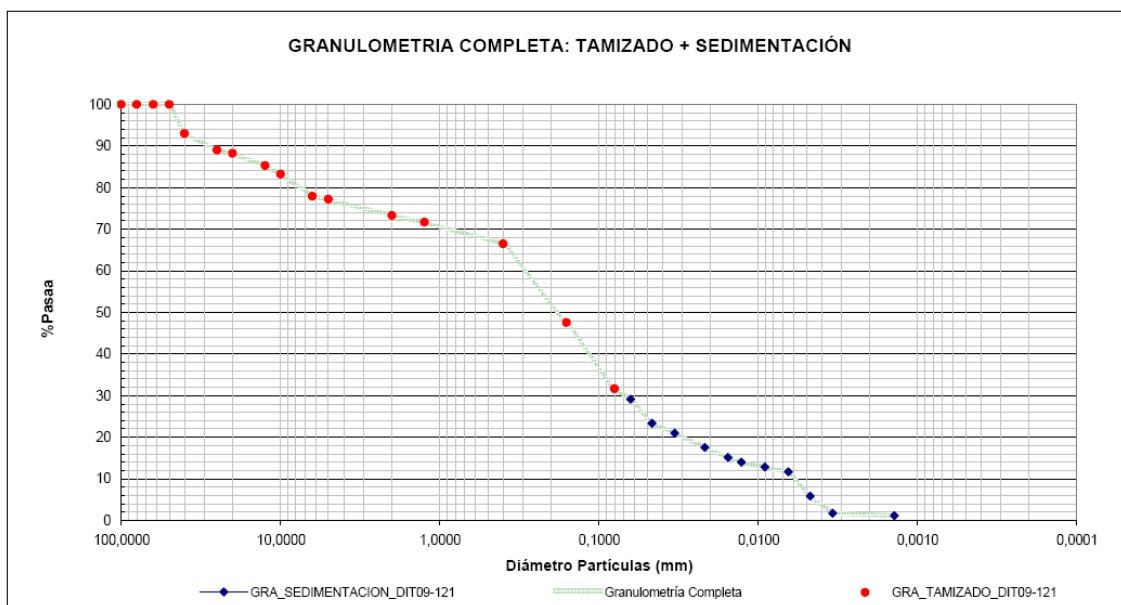
- Porcentaje de gravas (partículas con tamaño  $>2$  mm)
- Porcentaje de arenas ( $0,05$  mm  $<$  tamaño  $<2$  mm)
- Porcentaje de limos ( $0,002$  mm  $<$  tamaño  $<0,05$  mm)
- Porcentaje de arcilla (tamaño  $< 0,002$ )
- Porcentaje de materia orgánica

El valor de estos parámetros de suelo se obtiene mediante análisis de laboratorio. Los análisis realizados son los siguientes:

- *Análisis granulométrico por tamizado:* Se utiliza para determinar el porcentaje de limos y arcillas.
- *Análisis granulométrico de suelos finos por sedimentación/ método del densímetro:* Se utiliza para determinar el porcentaje de gravas y arenas.

- *Determinación del contenido de materia orgánica de un suelo por el método del permanganato potásico:* Se utiliza para determinar el porcentaje de materia orgánica.

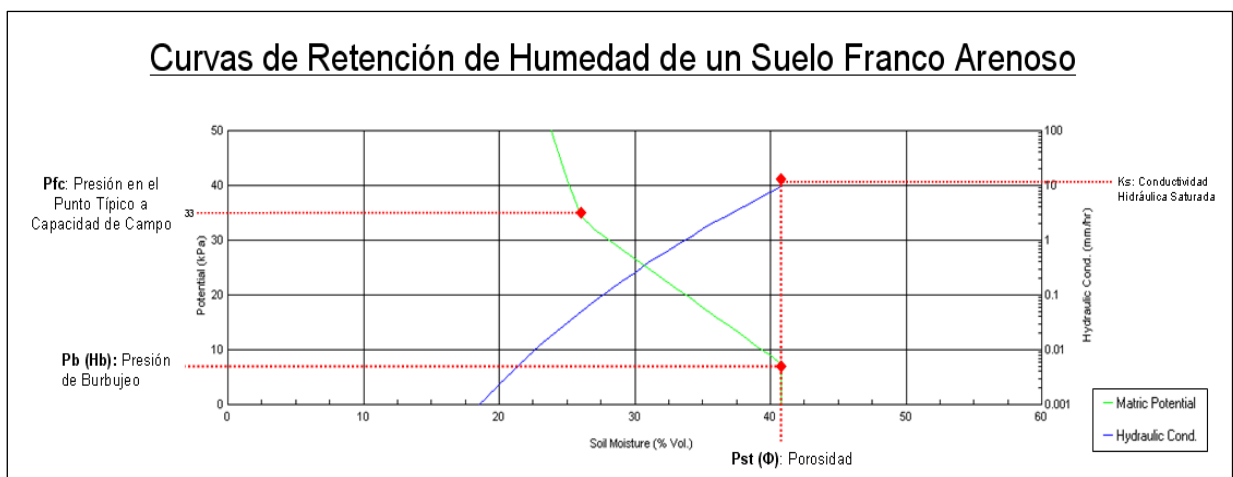
A continuación se muestra una gráfica donde se representa a modo de ejemplo el resultado de un análisis granulométrico tanto en el ensayo por sedimentación como por tamizado:



**Figura 4.48. Resultado del análisis granulométrico del suelo de la muestra 1 del tramo de Terde-1 (río Mijares).**

Posteriormente los parámetros básicos son introducidos en el Modelo: "Soil Water Characteristics." Este modelo ha sido desarrollado por Saxton et al 2006 para el USDA (Departamento de Agricultura de EEUU). Se ha escogido este modelo debido a que tiene en cuenta los porcentajes de gravas y de materia orgánica, parámetros que son muy importantes en los suelos de ribera. Además de los parámetros mencionados anteriormente, este modelo también puede contemplar otros parámetros de entrada: como la salinidad (dS/m) y el grado de compactación del suelo. La salinidad no se ha tenido en cuenta debido a que los tramos de estudio se encuentran en zonas medias y altas de los ríos, donde no hay problemas de intrusión marina y donde no se han observado otros tipos de contaminación por sales. El grado de compactación es un factor multiplicador que tampoco se ha tenido en cuenta, debido a que las muestras de suelo de la zona radicular tenían compacidades normales o estándar. Cabe decir que antes de introducir los porcentajes granulométricos en este modelo, hay que separar el porcentaje de gravas del resto debido a que este modelo toma el sumatorio de arena, limo y arcilla como 100. El modelo "Soil Water Characteristics" proporciona a partir de sus datos de salida y de sus curvas de humedad los siguientes parámetros del modelo:

- **Mfc ( $\theta_{FC}$ ):** Humedad a Capacidad de Campo ( ). Como se ha mencionado antes es la cantidad máxima que puede retener un suelo sin dañar a las plantas por anaerobiosis. Indica la capacidad máxima del tanque del modelo y se haya para el valor típico de humedad a 33Kpa.
- **Pst ( $\Phi$ ):** Porosidad ( ). Es la proporción del volumen de suelo ocupado por poros (huecos) en comparación con la fracción sólida. Los poros podrán ser ocupados tanto por aire como por agua dependiendo del grado de humedad.
- **Ks:** Conductividad Hidráulica Saturada del Suelo (mm/hr). En suelos no saturados, la Conductividad Hidráulica es variable y dependiente del grado de humedad. Aun así en condiciones de saturación la conductividad es constante para cada tipo de suelo. En términos generales se podría definir a la conductividad hidráulica como la distancia recorrida por el flujo de agua por unidad de tiempo.
- **Pb (Hb):** Presión de Burbujeo (en valor absoluto) (Kpa). Es la presión ejercida por las burbujas de aire en zonas donde el suelo está en condiciones de saturación. Esto es debido a que, aunque el suelo este saturado, siempre habrá zonas donde hayan quedado burbujas de aire atrapadas que todavía ejerzan una presión matricial residual.
- **Ip ( $\lambda$ ):** Índice de Porosidad ( ). El parámetro de la porosidad anteriormente descrita indica exclusivamente la proporción de poros con respecto a la fracción sólida. Por ello se utiliza adicionalmente el parámetro del índice de porosidad porque indica el tamaño y la interconectividad entre los poros. Debido a las tensiones capilares, este parámetro resulta



fundamental para determinar tanto la presión matricial como la conductividad hidráulica para suelo no saturado.

En la figura siguiente se puede observar los parámetros de suelo anteriormente citados representados gráficamente:

**Figura 4.49.** Curvas de retención de humedad según las ecuaciones de Campbell.

Por último se va mostrar a modo de ejemplo una tabla donde se muestra los parámetros de suelo del modelo para todas las muestras obtenidas en el tramo de Terde-1 (río Mijares). Las tablas de parámetros de suelo del modelo para los demás tramos se muestran en el anexo correspondiente.

**Tabla 4.15 Valores de los parámetros de suelo del tramo de Terde-1 (río Mijares)**

Código del Suelo	Porosidad	Índice de Porosidad	Presión de Burbujeo	Conductividad Hidráulica Saturada del Suelo	Humedad a capacidad de Campo	Profundidad a considerar en el Ascenso Hidraulico Capilar
	Pst ( )	Ip ( )	Pb (Kpa)	Ks (mm/hr)	Mfc ( )	Zcmin (m)
1	0.397	0.530	3.848	56.24	0.131	17
2	0.436	0.208	0.342	22.65	0.168	17
3	0.441	0.226	2.625	19.91	0.247	17
4	0.406	0.277	1.037	42.76	0.154	17
5	0.412	0.220	0.057	28.19	0.102	17
6	0.432	0.184	0.246	37.09	0.174	17
7	0.414	0.251	0.024	97.96	0.069	17
8	0.423	0.231	0.056	34.45	0.095	17
9	0.398	0.278	2.370	26.83	0.19	17
10	0.403	0.206	0.045	25.06	0.104	17

### Datos Hidrometeorológicos

Son los datos de entrada relacionados con los procesos hidrológicos o meteorológicos:

- (PP): Precipitación Diaria (mm). (descrito anteriormente)
- (ETP): Evapotranspiración Potencial (mm). (descrito anteriormente)
- Qcurve(j): Valores del Caudal de Río de la Curva de Gasto (m<sup>3</sup>/s).
- Hcurve(j): Valores de Nivel de Río de la Curva de Gasto (m)

- $Q_{daily}(t)$ : Caudal Diario Observado en el Río (m<sup>3</sup>/s). Son los datos del caudal de río que recojen diariamente las estaciones de aforo de las demarcaciones hidrográficas.
- $WTE(t)$ : Cota del Nivel Freático Calculada (m). Se calcula mediante interpolación lineal introduciendo los datos anteriores en la siguiente ecuación:

$$WTE_{(t)} = H_{curve_{(j-1)}} + \left( \frac{Q_{daily_{(t)}} - Q_{curve_{(j-1)}}}{Q_{curve_{(j)}} - Q_{curve_{(j-1)}}} \right) \cdot (H_{curve_{(j)}} - H_{curve_{(j-1)}})$$

### **Parámetros de Vegetación**

A continuación se describen los parámetros de vegetación que podrán variar según el tipo funcional de vegetación.

- $P_{crit}$ : Presión para el Punto de Humedad Crítica (KPa). Indica la presión en la cual la planta deja de evapotranspirar más aunque reciba más humedad. El valor típico es de 95Kpa.
- $P_{wp}$ : Presión para el Punto de Marchitez Permanente (KPa). Es la presión a la cual las plantas no pueden extraer más agua del suelo y pierden su turgencia. El valor típico es de 1500 Kpa.
- $Cov (\lambda v)$ : Fracción de Cobertura de las Plantas ( ). Indica el grado de cobertura que ejercen las plantas sobre la superficie del suelo. Aunque las plantas tiendan a colonizar toda la superficie del suelo hay zonas en las que o no pueden o lo realizar con mucha dificultad en casos como afloramientos rocosos o compactación del suelo por tránsito de animales.
- $Crt$ : Conductividad Máxima de Agua Raíz-Suelo (mmMpa-1h-1). Es un factor multiplicador que mide la eficiencia en el flujo de agua desde el suelo a las raíces para unas determinadas condiciones de presión.
- $Zr$ : Profundidad Máxima de Raíces (m). Indica la profundidad máxima a la que las raíces pueden acceder al agua del nivel freático.
- $Ze$ : Profundidad Efectiva de Raíces (m) Es la profundidad máxima a la que se encuentran las raíces que captan agua y evapotranspiran de la zona no saturada de suelo, que queda por encima del nivel freático.

- **Zsat:** Profundidad de Extinción por Saturación (m). Es la profundidad máxima a la cual las plantas pueden tolerar tener el suelo saturado de agua sin ser perjudicadas debido a la asfixia

#### 1.1.1.5.- Clasificación utilizada para los de tipos funcionales de vegetación

Para los tramos de estudio de la demarcación del río Júcar se ha realizado una clasificación con 4 tipos funcionales de vegetación. Esta clasificación está basada en el hábito de crecimiento de las especies vegetales, en la profundidad radicular, en la demanda de agua e indirectamente en la posición con respecto al cauce. Dicha clasificación es la siguiente:

- **Herbáceas de Ribera (RH):** Son las plantas sin tejidos leñosos que requieren suelos en condiciones de saturación permanente. Tienen un sistema radicular generalmente poco profundo y por ello se encuentran normalmente en zonas más cercanas a la lámina de agua del río, donde hay una elevación reducida con respecto al nivel freático. En este grupo se encuentran tanto las herbáceas de reducido tamaño (*Apium nodiflorum*) como otras de tamaño considerable como son las cañas autóctonas (*Arundo donax*) que pueden llegar a tener hasta los 5 metros de altura. Se considera el estado más regresivo de la vegetación de ribera después del suelo desnudo.
- **Juveniles de ribera, pequeños arbustos y enredaderas (RJ):** Los juveniles de ribera son individuos de pocos años de edad de especies leñosas (*Populus* spp y *Salix* spp) que durante las fases avanzadas de su desarrollo formarán parte de los estados más evolucionados del ecosistema fluvial. Aunque en esta fase tienen un porte todavía reducido, la longitud de las raíces es bastante elevada en comparación con la parte aérea, debido a su búsqueda del nivel freático. En este tipo funcional también se incluye a los pequeños arbustos (o matorrales) y a las plantas enredaderas (o lianas). Esto es debido a que estas plantas, también leñosas, tienen un desarrollo radicular que se asemeja más a las plantas juveniles que a las plantas arbóreas y arbustivas de gran tamaño.
- **Árboles adultos de ribera y grandes arbustos (RA):** En este grupo se incluyen las especies leñosas (tanto arbóreas como arbustivas) de gran tamaño y en estado maduro. Representan el estado más evolucionado del ecosistema de ribera, lo que se conoce como bosque de galería. Las raíces son de gran longitud y tienen una eficiencia bastante elevada para extraer agua del nivel freático. Por otro lado tienen cierta tolerancia a la asfixia radicular, pero aun así en menor grado que las herbáceas de ribera.

- **Vegetación terrestre (TV):** Se incluyen las especies vegetales (herbáceas, arbustivas y arbóreas), que no forman parte del ecosistema acuático propiamente dicho. Son las especies que poblarían la zona (dependiendo de las características climatológicas y edafológicas) si no hubiera un curso de agua y si el nivel freático no fuera somero. El límite entre la vegetación terrestre y la de ribera resulta fundamental para determinar el contorno de la zona de estudio. En algunas zonas existen parches de transición donde hay plantas tanto terrestres como riparias. Por ello estos parches intermedios son muy útiles para determinar la influencia que tiene la orografía sobre las series temporales de caudales del río.

En la siguiente tabla se muestra la vegetación observada en cada uno de los parches del tramo de estudio de Lorcha (río Sérpis). Para cada parche se muestra: las especies o grupos de especies más representativos, el tipo grupo funcional en el que han sido clasificados y el grado de cobertura vegetal de las copas.

**Tabla 4.16.** *Parches de Vegetación del tramo Lorcha del río Serpis*

Código Parche	Vegetación Observada más Representativa	Tipo Funcional Observado	Grado de Cobertura Vegetal de las Copas ( )
1	Herbazal de ribera en zona encharcada permanentemente	RH	1.00
2	Adelfas	RA	1.00
3	Herbazal de ribera	RH	1.00
4	Herbazal en isla con gran impacto de inundación	RH	0.90
5	Cañaveral	RH	1.00
7	Herbazal de ribera	RH	0.95
8	Carrizal con regeneración de Sauces (impacto de avenida alto)	RH-RJ	1.00
9	Herbazal de ribera	RH	0.75
10	Sauces (restos acumulados en troncos por arrastre en avenida)	RJ	1.00
11	Cañaveral	RH	1.00
12	Zarzal con Cañaveral	RH-RJ	1.00
13	Adelfas y Majuelos - Impacto de avenida	RA	1.00
14	Adelfas, Lentiscos y Pinos	TV	0.95
15	Herbáceas de ribera en lengua de grava	RH	0.80
16	Chopos, Sauces y Majuelos	RA	1.00
17	Carrizal y pequeña cantidad de Apium nodiflorum	RH	0.95
18	Herbazal mezclado con Chopos, Sauces y Adelfas	RH-RA	1.00
19	Herbazal húmedo de Apium nodiflorum	RH	0.80
20	Cañaveral en isla	RH	1.00
21	Mezcla de Chopos con vegetación terrestre (Pinos, Lentiscos...)	RA-TV	1.00
23	Sauces, Majuelos y Adelfas	RA	0.70
24	Lentiscos	TV	1.00
25	Majuelos y Zarzas	RJ-RA	1.00

Código Parche	Vegetación Observada más Representativa	Tipo Funcional Observado	Grado de Cobertura Vegetal de las Copas ( )
26	Lentiscos con Adelfas	RA-TV	1.00
27	Zarzal con Adelfas y Sauces	RJ-RA	1.00
29	Carrizal y Zarzas	RH-RJ	1.00
30	Lentiscos con Adelfas	TV	0.90
31	Adelfas, Zarzas, Majuelos y Rosal silvestre	RJ-RA	1.00
32	Zarzal	RJ	0.60
33	Carrizal mas herbazal de orilla	RH	1.00
34	Sauces, Chopos, Adelfas	RA	1.00
35	Carrizo y Herbazal	RH	1.00
36	Eneas	RH	1.00
37	Adelfas, Zarzas y algo de Equisetos	RJ-RA	1.00
38	Herbazal de ribera	RH	1.00
39	Chopos y Vegetación Terrestre	RA-TV	1.00
40	Chopos, Zarzas y otras Herbáceas	RA	1.00
41	Herbazal de ribera	RH	0.80
42	Regeneración de Chopos y Herbazal de ribera	RH-RJ	1.00
43	Adelfas y Juncos	RH-RA	1.00
44	Mezcla de Sauces juveniles y adultos	RJ-RA	1.00
45	Mezcla de Chopos adultos y juveniles	RJ-RA	0.85
46	Herbazal de ribera	RH	0.90
47	Sauces y Chopos adultos	RA	1.00
48	Herbazal de ribera	RH	1.00
49	Adelfar con Zarzas y Juncos	RJ-RA	1.00
50	Adelfar con Lentiscos y Aladiernos	RA-TV	1.00
51	Sauces y Zarzas	RJ-RA	0.95
52	Adelfas con Juncos	RH-RA	0.85
53	Zarzal	RH-RJ	1.00
54	Ecosistema forestal del contorno	TV	1.00
55	Ecosistema forestal del contorno	TV	1.00

En la siguiente tabla se muestran los nombres científicos de las especies mencionadas anteriormente:

**Tabla 4.17:** Nombres comunes y científicos del tramo Lorcha de río Sêrpis

Nombre Común	Nombre Científico
Adelfa	Nerium oleander
Aladierno	Rhamnus alaternus
Caña	Arundo donax



Carrizo	<i>Phragmites australis</i>
Chopos	<i>Populus sp.</i>
Enea	<i>Thypha latifolia</i>
Junco Común	<i>Scirpus holoschoenus</i>
Lentisco	<i>Pistacia lentiscus</i>
Majuelos	<i>Crataegus monogyna</i>
Pino	<i>Pinus halepensis</i>
Rosal Silvestre	<i>Rosa canina</i>
Sauces	<i>Salix sp.</i>
Sauco	<i>Sambucus sp.</i>
Zarzas (Zarzamora)	<i>Rubus ulmifolius</i>

---

A continuación se muestra un mapa del tramo del estudio de Lorcha con los parches de vegetación clasificados en tipos funcionales:

### 4.3.- ANÁLISIS DE SENSIBILIDAD

El análisis de sensibilidad es el primer paso para reconocer la incertidumbre. Es utilizado para examinar cómo el cambio en un parámetro (input) afecta a un resultado (output). Esto permite identificar las variables más críticas o construir escenarios posibles que permitirán analizar el comportamiento de un resultado bajo diferentes supuestos. Este análisis permite medir el cambio en un resultado dado, un cambio en un conjunto de variables. Todo ello tanto en términos relativos como en términos absolutos.

Los objetivos del análisis de sensibilidad son:

- Identificar las variables que más influyen en el modelo (más sensibles o críticas).
- Identificar dónde se debe dedicar más esfuerzos tanto en el proceso de planificación como en el de control y seguimiento de una decisión.
- Identificar las variables que deben ser incluidas en la creación de escenarios o en las simulaciones posteriores.

#### 4.3.1.- *Planteamiento del análisis de sensibilidad*

Para este modelo se va a realizar un análisis de sensibilidad donde se determinará la “importancia” de una serie de parámetros de vegetación, en cuanto a como influyen sobre el resultado final (variable de salida), y todo ello para varios escenarios.

##### 4.3.1.1.- *Parámetros de Vegetación*

Los parámetros de vegetación que serán evaluados en este análisis son:

- **Rj: Factor de Transpiración de la Zona Saturada** ( ). Es un factor multiplicador sobre la eficiencia en las raíces de las plantas de un determinado grupo funcional en los casos en los que estas se encuentran en la zona saturada (debido al ascenso temporal del nivel freático).
- **Ri: Factor de Transpiración de la Zona No Saturada** ( ). Es otro factor multiplicador sobre la eficiencia pero en los casos en los que las raíces se encuentran en la zona saturada.

- **CRT: Conductividad Máxima de Agua Raíz-Suelo** (mmMpa<sup>-1</sup>h<sup>-1</sup>). Es un factor multiplicador que mide la eficiencia en el flujo de agua desde el suelo a las raíces para unas determinadas condiciones de presión.
- **Zr: Profundidad Máxima de Raíces** (m). Es un parámetro específico para cada tipo funcional de vegetación. Aunque describe la profundidad máxima a la que llegan las raíces, las raíces más profundas no intervienen en todos los procesos fisiológicos que son considerados en el modelo.
- **Ze: Profundidad Efectiva de Raíces** (m). Es la profundidad máxima a la que se encuentran las raíces que intervienen en la Evapotranspiración Real (ETR) de forma considerable.
- **Zsat: Profundidad de Extinción por Saturación** (m). Es la cota máxima a la cual las plantas pueden tolerar tener el suelo saturado de agua sin ser perjudicadas debido a la asfixia radicular.

#### 4.3.1.2.- Variable de Salida Principal del Modelo

La variable de salida (output) más importante del modelo es la Evapotranspiración Real (ETR). Dicha variable expresa la cantidad de agua diaria (mm) que utiliza una planta bajo unas determinadas condiciones y en un determinado estado vegetativo. Su importancia dentro del modelo es debida a que indica el grado de adecuación de una planta (o grupo funcional) a unas condiciones ambientales, por lo que estará directamente relacionada con el valor observado en el campo, que es la ausencia/presencia de un determinado grupo funcional. Por ello esta variable de salida será fundamental cuando se deseen realizar las calibraciones, validaciones y simulaciones del modelo.

Para determinar la sensibilidad de la Evapotranspiración Real (ETR) se va a utilizar un Índice que será el cociente entre esta y la Evapotranspiración Potencial (ETP). La ETP indica la cantidad máxima de agua que puede evapotranspirar una planta en condiciones ambientales óptimas y está regulada solamente por condiciones meteorológicas. Por tanto la ecuación del Índice de Evapotranspiración ( $ET_{ind}$ ) es la siguiente:

$$ET_{ind} = \frac{ETR_{real}}{ETP_{potencial}}$$

Cuanto más varíe el ( $ET_{ind}$ ) ante la modificación de un determinado parámetro, se considerará a dicho parámetro como más sensible.

#### 4.3.1.3.- Escenarios Utilizados durante el Análisis de Sensibilidad

Para el análisis de sensibilidad se realizaron 40 simulaciones. Se tomaron 20 puntos de simulación de un transecto del tramo de estudio. Cada uno de esos puntos tenía una cota distinta (*y por tanto una elevación distinta sobre el nivel freático*). De esos 20 puntos 13 de ellos tenían un suelo limo-arenoso (llamado "Suelo 4") y 7 de ellos un suelo arenoso con muchas gravas (llamado "Suelo 10").

#### 4.3.1.3.1.- Escenarios de suelos

Los dos tipos de suelos tendrán unos parámetros básicos obtenidos en el laboratorio que son los % de: gravas, arena, arcillas y materia orgánica. Los parámetros básicos para los suelos utilizados en este análisis son los siguientes:

**Tabla 4.18. Parámetros Básicos de Suelo**

Código Muestra	Parámetros Básicos				Observaciones de los Suelos
	Gravas >2mm	Arenas 0,05<S < 2mm	Arcillas C< 0,002 mm	Materia Orgánica (% Peso)	
4	0.1	30.6	15.9	1.96	limo-arenoso
10	59.5	67.2	4.9	3.41	arenoso con muchas gravas

A partir de estos últimos parámetros se han obtenido otros (*mediante el modelo "Soil Water Characteristics"*) que serán los parámetros del modelo y son:

- Pst: Porosidad ( )
- Pb: Presión de Burbujeo (Kpa)
- Ip: Índice de Porosidad ( )
- Ks: Conductividad Hidráulica Saturada del Suelo (mm/hr)
- Mfc: Humedad a Capacidad de Campo ( )

Los parámetros de suelo del modelo para los suelos utilizados en este análisis son los siguientes:

**Tabla 4.19. Parámetros Básicos de Suelo**

Código Muestra	Parámetros del Modelo					Observaciones de los Suelos
	Presión de Burbujeo Pb (Kpa)	Porosidad Pst ( )	Índice de Poros Ip ( )	Conductividad Hidráulica Saturada Ks (mm/hr)	Humedad a Capacidad de Campo Mfc ( )	
4	4.343	0.451	0.238	15.81	0.277	limo-arenoso
10	0.516	0.486	0.26	40.02	0.164	arenoso con gravas

##### 4.3.1.3.1.1 Escenarios con distintos regímenes de caudal

En el modelo se incluyen datos de caudal de río de una estación de aforo cercana. En el análisis de sensibilidad se considerarán dos regímenes de caudal distintos (natural y alterado). Para el régimen natural se tomarán datos de caudales de años previos a la construcción de la presa de Beniarrés, concretamente de los años: 1943-1949. Para el régimen alterado se tomarán datos recientes de los años 2000-2006. En la tabla siguiente tabla se muestran las funciones estadísticas de los datos de

caudal tanto para régimen natural como alterado. Se observa que para el régimen natural (sin regulación de la presa) la medida de la dispersión de la varianza es bastante más elevada y el valor máximo mucho más extremo.

**Tabla 4.20. Comparación de Valores entre Régimen Natural y Alterado**

Función Estadística	Regímenes	
	Natural	Alterado
Media	4.431	1.151
Mediana	2.17	0.788
Varianza	304.1	6.579
Máximo	642.96	64.06
Mínimo	0.022	0.033

Los valores de caudal son fundamentales en el modelo, porque mediante las curvas de aforo permiten obtener los niveles de río diarios. Como los puntos de simulación son cercanos al curso fluvial, se considera que el nivel freático es horizontal y sus valores equivalentes a los del nivel de río. La profundidad del nivel freático y su variación temporal es una variable muy importante en el modelo debido a las adaptaciones que tienen los distintos grupos funcionales de vegetación para los periodos tanto de sequía como de inundación.

#### 4.3.1.3.2.- Escenarios para distintos puntos con distinta cota

La cota de los 20 puntos de simulación es también muy importante en este modelo porque es necesaria para determinar la afección del ascenso/descenso del nivel freático sobre la zona radicular de las plantas que componen los grupos funcionales de vegetación. En la siguiente tabla se muestra la cota de cada uno de los puntos en metros sobre el nivel del mar. Adicionalmente se muestra la elevación de los puntos con respecto al nivel medio de río para ambos regímenes de caudal. También se muestra el tipo de suelo que es considerado para cada uno de dichos puntos.

**Tabla 4.21. Datos sobre los puntos de simulación elegidos para el análisis de sensibilidad**

Cota del Punto de Simulación (m.s.n.m.)	Elevación sobre el Nivel del Río Promedio (m)		Código del Suelo del Punto de Simulación
	En Régimen Natural	En Régimen Alterado	
238.247	7.754	7.955	4
237.497	7.004	7.205	4
236.747	6.254	6.455	4
235.997	5.504	5.705	4
235.248	4.755	4.956	4
234.498	4.005	4.206	4
233.747	3.254	3.455	4
232.997	2.504	2.705	4

232.247	1.754	1.955	4
232.116	1.623	1.824	4
231.967	1.474	1.675	4
231.819	1.326	1.527	4
230.952	0.459	0.66	4
230.606	0.113	0.314	10
231.631	1.138	1.339	10
232.098	1.605	1.806	10
232.953	2.46	2.661	10
233.532	3.039	3.24	10
233.788	3.295	3.496	10
234.693	4.2	4.401	10

#### **4.3.2.- Modificaciones de los valores de los parámetros**

Para cada uno de los 4 tipos funcionales de vegetación (*TV/RA/RJ/RH*) se va a modificar 2 veces cada uno de los parámetros (*Zr/ Ze/ Zsat/ Ri/ Rj/ CRT*). En una de ellas se modificará el valor inicial un 30% por encima y en la otra un 30% por debajo del valor inicial. Al valor 30% superior se le aplicará el símbolo "+" y al valor 30% inferior el símbolo "-". A continuación se mostrará unas tablas para cada tipo funcional, donde aparecen los valores modificados (arriba/debajo) de los parámetros intervinientes en este análisis. Las celdas de los valores modificados han sido resaltadas en un color diferente.

Tabla 4.22. *Modificaciones de los parámetros del tipo funcional de Vegetación Terrestre (TV)*

TIPO FUNCIONAL:		TV (Vegetación Terrestre)				
Identificación Grupo	Profundidad Máxima de Raíces (m)	Profundidad Efectiva de Raíces (m)	Profundidad de Extinción por Saturación (m)	Factor de Transpiración de la Zona No Saturada ( )	Factor de Transpiración de la Zona Saturada ( )	Conductividad Máxima de Agua Raíz-Suelo (mmMpa-1h-1)
	Zr	Ze	Psat	Ri	Rj	CRT
Valores Iniciales	2	1.5	0.1	1	0.1	0.97
Zr+	2.6	1.5	0.1	1	0.1	0.97
Ze+	2	1.95	0.1	1	0.1	0.97
Psat+	2	1.5	0.13	1	0.1	0.97
Ri+	2	1.5	0.1	1.3	0.1	0.97
Rj+	2	1.5	0.1	1	0.13	0.97
CRT+	2	1.5	0.1	1	0.1	1.261
Zr-	1.4	1.5	0.1	1	0.1	0.97
Ze-	2	1.05	0.1	1	0.1	0.97
Psat-	2	1.5	0.07	1	0.1	0.97
Ri-	2	1.5	0.1	0.7	0.1	0.97
Rj-	2	1.5	0.1	1	0.07	0.97
CRT-	2	1.5	0.1	1	0.1	0.679

Tabla 4.23. *Modificaciones de los parámetros del tipo funcional de Arbóreas de Ribera (RA)*

TIPO FUNCIONAL:		RA (Arbóreas de Ribera)				
Identificación Grupo	Profundidad Máxima de Raíces (m)	Profundidad Efectiva de Raíces (m)	Profundidad de Extinción por Saturación (m)	Factor de Transpiración de la Zona No Saturada ( )	Factor de Transpiración de la Zona Saturada ( )	Conductividad Máxima de Agua Raíz-Suelo (mmMpa-1h-1)
	Zr	Ze	Psat	Ri	Rj	CRT
Valores Iniciales	6	1.5	-0.1	1	0.56	0.97
Zr+	7.8	1.5	-0.1	1	0.56	0.97
Ze+	6	1.95	-0.1	1	0.56	0.97
Psat+	6	1.5	-0.13	1	0.56	0.97
Ri+	6	1.5	-0.1	1.3	0.56	0.97
Rj+	6	1.5	-0.1	1	0.728	0.97
CRT+	6	1.5	-0.1	1	0.56	1.261
Zr-	4.2	1.5	-0.1	1	0.56	0.97
Ze-	6	1.05	-0.1	1	0.56	0.97
Psat-	6	1.5	-0.07	1	0.56	0.97
Ri-	6	1.5	-0.1	0.7	0.56	0.97
Rj-	6	1.5	-0.1	1	0.392	0.97
CRT-	6	1.5	-0.1	1	0.56	0.679



Tabla 4.24. *Modificaciones de los parámetros del tipo funcional de Juveniles de Ribera (RJ)*

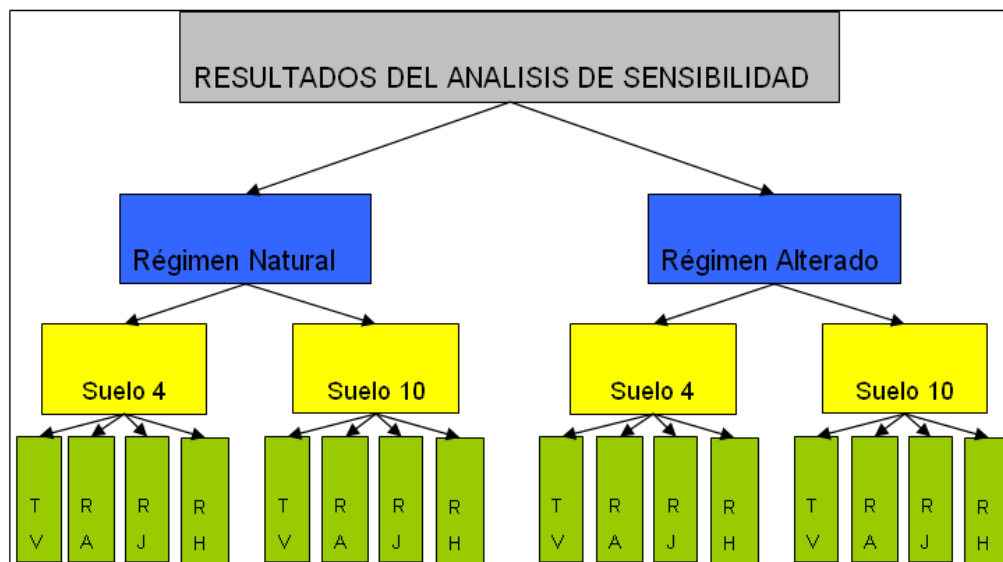
TIPO FUNCIONAL:		RJ (Juveniles de Ribera)				
Identificación Grupo	Profundidad Máxima de Raíces (m)	Profundidad Efectiva de Raíces (m)	Profundidad de Extinción por Saturación (m)	Factor de Transpiración de la Zona No Saturada ( )	Factor de Transpiración de la Zona Saturada ( )	Conductividad Máxima de Agua Raíz-Suelo (mmMpa-1h-1)
	Zr	Ze	Psat	Ri	Rj	CRT
Valores Iniciales	1.6	0.5	0	1	0.2	0.97
Zr+	2.08	0.5	0	1	0.2	0.97
Ze+	1.6	0.65	0	1	0.2	0.97
Psat+	1.6	0.5	0.3	1	0.2	0.97
Ri+	1.6	0.5	0	1.3	0.2	0.97
Rj+	1.6	0.5	0	1	0.26	0.97
CRT+	1.6	0.5	0	1	0.2	1.261
Zr-	1.12	0.5	0	1	0.2	0.97
Ze-	1.6	0.35	0	1	0.2	0.97
Psat-	1.6	0.5	-0.3	1	0.2	0.97
Ri-	1.6	0.5	0	0.7	0.2	0.97
Rj-	1.6	0.5	0	1	0.14	0.97
CRT-	1.6	0.5	0	1	0.2	0.679

Tabla 4.25. *Modificaciones de los parámetros del tipo funcional de Herbáceas de Ribera (RH)*

TIPO FUNCIONAL:		RH (Herbáceas de Ribera)				
Identificación Grupo	Profundidad Máxima de Raíces (m)	Profundidad Efectiva de Raíces (m)	Profundidad de Extinción por Saturación (m)	Factor de Transpiración de la Zona No Saturada ( )	Factor de Transpiración de la Zona Saturada ( )	Conductividad Máxima de Agua Raíz-Suelo (mmMpa-1h-1)
	Zr	Ze	Psat	Ri	Rj	CRT
Valores Iniciales	0.6	0.5	-0.75	1	0.2	0.97
Zr+	0.78	0.5	-0.75	1	0.2	0.97
Ze+	0.6	0.65	-0.75	1	0.2	0.97
Psat+	0.6	0.5	-0.975	1	0.2	0.97
Ri+	0.6	0.5	-0.75	1.3	0.2	0.97
Rj+	0.6	0.5	-0.75	1	0.26	0.97
CRT+	0.6	0.5	-0.75	1	0.2	1.261
Zr-	0.42	0.5	-0.75	1	0.2	0.97
Ze-	0.6	0.35	-0.75	1	0.2	0.97
Psat-	0.6	0.5	-0.525	1	0.2	0.97
Ri-	0.6	0.5	-0.75	0.7	0.2	0.97
Rj-	0.6	0.5	-0.75	1	0.14	0.97
CRT-	0.6	0.5	-0.75	1	0.2	0.679

### 4.3.3.- Resultados del análisis de sensibilidad

Este apartado trata sobre los resultados del análisis de sensibilidad presentados en 16 tablas. En la mitad de las tablas se habrá aplicado un “régimen natural” de caudales y en la otra un “régimen alterado”. Además 8 de las tablas tendrán “Suelo 4” (*limo-arenoso*) y las otras 8 un “Suelo 10” (*arenoso y gravoso*). Habrá 4 tablas por cada uno de los 4 tipos funcionales de vegetación: TV Vegetación Terrestre; RA-Arbóreas de Ribera; RJ-Juveniles de Ribera; RH-Herbáceas de Ribera. Las tablas de “Suelo 4” tendrán 13 puntos de simulación en representados en 13 filas o registros y las de “Suelo 10” tendrán 7. En primer lugar se va a representar un diagrama a modo de guía. En las tablas posteriores de resultados, se resaltará para cada fila el valor de la variación de parámetro más elevado con un color diferente.



**Figura 4.50.** Diagrama de las tablas utilizadas en el análisis de sensibilidad (ver anexo correspondiente)

A continuación se va a mostrar una tabla de ejemplo de los resultados del análisis de sensibilidad. El resto de las tablas quedarán representadas en el anexo correspondiente:

Tabla 4.26. **Ejemplo de tabla de resultados del análisis de sensibilidad**

Vegetación:		RH (Herbáceas de Ribera)							
Suelo:		4 (Limo-Arenoso)							
Régimen de Q:		Natural							
Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.754	0.310	0.00E+00	4.35E-02	0.00E+00	2.87E-02	0.00E+00	0.00E+00	Ze
237.497	7.004	0.311	0.00E+00	4.38E-02	0.00E+00	2.81E-02	0.00E+00	0.00E+00	Ze
236.747	6.254	0.313	0.00E+00	4.42E-02	0.00E+00	2.73E-02	0.00E+00	0.00E+00	Ze
235.997	5.504	0.314	0.00E+00	4.48E-02	0.00E+00	2.63E-02	0.00E+00	0.00E+00	Ze
235.248	4.755	0.317	0.00E+00	4.57E-02	0.00E+00	2.48E-02	0.00E+00	0.00E+00	Ze
234.498	4.005	0.321	0.00E+00	4.73E-02	0.00E+00	2.26E-02	0.00E+00	0.00E+00	Ze
233.747	3.254	0.328	0.00E+00	5.00E-02	0.00E+00	1.83E-02	0.00E+00	0.00E+00	Ze
232.997	2.504	0.343	2.18E-04	5.55E-02	0.00E+00	4.79E-03	0.00E+00	0.00E+00	Ze
232.247	1.754	0.417	1.59E-03	9.50E-02	1.79E-04	1.24E-03	0.00E+00	0.00E+00	Ze
232.116	1.623	0.434	1.34E-03	8.74E-02	1.79E-04	7.41E-03	0.00E+00	0.00E+00	Ze
231.967	1.474	0.454	1.80E-03	1.11E-01	1.79E-04	2.41E-02	0.00E+00	0.00E+00	Ze
231.819	1.326	0.487	2.43E-03	1.49E-01	2.25E-04	4.25E-03	0.00E+00	0.00E+00	Ze
230.952	0.459	0.999	7.45E-02	3.98E-01	0.00E+00	3.26E-01	0.00E+00	0.00E+00	Ze

#### 4.3.4.- Conclusiones del análisis de sensibilidad

Tras analizar los resultados se ha llegado a las siguientes conclusiones:

- “El parámetro de vegetación menos sensible es CRT”: Esto era ya predecible debido a que en la bibliografía consultada sobre el parámetro CRT se citaba casi siempre el valor fijo de:  $0.97 \text{ mm} \cdot \text{Mpa}^{-1} \cdot \text{h}^{-1}$
- “El parámetro de vegetación Zsat también es muy poco sensible”: Esto se aprecia sobretodo en los puntos de mayor cota o en las simulaciones con régimen alterado debido a que en estos casos la Cota de Extinción por Saturación (Cpsat) se alcanza con mayor dificultad.
- “Los parámetros claramente más sensibles son: Zr, Ze, Ri y Rj”: Para realizar las calibraciones se tendrá en cuenta especialmente a estos últimos parámetros pero también a Zsat.

#### 4.4.- CALIBRACIÓN DEL MODELO

En este apartado se describe el proceso mediante el cual se modifican los parámetros más sensibles, con el fin de que la vegetación simulada coincida lo máximo posible con la vegetación observada en campo. La vegetación ha sido clasificada en tipos funcionales de vegetación y el modelo indica, para cada punto de simulación, cuál de los tipos funcionales es más idóneo para las condiciones ambientales expuestas. Los parámetros utilizados en la calibración son todos parámetros de vegetación. Tras realizar consultas bibliográficas y tras realizar el análisis de sensibilidad se concluyó que los parámetros de vegetación más sensibles, y por tanto a utilizar en la calibración, eran los siguientes:

- Zr: Profundidad Máxima de Raíces (m)
- Ze: Profundidad Efectiva de Raíces (m)
- Zsat: Profundidad de Extinción por Saturación (m)
- Ri: Factor de Transpiración de la Zona No Saturada ( ).
- Rj: Factor de Transpiración de la Zona Saturada ( ).

La fracción de cobertura de la vegetación se ha considerado la unidad en los tipos funcionales RH, RA y TV mientras que en el caso de RJ se ha considerado 0.8 por presentar la vegetación juvenil menor densidad. La conductividad máxima de agua raíz-suelo se ha fijado en  $0.97 \text{ mmMpa}^{-1}\text{h}^{-1}$  para todos los tipos funcionales. La presión para el punto de humedad crítica se ha fijado en 500 Kpa en los tipos funcionales RH y RJ, 250 Kpa en RA y 95 Kpa en TV; mientras que la presión para el punto de marchitez permanente se ha mantenido en 1500 Kpa en todos los casos.

Se ha realizado la calibración previa del modelo en dos tramos con régimen de caudales diferentes con el fin de obtener unos parámetros de vegetación del modelo capaces de reflejar coherentemente la distribución de los tipos funcionales de vegetación en condiciones hidrológicas diversas. De este modo se ha realizado una primera calibración en el tramo Terde 1 del Río Mijares, el cual se encuentra en una zona de cabecera y con régimen natural de caudales. Paralelamente se ha realizado la calibración del modelo en el tramo Lorcha del Río Serpis, el cual está sometido a la regulación de caudales del Pantano de Beniarres. Finalmente, y teniendo en cuenta los resultados de ambas calibraciones, se han seleccionado los parámetros óptimos del modelo para la calibración definitiva del mismo.

#### 4.4.1.- **Matriz de confusión**

Para visualizar los resultados de cada uno de los pasos de la calibración se utilizará una matriz de confusión. La matriz de confusión es una herramienta de visualización donde las columnas de la matriz representan los valores simulados de cada clase, mientras que cada fila representa los valores reales.

En nuestro caso las clases de la matriz son los tipos funcionales de vegetación, tanto puros como mixtos. Para cada punto de simulación de nuestro tramo de estudio existe un valor de “*tipo de vegetación observado en campo*” y otro valor que es el de “*vegetación simulada por el modelo*”.

A medida de que se van obteniendo resultados de cada punto de simulación se añaden valores unitarios en las celdas cuyos índices de filas y columnas coinciden con las clases observadas y simuladas respectivamente.

La matriz de confusión para el proceso de calibración tiene 8 columnas y 8 filas. Como resultado habrá un total de 36 celdas que se agruparán de la siguiente manera:

- 16 celdas donde coinciden los valores simulados y observados exclusivamente para los tipos funcionales puros (*agrupación A de la tabla*).
- 16 celdas donde coinciden los tipos funcionales simulados puros con los observados mixtos (*agrupación B de la tabla*).
- 4 celdas donde coinciden los tipos funcionales simulados mixtos, con los observados mixtos (*agrupación C de la tabla*). Estas celdas solo contabilizan los valores a partir de las celdas de la agrupación B de la tabla, y tan solo en los casos en los que los valores simulados puros coincidan con algunos de los valores observados mixtos.

La diagonal principal de la matriz de confusión ha sido resaltada en gris ya que este conjunto de celdas se caracterizan porque los valores simulados coinciden con los observados. La calibración tiene como objetivo que estos valores sean lo más altos posible a costa de reducir los valores de las demás celdas (especialmente de las más alejadas de la diagonal).

Las tablas resultantes, para cada uno de los dos tramos seleccionados para la calibración del modelo, quedan representadas de la siguiente forma:

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH					-	-	-	-	
	RJ					-	-	-	-	
	RA					-	-	-	-	
	TV					-	-	-	-	
	RH+RJ									
	RH+RA									
	RJ+RA									
	RA+TV									
Total unicas									Tot obs	
Total todas									Tot sim	

Agrupación A
Agrupación B
Agrupación C

**Figura 4.51.** Matriz de confusión para tipos de vegetación mixtos.

A efectos de comprobación se ha incluido una columna adicional en el lado derecho de la tabla donde se suman los valores totales de para cada fila. Al mismo tiempo los valores totales de dichas filas son sumados en la celda inferior de dicha columna. El valor de dicha celda tiene que ser igual al número total de puntos de simulación. Igualmente se contabiliza la suma de cada uno de los tipos funcionales simulados y el total de puntos reflejados en la simulación. A partir de estos totales se han obtenido los porcentajes de acierto de cada simulación sobre cada uno de los tipos funcionales observados y el conjunto de todos ellos.

#### 4.4.2.- Proceso de Calibración en Lorcha del Río Serpis (Régimen alterado)

Aunque durante la calibración se han realizado multitud de simulaciones, a continuación sólo se van a mostrar aquí las consideradas de mayor importancia.

##### 4.4.2.1.- Simulación 1 de Lorcha

En la simulación 1ª, donde se tienen en cuenta la hipótesis inicial, el valor de los parámetros para los distintos tipos de vegetación es el siguiente:

**Tabla 4.27. Parámetros de la Simulación 1 de Lorcha**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Perit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/Mpa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.6	0.5	-0.75	1	0.2	1	0.97	1500	95
Riparian Juveniles and small Shrubs	RJ	1.6	0.5	0	1	0.2	1	0.97	1500	95
Riparian adults Trees and Shrubs	RA	6	1.5	-0.1	1	0.56	1	0.97	1500	95
Terrestrial Vegetation	TV	2	1.5	0.1	1	0	1	0.97	1500	95

Tras ejecutar el software y transferir los resultados del Índice de Evapotranspiración a la matriz de confusión, se obtienen los siguientes resultados:

Tabla 4.28. *Matriz de Simulación 1 de Lorcha*

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH	20	0	50	0	-	-	-	-	70
	RJ	0	0	5	0	-	-	-	-	5
	RA	0	0	18	0	-	-	-	-	18
	TV	0	0	120	4	-	-	-	-	124
	RH+RJ	2	0	18	0	2	-	-	-	20
	RH+RA	0	0	17	0	-	17	-	-	17
	RJ+RA	2	0	82	0	-	-	82	-	84
	RA+TV	0	0	92	0	-	-	-	92	92
	Total únicas	20	0	193	4	2	17	82	92	430
Total todas	24	0	402	4					430	

Tot obs

Tot sim

BALANCE	Simulados	Observados	% Aciertos
RH	20	70	28.57%
RJ	0	5	0.00%
RA	18	18	100.00%
TV	4	124	3.23%
RH+RJ	2	20	10.00%
RH+RA	17	17	100.00%
RJ+RA	82	84	97.62%
RA+TV	92	92	100.00%

En esta matriz se observa que los parámetros del tipo funcional RA (*Árboles Adultos y Grandes Arbustos de Ribera*), es favorecido enormemente por lo que habrá que penalizar este tipo funcional modificando sus parámetros. El resto de tipos funcionales no se simulan correctamente por lo que la hipótesis inicial se ha descartado automáticamente.

#### 4.4.2.2.- Simulación 2 de Lorcha

Debido a la gran predominancia del tipo RA, en la Simulación se han reducido los parámetros Zr (*Profundidad Máxima de Raíces*) y Rj (*Factor de Transpiración de la Zona Saturada*) de este tipo funcional. Además se han incluido diferentes modificaciones en los demás tipos funcionales con el fin de favorecerlos al máximo tras varios pasos intermedios. Los resultados evidencian mejoras sustanciales respecto a los obtenidos anteriormente con los parámetros de la hipótesis inicial. Los parámetros de calibración en este punto del proceso se han establecido como se muestra en la siguiente tabla.

Tabla 4.29. **Parámetros de la Simulación 2 de Lorcha**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Pcrit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [m m/Mpalh]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	1.2	0.6	-0.75	0.8	0.6	1	0.97	1500	95
Riparian Juveniles and small Schrubbs	RJ	1.5	0.8	0	0.9	0.35	1	0.97	1500	95
Riparian adults Trees and Schrubbs	RA	4	0.8	-0.1	0.9	0.35	1	0.97	1500	95
Terrestrial Vegetation	TV	2.2	2	0.3	1	0	1	0.97	1500	95

Tras realizar la simulación con los parámetros antes expuestos se obtienen los siguientes resultados:

Tabla 4.30. **Matriz de Simulación 2 del Lorcha**

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH	24	3	38	5	-	-	-	-	70
	RJ	0	1	2	2	-	-	-	-	5
	RA	0	0	17	1	-	-	-	-	18
	TV	0	0	12	113	-	-	-	-	125
	RH+RJ	3	3	8	6	6	-	-	-	20
	RH+RA	0	1	12	4	-	12	-	-	17
	RJ+RA	2	1	22	59	-	-	23	-	84
	RA+TV	0	3	40	49	-	-	-	89	92
	Total únicas	24	4	69	121	6	12	23	89	431
Total todas	29	12	151	239					431	

Tot obs  
Tot sim

BALANCE	Simulados	Observados	% Aciertos
RH	24	70	34.29%
RJ	1	5	20.00%
RA	17	18	94.44%
TV	113	125	90.40%
RH+RJ	6	20	30.00%
RH+RA	12	17	70.59%
RJ+RA	23	84	27.38%
RA+TV	89	92	96.74%

En esta simulación se observa que los aciertos han ido también en aumento. El tipo funcional más beneficiado ha sido el de vegetación terrestre (TV). Por otro lado los tipos RH y RJ también han mejorado sensiblemente sin perjudicar al tipo RA. Tan solo sale perjudicado el tipo RJ+RA donde varios aciertos anteriores irán a parar al tipo TV. Dadas las notables mejoras producidas en la



simulación con los parámetros aquí definidos, estos se han tenido en consideración al decidir los parámetros definitivos del modelo. Sin embargo, y a pesar de haberse obtenido aquí resultados muy satisfactorios para los tipos funcionales RA y TV principalmente, se ha decidido continuar con el proceso de calibración para lograr mejores resultados en los tipos funcionales restantes.

#### 4.4.2.3.- Simulación 3 de Lorcha

En esta simulación se realizarán los siguientes cambios a los parámetros de vegetación: El tipo funcional RH se ha limitado mediante el parámetro Zr del mismo modo que se ha tratado de compensar este efecto incrementando ligeramente su parámetro Ze. El resultado de los puntos simulados respecto a este parámetro se modifica ligeramente logrando simular un punto de simulación más que en el caso anterior.

El balance se modifica ligeramente, beneficiándose RH respecto al caso anterior. Se ha seguido el mismo razonamiento en las modificaciones realizadas a Zr y Ze en los tipos funcionales RJ y RA lográndose mejores resultados tanto en los tipos funcionales puros como en los mixtos. El tipo funcional TV ha visto reducido aquí su Zr y su Ze, a pesar de ello no se observan resultados mucho peores por verse beneficiado respecto al resto de tipos funcionales en la eficiencia de transpiración en la zona no saturada. Con estos cambios se observan mejoras en el balance de la mayoría de tipos funcionales y en ninguno de ellos se empeora notablemente el resultado. Aún así se ha tratado de afinar más en la calibración reduciendo el grado de cobertura de RJ, lo cual refleja mejor la realidad observada, y aumentando la presión para el punto de humedad crítica (Pcrit) en los tipos funcionales RH, RJ y RA. Como se describió en el apartado de análisis de sensibilidad, el modelo no es especialmente sensible a estos dos parámetros, sin embargo si se ha logrado matizar ligeramente los resultados de la simulación con estas modificaciones.

**Tabla 4.31. Parámetros de la Simulación 3 de Lorcha**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Pcrit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/MPa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.8	0.7	-0.3	0.8	0.6	1	0.97	1500	500
Riparian Juveniles and small Shrubs	RJ	1.25	0.9	0	0.9	0.35	0.8	0.97	1500	500
Riparian adults Trees and Shrubs	RA	3.5	0.9	-0.1	0.9	0.35	1	0.97	1500	250
Terrestrial Vegetation	TV	2	1.9	0.3	1	0	1	0.97	1500	95

Tras ejecutar el software con los anteriores parámetros se obtiene la siguiente matriz:

Tabla 4.32. *Matriz de Simulación 3 de Lorcha*

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH	25	3	39	3	-	-	-	-	70
	RJ	0	2	1	2	-	-	-	-	5
	RA	0	0	17	1	-	-	-	-	18
	TV	0	0	15	110	-	-	-	-	125
	RH+RJ	3	5	5	7	8	-	-	-	20
	RH+RA	0	1	12	4	-	12	-	-	17
	RJ+RA	2	2	29	51	-	-	31	-	84
	RA+TV	0	0	51	41	-	-	-	92	92
	Total únicas	25								92
Total todas	30									431

BALANCE	Simulados	Observados	% Aciertos
RH	25	70	35.71%
RJ	2	5	40.00%
RA	17	18	94.44%
TV	110	125	88.00%
RH+RJ	8	20	40.00%
RH+RA	12	17	70.59%
RJ+RA	31	84	36.90%
RA+TV	92	92	100.00%

Tot obs  
Tot sim

Como se ha comentado anteriormente la mayoría de los tipos funcionales se simulan mejor que en el caso anterior sin haberse reducido considerablemente ninguno de los porcentajes resultantes del balance. Se puede observar que la cantidad de aciertos de RH solo ha mejorado ligeramente y aunque los aciertos de RJ también han mejorado, hay que tener en cuenta que había muy pocos puntos de simulación donde se hubiera observado vegetación pura de RJ.

Por otro lado la cantidad de aciertos de RA se mantiene constante respecto al caso anterior. Se produce una reducción despreciable de los aciertos en TV que se compensa por las notables mejoras que ha alcanzado el modelo en el resto de tipos funcionales simulados. Observando los tipos mixtos han mejorado tanto RH+RJ como RJ+RA ligeramente sin verse perjudicados los demás. Estos parámetros ofrecen un buen resultado y por ello se han tenido en cuenta a la hora de decidir los valores definitivos de los parámetros del modelo.

#### 1.1.1.6.- Simulación 4 de Lorcha

Tomando como referencia los valores de los parámetros correspondientes a la simulación anterior, se ha tratado de obtener una mejora de los resultados o, al menos, un conjunto de alternativas que ofreciesen balances similares al anterior. Para ello, en esta cuarta simulación se han modificado las

cotas de extinción por saturación de los diferentes tipos funcionales dotando a todos ellos de una mayor resistencia.

**Tabla 4.33. Parámetros de la Simulación 4 de Lorcha**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Perit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/MPa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.8	0.7	-0.75	0.8	0.6	1	0.97	1500	500
Riparian Juveniles and small Shrubs	RJ	1.25	0.9	-0.1	0.9	0.35	0.8	0.97	1500	500
Riparian adults Trees and Shrubs	RA	3.5	0.9	-0.3	0.9	0.35	1	0.97	1500	250
Terrestrial Vegetation	TV	2	1.9	0.15	1	0	1	0.97	1500	95

Tras ejecutar el software con los anteriores parámetros se obtiene la siguiente matriz:

**Tabla 4.34. Matriz de Simulación 4 de Lorcha**

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH	25	3	38	4	-	-	-	-	70
	RJ	0	2	1	2	-	-	-	-	5
	RA	0	0	17	1	-	-	-	-	18
	TV	0	0	15	110	-	-	-	-	125
	RH+RJ	3	5	4	8	8	-	-	-	20
	RH+RA	0	1	12	4	-	12	-	-	17
	RJ+RA	2	2	28	52	-	-	30	-	84
	RA+TV	0	0	51	41	-	-	-	92	92
	Total únicas	25	5	71	117	8	12	30	92	431
										Tot obs
Total todas	30	13	166	222						431
										Tot sim

BALANCE	Simulados	Observados	% Aciertos
RH	25	70	35.71%
RJ	2	5	40.00%
RA	17	18	94.44%
TV	110	125	88.00%
RH+RJ	8	20	40.00%
RH+RA	12	17	70.59%
RJ+RA	30	84	35.71%
RA+TV	92	92	100.00%

Se obtiene como resultado el mismo balance que en la simulación anterior en todos los tipos funcionales excepto en el tipo mixto RJ+RA donde se reduce ligeramente por simularse como TV uno de los puntos simulados como RA en la simulación anterior. A pesar de este ligero

empeoramiento los parámetros mostrados en esta simulación se han tenido en cuenta a la hora de decidir los parámetros definitivos del modelo por ser, al menos en los tres primeros tipos funcionales (RH, RJ y RA), más coherentes con la realidad observada.

#### 4.4.2.4.- Parámetros de calibración en Lorcha

Se ha tratado aquí de lograr de nuevo los mejores resultados posibles del balance entre los tipos funcionales observados y simulados, obtenidos en la simulación 3, manteniendo las profundidades de extinción por saturación (Zsat) lo más similares posible a los mostrados en la simulación 4. De este modo, ha sido únicamente necesaria la penalización del tipo funcional TV, al cual se le ha dado el mismo valor ya calibrado en la simulación 3, es decir Zsat igual a 0.3. Los parámetros de calibración que se han considerados óptimos para el tramo Lorcha del Río Sèrpis, el cual se encuentra en régimen alterado de caudales, son los que se muestran a continuación:

Tabla 4.35. Parámetros de calibración en Lorcha

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Pcrit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/Mpa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.8	0.7	-0.75	0.8	0.6	1	0.97	1500	500
Riparian Juveniles and small Shrubs	RJ	1.25	0.9	-0.1	0.9	0.35	0.8	0.97	1500	500
Riparian adults Trees and Shrubs	RA	3.5	0.9	-0.3	0.9	0.35	1	0.97	1500	250
Terrestrial Vegetation	TV	2	1.9	0.3	1	0	1	0.97	1500	95

Tras ejecutar el software con los anteriores parámetros se obtiene la siguiente matriz:

Tabla 4.36. Matriz de calibración en Lorcha

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH	25	3	39	3	-	-	-	-	70
	RJ	0	2	1	2	-	-	-	-	5
	RA	0	0	17	1	-	-	-	-	18
	TV	0	0	15	110	-	-	-	-	125
	RH+RJ	3	5	5	7	8	-	-	-	20
	RH+RA	0	1	12	4	-	12	-	-	17
	RJ+RA	2	2	29	51	-	-	31	-	84
	RA+TV	0	0	51	41	-	-	-	92	92
	Total únicas	25	5	72	116	8	12	31	92	431
Total todas	30	13	169	219					431	

Tot obs

Tot sim

BALANCE	Simulados	Observados	% Aciertos
RH	25	70	<b>35.71%</b>
RJ	2	5	<b>40.00%</b>
RA	17	18	<b>94.44%</b>
TV	110	125	<b>88.00%</b>
RH+RJ	8	20	<b>40.00%</b>
RH+RA	12	17	<b>70.59%</b>
RJ+RA	31	84	<b>36.90%</b>
RA+TV	92	92	<b>100.00%</b>

Se observa en los resultados como el tipo funcional mixto RJ+RA ha recuperado el balance que se había obtenido en la simulación tres únicamente con la penalización de Zsat en TV descrita anteriormente. Son las herbáceas de ribera (RH) y las riparias juveniles (RJ) las que mayores problemas muestran en la calibración. Sin embargo, es poco representativo el caso de RJ por existir únicamente 5 puntos en los que este tipo funcional ha sido observado de forma aislada. No se ha logrado mejorar el balance de RH sin penalizar los resultados de los demás tipos funcionales considerablemente.

Se ha considerado esta calibración la óptima para este tramo por ser capaz el modelo de simular, con estos parámetros, los tipos funcionales principales con buenos resultados. Además, el tipo funcional más diferenciado, TV, se simula con una fiabilidad (88%) diferenciándose de los tipos funcionales de vegetación de ribera propiamente dicha (RA, RJ y RH). Incluso en estos tres tipos se ha logrado alcanzar una diferenciación considerada muy satisfactoria, especialmente el para el tipo RA, el cual es simulado en un 94.44% de los casos. Los tipos funcionales mixtos también muestran buenos resultados.

#### **4.4.3.- Proceso de Calibración en Terde del Río Mijares (Régimen Natural)**

Del mismo modo que ocurría en el apartado anterior de calibración del modelo en Lorcha, aunque durante el proceso de calibración del tramo Terde se han realizado un elevado número de simulaciones, se van a mostrar a continuación las que han ofrecido resultados más representativos.

En este tramo sólo se han observado los tipos funcionales TV, RA y RA+TV. Por este motivo se ha adaptado la tabla resumen de tal modo que sean estos tipos funcionales los que queden reflejados en el proceso de calibración.

### 1.1.1.7.- Simulación 1 de Terde

No se muestra en este apartado la simulación correspondiente a la hipótesis inicial por no ser representativos los resultados, del mismo modo que sucedía en el tramo lorcha. Se ha seleccionado la siguiente simulación por ofrecer unos buenos resultados de calibración de la vegetación terrestre. Los parámetros de vegetación simulados aquí son los mostrados a continuación.

**Tabla 4.37. Parámetros de la Simulación 1 de Terde 1**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Perit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/MPa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.8	0.7	-0.3	0.8	0.6	1	0.97	1500	500
Riparian Juveniles and small Shrubs	RJ	1.25	0.9	0	0.9	0.35	0.8	0.97	1500	500
Riparian adults Trees and Shrubs	RA	3.5	0.9	-0.1	0.9	0.35	1	0.97	1500	250
Terrestrial Vegetation	TV	2	1.9	0.3	1	0	1	0.97	1500	95

Como puede observarse la profundidad máxima de raíces es máxima para la vegetación adulta de ribera (RA), viéndose muy reducida en comparación la del tipo funcional que engloba la vegetación herbácea (RH). Es la vegetación terrestre (TV) la que presenta en cambio una mayor profundidad efectiva de raíces, necesaria para compensar el número de puntos de simulación donde este tipo funcional no se simula a consecuencia de la anulación de la transpiración en zona saturada.

Es TV el tipo funcional más vulnerable respecto al parámetro de profundidad de extinción por saturación. Las cotas negativas implican resistencia cuando la saturación se encuentra por encima de la superficie del suelo. En este sentido se han establecido los parámetros de modo que sean las herbáceas (RH) las más resistentes, fijando su Zsat en 30 cm por encima de la superficie del suelo.

Los factores de transpiración, tanto en zona saturada como en zona no saturada, se han igualado en los tipos funcionales de riparias adultas y juveniles (RA y RJ) mientras que la fracción de cobertura se ha reducido en RJ considerándose igual a la unidad (cobertura total) en el resto de tipos funcionales.

La presión para el punto de humedad crítica se ha establecido en 500 Kpa para RH y RJ, 250 Kpa para RA y 95 Kpa para TV. Dado que este no es un parámetro frente al cual el modelo se haya mostrado especialmente sensible, estos valores se han mantenido constantes a lo largo de todo el proceso de calibración.

Tras ejecutar el software con los anteriores parámetros se obtiene la siguiente matriz:

Tabla 4.38. Matriz de Simulación 1 de Terde 1

		SIMULADOS								Total	
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV		
OBSERVADOS	RH	0	0	0	0					0	Total ripario
	RJ	0	0	0	0					0	83
	RA	30	11	38	4					83	Total puros
	TV	0	45	21	6					72	155
	RH+RJ	0	0	0	0	0				0	
	RH+RA	0	0	0	0		0			0	
	RJ+RA	0	0	0	0			0		0	Total mixtos
	RA+TV	0	2	3	5				8	10	10
	Total puros	30	56	59	10				Total obs	165	
	Total mixtos	0	2	3	5						
Total	30	58	62	15				Total sim	165		
RIPARIO VS TERRESTRE puros		145		10							
RIPARIO VS TERRESTRE total		150		15							

BALANCE	Simulados	Observados	% Aciertos
RA	38	83	45.8%
TV	6	72	8.3%
RA+TV	8	10	80.0%
TOTAL	52	165	31.5%

Como se ha comentado anteriormente, en este tramo sólo se han observado los tipos funcionales TV, RA y RA+TV. Por este motivo son nulos los valores correspondientes a las filas de los tipos funcionales RH, RJ, RH+RJ, RH+RA y RJ+RA. Del mismo modo, se ha adaptado la tabla resumen de tal modo que sean los tipos funcionales observados los que queden reflejados en la tabla del balance.

En el balance total puede observarse como el 44.71% de los puntos de simulación no ha conseguido simularse como alguno de los tipos funcionales observados. Esto es consecuencia de que 13 puntos de simulación en los que el tipo funcional observado es RA, se han simulado como RH y 12 puntos como RJ. Como consecuencia, el porcentaje de aciertos de los puntos de simulación correspondientes a RA es bajo en comparación con los obtenidos para los otros dos tipos funcionales presentes.

Al observar el comportamiento del modelo, al simular la vegetación terrestre (TV) y el tipo mixto (RA+TV), se concluye que el resultado es bastante satisfactorio.

#### 4.4.3.1.- Simulación 2 de Terde

Dado que las deficiencias de la simulación anterior se ven reflejadas en los puntos de RA se ha decidido favorecer este tipo funcional ampliando su Zsat. Las pruebas intermedias han mostrado que con esta modificación el resultado se ve perjudicado, por lo que se han recalibrado los Zsat de todos los tipos funcionales de ribera.

Tabla 4.39. **Parámetros de la Simulación 2 de Terde 1**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Perit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/MPa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.8	0.7	-0.75	0.8	0.6	1	0.97	1500	500
Riparian Juveniles and small Shrubs	RJ	1.25	0.9	-0.1	0.9	0.35	0.8	0.97	1500	500
Riparian adults Trees and Shrubs	RA	3.5	0.9	-0.3	0.9	0.35	1	0.97	1500	250
Terrestrial Vegetation	TV	2	1.9	0.3	1	0	1	0.97	1500	95

Tras ejecutar el software con los anteriores parámetros se obtiene la siguiente matriz:

Tabla 4.40. **Matriz de Simulación 2 de Terde 1**

		SIMULADOS								Total	
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV		
OBSERVADOS	RH	0	0	0	0					0	Total ripario
	RJ	0	0	0	0					0	83
	RA	29	9	41	4					83	Total puros
	TV	0	45	21	6					72	155
	RH+RJ	0	0	0	0	0				0	
	RH+RA	0	0	0	0		0			0	
	RJ+RA	0	0	0	0			0		0	Total mixtos
	RA+TV	0	1	4	5				9	10	10
	Total puros	29	54	62	10				Total obs	165	
Total mixtos	0	1	4	5							
Total	29	55	66	15				Total sim	165		
RIPARIO VS TERRESTRE puros		145			10						
RIPARIO VS TERRESTRE total		150			15						

BALANCE	Simulados	Observados	% Aciertos
RA	41	83	49.40%
TV	6	72	8.33%
RA+TV	9	10	90.00%
TOTAL	56	165	33.94%

Los resultados muestran como los cambios efectuados en el parámetro Zsat de los diferentes tipos funcionales mejoran el balance de simulación en un 4.34% para el tipo funcional RA. Este incremento se produce a consecuencia de la reducción de dos de los puntos simulados como RJ en la simulación anterior. Esto provoca que se incremente además el balance total de puntos simulados.



#### 4.4.3.2.- Simulación 3 de Terde

En esta simulación se han logrado los mejores resultados entre todas las realizadas para el tramo Terde. Se han reducido la profundidad efectiva de raíces de los tipos funcionales RA, RJ y RH. Se han vuelto a modificar los parámetros Zsat de estos tres tipos funcionales y se han diferenciado los factores de transpiración en zona no saturada para los tipos funcionales de riparias adultas y juveniles (RA y RJ), reduciéndose ligeramente esta última.

**Tabla 4.41. Parámetros de calibración en Terde**

Parameter		Zr	Ze	Zsat	Ri	Rj	Cov	CRT	Pwp	Pcrit
		Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Vegetation density [ ]	Maximum Soil-Root Water Conductance [mm/MPa/h]	Wilting Point Matrix Potencial [Kpa]	Critical matrix potential [Kpa]
Riparian Herbs	RH	0.8	0.4	-0.3	0.8	0.6	1	0.97	1500	500
Riparian Juveniles and small Shrubs	RJ	1.6	0.75	0	0.7	0.35	1	0.97	1500	500
Riparian adults Trees and Shrubs	RA	3.5	0.75	-0.1	0.9	0.35	1	0.97	1500	250
Terrestrial Vegetation	TV	2	1.9	0.3	1	0	1	0.97	1500	95

Tras ejecutar el software con los anteriores parámetros se obtiene la siguiente matriz:

**Tabla 4.42. Matriz de calibración en Terde**

		SIMULADOS								Total	
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV		
OBSERVADOS	RH	0	0	0	0					0	Total ripario
	RJ	0	0	0	0					0	83
	RA	20	15	39	9					83	Total puros
	TV	0	1	17	54					72	155
	RH+RJ	0	0	0	0	0				0	
	RH+RA	0	0	0	0		0			0	
	RJ+RA	0	0	0	0			0		0	Total mixtos
	RA+TV	0	0	4	6				10	10	10
Total puros		20	16	56	63				Total obs	165	
Total mixtos		0	0	4	6						
Total		20	16	60	69				Total sim	165	
RIPARIO VS TERRESTRE puros		92			63						
RIPARIO VS TERRESTRE total		96			69						

BALANCE	Simulados	Observados	% Aciertos
RA	39	83	46.99%
TV	54	72	75.00%
RA+TV	10	10	100.00%
TOTAL	103	165	62.42%

Estos resultados son los mejores de todos los obtenidos en el proceso de calibración del modelo en este tramo, por ello se han seleccionado estos parámetros como parámetros de calibración del modelo en este apartado.

#### 4.5.- VALIDACIÓN DEL MODELO

Tras haber realizado el proceso de calibración, y tomando como punto de partida las diversas conclusiones que de él se extrajeron, se ha procedido a efectuar la validación del modelo. Para ello se han seleccionado los valores de los parámetros calibrados para el tramo Lorcha del río Serpis (tramo de régimen natural) pero realizando la simulación en el subtramo Terde 2 del río Mijares. Los resultados de la matriz de confusión se muestran a continuación.

**Tabla 4.43.** Resultados de validación del modelo en el tramo Terde 2 del río Mijares

		SIMULADOS								Total
		RH	RJ	RA	TV	RH+RJ	RH+RA	RJ+RA	RA+TV	
OBSERVADOS	RH	0	0	0	0	-	-	-	-	0
	RJ	0	0	0	0	-	-	-	-	0
	RA	10	7	15	5	-	-	-	-	37
	TV	0	0	12	26	-	-	-	-	38
	RH+RJ	0	0	0	0	0	-	-	-	0
	RH+RA	0	0	0	0	-	0	-	-	0
	RJ+RA	0	0	0	0	-	-	0	-	0
	RA+TV	0	0	1	4	-	-	-	5	5
	Total únicas	10	7	27	31	0	0	0	5	80
										Tot obs
Total todas	10	7	28	35						80
										Tot sim

BALANCE	Simulados	Observados	% Aciertos
RA	15	37	40.54%
TV	26	38	68.42%
RA+TV	5	5	100.00%
TOTAL	46	80	57.50%

BALANCE	Simulados	Observados	% Aciertos
RIBERA	32	37	86.49%
TERRESTRE	26	38	68.42%
MIXTA	5	5	100.00%
TOTAL	63	80	78.75%

Se ha observado en el balance global como el porcentaje de vegetación terrestre se simula correctamente respecto a lo observado, así como la vegetación de ribera en su conjunto. Se observó, además que el modelo se comportaba correctamente en la simulación de grupos funcionales combinados correspondiéndose con lo observado en los diferentes puntos de simulación. Al analizar el 100% de los puntos simulados observamos que en total se simulan 44 puntos RIBERA y 31 puntos TERRESTRE; éstos frente a 37 y 38 puntos observados respectivamente, suponen la simulación del 118.91% y el 81.57% de los puntos de simulación en cada caso. Es decir, que el modelo globalmente desplaza la simulación ligeramente hacia la vegetación de ribera frente a la vegetación terrestre.

## **5.- CONCLUSIONES Y POSIBLES LÍNEAS DE ACTUACIÓN FUTURA**

## 5.1.- CONCLUSIONES

El análisis de sensibilidad manual ha dado como resultado que los parámetros de vegetación a calibrar más sensibles del modelo son los relativos a la profundidad de las raíces (tanto en zona saturada como en no saturada) y a la eficiencia en la evapotranspiración de éstas. Aun así se ha observado en el posterior proceso de calibración, que el potencial mátrico para el punto crítico es otro parámetro relativamente sensible que también debe de ser tenido en cuenta. Adicionalmente, en el futuro sería conveniente evaluar también la sensibilidad de los parámetros de suelo estimados. Esto sería útil para determinar la precisión adecuada de estos parámetros tanto en la recogida de muestras de suelo como en su posterior obtención en el laboratorio de los parámetros básicos de texturas y la posterior estimación de los parámetros del modelo de las curvas de retención de humedad. También sería interesante evaluar la sensibilidad de los parámetros de entrada hidrometeorológicos no solo para exigir un mínimo de precisión a los datos recogidos en las estaciones meteorológicas sino también para poder predecir la variabilidad de las simulaciones para los diferentes escenarios de cambio climático. Por ello se propondría realizar un segundo análisis de sensibilidad pero esta vez más detallado y utilizando una metodología más elaborada como las simulaciones de Montecarlo utilizando el método GLUE (General Likelihood Uncertainty Estimation).

Tras realizar el proceso de calibración del modelo, se ha observado que existe un gran número de aciertos para distinguir entre el tipo funcional de vegetación terrestre (TV) y los demás tipos de vegetación riparia (tanto arbórea-RA, juveniles-RJ y herbácea-RH).

Aun así, desafortunadamente el modelo Ribav no distingue demasiado bien entre los distintos tipos de vegetación riparia, especialmente entre la arbórea (RA) y la herbácea (RH). Esto puede ser debido a que es un modelo estático que no contempla procesos tales como el esfuerzo cortante de la corriente del río que afecta de manera diferente a las plantas herbáceas que a árboles adultos. Esta es una de las razones por las que se decidió fusionar Ribav2D con Ripflow v.2 para formar Ripflow v.3.

## 5.2.- LÍNEAS DE ACTUACIÓN CON RESPECTO AL MODELO

### a. Nivel freático

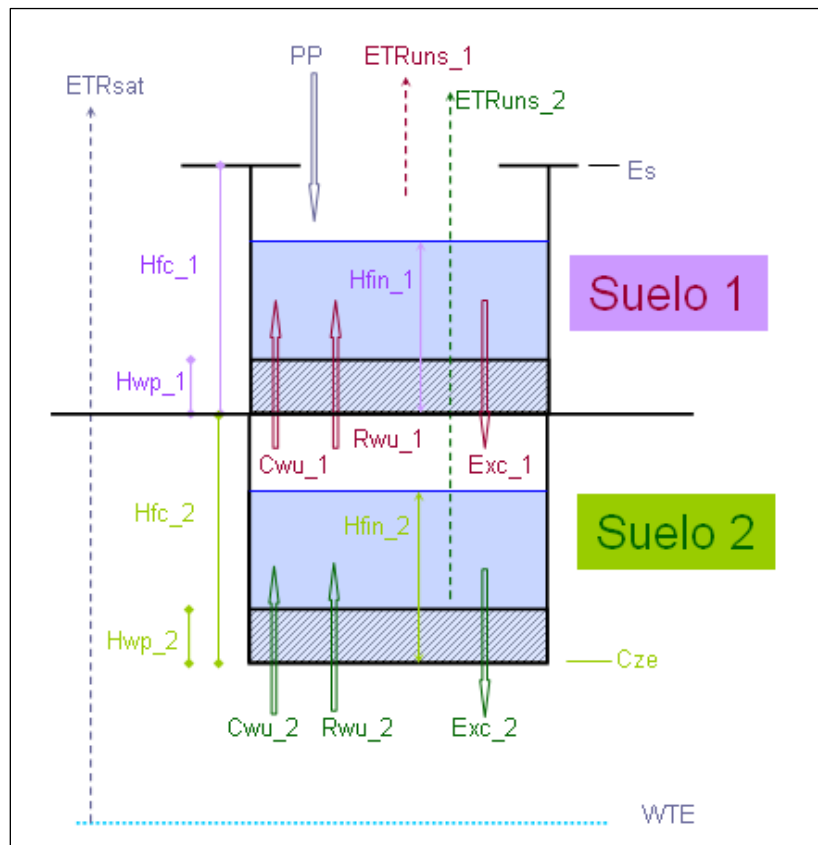
En el modelo Ribav se ha considerado que el nivel freático estaba en el mismo plano que la lámina de agua del río debido a que los suelos riparios suelen tener texturas gruesas de arenas o gravas, donde el agua se infiltra rápidamente. Aun así el intervalo de tiempo del modelo Ribav (no de Ripflow) es diario. Al tener un intervalo temporal tan reducido, es posible que el nivel freático no sea horizontal y que en las zonas más alejadas del cauce, el agua infiltrada tarde más tiempo en llegar al plano definido por las cotas de la lámina de agua. Para determinar la posible variación de la cota del nivel freático con respecto a la distancia horizontal del cauce se podría realizar un estudio mediante catas. En dicho estudio se realizarían catas para varios transectos de los tramos de estudio. Se realizarían mediciones de la profundidad de lámina de agua de cada una de las catas para distintos días con distintos niveles de río y distintos niveles de precipitaciones. Con los datos obtenidos se podría realizar una correlación donde se podría evaluar si existiera una variación significativa entre la cota de lámina de agua y la cota del nivel freático. En caso de existir una diferencia significativa se podría obtener una curva ajustada que podría utilizarse para otros transectos con tipos de suelos con texturas similares.

### b. Profundidad de la roca madre

Hay otro factor que no se ha tenido en cuenta en el modelo que es la profundidad de la roca madre. En el análisis de sensibilidad se ha observado que la profundidad de las raíces de las plantas es un parámetro del modelo bastante sensible. Si la profundidad de la roca madre es menor a la profundidad de las raíces de un determinado tipo funcional de vegetación, entonces las raíces no pueden alcanzar la profundidad de los valores calibrados. Esto podría causar que las simulaciones del modelo no simularán adecuadamente en las zonas donde la roca madre está cercana a la superficie del suelo. Una solución a este posible problema sería introducir un nuevo parámetro de suelo que fuera la profundidad de la roca madre. Para estimar este parámetro se podrían utilizar métodos geo-físicos. Estos métodos utilizan una corriente eléctrica que fluye por la zona de suelo contenida entre dos electrodos que irían situados en los extremos del transecto. Los datos obtenidos mediante estos métodos sirven para determinar la dureza o cohesión de las distintas capas del suelo y de los estratos rocosos y la profundidad a la que se encuentran.

### c. Varias capas de suelo

Las versiones actuales del modelo Ribav solo consideran una capa de suelo en la conceptualización del tanque estático. Si tras realizar el análisis de sensibilidad más detallado se concluyera que los parámetros de suelo estimados fueran bastante sensibles se podría modificar el modelo Ribav para que contemplase 2 ó más capas de suelo de características distintas. El modelo entonces tendría un tanque estático para cada una de las capas de suelo existentes. Para una zona con dos capas de suelo el diagrama de flujos quedaría de la siguiente forma estando los procesos ya descritos en el apartado 2.1 donde se describe el modelo Ribav.



**Figura 4.52.** Diagrama de flujos del modelo Ribav en caso de que se utilizaran 2 tanques estáticos para simular el efecto de 2 capas de suelo con características notablemente diferentes.

#### d. Interacción Ripflow-Ribav

El intervalo temporal de cálculo del modelo Ribav (tanto la versión 1D como la 2D) es el día mientras que en el modelo Ripflow es el año. Aun así Ripflow clasifica los años según el caudal del río (ej. año seco, normal o húmedo) basándose en los datos diarios recogidos en la estaciones de aforo. Esta clasificación de años es utilizada para determinar el mapa de niveles freáticos y el mapa con la cantidad de días de inundación de cada celda asociados a cada grupo de años. Esta clasificación de los tipos de años se realiza de forma externa al modelo Ripflow y por regla general de forma muy poco precisa en caso de que no haya una gran cantidad de tipos de año. Esta forma de clasificar los años probablemente sea aceptable para la una primera fase de integración entre los modelos Ribav y Ripflow . Aun así en versiones posteriores a la actual sería recomendable aprovechar los cálculos diarios de Ribav para obtener datos más precisos para los módulos de Ripflow. Ribav ya calcula internamente el nivel freático de cada celda para cada día de la simulación mediante un algoritmo de interpolación lineal. Los resultados de dichas interpolaciones también podrían ser utilizados como inputs en el modulo de reclutamiento. En caso de que para una determinada especie solo se pueda producir reclutamiento durante una determinada estación del año, entonces solo se utilizarían como inputs los datos diarios (individuales o promediados) de los meses pertenecientes a dicha estación.

El algoritmo que calculase en número de días al año en los que una determinada celda resultara inundada sería también bastante simple. Consistiría tan solo en restar la cota de la superficie del suelo (del MDT) de la cota de la lámina de suelo. Si el valor resultase negativo, se añadiría un contador que iría sumándose para todos los días en los que se diese este caso. Al final de año se obtendría un mapa donde cada celda tendría almacenado un número entero que indicaría el número de días en los que ha habido inundación en dicha porción de suelo.

### 5.3.- LÍNEAS DE ACTUACIÓN CON RESPECTO AL SOFTWARE

#### a. Coste computacional del modelo e interactividad entre lenguajes de programación

Tras realizar las primeras pruebas del software del modelo conjunto “Ribav2D-Ripflow” se ha observado que los componentes de Ripflow v.2 han resultado ser considerablemente más lentos en ser ejecutados que el componente Ribav2D. Esto es debido a que Ripflow v.2 fue programado con Python pero dentro de la aplicación Model-Builder de ArcGIS. Aunque Python no es por si solo un lenguaje lento, en este caso al estar siendo utilizado dentro de la plataforma del “Model Builder” resulta actuar como un lenguaje de muy alto nivel que pierde muchísima eficiencia al realizar los cálculos. Cabe destacar que la utilización de scripts del Model-Builder no tiene porque ser siempre desaconsejable debido a que también poseen ciertas ventajas. Las ventajas son la rapidez de desarrollo y la facilidad de actualización. Por tanto la utilización de Model-Builder se recomienda para modelos simples o modelos que estén en sus primeras fases de desarrollo. Como Ripflow o la combinación Ribav2D-Ripflow tiene ya en esta versión cierta complejidad se recomienda que las instrucciones dependientes de Model-Builder sean reprogramadas en otro lenguaje de programación y con un compilador que cree archivos ejecutables más eficientes. No es necesario que el nuevo código sea programado en Visual Basic al igual que el modulo Ribav2D. Esto es debido a que como Visual Basic es un lenguaje con tecnología “.Net” de la plataforma Visual Studio. Esta plataforma es multi-lenguaje puesto que utiliza el “Common Language Runtime” (Lenguaje Común en tiempo de ejecución) que actúa de intérprete entre los distintos lenguajes compatibles que pueda tener un proyecto diseñado dentro de Visual Studio. Algunos de estos lenguajes compatibles son: C++, C#, J# (implementación de JAVA), Visual Fortran e incluso IronPython (que es una implementación de Python).

#### b. Utilidades futuras con la tecnología COM

En este proyecto se ha utilizado la tecnología COM (*Component Object Model*) como una necesidad para poder unir dos aplicaciones Ribav2D y Ripflow v.2, debido a que fueron programados en lenguajes y plataformas distintas. Aun así al utilizar dicha tecnología se ha visto que puede otras traer ventajas al ser utilizada en otro tipo de tareas. Un ejemplo de estas tareas sería la realización de un análisis de sensibilidad con Ripflow mediante métodos de Montecarlo. Este tipo de análisis de sensibilidad consiste en ejecutar el modelo con distintos valores de los parámetros que se quieran analizar. Para ello se tiene que obtener una cantidad elevada de números al azar que serán utilizados por el programa para dar valores distintos a los parámetros durante cada ejecución. Aunque esta operación puede llevarse a cabo de distintas formas, al utilizar la tecnología COM no se exigiría modificar mucho el código. Tan solo habría que añadir una Clase COM al código fuente tal como se describe en el apartado 3.2.3. Adicionalmente el código fuente no se compilaría a un archivo “.exe” (ejecutable) sino a un archivo “.dll” (dynamic link library). Los archivos DLL permiten ser ejecutados y recibir información externamente desde otros programas. Esto sería útil en el análisis de sensibilidad porque mediante un programa externo no muy complejo se podrían crear números aleatorios que serían enviados periódicamente a la librería DLL. Dichos números aleatorios serían utilizados como los valores de los parámetros a analizar del modelo en cada una de las muchas ejecuciones del modelo que requeriría el análisis de sensibilidad. La tecnología COM



también podría ser utilizada de forma similar para realizar calibraciones automáticas del modelo o para ejecutar el modelo con gran cantidad de escenarios con parámetros de entrada diferentes.

### c. Interfaz gráfica de usuario (GUI)

Visual Basic es un lenguaje que en sus inicios se hizo muy popular por la facilidad que tenía para crear “interfaces gráficas de usuario” (GUI.) Aun así este lenguaje recibió muchas críticas debido a que en sus primeras versiones obligaba a mezclar el código de la GUI con la “lógica del programa” (es decir el código que se encarga de realizar los cálculos que requieren las ecuaciones del modelo). Por el contrario las versiones modernas de Visual Basic permiten que el usuario pueda separar ambos tipos de código de forma relativamente fácil al crear aplicaciones de consola de forma independiente. Para que Ripflow pudiera utilizar el código de Ribav, se tuvo que despojar a Ribav2D de su GUI debido a que Ripflow ya poseía una GUI propia que penaliza el tiempo de computación o incluso podría causar conflictos. Por ello sería deseable que en las versiones siguientes de Ribav2D se separe el código de la GUI del código de la lógica del programa. En las versiones modernas de Visual Studio esto se realizaría mediante una única “solución” que englobe dos “proyectos” distintos (uno para la GUI y otro para la lógica). Para otras aplicaciones informáticas como páginas web, esta separación no es tan necesaria, pero en la programación científica la lógica del programa resulta ser bastante más compleja. Esto trae como ventaja que el código fuente sea más ordenado lo que facilitaría las posibles modificaciones o mejoras de la conceptualización matemática del modelo. Además permitiría cambiar o modificar la GUI en el futuro (cuando surgieran nuevas tecnologías gráficas que mejorasen considerablemente la estética o la funcionalidad de los programas), sin que hubiera posibilidad de modificar por error el código de la lógica del programa.

### d. Utilización de programación orientada a objetos

Como última mejora se pretende que las versiones posteriores del programa tengan un paradigma de programación orientado a objetos (POO). La programación orientada a objetos tal como su nombre indica utiliza objetos, que son estructuras de datos que contienen tanto propiedades (diferentes tipos de variables) como métodos (funciones y subrutinas). Fue utilizada por primera vez durante la década de los sesenta por el lenguaje “Simula 67”, para procesos de simulación física. Aun así este paradigma de programación no se extendió hasta mitad de la década de los ochenta con lenguajes de programación como C++, Smalltalk o Eiffel. A principios de los noventa se hizo el paradigma de programación dominante con la irrupción de las interfaces gráficas de usuario (GUI) y con el desarrollo de aplicaciones web con lenguajes como JAVA. En la actualidad las nuevas versiones de muchos lenguajes que inicialmente tenían un paradigma de programación procedural como BASIC (u otros como Pascal o Fortran) tienen propiedades que les permiten funcionar enteramente como lenguajes orientados a objetos. Esto permitiría que el código programado anteriormente en Visual Basic pudiese ser reciclado con relativa facilidad.

La programación orientada a objetos (POO) se diferencia de la programación procedural en que se hace más énfasis en los datos (objetos) que en las instrucciones (procedimientos). Se utiliza también la abstracción de datos donde tan solo se representa externamente la parte fundamental

de cada una de las partes (objetos) del programa. Esto trae como ventaja que una vez se domina este paradigma de programación, existe una mayor relación entre el planteamiento del modelo y el código fuente del software, lo que facilita la comprensión tanto del código fuente como del modelo. A continuación se muestra un ejemplo simplificado de la utilización del tanque estático del modelo Ribav. En ella habría una clase llamada “StaticTankClass” a partir de la cual se crearían los objetos de tanque estático. Dicha clase tendría variables (“TankSoilMoisture”, “FieldCapacity” y “WiltingPoint”) que representarían las propiedades (características) del tanque. Además habría asociado a dicha clase (y objetos creados) dos métodos (las subrutinas: “MatrixPressure” y “RootWaterUptake”), que representarían los procesos que tienen lugar. Las propiedades y métodos que tendrían interacción con otros objetos externos como podría ser un objeto de “suelo saturado” son: (“TankSoilMoisture” y “RootWaterUptake”) se declaran como públicas (“Public”) y las que funcionan solo internamente dentro del objeto del tanque (“FieldCapacity”, “WiltingPoint” y “MatrixPressure”) se declaran como privadas (“Dim” y “Private”).

```
Public Class StaticTankClass

    Public TankSoilMoisture As Single
    Dim FieldCapacity As Single
    Dim WiltingPoint As Single

    Private Sub MatrixPressure(ByVal Pb As Single, ByVal Psoil As Single, ByVal Pst As Single, ByVal Ip As Single)
        Dim Pmat As Single 'Matrix Pressure output variable
        Pmat = Pb / (Psoil / Pst) ^ (1 / Ip)
    End Sub

    Public Sub RootWaterUptake(ByVal Crt As Single, ByVal Pfc As Single, ByVal Pmat As Single, ByVal Pm50 As Single,
        ByVal Rj As Single)
        Dim rwu As Single 'Root Water Uptake output variable
        rwu = Math.Max(0, (-Crt * 10 / 1000) * (Pfc - Pmat) * (1 / (1 + ((Pmat / Pm50) ^ 3.22))) * Rj)
    End Sub

End Class
```

**Figura 4.53.** Ejemplo de utilización de programación orientada a objetos donde se representa algunos de los procesos (métodos) y características (propiedades) de un objeto que representaría al tanque estático de Ribav.

Otra ventaja de la PPO es la herencia. La herencia permite que las propiedades y métodos de una determinada clase de objeto puedan ser transferidos (heredados) a otra clase y a sus objetos asociados. Esto mejora las posibilidades de mantenimiento del código y facilita su actualización futura porque se pueden ampliar las capacidades de un objeto sin tener miedo a estropear el núcleo inicial.

La OPP tiene también como característica avanzada el polimorfismo. El polimorfismo consiste en que un operador (ej: “+”, “-” ó “<”) y también un método pueda actuar de distinta forma aun teniendo el mismo nombre según la situación lo requiera. A esto se le llama sobrecarga de operadores y métodos respectivamente. En nuestro caso esto podría ser útil porque a modo de ejemplo el operador “+” podría utilizarse no solo para sumar el valor de dos escalares sino también los valores de matrices o incluso de mapas grid. Esto permitiría que cuando se realizan cálculos utilizando datos inputs de varios mapas grid no sea necesario utilizar bucles dobles. Existen lenguajes de más alto nivel como Matlab o ModelBuilder que realizan esto de forma automática. El polimorfismo facilita no solo la comprensión del código sino la búsqueda de errores lógicos.

A modo de conclusión, aunque la programación orientada a objetos puede resultar al principio difícil de aprender por su elevado grado de abstracción, permite crear código fuente más relacionado con el modelo y la realidad que simula. Esto permite que una determinada aplicación informática pueda tener un mayor tamaño y pueda ser mantenida y actualizada con mayor facilidad, y mientras tanto ser gestionada por un número mayor de personas.

## **6.- REFERENCIAS**

- Altier, L.S.; Lowrance, R.; Williams, R.G.; Inamdar, S.P.; Bosch, D.D.; Sheridan, J.M.; Hubbard, K.; Thomas, D.L. (2002). *Riparian Ecosystem Management Model: simulator for ecological processes in riparian zones*. USDA, Conservation Research Report 46. 216 p.
- Altier, L.S.; Lowrance, R.; Williams, R.G.; Inamdar, S.P.; Bosch, D.D.; Sheridan, J.M.; Hubbard, K.; Thomas, D.L. (2002). *Riparian Ecosystem Management Model: simulator for ecological processes in riparian zones*. USDA, Conservation Research Report 46. 216 p.
- Baird, K.J.; Maddock III, T. (2005). *Simulating riparian evapotranspiration: a new methodology and application for groundwater models*. Journal of Hydrology 312: 176-190.
- Baird, K.J.; Maddock III, T. 2005. Simulating riparian evapotranspiration: a new methodology and application for groundwater models. Journal of Hydrology 312: 176-190.
- Benjankar R. (2006). Quantification of reservoir operational-based losses to Floodplain physical processes and impact on the Floodplain vegetation at the Kootenai River, USA .Research proposal towards fulfillment requirements for Doctoral degree, Idaho University, Boise.
- Campbell, G.S. (1974). *A simple method for determining unsaturated conductivity from moisture retention data*. Soil Science, Vol. (117), No. 6, 311-314 p.
- Cohen, Jacob. (1960) A coefficient of agreement for nominal scales, *Educational and Psychological Measurement* Vol.20, No.1, pp. 37–46.
- Dingman, S.L. 2002. Physical hydrology. New Jersey, U.S.A., Prentice Hall. 646 p.
- Ferreira, M.T. y F.C.F. Aguiar. 2005. Metodologia para amostragem de macrófitos em sistemas fluviais para implementação da Directiva Quadro da Água. *Palestra proferida no 1º Workshop sobre Metodologias de Amostragem de Elementos Biológicos e Hidromorfológicos no âmbito da Directiva Quadro*. 6-8 Março. Universidade de Évora. Évora.
- González del Tánago, M. 1998. "Las riberas, elementos clave del paisaje y en la gestión del agua", I *Congreso Ibérico sobre Gestión y Planificación de Aguas*, Zaragoza, 14-18 de Septiembre, Prensas Universitarias de Zaragoza, Zaragoza.
- Goodwin, C.N.; Tarboton, D.G. (2001). An ecohydrological approach for modelling vegetation cover based upon equilibrium soil moisture availability. *AGU Chapman Conference on State-of-the-Art Hillslope Hydrology*. Sunriver, Oregon, October 8-12, 2001.
- Horton, J.L.; Kolb, T.E.; Hart, S.C. (2001). *Responses of riparian trees to interannual variation in ground water depth in a semi arid river basin*. Plant, Cell and Environment 24, 293–304.
- Ibero, C. (1996). *Ríos de vida. El estado de conservación de las riberas fluviales en España*. SEO/Bird Life.
- Lite, S.J.; Stromberg, J.C. (2005). Surface water and ground-water thresholds for maintaining Populus–Salix forests, San Pedro River, Arizona. *Biological Conservation* 125 (2005) 153–167.
- Richards, K.S.; Hughes, F.M.R.; El-Hames, A.S.; Harris, T.; Pautou, G.; Peiry, J.; Girel, J. (1996). Integrated field, laboratory and numerical investigations of hydrological influences on the establishment of riparian tree species. 611-635p. In: Anderson, M.; Walling, D.E.; Bates, P.D. 1996. *Floodplain processes*.
- Rivas, R.E. 2004. Propuesta de un modelo operativo para la estimación de la evapotranspiración. Tesis doctoral Facultad de Física / Universidad de Valencia. 140 p.
- 
- Rodríguez-González, P.M.; Ferreira, M.T.; Rego, P. R. (2004). Vegetation types and within-stand structure. *Forest Ecology and Management* 203: 261–272.
- Rodríguez-Iturbe, I.; Porporato, A. (2004). *Ecohydrology of water-controlled ecosystems: soil moisture and plant dynamics*. Cambridge University Press. 442 p.

Ryel, R.; Caldwell, M.; Yoder, C.; Or, D.; Leffel, A.; (2002). Hydraulic redistribution in a stand of *Artemisia tridentata*: evaluation of benefits to transpiration assessed with a simulation model. *Oecologia*- Springer

---

Shaw, E.M. 1994. *Hydrology in practice*. 3ª. Ed., Stanley Thornes, U.K. 569 p.

Shuttleworth, W.J. 1993. Evaporation. In: Maidment, D.R. (Ed.). 1993. *Handbook of hydrology*. McGraw-Hill, U.S.A. p. 4.1 – 4.53 p.

Tabacchi, E., Correll, D.L., Hauer, R., Pinay, G., Planty-Tabacchi, A., Wissmar, R.C., (1998). Development, maintenance and role of riparian vegetation in the river landscape. *Freshwater Biol.* 40, 497–516.

Tabacchi, E.; Planty-Tabacchi, A.-M. (2003). Recent changes in riparian vegetation: possible consequences on dead wood processing along rivers. *River Res. Applic.* 19: 251–263.

Terradas, J. (2001). *Ecología de la vegetación: de la ecofisiología de las plantas a la dinámica de comunidades y paisajes*. España, Ediciones Omega. 703 p.

Ward, R.C.; Robinson, M. 2000. *Principles of hydrology*. 4a. Ed. McGraw-Hill, U.K. 450 p.

Ward, R.C.; Robinson, M. 2000. *Principles of hydrology*. 4a. Ed. McGraw-Hill, U.K. 450 p.

Webb, R.H.; Leake, S.A. (2006). Ground-water surface-water interactions and long-term change in riverine riparian vegetation in the southwestern United States. *Journal of Hydrology* 320 (2006) 302–323.

Weltya, J., Beechieb, T., Sullivanc, K., Hyinka, D., Bilbya, R., Andrusd, C., Pess, G. 2002. Riparian aquatic interaction simulator (RAIS): a model of riparian forest dynamics for the generation of large woody debris and shade. *Forest Ecology and Management* 162 (2002) 299–318.

Wild, A. 1992. *Condiciones del suelo y desarrollo de las plantas según Russell*. Ediciones Mundi-Prensa. 1045 p.

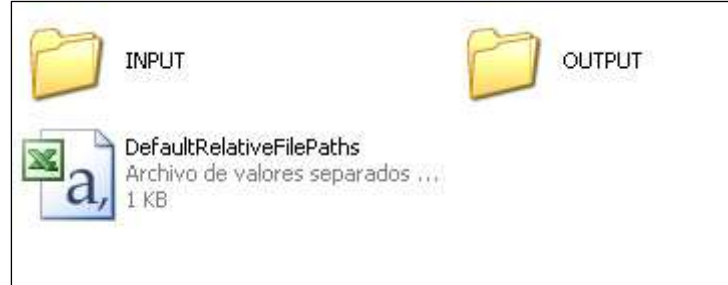
# **ANEXO I TUTORIAL DE UTILIZACIÓN DEL MODULO RIBAV 2D (INGLÉS)**

## (1) Introduction

Ribav2D has been included as an additional module to the Ripflow Model. Ribav2D has the same mathematical conceptualization as the Ribav1D model but instead of working with points of simulation, it works with map grids. Both versions simulate the vegetation functional type distribution from a certain river stretch. To carry out these predictions Ribav uses physically-based soil moisture equations. The main inputs from the Ribav model are the Initial Vegetation Types Distributions, Soil Types Distribution, Topography and Hydro-meteorological data. The yearly output from the Ribav2D module is a map, which returns a value of the Evapotranspiration Index (ETindex) for each cell of the grid. The ETindex is the quotient between the Actual Evapotranspiration (ETR) and the Potential Evapotranspiration (ETP) of the vegetation of each cell. The closest to 1 the value of the ETindex is, the smaller is the difference between ETR and ETP. Due to that, a plant with a high ETindex will have more possibilities of advancing to a higher or more advanced succession phase, and a plant with a low ETindex will have more possibilities to retrogress to a lower or previous succession phase, but always taking into account the factors of the other modules of the Ripflow model.

## (2) File Organization

When the Data directory is opened: 2 subdirectories and 1 CSV file appear. The file is called: “DefaultRelativeFilepaths.csv” and stores the relative paths of the input and output files of Ripflow. The subdirectories are called: “INPUT” and “OUTPUT” and as their name says, they store the input and output files respectively.



**Figure I.54.** File and subdirectories found in the Data directory within Ribav.

When the “DefaultRelativeFilepaths.csv” is opened, 7 relative file paths appear (1 per row). The significance of these file paths is shown on the following table:

**Table I.b Records of the “DefaultRelativeFilepaths.csv” file**

Row	Referring to:	Path given by default:
1	Soil Type Maps	\\Data\\INPUT\\Maps\\SoilMap.asc
2	Digital Elevation Model (DEM) Map	\\Data\\INPUT\\Maps\\DemMap.asc
3	Soil Parameters File	\\Data\\INPUT\\Parameters\\SoilParameters.csv
4	Vegetation Parameters File	\\Data\\INPUT\\Parameters\\VegetationParameters.csv
5	Hydro-meteorology File	\\Data\\INPUT\\Hydromet\\Hydromet.csv
6	File which relates the different Water Table Elevation	\\Data\\INPUT\\Hydromet\\WteMaps.csv



	Files with the river flow	
7	Subdirectory where ETindex Output Files are stored	\\Data\OUTPUT\ETindexMaps\

```

\\Data\INPUT\Maps\SoilMap.asc
\\Data\INPUT\Maps\DemMap.asc
\\Data\INPUT\Parameters\SoilParameters.csv
\\Data\INPUT\Parameters\VegetationParameters.csv
\\Data\INPUT\Hydromet\Hydromet.csv
\\Data\INPUT\Hydromet\wteMaps.csv
\\Data\OUTPUT\ETindexMaps\

```

**Figure I.55.** Default contents of the “DefaultRelativeFilepaths.csv” file

### (3) File Structure

Two different types of files are used in the Ribav2D module: the ASCII grid files and the CSV files.

#### a) ASCII grid files

ASCII grid files have been used because they are human readable, hardware independent, widely supported and easy to import/export from most GIS software. To avoid errors all the grids from the Ribav2D simulation require the same dimensions and cell sizes. In case a map has cells with unavailable or irrelevant information, then they should be filled with “No Data Values”. The cells within a row are separated by a blank space. ASCII grid files have a header of 6 lines with the following information where X represents a number:

**Table III.c Header of an ASCII grid file**

Row	Reference	Explanation
1	ncols X	Number of columns from the grid
2	nrows X	Number of rows from the grid
3	xllcorner X	Western (left) x-coordinate
5	yllcorner X	Southern (bottom) y-coordinate
5	cellsize X	Length of the sides of the squares
6	NODATA_value X	Value that is regarded as missing or not applicable

#### b) CSV files

CSV stands for Comma Separated Values. They are a type of text file in which the columns are separated by commas “,” or semicolon “;”. In our case we have chosen to use a semicolon to prevent confusions with the comma as decimal separator. Even so, in all types of files the point “.” is used as a decimal separator. These files don’t have any headers and the columns represent the different parameters. By the other hand the rows represent the different parameter records.





river flow value should be 0 m<sup>3</sup>/s and the highest flow value should be higher than the highest value of flow from the hydro-meteorological time series. This file is located by default in the “Data\Input\Hydromet\WteMaps.csv” subdirectory.

```
0;\Data\INPUT\Hydromet\wte-q0.asc
0.25;\Data\INPUT\Hydromet\wte-q0,25.asc
0.47;\Data\INPUT\Hydromet\wte-q0,47.asc
1;\Data\INPUT\Hydromet\wte-q1.asc
2;\Data\INPUT\Hydromet\wte-q2.asc
5;\Data\INPUT\Hydromet\wte-q5.asc
10;\Data\INPUT\Hydromet\wte-q10.asc
30;\Data\INPUT\Hydromet\wte-q30.asc
150;\Data\INPUT\Hydromet\wte-q150.asc
300;\Data\INPUT\Hydromet\wte-q300.asc
650;\Data\INPUT\Hydromet\wte-q650.asc
```

Figure I.59. Content of a WTE CSV file visualised with a text editor.

e) Soil parameter’s CSV file

In this file the values of all the parameters of each of the numbered soil types that appear on the soil grid will be stored. This file is located by default in the “Data\Input\Parameters\SoilParameters.csv” subdirectory. In this file the following fields (columns) are located from left to right:

- n: Numerical key from the table which relates the row number with the number of the soil type map.
- Pst: Porosity ( )
- Ip: Porosity index( )
- Pb: Bubble Preassure (Kpa)
- Ks: Saturated Conductivity of the Unsaturated Soil(mm/hr)
- Mfc: Moisture at Field Capacity ( )
- Zcmin: Depth to be considered in the Capillary Hydraulic Lift (m)
- Descr: It is a description of the soil / It doesn’t take part in the calculations (character string).

```
1;0.401;0.4556;1.837;36.78;0.18;Suelo arenosos y limoso;17
2;0.412;0.31446;4.40324;22.62;0.221;Suelo bastante limoso;17
3;0.407028884;0.266172308;2.47510017;32;0.205;Suelo arenosos y limoso;17
4;0.45;0.23833779;4.34325797;15.81;0.277;Suelo limoso sin gravas;17
5;0.501831;0.232989037;1.253170442;46.37691;0.212454;suelo organico con mucha grava;17
6;0.458;0.206439561;1.196334864;14.32;0.23;Suelo gravoso y arenoso;17
7;0.441;0.214628488;0.187422871;30.53;0.145;Suelo gravoso y arenoso;17
8;0.435;0.238123083;0.983633766;40.39;0.189;Suelo arenosos y limoso;17
9;0.412;0.208738261;0.021171;38.82;0.066;Suelo gravoso y muy arenoso;17
10;0.486;0.260239852;0.516127645;40.02;0.164;Suelo gravoso y arenoso con bastante M. organica;17
```

SOIL PARAMETERS							
SOIL SAMPLE	● Porosity [ ]	▲ Porosity Index	Hb Bubble Pressure [Kpa]	Saturated Hydraulic Conductivity [mm/hr]	●CC Field Capacity Moisture at 33Kpa [ ]	Soil Description	Minimum Capillary Depth to be Considered [m]
n	Pst	Ip	Pb	Ks	Mfc	Descr	ZCmin
1	0.401	0.4556	5.837	36.78	0.18	Suelo arenosos y limo	17
2	0.412	0.31446	4.40324	22.62	0.221	Suelo bastante limoso	17
3	0.407028888	0.266172308	2.47510017	32	0.205	Suelo arenosos y limo	17
4	0.45	0.23833779	4.34325797	15.81	0.277	Suelo limoso sin gravi	17
5	0.501831	0.23298904	1.25317044	46.37691	0.212454	Dos horizontes le sup	17
6	0.458	0.20643956	1.19633486	14.32	0.23	Suelo gravoso y aren	17
7	0.441	0.21462849	0.18742287	30.53	0.145	Suelo gravoso y aren	17
8	0.435	0.23812308	0.98363377	40.39	0.189	Suelo arenosos y limo	17
9	0.412	0.20873826	0.021171	38.82	0.066	Suelo gravoso y muy	17
10	0.486	0.26023985	0.51612765	40.02	0.164	Suelo gravoso y aren	17

**Figure I.60.** Soil parameters in a file in CSV format (upper figure) and with header in a spreadsheet (lower figure).

f) Vegetation parameter's CSV file

In this file the values of all the parameters of each of the vegetation types will be stored. This file is located by default in the "Data\Input\Parameters\VegetationParameters.csv" subdirectory. In this file the following fields (columns) are located from left to right:

- n: Numerical key from the table which relates the row number with the number of the soil type map.
- Zr: Maximum Root Depth (m)
- Ze: Effective Root Depth (m)
- Zsat: Extinction at Saturation [m]
- Ri: Transpiration Factor from Unsaturated Zone ( )
- Rj: Transpiration Factor from Saturated Zone ( )
- Cov: Plant Cover ( )
- Crt: Maximum Soil-Root Water Conductance (mmMpa-1h-1)
- Pwp: Wilting Point Pressure (Kpa)
- Pcrit: Critical Point Pressure (Kpa)
- Acron: Acronym representing the plant type / it does not take part in the calculations (character string).
- Descr: It is a description of the plant type / it doesn't take part in the calculations (character string).

```
1;0.8;0.7;-0.75;0.8;0.6;1;0.97;1500;500;RH;Riparian Herbs
2;1.25;0.9;-0.1;0.9;0.35;0.8;0.97;1500;500;RJ;Riparian Juveniles and Small Schrubbs
3;3.5;0.9;-0.3;0.9;0.35;1;0.97;1500;250;RA;Riparian Trees and Big Schrubbs
4;2;1.9;0.3;1;0;1;0.97;1500;95;TV;Terrestrial Vegetation
```

B. VEGETATION BASIC PARAMETERS											
Vegetation Type Number	Maximum Root Depth [m]	Effective Root Depth [m]	Extinction at Saturation [m]	Transpiration Factor from Unsaturated Zone [ ]	Transpiration Factor from Saturated Zone [ ]	Av Plant Cover [ ]	Maximum Soil Root Water Conductance [mmMpa-1h-1]	Wilting Point Pressure [Kpa]	Critical Point Pressure [Kpa]	Vegetation Type Acronym	Vegetation Type Description
n	Zr	Ze	Zsat	Ri	Rj	Cov	Crt	Pwp	Pcrit	Acron	Descr
1	0.8	0.7	-0.75	0.8	0.6	1	0.97	1500	500	RH	Riparian Herbs
2	1.25	0.9	-0.1	0.9	0.35	0.8	0.97	1500	500	RJ	Riparian Juveniles and Small Schrubbs
3	3.5	0.9	-0.3	0.9	0.35	1	0.97	1500	250	RA	Riparian Trees and Big Schrubbs
4	2	1.9	0.3	1	0	1	0.97	1500	95	TV	Terrestrial Vegetation

**Figure I.61.** Vegetation parameters in a file in CSV format (upper figure) and with header in a spreadsheet (lower figure).

g) Hidro-meteorological data file

This file is located by default in the "Data\Input\Hydromet\Hydromet.csv" subdirectory. In this file the following fields (columns) are located from left to right:

- Date (dd/mm/aaaa)

- Daily Precipitation (mm)
- Potential Evapotranspiration (mm)
- Daily Flow from the River Stretch (m3/s)

C. DAILY HYDROMETEOROLOGICAL INPUTS			
Date	Daily Precipitation [mm]	Potential Evapotranspiration [mm]	Daily Flow [m3/s]
	c.1	c.2	c.3
	PP	ETP	Q
01/01/1999	1.700	1.174	0.791
02/01/1999	0.717	1.17	0.783
03/01/1999	0	1.192	0.718
04/01/1999	0	1.172	0.672
05/01/1999	0	1.198	0.728
06/01/1999	0	1.18	0.865
07/01/1999	0	1.182	1.145
d	.....	.....	.....

```

01/01/1999;1.7004;1.174;0.586
02/01/1999;0.7167;1.17;0.578
03/01/1999;0;1.192;0.573
04/01/1999;0;1.172;0.569
05/01/1999;0;1.198;0.564
06/01/1999;0;1.18;0.573
07/01/1999;0;1.182;0.558
08/01/1999;0;1.178;0.558
09/01/1999;0;1.181;0.565
10/01/1999;0;1.155;0.554
11/01/1999;0;1.12;0.558
12/01/1999;0;0.721;0.559
13/01/1999;0;1.401;0.564
14/01/1999;0.4962;1.057;0.554
15/01/1999;0.9924;1.165;0.555
16/01/1999;0;1.382;0.556
17/01/1999;2.3352;1.107;0.551
18/01/1999;0;1.014;0.549
19/01/1999;0;1.131;0.546
20/01/1999;0;1.107;0.555
21/01/1999;0;1.029;0.551

```

**Figure I.62.** Hydro-meteorological data with header in a spreadsheet (left) and in a file in CSV format (right).

h) Vegetation types data file (created by Ripflow)

This file is created by the other Ripflow modules in a yearly step. The only time the user needs to prepare this type of file is to compare observed vegetation with simulated vegetation in processes like calibration/validation. This is a type of ASCII type grid where **each cell is given an integer number which represents the vegetation type**. That number will be related to the row number of the vegetation parameter files. This file is located by default in the "Data\Input\Maps\VegetationMap.asc" subdirectory.

ncols															572
nrows															541
xllcorner					733225,7372										
yllcorner					4304029,0124										
cellsize										0,5					
NODATA_value															
	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999
	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999
	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999	-9999
	-9999	-9999	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	1	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	3	1	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	3	3	1	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	3	3	3	1	1	1	1	1	1	1	1	1	1
	-9999	-9999	3	3	3	3	1	1	1	1	1	1	1	1	1
	-9999	-9999	3	3	3	3	3	1	1	1	1	1	1	1	1
	-9999	-9999	3	3	3	3	3	3	1	1	1	1	1	1	1
	-9999	-9999	3	3	3	3	3	3	3	1	1	1	1	1	1
	-9999	-9999	3	3	3	3	3	3	3	3	1	1	1	1	1

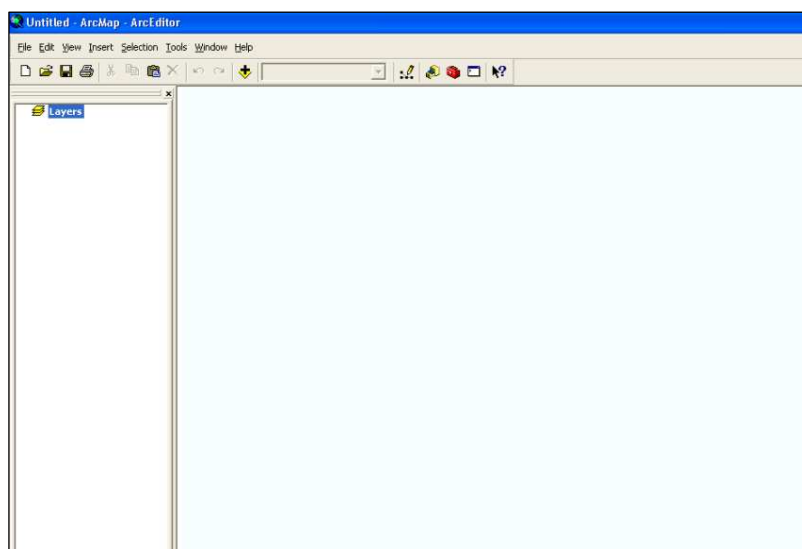
**Figure I.63.** Example of a Vegetation Type Grid showing vegetation types 1 and 3 and also NODATA\_values (represented in this case by number: -9999)

## (5) File Conversion and Visualization

### 5.1 ASCII conversion to ESRI-Raster and viceversa

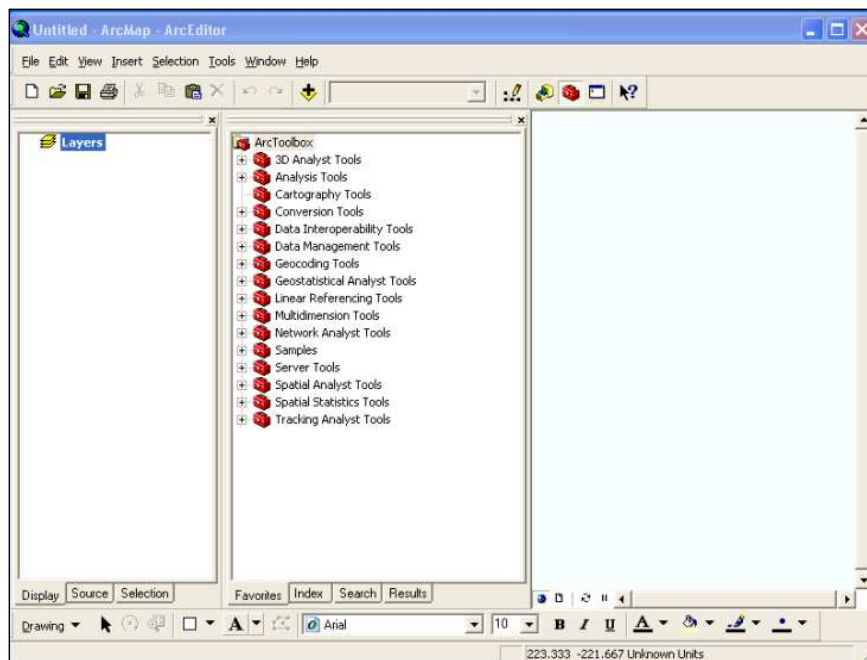
The grid files used by the Ribav2D are ASCII format and the grid files of the other Ripflow have an ESRI-Raster format. Even so the conversion between ASCII and ESRI-Raster is very simple. The steps needed to carry out this conversion are the following:

- a) Open ArcGIS 9.2



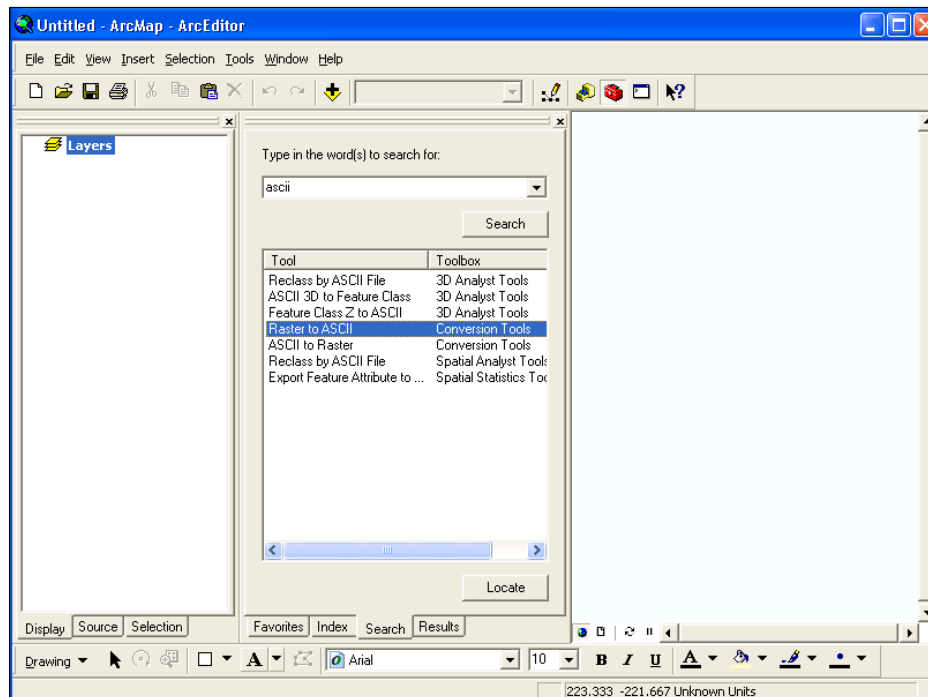
**Figure I.64.** ARCGIS 9.2 initial window for an empty project

b) Press on the ArcToolbox icons so the tool menu appears



*Figure I.65. ArcToolbox menu from ARCGIS 9.2*

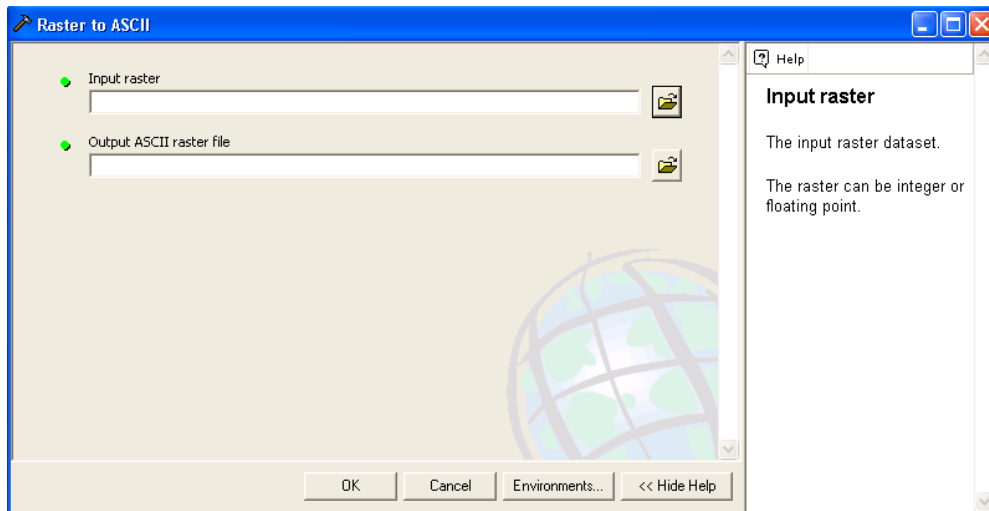
c) Press on the search tab from the Arctoolbox, write "ASCII" in the textbox and click on the Search button. All the tools with "ASCII" in their title will appear.



*Figure I.66. Tools related to ASCII in ARCGIS 9.2*

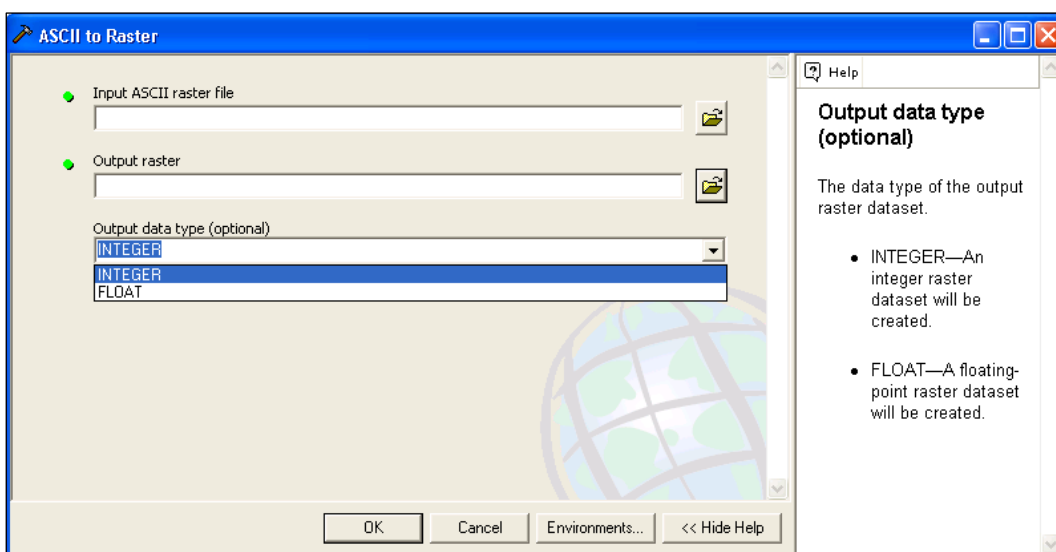


- d) Press on the “Raster to ASCII” if you want to make this type of conversion. Two textboxes will appear. In the first one we will need to introduce the path of the input ESRI-Raster file to be converted. In the second textbox we will introduce the path of the ASCII output file that will appear due to the conversion.



**Figure I.67.** Raster to ASCII conversion toolbox.

- e) On the other hand, if you want to make an ASCII to ESRI-Raster conversion press on the “Raster to ASCII” tool. Two textboxes and a combo-box will appear. In the first textbox we will introduce the path of the input ASCII file to be converted. In the second textbox we will introduce the path of the output ESRI-Raster file that will appear due to the conversion. In the combo-box we will optionally select if we want the data types of the new file to be integer or float (decimal) data types.



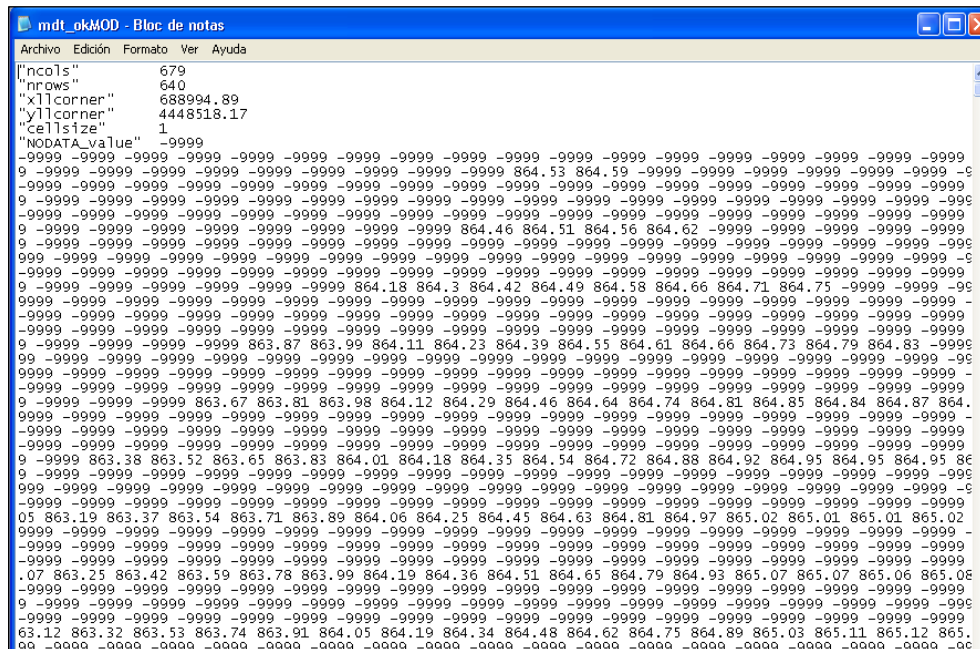
**Figure I.68.** ASCII to Raster conversion toolbox.

## 5.2 File visualization

The ASCII and CSV files can be visualized in texts editors (like Notepad) or spreadsheets (like Open-office Calc 3.2 or MS Excel 2003).

### a) Text Editor Visualization

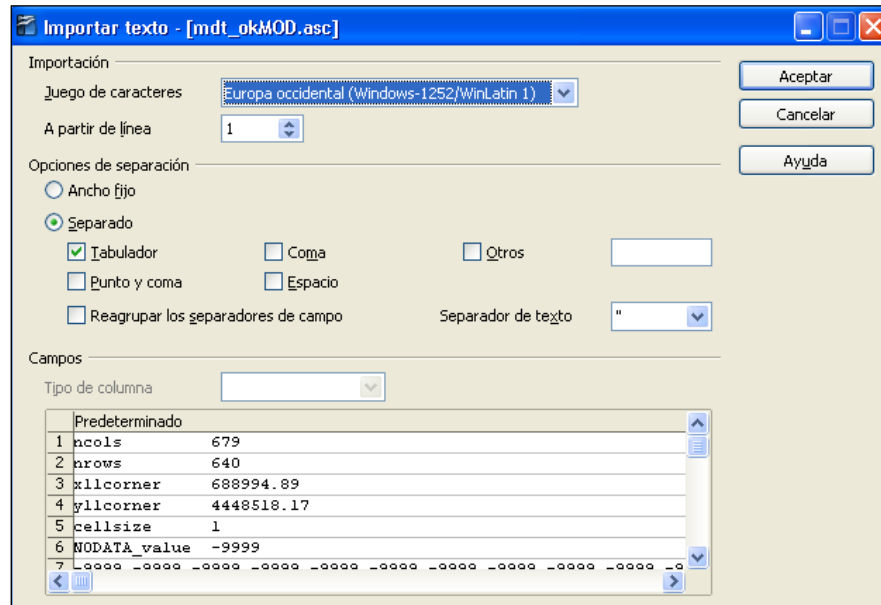
The simplest way to open these files is with a text editor like Notepad which is included in Windows Xp. In this way you can visualize a lot of details from the file format (like the type of column separator that the cells have) that would not be noticed in a spreadsheet visualization. The main inconvenient with this type of visualization is that it brings a lot of difficulties in managing large amounts of data.



**Figure I.69.** Example of a MDT of study site in the river Mijares visualized with a text editor.

### b) Open-Office Calc 3.2 Visualization

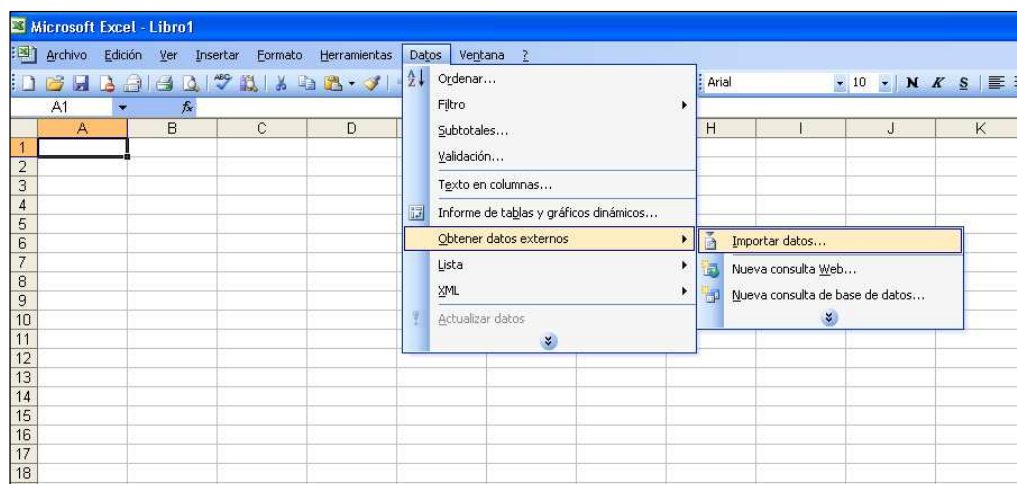
The way to open a CSV or ASCII file through the Calc 3.2 spreadsheet is very simple. When the file is opened a dialog box will appear. You will need to select the type of column separator: semicolon for CSV files or space for ASCII files. Additionally if you want headers to be ignored you can select to carry out the visualization from a row different as row 1. To avoid errors it is very important to leave blank the combo-box labelled “text separator”, because the default “double quotation marks” character can cause problems when the ASCII files are loaded within SIG software.



**Figure I.70.** Dialog box used in Open-Office Calc 3.2 to load textfiles like ASCII or CSV.

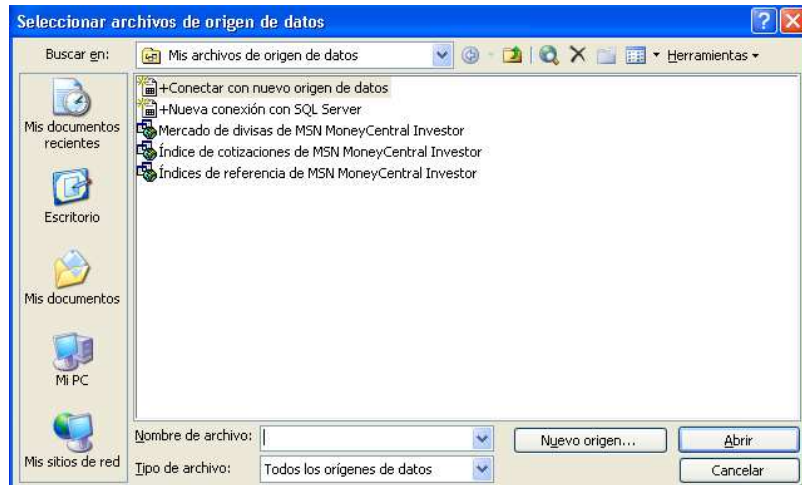
c) MS Excel 2003 Visualization

In Excel 2003 CSV files are opened automatically but ASCII files require a special operation. Therefore the “Import data” application from the “Data” menu should be used.



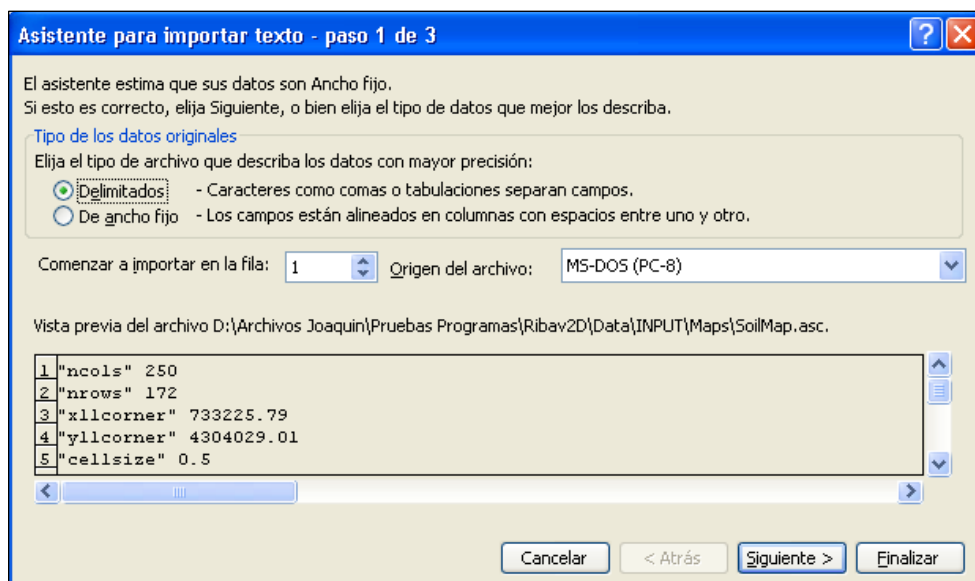
**Figure I.71.** Data menu in Office 2003 showing the “Import Data” application.

First of all we will select the data file that we want to import.



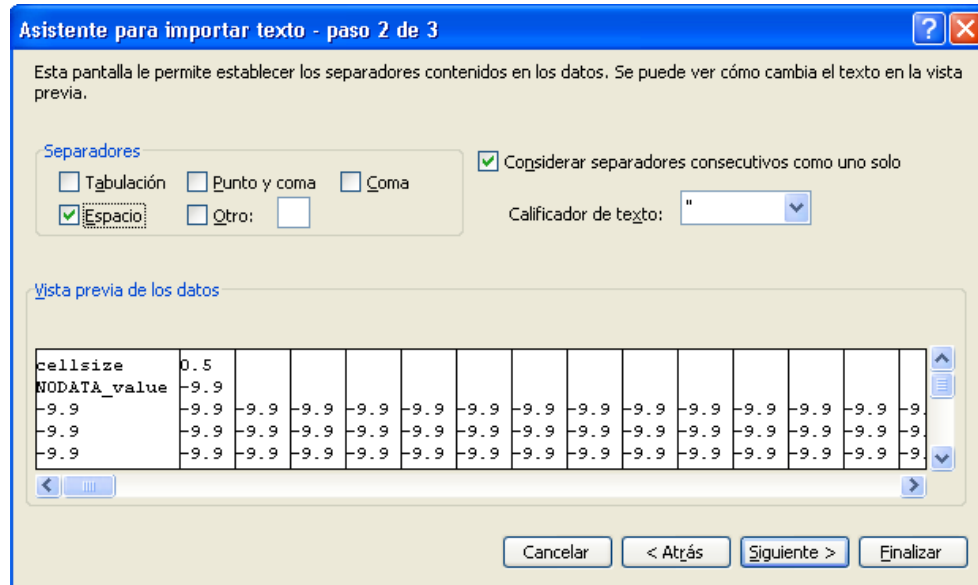
**Figure I.72.** Form where the file to be imported is selected is Excel 2003.

Then a dialog box will appear and we will select the radio button (*circular shape*) named: “Delimited”. In this dialog box we can also select the line from which to start the importation, in case we want to ignore previous headers.



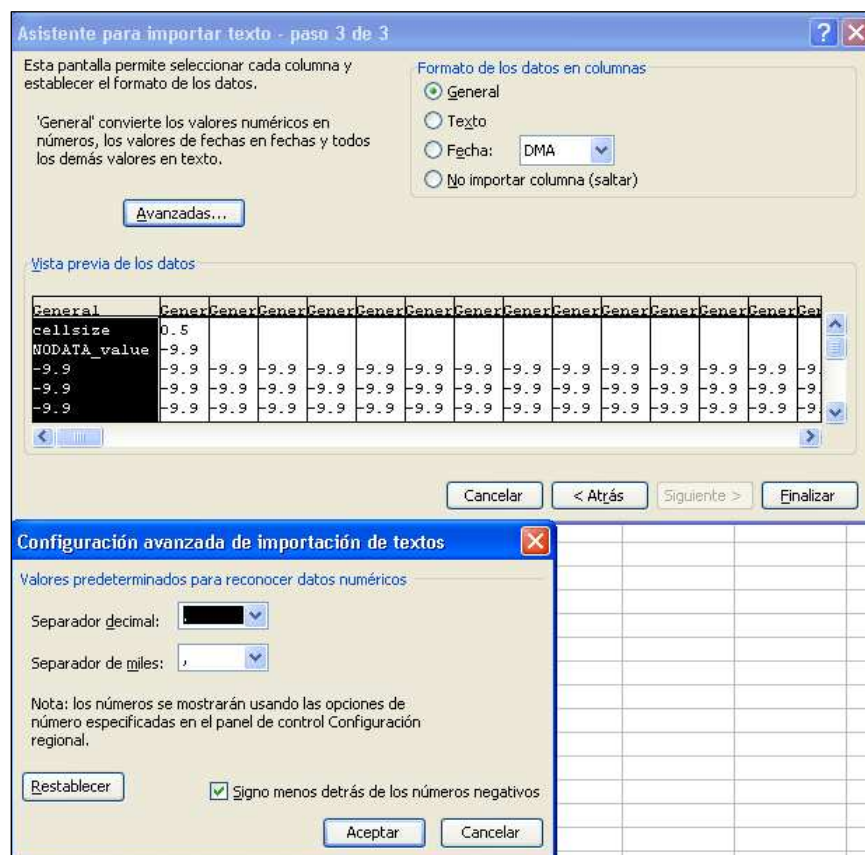
**Figure I.73.** Step 1 of 3 of the “Import Data” application of Excel 2003.

In the next dialog box we select the type of column separator. For ASCII files it will be a space and for CSV files a semicolon. In the bottom part of the dialog box a diagram will appear to let us see if the selected option is adequate or not.



**Figure I.74.** Step 2 of 3 of the “Import Data” application of Excel 2003.

In the last dialog box we can press the button “advanced” if we want to change the definition of decimal separator or thousands separator. In our case we should always have a point as a decimal separator.



**Figure I.75.** Step 3 of 3 of the “Import Data” application of Excel 2003.

**ANEXO IV TUTORIAL DE UTILIZACIÓN DE LA  
APLICACIÓN PARA EL CÁLCULO DEL  
COEFICIENTE DE COHEN PARA MAPAS DE  
VEGETACIÓN (INGLÉS)**

## 1. Introduction

This tutorial explains how to use the software application that calculates the Cohen Kappa Coefficient between observed and simulated vegetation maps. The inputs of this program are the observed and simulated vegetation grid maps, which should have the same dimensions and the same coding of the vegetation types. In the other hand, there are 2 output files produced by this program: a CSV file that stores the results from the calculations and also an ASCII map file, which evaluates the accuracy between the observed and simulated maps. Additionally there is a CSV file, which shows the data loaded by default when the program is executed. The default names and paths of the program files are shown in the following table:

Table II.e Files from the Cohen Coefficient Calculation Software

Default File Name	File Type	Default Relative File Paths	File Function
ObservedVegetationMap.asc	Input	/Data/INPUT/	Grid map of the vegetation observed in the field
SimulatedVegetationMap.asc			Grid map of the vegetation simulated through a model
AccuracyMap.asc	Output	/Data/OUTPUT/	Grid map which indicates the accuracy of the simulation
DataFile.csv			Data file which contains the confusion matrix and the Cohen Coefficient calculation results
DefaultInputs.csv	Default	/Data/Default/	Data file which contains the paths and names of the files loaded by default and also the number of vegetation types, which indicate the size of the confusion matrix

## 2. Equations used in the Calculations

The Cohen Kappa Coefficient evaluates the rate of agreement. It is used in qualitative items and takes into account the effect caused by chance. Specifically in this program the evaluated agreement is between the observed and the simulated vegetation. To carry out this evaluation the program will first build a confusion matrix where the columns indicate the observed items ( $m_1, m_2, m_3$ ) and the rows the simulated items ( $n_1, n_2, n_3$ ). Each cell from the study site will compare the vegetation type from the observed map and the vegetation type from the simulated map, increasing by one the value of the table cell with the correspondent row and column. The number of rows and

the number of columns will be equal to the number of vegetation type items. As an example we are going to assume that a certain study site has 3 vegetation types which will produce the following confusion matrix:

Table II.f **Confusion matrix for 3 vegetation types used for evaluating the Cohen coefficient.**

		Observed			
		m <sub>1</sub>	m <sub>2</sub>	m <sub>3</sub>	
Simulated	n <sub>1</sub>	V <sub>11</sub>	V <sub>21</sub>	V <sub>31</sub>	∑V <sub>m,1</sub>
	n <sub>2</sub>	V <sub>12</sub>	V <sub>22</sub>	V <sub>32</sub>	∑V <sub>m,2</sub>
	n <sub>3</sub>	V <sub>13</sub>	V <sub>23</sub>	V <sub>33</sub>	∑V <sub>m,3</sub>
		∑V <sub>1,n</sub>	∑V <sub>2,n</sub>	∑V <sub>3,n</sub>	∑V <sub>m,n</sub>

After building the matrix, the following summation values are calculated:

- ⇒ Sum of the values of each **row** (e.g. ∑V<sub>m,1</sub>=V<sub>11</sub>+V<sub>21</sub>+V<sub>31</sub>)
- ⇒ Sum of the values of each **column** (e.g. ∑V<sub>1,n</sub>= V<sub>11</sub>+V<sub>12</sub>+V<sub>13</sub>)
- ⇒ Sum of the values of the **main diagonal** (∑V<sub>m,n</sub>= V<sub>11</sub>+V<sub>22</sub>+V<sub>33</sub>), which are the number of accurate results. To avoid confusions with the summation notation, we will call this variable (**fo**).
- ⇒ Sum of **all** the values of the matrix (∑V<sub>m,n</sub>= V<sub>m,1</sub> + V<sub>m,2</sub> + V<sub>m,3</sub>) or (∑V<sub>m,n</sub>= V<sub>1,n</sub> + V<sub>2,n</sub> + V<sub>3,n</sub>). To avoid confusions with the summation notation, we will call this variable (**N**).

Then, the expected frequencies fe<sub>(i)</sub> of each vegetation type are calculated. This is calculated by **multiplying**: [the sum of the values of each **row** which intersects the plant type] **times** [the sum of the values of each **column** which intersects the plant type] and **dividing** into [the sum of all the values of the matrix (N)].

$$\Rightarrow \text{for vegetation 1: } fe_{(1)} = \frac{\sum V_{m,1} \cdot \sum V_{1,n}}{N}$$

$$\Rightarrow \text{for vegetation 2: } fe_{(2)} = \frac{\sum V_{m,2} \cdot \sum V_{2,n}}{N}$$

$$\Rightarrow \text{for vegetation 3: } fe_{(3)} = \frac{\sum V_{m,3} \cdot \sum V_{3,n}}{N}$$



Next, the sum all the expected frequencies is also calculated:

$$fe = \sum fe_{(i)} = fe_{(1)} + fe_{(2)} + fe_{(3)}$$

Afterwards we have all the available values (fo, fe, N) required to obtain the Cohen Kappa coefficient (k), which is calculated by the following equation:

$$k = \frac{fo - fe}{N - fe}$$

If the total number of values (N) of the matrix is high, then it can be assumed that it pertains to a normal distribution. In that case the standard error and the confidence limits (95% and 99%) could also be calculated:

$$\Rightarrow \text{Standard Error } (\sigma_x): \quad \sigma_x = \frac{\sqrt{fo \cdot \left(1 - \frac{fo}{n}\right)}}{N - fe}$$

$$\Rightarrow 95\% \text{ confidence limit: } k \pm 1,96 \cdot \sigma_x$$

$$\Rightarrow 99\% \text{ confidence limit: } k \pm 2,58 \cdot \sigma_x$$

The maximum value of the Kappa coefficient (k) is 1, and the minimum value is 0. If k=1 then all the cases are in the main diagonal and the rest of the cells of the confusion matrix would be empty. If k=0 then it means that there isn't any non-random degree of agreement between the observed and simulated items. The ranges of values of the Kappa coefficient are considered in the following way:

$$\Rightarrow 0.80 < k < 1 : \quad \text{Excellent}$$

$$\Rightarrow 0.60 < k < 0.80 : \quad \text{Good}$$

$$\Rightarrow 0.40 < k < 0.60 : \quad \text{Acceptable}$$

$$\Rightarrow 0.20 < k < 0.40 : \quad \text{Inadequate}$$

$$\Rightarrow 0 < k < 0.20 : \quad \text{Unacceptable}$$

### 3. Program Use Procedure

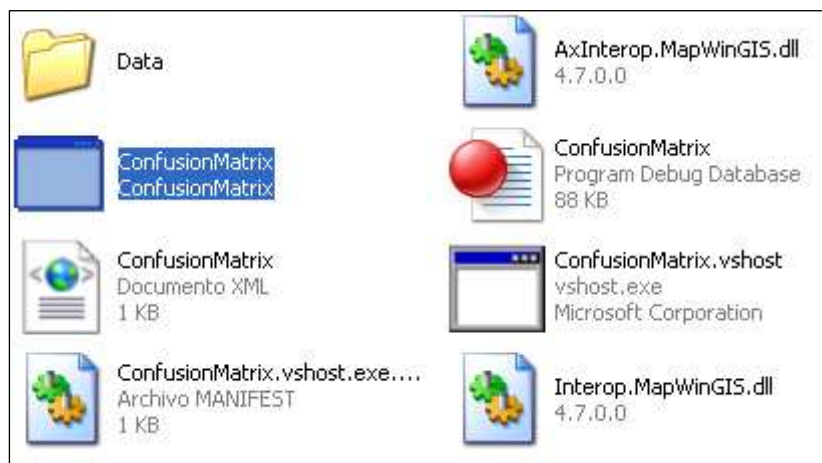
To use the program the following procedure is recommended:

a) First of all, access the “/Data/Default/” subdirectory and open the “DataInputs.csv” file. In the first line an integer number indicates the number of different vegetation types in the simulation. The second and third rows show the default relative paths and names of the input files. The fourth and fifth rows show the default relative paths and names of the output files, which the program is going to create.

```
4
Data\INPUT\ObservedVegetationMap.asc
Data\INPUT\SimulatedVegetationMap.asc
Data\OUTPUT\DataFile.csv
Data\OUTPUT\AccuracyMap.asc
```

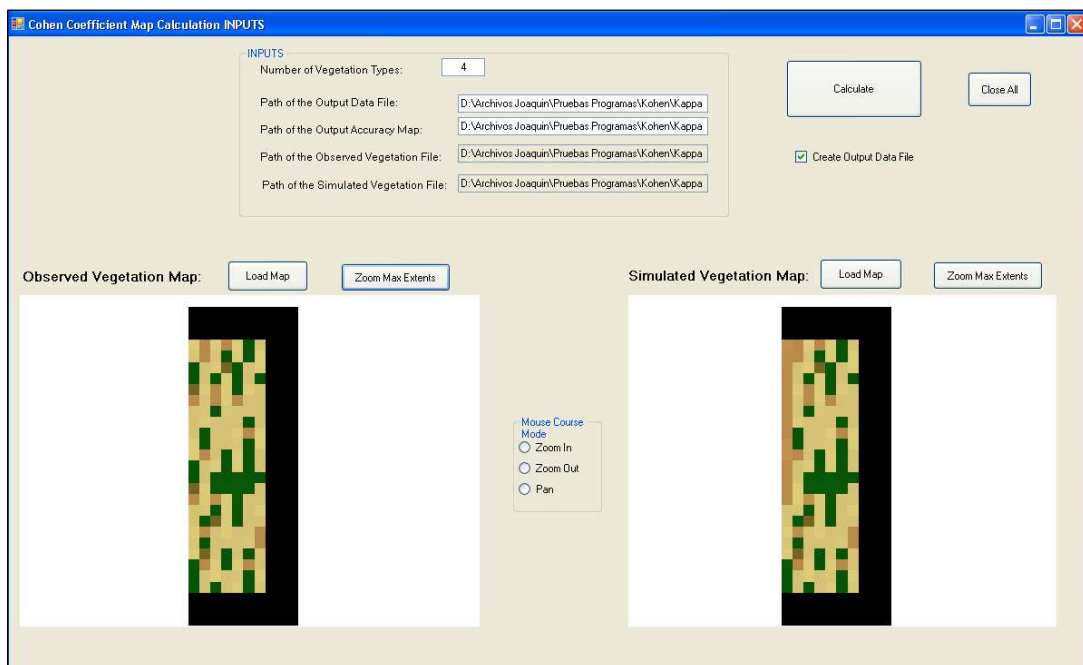
**Figure 11.76.** Data from the “DataInputs.csv” file

b) Double-click on the program’s executable file “ConfusionMatrix.exe” from the main program directory.



**Figure 11.77.** Main directory of the program with the executable file and the “/Data/” folder

c) Once the program is loaded, the first form to appear will be the one of the program’s inputs. Verify then that the observed and simulated vegetation maps are the ones in which you want to carry out the calculations. If not, the loaded maps can be modified by pressing the “Load Map” Button. If you do not want the output file: “DataFile.csv” to be created, then un-check the checkbox with the label: “Create Output Data File.” Additionally, to visualize the maps you can press on the radio buttons from the “Mouse Course Mode” group-box with the “Zoom In”, “Zoom Out” and “Pan” options. Finally, press on the “Calculate” button to carry out the calculations. To prevent errors from happening, the program has to be closed and executed again each time a calculation is made.

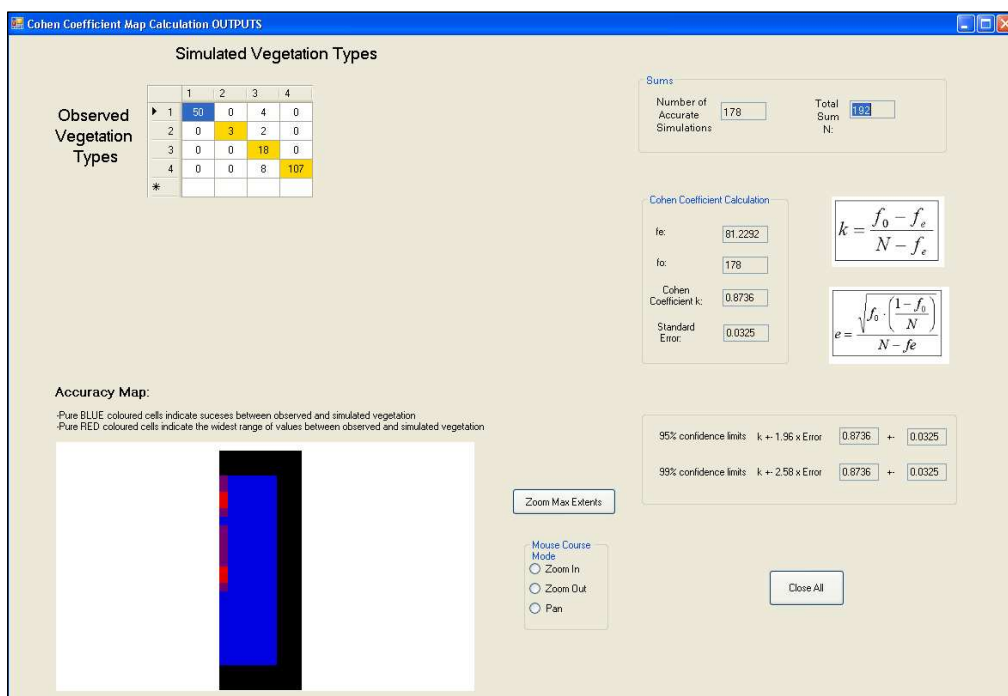


**Figure 11.78.** First form from the program, which is dedicated to the input data.

d) When the calculations finish, the second form of the program appears. This form is dedicated to the output data. At the same time, the subdirectory where the output files are created will open so the user can also analyse these files. On the upper-left part of the form the confusion matrix will appear. In this matrix the rows represent the observed vegetation types and the columns the simulated vegetation types. The main diagonal, where the coincident results are, is highlighted in orange.

On the bottom-left part of the form the “Accuracy Map” appears. This map is the result of the calculation of the absolute value of the subtraction from the plant type numbers between the observed vegetation and the simulated vegetation. The cells with a pure blue colour indicate success in the simulation, which means a subtraction value of 0. In contrast, the cells with a pure red colour will indicate the widest discordance between the observed and the simulated values.

On the right part of the form the numerical results of the Cohen Kappa Coefficient calculation appear. These results include variables as the number of cells with accurate simulations, the total number of cells, the Cohen Coefficient’s value or the standard error.



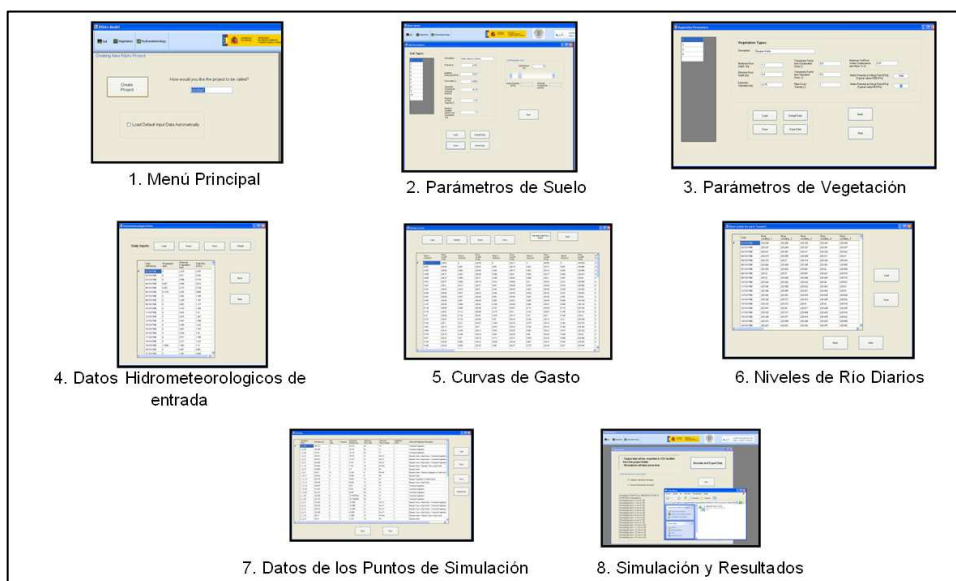
**Figure 11.79.** Second form from the program, which is dedicated to the output data.

**ANEXO VII CÓDIGO FUENTE DEL PROGRAMA  
RIBAV-1D**

En este anexo se presenta el código fuente del software de Ribav1D. El código fuente se presenta distribuido en dos tipos de archivo: los archivos de modulo y los archivos "Windows-Form". Estos se diferencian entre si en que los archivos "Windows-Form" pertenecen a un formulario de Windows de la interfaz gráfica de usuario mientras que los archivos de modulo tan solo contienen instrucciones que realizan cálculos del modelo y no están relacionados con dicha interfaz. Para diferenciarlos se ha aplicado el prefijo "frm" para los nombres de los archivos "Windows-Form" y el prefijo "module" para los archivos de modulo. Los archivos que contienen el código fuente de Ribav 1D son los siguientes:

1. Menú Principal: *"frmMDIParent1.vb"*
2. Parámetros de Suelo: *"FrmSoilParam.vb"*
3. Parámetros de Vegetación: *"FrmVegetParam.vb"*
4. Datos Hidrometeorologicos de entrada: *"FrmHydromet.vb"*
5. Curvas de Gasto: *"FrmRatingCurves.vb"*
6. Niveles de Río Diarios: *"FrmRiverLevels.vb"*
7. Datos de los Puntos de Simulación: *"frmSimPoints.vb"*
8. Simulación y Resultados: *"frmSimPoints.vb"*
9. Modulo 1: *"Module1"*
10. Modulo 2: *"Module2"*

A continuación se muestran los formularios del programa Ribav1D según el orden de aparición al ejecutar el programa:



**Figura III.80.** Formulario del software de Ribav1D

El código fuente de Ribav1D se muestra a continuación distribuido entre sus distintos archivos y formularios:

## 1. Menú Principal: "frmMDIParent1.vb"



```
Imports System.Windows.Forms
Imports System.IO
Imports System.Math
```

```
Public Class frmMDIParent1
```

```
    Private Sub ShowNewForm(ByVal sender As Object, ByVal e As EventArgs)
        ' Cree una nueva instancia del formulario secundario.
        Dim ChildForm As New System.Windows.Forms.Form
        ' Conviértalo en un elemento secundario de este formulario MDI
        antes de mostrarlo.
```

```
        ChildForm.MdiParent = Me
```

```
        m_ChildFormNumber += 1
```

```
        ChildForm.Text = "Ventana " & m_ChildFormNumber
```

```
        ChildForm.Show()
```

```
    End Sub
```

```
    Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)
```

```
        Dim OpenFileDialog As New OpenFileDialog
```

```
        OpenFileDialog.InitialDirectory =
```

```
My.Computer.FileSystem.SpecialDirectories.MyDocuments
```

```
        OpenFileDialog.Filter = "CSV Files (*.csv)|*.csv|All Files
```

```
(*.*)|*.*"
```

```
        If (OpenFileDialog.ShowDialog(Me) =
```

```
System.Windows.Forms.DialogResult.OK) Then
```

```
            Dim FileName As String = OpenFileDialog.FileName
```

```
        End If
```

```
    End Sub
```

```
    Private Sub SaveAsToolStripMenuItem_Click(ByVal sender As Object,
        ByVal e As EventArgs)
```

```
        Dim SaveFileDialog As New SaveFileDialog
```

```
        SaveFileDialog.InitialDirectory =
```

```
My.Computer.FileSystem.SpecialDirectories.MyDocuments
```

```
        SaveFileDialog.Filter = "CSV Files (*.csv)|*.csv|All Files
```

```
(*.*)|*.*"
```



```

        If (SaveFileDialog.ShowDialog(Me) =
System.Windows.Forms.DialogResult.OK) Then
            Dim FileName As String = SaveFileDialog.FileName
        End If
    End Sub

    Private Sub ExitToolsStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs)
        Me.Close()
    End Sub

    Private Sub CascadeToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.Cascade)
    End Sub

    Private Sub TileVerticalToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.TileVertical)
    End Sub

    Private Sub TileHorizontalToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.TileHorizontal)
    End Sub

    Private Sub ArrangeIconsToolStripMenuItem_Click(ByVal sender As
Object, ByVal e As EventArgs)
        Me.LayoutMdi(MdiLayout.ArrangeIcons)
    End Sub

    Private Sub CloseAllToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As EventArgs)
        ' Cierre todos los formularios secundarios del principal.
        For Each ChildForm As Form In Me.MdiChildren
            ChildForm.Close()
        Next
    End Sub

    Private m_ChildFormNumber As Integer

    Private Sub MDIParent1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim rand As Integer
        Dim random As New Random
        rand = random.Next(1, 9999)
        txtProyName.Text = "project_" + Str(rand) + "\"

        FrmHydromet.MdiParent = Me
        frmSimPoints.MdiParent = Me
        FrmRatingCurves.MdiParent = Me
        FrmRiverLevels.MdiParent = Me
        FrmSimulations.MdiParent = Me
        FrmSoilParam.MdiParent = Me
        FrmVegetParam.MdiParent = Me
    End Sub

    Private Sub TsbSoilParameters_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TsbSoilParameters.Click
        FrmSoilParam.Show()
    End Sub

```

```

End Sub

Private Sub TsbVegetationParameters_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TsbVegetationParameters.Click
    FrmVegetParam.Show()
End Sub

Private Sub TsbClimaticParameters_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
TsbClimaticParameters.Click
    FrmHydromet.Show()
End Sub

Private Sub btCreateProyect_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs)
    ProjAddress = System.Environment.CurrentDirectory +
"\Data\Projects\" + txtProyName.Text + "\"
    Directory.CreateDirectory(ProjAddress)
    gboxProject.Hide()
    FrmSoilParam.Show()
End Sub

Private Sub btCreateProyect_Click_1(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btCreateProyect.Click
    ProjAddress = System.Environment.CurrentDirectory +
"\Data\Projects\" + txtProyName.Text + "\"
    Directory.CreateDirectory(ProjAddress)

    InputAddress = ProjAddress + "InputData"
    OutputAddress = ProjAddress + "OutputData"
    Directory.CreateDirectory(InputAddress)
    Directory.CreateDirectory(OutputAddress)

    FrmSoilParam.Show()
    gboxProject.Hide()
    If chkAutomatic.Checked = True Then
        FrmSoilParam.Show()
        Call Automatic()
    End If
End Sub

Private Sub PictureBox3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox3.Click
    Process.Start("www.upv.es")
End Sub

Private Sub PictureBox2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles PictureBox2.Click
    Process.Start("www.iiama.upv.es")
End Sub

End Class

```

## 2. Parámetros de Suelo: "FrmSoilParam.vb"

```

Public Class FrmSoilParam

    Public SoilFileName As String

    Private Sub btNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btNext.Click
        Dim r As Integer
        'we store in a variable the current row index
        r = SoilDataGridView.CurrentRow.Index
        FrmVegetParam.Show()
        Me.Hide()

    End Sub

    Private Sub FrmVegetParam_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        Call DefaultFile()
    End Sub

    Private Sub DefaultFile()
        SoilFileName = (System.Environment.CurrentDirectory +
"\data\Default\SoilParameters.csv")

        SoilDataGridView.Rows.Add()
        Dim TextLine As String = ""
        Dim SplitLine() As String
    
```

```

If System.IO.File.Exists(SoilFileName) = True Then
    Dim objReader As New System.IO.StreamReader(SoilFileName)
    If SoilDataGridView.RowCount <> 0 Then
        SoilDataGridView.Rows.Clear()
    End If

    Do While objReader.Peek() <> -1
        TextLine = objReader.ReadLine()
        SplitLine = Split(TextLine, ";")
        SoilDataGridView.Rows.Add(SplitLine)
    Loop
Else
    MessageBox.Show("Default Vegetation Parameter's File has been
Removed or Corrupted", "Removed File", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
End If
Call TextboxUpdate()

End Sub

Private Sub TextboxUpdate()
    Dim i As Integer
    i = SoilDataGridView.CurrentRow.Index

    txtPst.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colPst").Value()
'Max Root Depth Zr [m]
    txtIp.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colIp").Value()
'Effective Root Depth Ze [m]
    txtPb.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colPb").Value()
'Extinction Saturation Psat [m]
    txtKs.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colKs").Value()
'Moisture at Wilting Point(cm3/cm3)
    txtMfc.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colMfc").Value()
'Fraction of Active Roots below
    txtSoilDescription.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colSoilDescription").Value()
'Brief Soil Description
    txtZmin.Text =
SoilDataGridView.Rows.Item(i).Cells.Item("colZmin").Value()
'Minimum Capillary Depth to be Considered [m]

End Sub

Private Sub btErase_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btErase.Click

    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then

```

```

        'We clear the Textboxes without Saving
        Call EraseValues()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSave.PerformClick()
        'And now after Saving the data we Clear the Textboxes
        Call EraseValues()
    End If

End Sub

Private Sub EraseValues()

    Dim rowcount As Integer
    rowcount = SoilDataGridView.RowCount
    For i = 0 To rowcount - 2
        SoilDataGridView.Rows.Clear()
    Next i

    txtPst.Text = 0
    txtIp.Text = 0
    txtPb.Text = 0
    txtKs.Text = 0
    txtMfc.Text = 0
    txtSoilDescription.Text = 0
    txtZmin.Text = 0

End Sub

Private Sub LoadValues()
    SoilFileName = ""
    OpenFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\Default")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    OpenFileDialog1.RestoreDirectory = True

    If (OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK) Then
        SoilFileName = OpenFileDialog1.FileName
    End If

    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(SoilFileName) = True Then
        Dim objReader As New System.IO.StreamReader(SoilFileName)
        If SoilDataGridView.RowCount <> 0 Then
            SoilDataGridView.Rows.Clear()
        End If
        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            SoilDataGridView.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("File Does Not Exist")
    End If

    Call TextboxUpdate()
End Sub

```

```

Private Sub btLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btLoad.Click

    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
        MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We the new data without Saving
        Call LoadValues()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btnSave.PerformClick()
        'And now after Saving the data we load the new data
        Call LoadValues()
    End If

End Sub

```

```

Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
    Dim rowcount As Integer
    rowcount = SoilDataGridView.RowCount
    Dim i As Integer
    Dim j As Integer
    Dim record(rowcount - 2) As Object

    SaveFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    SaveFileDialog1.Title = "Save File"
    SaveFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    SaveFileDialog1.FileName = "SoilParameters_1"
    SaveFileDialog1.OverwritePrompt = True
    SaveFileDialog1.ShowDialog()

    SoilFileName = SaveFileDialog1.FileName
    Dim objWriter As New System.IO.StreamWriter(SoilFileName)
    j = SoilDataGridView.CurrentRow.Index

    SoilDataGridView.Rows.Item(j).Cells.Item(1).Value =
Val(txtPst.Text)
    SoilDataGridView.Rows.Item(j).Cells.Item(2).Value =
Val(txtIp.Text)
    SoilDataGridView.Rows.Item(j).Cells.Item(3).Value =
Val(txtPb.Text)
    SoilDataGridView.Rows.Item(j).Cells.Item(4).Value =
Val(txtKs.Text)
    SoilDataGridView.Rows.Item(j).Cells.Item(5).Value =
Val(txtMfc.Text)
    SoilDataGridView.Rows.Item(j).Cells.Item(6).Value =
Val(txtSoilDescription.Text)

    For i = 0 To rowcount - 2

```

```

        record(i) =
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(0).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(1).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(2).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(3).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(4).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(5).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(6).Value) & _
        "; " &
CStr(SoilDataGridView.Rows.Item(i).Cells.Item(7).Value)

        objWriter.WriteLine(record(i))

    Next i
    objWriter.Close()

End Sub

Private Sub btDefault_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btDefault.Click

    If SoilDataGridView.RowCount <= 1 Then
        'There are no rows with values
        Call DefaultFile()
    Else
        'A dialogbox appears in case the user wants to save the
previous data
        Dim reply As DialogResult = MessageBox.Show("Do you want to
save the current data?", "", _
                                                    MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
        If reply = DialogResult.Cancel Then
            'Nothing happens
        ElseIf reply = DialogResult.No Then
            'We load the default values without Saving
            Call DefaultFile()
        ElseIf reply = DialogResult.Yes Then
            'We go to the SAVE DIALOG to SAVE the Data from the
Textboxes
            btSave.PerformClick()
            'And now after Saving the data we load the default values
            Call DefaultFile()
        End If
    End If

End Sub

Private Sub SoilDataGridView_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles SoilDataGridView.Click
    Call TextBoxUpdate()
End Sub

```

```

Private Sub hsbTest_Scroll(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.ScrollEventArgs) Handles hsbTest.Scroll

    hsbTest.Maximum = Val(txtPst.Text) * 100

    Dim PmatTst As Single = Val(txtPmatTst.Text)           'Campbell
Matrix Potential being used for testing results (mm)
    Dim PsoilTst As Single = Val(txtH.Text)               'Soil
Moisture (mm/mm)
    Dim i As Integer
    i = SoilDataGridView.CurrentRow.Index '+ 1

    Dim PstT As Single      'Φ Porosity (cm3/cm3)
    Dim IpT As Single       'λ=Pore -size index
    Dim PbT As Single       'Bubbling capillary pressure (mm)
    Dim KsT As Single       'Fully Saturated conductivity (mm/hr)

    PstT = SoilDataGridView.Rows.Item(i).Cells.Item("colPst").Value()
    IpT = SoilDataGridView.Rows.Item(i).Cells.Item("colIp").Value()
    PbT = SoilDataGridView.Rows.Item(i).Cells.Item("colPb").Value()
    KsT = SoilDataGridView.Rows.Item(i).Cells.Item("colKs").Value()

    txtH.Text = hsbTest.Value
    txtPmatTst.Text = Math.Round(PbT / ((hsbTest.Value / PstT / 100)
^ (1 / IpT)), 3)
    txtKsTst.Text = KsT * (PbT / Val(txtPmatTst.Text)) ^ (3 * IpT +
2)
End Sub

End Class

```



### 3. Parámetros de Vegetación: "FrmVegetParam.vb"

```
Public Class FrmVegetParam
```

```
    Public VegetFileName As String
```

```
    Private Sub FrmVegetParam_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        VegetationDataGridView.Columns.Add("colAcronim", "colAcronim")
        VegetationDataGridView.Columns.Add("colVegCode", "colVegCode")
        VegetationDataGridView.Columns.Add("colZr", "colZr")
        VegetationDataGridView.Columns.Add("colZe", "colZe")
        VegetationDataGridView.Columns.Add("colZsat", "colZsat")
        VegetationDataGridView.Columns.Add("colRi", "colRi")
        VegetationDataGridView.Columns.Add("colRj", "colRj")
        VegetationDataGridView.Columns.Add("colCrt", "colCrt")
        VegetationDataGridView.Columns.Add("colPwp", "colPwp")
        VegetationDataGridView.Columns.Add("colPcrit", "colPcrit")
        VegetationDataGridView.Columns.Add("colVegDescription",
"colVegDescription")
```

```
        VegetationDataGridView.Columns.Add("colCov0", "colCov0")
        VegetationDataGridView.Columns.Add("colCov1", "colCov1")
        VegetationDataGridView.Columns.Add("colCov2", "colCov2")
        VegetationDataGridView.Columns.Add("colCov3", "colCov3")
        VegetationDataGridView.Columns.Add("colCov4", "colCov4")
        VegetationDataGridView.Columns.Add("colCov5", "colCov5")
        VegetationDataGridView.Columns.Add("colCov6", "colCov6")
        VegetationDataGridView.Columns.Add("colCov7", "colCov7")
        VegetationDataGridView.Columns.Add("colCov8", "colCov8")
        VegetationDataGridView.Columns.Add("colCov9", "colCov9")
        VegetationDataGridView.Columns.Add("colCov10", "colCov10")
        VegetationDataGridView.Columns.Add("colCov11", "colCov11")
```

```

VegetationDataGridView.Columns("colAcronim").Visible = True
VegetationDataGridView.Columns("colAcronim").Width = 65

VegetationDataGridView.Columns("colVegCode").Visible = False
VegetationDataGridView.Columns("colZr").Visible = False
VegetationDataGridView.Columns("colZe").Visible = False
VegetationDataGridView.Columns("colZsat").Visible = False
VegetationDataGridView.Columns("colRi").Visible = False
VegetationDataGridView.Columns("colRj").Visible = False
VegetationDataGridView.Columns("colCrt").Visible = False
VegetationDataGridView.Columns("colPwp").Visible = False
VegetationDataGridView.Columns("colPcrit").Visible = False
VegetationDataGridView.Columns("colVegDescription").Visible =
False

VegetationDataGridView.Columns("colCov0").Visible = False
VegetationDataGridView.Columns("colCov1").Visible = False
VegetationDataGridView.Columns("colCov2").Visible = False
VegetationDataGridView.Columns("colCov3").Visible = False
VegetationDataGridView.Columns("colCov4").Visible = False
VegetationDataGridView.Columns("colCov5").Visible = False
VegetationDataGridView.Columns("colCov6").Visible = False
VegetationDataGridView.Columns("colCov7").Visible = False
VegetationDataGridView.Columns("colCov8").Visible = False
VegetationDataGridView.Columns("colCov9").Visible = False
VegetationDataGridView.Columns("colCov10").Visible = False
VegetationDataGridView.Columns("colCov11").Visible = False

Call DefaultVegFile()

End Sub

Private Sub DefaultVegFile()
    VegetFileName = (System.Environment.CurrentDirectory +
"\data\Default\VegetationParameters.csv")
    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(VegetFileName) = True Then
        ' Call EraseDaily_DataGridView()
        Dim objReader As New System.IO.StreamReader(VegetFileName)
        If VegetationDataGridView.RowCount <> 0 Then
            VegetationDataGridView.Rows.Clear()
        End If

        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            VegetationDataGridView.Rows.Add(SplitLine)
        Loop
    Else
        MessageBox.Show("Default Vegetation Parameter's File has been
Removed or Corrupted", "Removed File", MessageBoxButtons.OK,
MessageBoxIcon.Warning)
    End If
End Sub

```

```

    Call TextBoxUpdate()

End Sub

Private Sub btNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btNext.Click

    Call DataTableUpdate()

    'Before passing to the next form 2 data tests are made to prevent
errors
    Dim reply As DialogResult
    'In Test 1 we see if Zr <= Ze for each Vegetation Type (j)
    For i = 0 To VegetationDataGridView.RowCount - 2
        If
VegetationDataGridView.Rows.Item(i).Cells.Item("colZr").Value() <=
VegetationDataGridView.Rows.Item(i).Cells.Item("colZe").Value() Then
            'We inform about the error
            MsgBox("The Maximum root depth (Zr)should be GREATER than
the Effective root depth (Ze). An error can occur!")
            'We give the user the possibility to leave the program
            reply = MessageBox.Show("Do you want to exit the program
to correct the vegetation input data?", "Error", MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
            If reply = DialogResult.Yes Then
                MdiParent.Close()
            Else
                GoTo test2
            End If
        End If
    Next i

test2:
    'In Test 2 we see if Cov<=0.05 for each Vegetation Type (j) and
each Month (k).
    For j = 0 To VegetationDataGridView.RowCount - 2
        For k = 0 To 11
            If
VegetationDataGridView.Rows.Item(j).Cells.Item("colCov" +
CStr(k)).Value() < 0.05 Then
                'We inform about the error
                MsgBox("The Plant Cover (Cov) from ALL Vegetation
Types and ALL Months should be GREATER than 0.05. An error can occur!")
                'We give the user the possibility to leave the
program
                reply = MessageBox.Show("Do you want to exit the
program to correct the vegetation input data?", "Error",
MessageBoxButtons.YesNo, MessageBoxIcon.Question)
                If reply = DialogResult.Yes Then
                    Me.Close()
                End If
            Else
                'If we pass both test we go to the next form
                FrmHydromet.Show()
                Me.Hide()
            End If
        Next k
    Next j

End Sub

```

```

Private Sub TextboxUpdate()
    Dim i As Integer
    i = VegetationDataGridView.CurrentRow.Index

    txtVegDescription.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colVegDescription").Value
() _
    + " " +
VegetationDataGridView.Rows.Item(i).Cells.Item("colVegCode").Value()
'Vegetation Type Name

    txtZr.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colZr").Value()
'Maximum Root Depth Zr [m]
    txtZe.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colZe").Value()
'Effective Root Depth Ze [m]
    txtPsat.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colZsat").Value()
'Extinction Saturation Psat [m]
    txtRi.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colRi").Value()
'Transpiration Factor from Unsaturated Zone [ ]
    txtRj.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colRj").Value()
'Transpiration Factor from Saturated Zone [ ]
    'txtCov.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov").Value()
'Plant Cover Cv_yearly [m2/m2]
    txtCrt.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCrt").Value()
'Maximum Soil-Root Water Conductance CRT [mmMpa-1h-1]
    txtPwp.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colPwp").Value()
'Wilting Point Matrix Potencial (Kpa)
    txtPcrit.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colPcrit").Value()
'Critical Point Matrix Potencial (Kpa)

    txtCov0.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov0").Value()
    txtCov1.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov1").Value()
    txtCov2.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov2").Value()
    txtCov3.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov3").Value()
    txtCov4.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov4").Value()
    txtCov5.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov5").Value()
    txtCov6.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov6").Value()
    txtCov7.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov7").Value()
    txtCov8.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov8").Value()
    txtCov9.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov9").Value()

```

```

        txtCov10.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov10").Value()
        txtCov11.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov11").Value()

End Sub

Private Sub DataTableUpdate()
    Dim i As Integer
    i = VegetationDataGridView.CurrentRow.Index

    VegetationDataGridView.Rows.Item(i).Cells.Item("colZr").Value =
txtZr.Text                'Maximum Root Depth Zr [m]
    VegetationDataGridView.Rows.Item(i).Cells.Item("colZe").Value =
txtZe.Text                'Effective Root Depth Ze [m]
    VegetationDataGridView.Rows.Item(i).Cells.Item("colZsat").Value =
txtPsat.Text             'Extinction Saturation Psat [m]
    VegetationDataGridView.Rows.Item(i).Cells.Item("colRi").Value =
txtRi.Text                'Transpiration Factor from Unsaturated
Zone [ ]
    VegetationDataGridView.Rows.Item(i).Cells.Item("colRj").Value =
txtRj.Text                'Transpiration Factor from Saturated
Zone [ ]
    'txtCov.Text =
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov").Value()
'Plant Cover Cv_yearly [m2/m2]
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCrt").Value =
txtCrt.Text              'Maximum Soil-Root Water Conductance CRT
[mmMpa-1h-1]
    VegetationDataGridView.Rows.Item(i).Cells.Item("colPwp").Value =
txtPwp.Text              'Wilting Point Matrix Potencial (Kpa)
    VegetationDataGridView.Rows.Item(i).Cells.Item("colPcrit").Value
= txtPcrit.Text          'Critical Point Matrix Potencial (Kpa)

    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov0").Value =
txtCov0.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov1").Value =
txtCov1.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov2").Value =
txtCov2.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov3").Value =
txtCov3.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov4").Value =
txtCov4.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov5").Value =
txtCov5.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov6").Value =
txtCov6.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov7").Value =
txtCov7.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov8").Value =
txtCov8.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov9").Value =
txtCov9.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov10").Value
= txtCov10.Text
    VegetationDataGridView.Rows.Item(i).Cells.Item("colCov11").Value
= txtCov11.Text

End Sub

```

```

Private Sub VegetationDataGridView_Click(ByVal sender As Object,
ByVal e As System.EventArgs)
    Call TextBoxUpdate()
End Sub

Private Sub btEraseVeg_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btEraseVeg.Click

    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call EraseVeg()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSaveVeg.PerformClick()
        'And now after Saving the data we Clear the Textboxes
        Call EraseVeg()
    End If

End Sub

Private Sub EraseVeg()

    Dim rowcount As Integer
    rowcount = VegetationDataGridView.RowCount
    For i = 0 To rowcount - 2
        VegetationDataGridView.Rows.Clear()
    Next i
    txtZr.Text = 0
    txtZe.Text = 0
    txtPsat.Text = 0
    txtRi.Text = 0
    txtRj.Text = 0
    txtCrt.Text = 0
    txtPwp.Text = 0
    txtPcrit.Text = 0
    txtVegDescription.Text = 0

    txtCov0.Text = 0
    txtCov1.Text = 0
    txtCov2.Text = 0
    txtCov3.Text = 0
    txtCov4.Text = 0
    txtCov5.Text = 0
    txtCov6.Text = 0
    txtCov7.Text = 0
    txtCov8.Text = 0
    txtCov9.Text = 0
    txtCov10.Text = 0
    txtCov11.Text = 0

End Sub

```

```

Private Sub LoadValues()

    OpenFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    OpenFileDialog1.RestoreDirectory = True
    If (OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK) Then
        VegetFileName = OpenFileDialog1.FileName
    End If

    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(VegetFileName) = True Then
        Dim objReader As New System.IO.StreamReader(VegetFileName)
        If VegetationDataGridView.RowCount <> 0 Then
            VegetationDataGridView.Rows.Clear()
        End If

        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            VegetationDataGridView.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("File Does Not Exist")
    End If
    Call TextboxUpdate()
End Sub

Private Sub btLoadVeg_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btLoadVeg.Click
    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call LoadValues()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSaveVeg.PerformClick()
        'And now after saving the data, we clear the Textboxes
        Call LoadValues()
    End If

End Sub

Private Sub btSaveVeg_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btSaveVeg.Click

    Dim rowcount As Integer
    rowcount = VegetationDataGridView.RowCount

```

```

Dim i As Integer
Dim j As Integer
Dim record(rowcount - 2) As Object

SaveFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
SaveFileDialog1.Title = "Save File"
SaveFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
SaveFileDialog1.FileName = "VegetationParameters_1"
SaveFileDialog1.OverwritePrompt = True
SaveFileDialog1.ShowDialog()

Dim FILE_NAME As String = SaveFileDialog1.FileName
Dim objWriter As New System.IO.StreamWriter(FILE_NAME)

j = VegetationDataGridView.CurrentRow.Index

VegetationDataGridView.Rows.Item(j).Cells.Item("colVegDescription").Value
= Val(txtVegDescription.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colZr").Value =
Val(txtZr.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colZe").Value =
Val(txtZe.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colZsat").Value =
Val(txtPsat.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colRi").Value =
Val(txtRi.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colRj").Value =
Val(txtRj.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colCrt").Value =
Val(txtCrt.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colPwp").Value =
Val(txtPwp.Text)
VegetationDataGridView.Rows.Item(j).Cells.Item("colPcrit").Value
= Val(txtPcrit.Text)

VegetationDataGridView.Rows.Item(i).Cells.Item("colCov0").Value =
Val(txtCov0.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov1").Value =
Val(txtCov1.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov2").Value =
Val(txtCov2.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov3").Value =
Val(txtCov3.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov4").Value =
Val(txtCov4.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov5").Value =
Val(txtCov5.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov6").Value =
Val(txtCov6.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov7").Value =
Val(txtCov7.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov8").Value =
Val(txtCov8.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov9").Value =
Val(txtCov9.Text)
VegetationDataGridView.Rows.Item(i).Cells.Item("colCov10").Value
= Val(txtCov10.Text)

```



```

VegetationDataGridView.Rows.Item(i).Cells.Item("colCov11").Value
= Val(txtCov11.Text)

For i = 0 To rowcount - 2
    record(i) =
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colVegCode").Value)
& _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colAcronim").Value)
& _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colVegDescription").
Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colZr").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colZe").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colZsat").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colRi").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colRj").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCrt").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colPwp").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colPcrit").Value) &
_
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov0").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov1").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov2").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov3").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov4").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov5").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov6").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov7").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov8").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov9").Value) & _
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov10").Value) &
_
    "; " &
CStr(VegetationDataGridView.Rows.Item(i).Cells.Item("colCov11").Value)

```

```

        objWriter.WriteLine(record(i))
    Next i

    objWriter.Close()

End Sub

Private Sub btDefaultVeg_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btDefaultVeg.Click

    If VegetationDataGridView.RowCount <= 1 Then
        'There are no rows with values
        Call DefaultVegFile()
    Else
        'A dialogbox appears in case the user wants to save the
previous data
        Dim reply As DialogResult = MessageBox.Show("Do you want to
save the current data?", "", _
                                                    MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
        If reply = DialogResult.Cancel Then
            'Nothing happens
        ElseIf reply = DialogResult.No Then
            'We load the default values without Saving
            Call DefaultVegFile()
        ElseIf reply = DialogResult.Yes Then
            'We go to the SAVE DIALOG to SAVE the Data from the
Textboxes
            btSaveVeg.PerformClick()
            'And now after Saving the data we load the default values
            Call DefaultVegFile()
        End If
    End If

End Sub

Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
    FrmSoilParam.Show()
    Me.Hide()
End Sub

Private Sub VegetationDataGridView_CellContentClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles VegetationDataGridView.CellContentClick
    Call TextboxUpdate()
End Sub

Private Sub VegetationDataGridView_CellClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles VegetationDataGridView.CellClick
    Call TextboxUpdate()
End Sub

End Class

```

#### 4. Datos Hidrometeorológicos de entrada: "FrmHydromet.vb"

Date [dd/mm/yy]	Precipitation (mm)	Potencial Evaporation (mm)	Daily Flow (m3/s)
13/10/1951	0.49284	1.798635838	1.48
14/10/1951	0	1.824815096	1.48
15/10/1951	0	1.763880272	1.48
16/10/1951	0	1.720429758	1.48
17/10/1951	0	1.69909119	1.48
18/10/1951	5.7317	1.708437904	1.12
19/10/1951	0.83237	1.805190524	1
20/10/1951	17.46026	1.838053174	1
21/10/1951	2.18209	1.79135129	1.12
22/10/1951	1.5318	1.818678516	1
23/10/1951	0	1.70819316	1
24/10/1951	0	1.705945078	1
25/10/1951	0	1.838555323	0.759
26/10/1951	0	1.813606958	0.879
27/10/1951	0	1.821516257	0.818
28/10/1951	0	1.854970639	7.44
29/10/1951	0	1.82348314	3.94
30/10/1951	0.90504	1.851653707	1.9
31/10/1951	0	1.802824912	1.36
01/11/1951	0.32054	1.428743475	1.24
02/11/1951	18.9587	1.390911594	1.24

```
Imports System.Math
```

```
Public Class FrmHydromet
```

```
Public DailyClimaticFileName As String
```

```
Private Sub NextForm()
```

```
rowcount_daily = DailyInputsTable.RowCount
```

```
ReDim TimeD(rowcount_daily)
```

```
ReDim PPd(rowcount_daily)
```

```
ReDim ETPd(rowcount_daily)
```

```
ReDim MonthNumber(rowcount_daily)
```

```
For i = 0 To rowcount_daily - 2
```

```
TimeD(i) =
```

```
DailyInputsTable.Rows.Item(i).Cells.Item(0).Value()
```

```
PPd(i) = DailyInputsTable.Rows.Item(i).Cells.Item(1).Value()
```

```
ETPd(i) = DailyInputsTable.Rows.Item(i).Cells.Item(2).Value()
```

```

        MonthNumber(i) =
Month(DailyInputsTable.Rows.Item(i).Cells.Item(0).Value)
    Next i

    FrmRatingCurves.Show()
    Me.Hide()

End Sub

Private Sub btNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btNext.Click
    Call NextForm()
End Sub

Private Sub btLoadDaily_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btLoadDaily.Click
    OpenFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    OpenFileDialog1.RestoreDirectory = True

    If (OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK) Then
        DailyClimaticFileName = OpenFileDialog1.FileName
    End If

    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(DailyClimaticFileName) = True Then
        Dim objReader As New
System.IO.StreamReader(DailyClimaticFileName)
        If DailyInputsTable.RowCount <> 0 Then
            DailyInputsTable.Rows.Clear()
        End If
        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            DailyInputsTable.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("File Does Not Exist")
    End If

End Sub

Private Sub btEraseDaily_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btEraseDaily.Click
    Dim rowcount_daily As Integer
    rowcount_daily = DailyInputsTable.RowCount
    For i = 0 To rowcount_daily - 2
        DailyInputsTable.Rows.Item(i).Cells.Item(0).Value = ""
        DailyInputsTable.Rows.Item(i).Cells.Item(1).Value = 0
        DailyInputsTable.Rows.Item(i).Cells.Item(2).Value = 0
        DailyInputsTable.Rows.Item(i).Cells.Item(3).Value = 0
        DailyInputsTable.Rows.Item(i).Cells.Item(4).Value = 0
    Next i

```

```

        DailyInputsTable.Rows.Clear()
    End Sub

    Private Sub btSaveDaily_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btSaveDaily.Click
        rowcount_daily = DailyInputsTable.RowCount
        Dim i As Integer
        Dim record(rowcount_daily - 2) As Object

        SaveFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
        SaveFileDialog1.Title = "Save File"
        SaveFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
        SaveFileDialog1.FileName = "Daily Met_1"
        SaveFileDialog1.OverwritePrompt = True
        SaveFileDialog1.ShowDialog()

        DailyClimaticFileName = ""
        DailyClimaticFileName = SaveFileDialog1.FileName
        Dim objWriter As New
System.IO.StreamWriter(DailyClimaticFileName)

        For i = 0 To rowcount_daily - 2
            record(i) =
CStr(DailyInputsTable.Rows.Item(i).Cells.Item(0).Value) & _
                "; " &
CStr(DailyInputsTable.Rows.Item(i).Cells.Item(1).Value) & _
                "; " &
CStr(DailyInputsTable.Rows.Item(i).Cells.Item(2).Value) & _
                "; " &
CStr(DailyInputsTable.Rows.Item(i).Cells.Item(3).Value) & _
                "; " &
CStr(DailyInputsTable.Rows.Item(i).Cells.Item(4).Value)
            objWriter.WriteLine(record(i))
        Next i

        objWriter.Close()

    End Sub

    Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
        FrmVegetParam.Show()
        Me.Hide()
    End Sub

    Private Sub FrmHydromet_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        Call QuickLoad_DailyInputsTable()

    End Sub

    Private Sub QuickLoad_DailyInputsTable()

        DailyClimaticFileName = ""
        DailyClimaticFileName = (System.Environment.CurrentDirectory +
"\data\Default\Hydromet.csv")

```

```

    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(DailyClimaticFileName) = True Then
        ' Call EraseDaily_DataGridView()
        Dim objReader As New
System.IO.StreamReader(DailyClimaticFileName)
        If DailyInputsTable.RowCount <> 0 Then
            DailyInputsTable.Rows.Clear()
        End If
        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            DailyInputsTable.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("File Does Not Exist")
    End If

End Sub

Private Sub btDefault_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btDefault.Click
    Call QuickLoad_DailyInputsTable()
End Sub

End Class

```

## 5. Curvas de Gasto: "FrmRatingCurves.vb"

Flow 1 Q (m3/s)	River Level1 H (m)	Flow 2 Q (m3/s)	River Level2 H (m)	Flow 3 Q (m3/s)	River Level3 H (m)	Flow 4 Q (m3/s)	River Level4 H (m)	Flow 5 Q (m3/s)	River Level5 H (m)	F C
0	847.71	0	847.64	0	847.68	0	847.55	0	848.36	0
0.25	847.85	0.25	847.91	0.25	847.96	0.25	848.08	0.25	848.6	0.
0.47	847.91	0.47	847.99	0.47	848.04	0.47	848.17	0.47	848.65	0.
1	848.03	1	848.13	1	848.19	1	848.32	1	848.74	1
2	848.19	2	848.31	2	848.38	2	848.5	2	848.85	2
3	848.31	3	848.45	3	848.53	3	848.63	3	848.94	3
4	848.41	4	848.56	4	848.64	4	848.74	4	849.02	4
5	848.49	5	848.65	5	848.74	5	848.83	5	849.1	5
6	848.56	6	848.73	6	848.83	6	848.91	6	849.16	6
7	848.62	7	848.8	7	848.92	7	848.99	7	849.25	7
7.5	848.65	7.5	848.84	7.5	848.95	7.5	849.03	7.5	849.27	7.
8	848.68	8	848.87	8	848.99	8	849.06	8	849.33	8
10	848.79	10	849	10	849.13	10	849.19	10	849.42	10
15	849.05	15	849.27	15	849.43	15	849.51	15	849.61	15
20	849.28	20	849.52	20	849.68	20	849.77	20	849.86	20
25	849.47	25	849.71	25	849.87	25	849.98	25	850.06	25
30	849.6	30	849.85	30	850	30	850.13	30	850.22	30
40	849.85	40	850.09	40	850.24	40	850.39	40	850.48	40
50	850.04	50	850.25	50	850.4	50	850.58	50	850.69	50
75	850.33	75	850.54	75	850.7	75	850.98	75	851.13	75
100	850.56	100	850.77	100	850.92	100	851.3	100	851.46	100
150	850.98	150	851.16	150	851.32	150	851.78	150	851.97	150
200	851.37	200	851.5	200	851.66	200	852.19	200	852.39	200

```
Imports System.IO
```

```
Public Class FrmRatingCurves
```

```
    Public RatingCurveFileName As String
```

```
    Private Sub btNext_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
        FrmSimulations.Show()
```

```
        Me.Hide()
```

```
    End Sub
```

```
    Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
        FrmHydromet.Show()
```

```
        Me.Hide()
```

```
    End Sub
```

```
    Private Sub btLoadDefault_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btDefault.Click
```

```
        'Relative path of the raing curves default file
```

```
        Dim InputFile As String = "\Data\Default\RatingCurves.csv"
```

```
        'The absolute file path
```

```
        RatingCurveFileName = System.Environment.CurrentDirectory + InputFile
```

```

'A datatable is declared to retrieve the data from the CSV file
Dim RatingCurvesTable As DataTable
RatingCurvesTable = New DataTable

'Auxiliary variable that counts number of column separators per
row
Dim SeparatorsPerRow As Integer
'Vector which counts number of column separators per row
Dim columnCount() As Integer
'Character which acts as column separator
Dim columnSeparator As String = ";"

'String vector which stores line data from array
Dim Lines As String() = IO.File.ReadAllLines(RatingCurveFileName)
'Number of lines from the file
Dim NumLines As Integer = Lines.Length
ReDim columnCount(NumLines)
Dim i As Integer = 0

For Each line In Lines
    For Each ch As Char In line
        If ch = columnSeparator Then
            SeparatorsPerRow = SeparatorsPerRow + 1
        End If
    Next
    columnCount(i) = SeparatorsPerRow
    SeparatorsPerRow = 0
    i = i + 1
Next

'Maximum number of columns is calculated
Dim MaxColumnCount As Integer = columnCount.Max

'The datacolumns are added to the datatable
Dim n As Integer = 1 'count the rating curve number of the
pair (Q-H)
For i = 0 To MaxColumnCount
    If i Mod 2 = 0 Then
        RatingCurvesTable.Columns.Add("Flow " + n.ToString +
vbCrLf + "Q (m3/s)")
    Else
        RatingCurvesTable.Columns.Add("River" + vbCrLf + "Level"
+ n.ToString + vbCrLf + "H (m)")
        n = n + 1
    End If
Next

'String that stores each file line
Dim TextLine As String = ""
'Auxiliary string vector which stores each line file for
different columns
Dim SplitLine() As String

'The fileline are read again and its values passed to the
correspondent row and column
If System.IO.File.Exists(RatingCurveFileName) = True Then

```



```

        Dim objReader As New
System.IO.StreamReader(RatingCurveFileName)
        Dim a As Integer = RatingCurvesTable.Columns.Count
        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, columnSeparator)
            RatingCurvesTable.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("Rating Curves File Does Not Exist")
    End If

    'The datatable is transfered to the datagridview
    RatingCurvesDatagrid.DataSource = RatingCurvesTable.DefaultView
    btCalculateDailyFlows.Enabled = True

End Sub

Private Sub btCalculateDailyFlows_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btCalculateDailyFlows.Click
    FrmRiverLevels.Show()
    Me.Hide()
End Sub

Private Sub btBack_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
    FrmHydromet.Show()
    Me.Hide()

End Sub

Private Sub btNext_Click_1(ByVal sender As System.Object, ByVal e As
System.EventArgs)

    FrmRiverLevels.Show()
    Me.Hide()

End Sub

Private Sub FrmRatingCurves_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
    btDefault.PerformClick()
End Sub

Private Sub btSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btSave.Click

    SaveFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    SaveFileDialog1.Title = "Save File"
    SaveFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    SaveFileDialog1.FileName = "Rating Curves_1"
    SaveFileDialog1.OverwritePrompt = True
    SaveFileDialog1.ShowDialog()

    Dim FILE_NAME As String = SaveFileDialog1.FileName
    Dim numCols As Integer = RatingCurvesDatagrid.ColumnCount
    Dim numRows As Integer = RatingCurvesDatagrid.RowCount - 1

```

```

Using tw As TextWriter = New StreamWriter(FILE_NAME)

    For indexRows As Integer = 0 To numRows - 1
        'print all column values for a row
        For indexCols As Integer = 0 To numCols - 1

tw.Write(RatingCurvesDatagrid.Rows(indexRows).Cells(indexCols).Value)
            If (indexCols <> numCols) Then
                tw.Write(";")
            End If
        Next
        tw.WriteLine()
    Next
End Using
End Sub

Private Sub btLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btLoad.Click
    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call LoadValues()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSave.PerformClick()
        'And now after Saving the data we Clear the Textboxes
        Call LoadValues()
    End If
End Sub

Private Sub LoadValues()

    '***** This subroutine reads the loaded rating curves file
and tranfers it to a datagridview *****

    'Properties are assign to the OpenFileDialog
    OpenFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    OpenFileDialog1.RestoreDirectory = True

    If (OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK) Then
        RatingCurveFileName = OpenFileDialog1.FileName
    End If

    'A datatable is declared to retrieve the data from the CSV file
Dim RatingCurvesTable As DataTable
RatingCurvesTable = New DataTable

    'Auxiliary variable that counts number of column separators per
row
Dim SeparatorsPerRow As Integer
'Vector which counts number of column separators per row

```

```

Dim columnCount() As Integer
'Character which acts as column separator
Dim columnSeparator As String = ";"

'String vector which stores line data from array
Dim Lines As String() = IO.File.ReadAllLines(RatingCurveFileName)
'Number of lines from the file
Dim NumLines As Integer = Lines.Length
ReDim columnCount(NumLines)
Dim i As Integer = 0

For Each line In Lines
    For Each ch As Char In line
        If ch = columnSeparator Then
            SeparatorsPerRow = SeparatorsPerRow + 1
        End If
    Next
    columnCount(i) = SeparatorsPerRow
    SeparatorsPerRow = 0
    i = i + 1
Next

'Maximum number of columns is calculated
Dim MaxColumnCount As Integer = columnCount.Max

'The datacolumns are added to the datatable
Dim n As Integer = 1 'count the rating curve number of the
pair (Q-H)
For i = 0 To MaxColumnCount
    If i Mod 2 = 0 Then
        RatingCurvesTable.Columns.Add("Flow " + n.ToString +
vbCrLf + "Q (m3/s)")
    Else
        RatingCurvesTable.Columns.Add("River" + vbCrLf + "Level"
+ n.ToString + vbCrLf + "H (m)")
        n = n + 1
    End If
Next

'String that stores each file line
Dim TextLine As String = ""
'Auxiliary string vector which stores each line file for
different columns
Dim SplitLine() As String

'The fileline are read again and its values passed to the
correspondent row and column
If System.IO.File.Exists(RatingCurveFileName) = True Then
    Dim objReader As New
System.IO.StreamReader(RatingCurveFileName)
    Dim a As Integer = RatingCurvesTable.Columns.Count
    Do While objReader.Peek() <> -1
        TextLine = objReader.ReadLine()
        SplitLine = Split(TextLine, columnSeparator)
        RatingCurvesTable.Rows.Add(SplitLine)
    Loop
Else

```

```

        MsgBox("Rating Curves File Does Not Exist")
    End If

    'The datatable is transfered to the datagridview
    RatingCurvesDatagrid.DataSource = RatingCurvesTable.DefaultView
    btCalculateDailyFlows.Enabled = True

End Sub

Private Sub btErase_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btErase.Click
    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call EraseVeg()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSave.PerformClick()
        'And now after Saving the data we Clear the Textboxes and we
dissable the calculation button
        btCalculateDailyFlows.Enabled = False
        Call EraseVeg()
    End If

End Sub

Private Sub EraseVeg()
    RatingCurvesDatagrid.DataSource = Nothing
    RatingCurvesDatagrid.Refresh()
End Sub

End Class

```

## 6. Niveles de Río Diarios: "FrmRiverLevels.vb"

River Levels for each Transect

Date	River Level(m)_1	River Level(m)_2	River Level(m)_3	River Level(m)_4	River Level(m)_5	River Level(m)_6	River Level(m)_7	River Level(m)_8	River Level(m)_9	River Level(m)_10
01/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
02/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
03/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
04/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
05/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
06/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
07/08/1948	847.869	847.935	847.985	848.108	848.616	849.155	849.522	850.062	850.235	850.4
08/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
09/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
10/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
11/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
12/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
13/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
14/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
15/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
16/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
17/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
18/08/1948	847.905	847.983	848.033	848.162	848.646	849.203	849.564	850.104	850.283	850.5
19/08/1948	848.32	848.461	848.541	848.641	848.948	849.6	849.99	850.508	850.751	851.0
20/08/1948	847.948	848.034	848.088	848.218	848.679	849.245	849.611	850.148	850.334	850.5

Buttons: Load, Save, Back, Next

```
Imports System.IO
```

```
Public Class FrmRiverLevels
```

```
    Public RiverLevelsFileName As String
```

```
    Private Sub btNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btNext.Click
        frmSimPoints.Show()
        Me.Hide()
    End Sub
```

```
    Private Sub FrmRiverLevels_Load(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles MyBase.Load
        Call Calculate()
    End Sub
```

```
Private Sub Calculate()
    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim rowcount_curve As Integer
    Dim column_count As Integer

    Dim WTEday(,) As Single
```

```
'we count the number of rows for each of the datagrids
rowcount_daily = FrmHydromet.DailyInputsTable.RowCount
rowcount_curve = FrmRatingCurves.RatingCurvesDatagrid.RowCount '-
```

1

```

        column_count =
Int((FrmRatingCurves.RatingCurvesDatagrid.ColumnCount - 1) / 2)

    'We give the dimensions to the array
    ReDim Qday(rowcount_daily)
    ReDim Qcurve(rowcount_curve, column_count)
    ReDim Hcurve(rowcount_curve, column_count)
    ReDim WTEday(rowcount_daily, column_count)

    'We load the data from the DailyInputsTable datagrids to the
Qdaily array
    For i = 0 To rowcount_daily - 2
        Qday(i) =
FrmHydromet.DailyInputsTable.Rows.Item(i).Cells.Item("Qdaily").Value()
    Next i

    'We load the data from the Rating Curve datagrids to the Qcurve
and Hcurve Single arrays for each transect
    For j = 0 To rowcount_curve - 2
        For k = 0 To column_count
            Qcurve(j, k) =
CDBl(CheckDBNull(FrmRatingCurves.RatingCurvesDatagrid.Rows.Item(j).Cells.
Item(2 * k).Value())) ', 0))
            Hcurve(j, k) =
CDBl(CheckDBNull(FrmRatingCurves.RatingCurvesDatagrid.Rows.Item(j).Cells.
Item(2 * k + 1).Value())) ', 0))
        Next k
    Next j

    ' Now we use the "Forecast" function to do the linear
interpolation to calculate the daily water level Hdaily
    For k = 0 To column_count
        For i = 0 To rowcount_daily - 2
            For j = 0 To rowcount_curve - 2
                If Math.Abs(Qday(i) - Qcurve(j, k)) < 0.0000001 Then
                    WTEday(i, k) = Hcurve(j, k)
                    GoTo break
                Else
                    If Qday(i) < Qcurve(j, k) Then
                        WTEday(i, k) = Hcurve(j - 1, k) + (Qday(i) -
Qcurve(j - 1, k)) / (Qcurve(j, k) - Qcurve(j - 1, k)) * (Hcurve(j, k) -
Hcurve(j - 1, k))
                    End If
                End If
            Next j
        Next i
    Next k

break:
    Next i
Next k

    tableRiverLevels.ColumnCount = column_count + 2 'Two extra
columns are added one for the dates and the other to compensate 0-based
array
    tableRiverLevels.Rows.Add(rowcount_daily - 1)

    For i = 0 To rowcount_daily - 2
        tableRiverLevels.Rows.Item(i).Cells.Item(0).Value() =
FrmHydromet.DailyInputsTable.Rows.Item(i).Cells.Item("Time").Value()
    Next i

```

```

        tableRiverLevels.Columns(0).HeaderText = "Date"
        For k = 0 To column_count - 1
            For i = 0 To rowcount_daily - 2
                tableRiverLevels.Rows.Item(i).Cells.Item(k + 1).Value() =
Math.Round(WTEday(i, k), 3)
            Next i
            tableRiverLevels.Columns(k + 1).HeaderText = "River
Level(m)_" + Str(k + 1)
        Next k

```

```

End Sub

```

```

Private Sub btSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btSave.Click

```

```

        SaveFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
        SaveFileDialog1.Title = "Save File"
        SaveFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
        SaveFileDialog1.FileName = "RiverLevels_1"
        SaveFileDialog1.OverwritePrompt = True
        SaveFileDialog1.ShowDialog()

```

```

        Dim FILE_NAME As String = SaveFileDialog1.FileName
        Dim numCols As Integer = tableRiverLevels.ColumnCount
        Dim numRows As Integer = tableRiverLevels.RowCount - 1

```

```

        Using tw As TextWriter = New StreamWriter(FILE_NAME)
            'writing the data
            For indexRows As Integer = 0 To numRows - 1
                'print all column values for a row
                For indexCols As Integer = 0 To numCols - 1
                    tw.Write(tableRiverLevels.Rows(indexRows).Cells(indexCols).Value)
                    If (indexCols <> numCols) Then
                        tw.Write(";")
                    End If
                Next
                tw.WriteLine()
            Next
        End Using
    End Sub

```

```

Private Sub btLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btLoad.Click
    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call LoadValues()
    End If

```

```

ElseIf reply = DialogResult.Yes Then
    'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
    btSave.PerformClick()
    'And now after Saving the data we Clear the Textboxes
    Call LoadValues()
End If
End Sub

Private Sub LoadValues()

    Dim RiverLevelsFileName As String = ""
    OpenFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    OpenFileDialog1.RestoreDirectory = True

    If (OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK) Then
        RiverLevelsFileName = OpenFileDialog1.FileName
    End If

    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(RiverLevelsFileName) = True Then
        Dim objReader As New
System.IO.StreamReader(RiverLevelsFileName)
        If tableRiverLevels.RowCount <> 0 Then
            tableRiverLevels.Rows.Clear()
        End If

        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            tableRiverLevels.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("File Does Not Exist")
    End If

End Sub

Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
    FrmRatingCurves.Show()
    Me.Hide()
End Sub

Enum enumObjectType
    StrType = 0
    IntType = 1
    DblType = 2
End Enum

Public Function CheckDBNull(ByVal value As Object)

    Dim DblReturn As Single
    If value Is "" Or IsDBNull(value) Then
        value = 0
        DblReturn = value
    Else

```



```
        DblReturn = (value)
    End If
    Return value

End Function

End Class
```

## 7. Datos de los Puntos de Simulación: "frmSimPoints.vb"

Simulation Point	Elevation (m)	Soil Type	Transect	Horizontal Distance (m)	Observed Veg. Code	Observed Veg. Acronym	Observed Vegetation Description
1_1_1	854.703	3	1	6.438893547	1	TV	1
1_2_2	851.1595	7	1	17.46003639	2	TV	1
1_3_3	849.86	10	1	24.61017362	3	RA+TV	1
1_4_4	849.4955	10	1	30.09970945	4	RA	1
1_5_5	848.598	10	1	35.87049408	5	RA	1
1_6_6	849.056	10	1	47.10789847	6	RA	1
2_1_1	857.741	3	2	3.078613512	1	TV	1
2_2_2	853.7695	3	2	12.42391059	2	TV	1
2_3_3	851.826	7	2	26.05797699	3	RA	1
2_4_4	850.8575	10	2	29.34843959	4	RA	1
2_5_5	849.6935	6	2	35.02121855	5	RA	1
2_6_6	848.8955	6	2	40.50386216	6	RA	1
2_7_7	849.223	7	2	51.91153358	7	RA	1
2_8_8	850.2685	9	2	55.24044612	8	RA+TV	1
2_9_9	851.9365	9	2	58.40410272	9	TV	1
3_1_1	855.266	3	3	7.629771279	1	TV	1
3_2_2	851.329	7	3	25.63550029	2	TV	1
3_3_3	850.924	5	3	28.53672982	3	RA+TV	1
3_4_4	850.7855	5	3	33.04266696	4	RA	1
3_5_5	849.953	5	3	37.26780455	5	RA	1
3_6_6	849.403	7	3	46.09073395	6	RA	1
3_7_7	852.077	9	3	55.2508404	7	RA	1
4_1_1	854.6455	3	4	12.30845022	1	TV	1

```
Imports System.IO
```

```
Public Class frmSimPoints
```

```
    Public PatchesFileName As String
```

```
    Private Sub DefaultSimulationPoints()
```

```
        PatchesFileName = (System.Environment.CurrentDirectory +  
        "\data\Default\DefaultSimulationPoints.csv")
```

```
        Dim TextLine As String = ""
```

```
        Dim SplitLine() As String
```

```
        If System.IO.File.Exists(PatchesFileName) = True Then
```

```
            Dim objReader As New System.IO.StreamReader(PatchesFileName)
```

```
            If PatchTable.RowCount <> 0 Then
```

```
                PatchTable.Rows.Clear()
```

```
            End If
```

```
            Do While objReader.Peek() <> -1
```

```
                TextLine = objReader.ReadLine()
```

```
                SplitLine = Split(TextLine, ";")
```

```
                PatchTable.Rows.Add(SplitLine)
```

```
            Loop
```

```
        Else
```

```
            MessageBox.Show("Default Patches File has been Removed or  
Corrupted", "Removed File", MessageBoxButtons.OK, MessageBoxIcon.Warning)
```

```
        End If
```

```
    End Sub
```

```

Private Sub btNext_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btNext.Click

    Dim i As Integer
    i = PatchTable.CurrentRow.Index
    FrmSimulations.Show()
    Me.Hide()

End Sub

Private Sub btErase_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btErase.Click

    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call EraseVeg()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSave.PerformClick()
        'And now after Saving the data we Clear the Textboxes
        Call EraseVeg()
    End If

End Sub

Private Sub EraseVeg()

    Dim rowcount As Integer
    rowcount = PatchTable.RowCount
    For i = 0 To rowcount - 2
        PatchTable.Rows.Clear()
    Next i

End Sub

Private Sub LoadValues()
    OpenFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    OpenFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    OpenFileDialog1.RestoreDirectory = True

    If (OpenFileDialog1.ShowDialog() =
System.Windows.Forms.DialogResult.OK) Then
        PatchesFileName = OpenFileDialog1.FileName
    End If

    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(PatchesFileName) = True Then
        Dim objReader As New System.IO.StreamReader(PatchesFileName)
        If PatchTable.RowCount <> 0 Then
            PatchTable.Rows.Clear()

```

```

        End If

        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            PatchTable.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("File Does Not Exist")
    End If

End Sub

Private Sub btLoad_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btLoad.Click
    Dim reply As DialogResult = MessageBox.Show("Do you want to save
the current data?", "", _
                                                MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
    If reply = DialogResult.Cancel Then
        'Nothing happens
    ElseIf reply = DialogResult.No Then
        'We clear the Textboxes without Saving
        Call LoadValues()
    ElseIf reply = DialogResult.Yes Then
        'We go to the SAVE DIALOG to SAVE the Data from the Textboxes
        btSave.PerformClick()
        'And now after Saving the data we Clear the Textboxes
        Call LoadValues()
    End If

End Sub

Private Sub btSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btSave.Click
    Dim rowcount As Integer
    rowcount = PatchTable.RowCount
    Dim i As Integer
    Dim j As Integer
    Dim record(rowcount - 2) As Object

    SaveFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + "\data\")
    SaveFileDialog1.Title = "Save File"
    SaveFileDialog1.Filter = "CSV Files (*.csv)|*.csv|All Files
(*.*)|*.*"
    SaveFileDialog1.FileName = "PatchData_1"
    SaveFileDialog1.OverwritePrompt = True
    SaveFileDialog1.ShowDialog()

    Dim FILE_NAME As String = SaveFileDialog1.FileName
    Dim objWriter As New System.IO.StreamWriter(FILE_NAME)

    j = PatchTable.CurrentRow.Index

    For i = 0 To rowcount - 2
        record(i) = CStr(PatchTable.Rows.Item(i).Cells.Item(0).Value)
    & _

```

```

        "; " & CStr(PatchTable.Rows.Item(i).Cells.Item(1).Value) & _
        "; " & CStr(PatchTable.Rows.Item(i).Cells.Item(2).Value) & _
        "; " & CStr(PatchTable.Rows.Item(i).Cells.Item(3).Value) & _
-
        "; " & CStr(PatchTable.Rows.Item(i).Cells.Item(4).Value)

objWriter.WriteLine(record(i))

Next i
objWriter.Close()

End Sub

Private Sub btDefaultPatch_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btDefaultPatch.Click

    If PatchTable.RowCount <= 1 Then
        'There are no rows with values
        Call DefaultSimulationPoints()
    Else
        'A dialogbox appears in case the user wants to save the
previous data
        Dim reply As DialogResult = MessageBox.Show("Do you want to
save the current data?", "", _
            MessageBoxButtons.YesNoCancel,
MessageBoxIcon.Question, MessageBoxDefaultButton.Button1)
        If reply = DialogResult.Cancel Then
            'Nothing happens
        ElseIf reply = DialogResult.No Then
            'We load the default values without Saving
            Call DefaultSimulationPoints()
        ElseIf reply = DialogResult.Yes Then
            'We go to the SAVE DIALOG to SAVE the Data from the
Textboxes
            btSave.PerformClick()
            'And now after Saving the data we load the default values
            Call DefaultSimulationPoints()
        End If
    End If

End Sub

Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
    FrmRiverLevels.Show()
    Me.Hide()
End Sub

Private Sub PatchTable_CellContentClick(ByVal sender As
System.Object, ByVal e As System.Windows.Forms.DataGridViewCellEventArgs)
Handles PatchTable.CellContentClick

End Sub

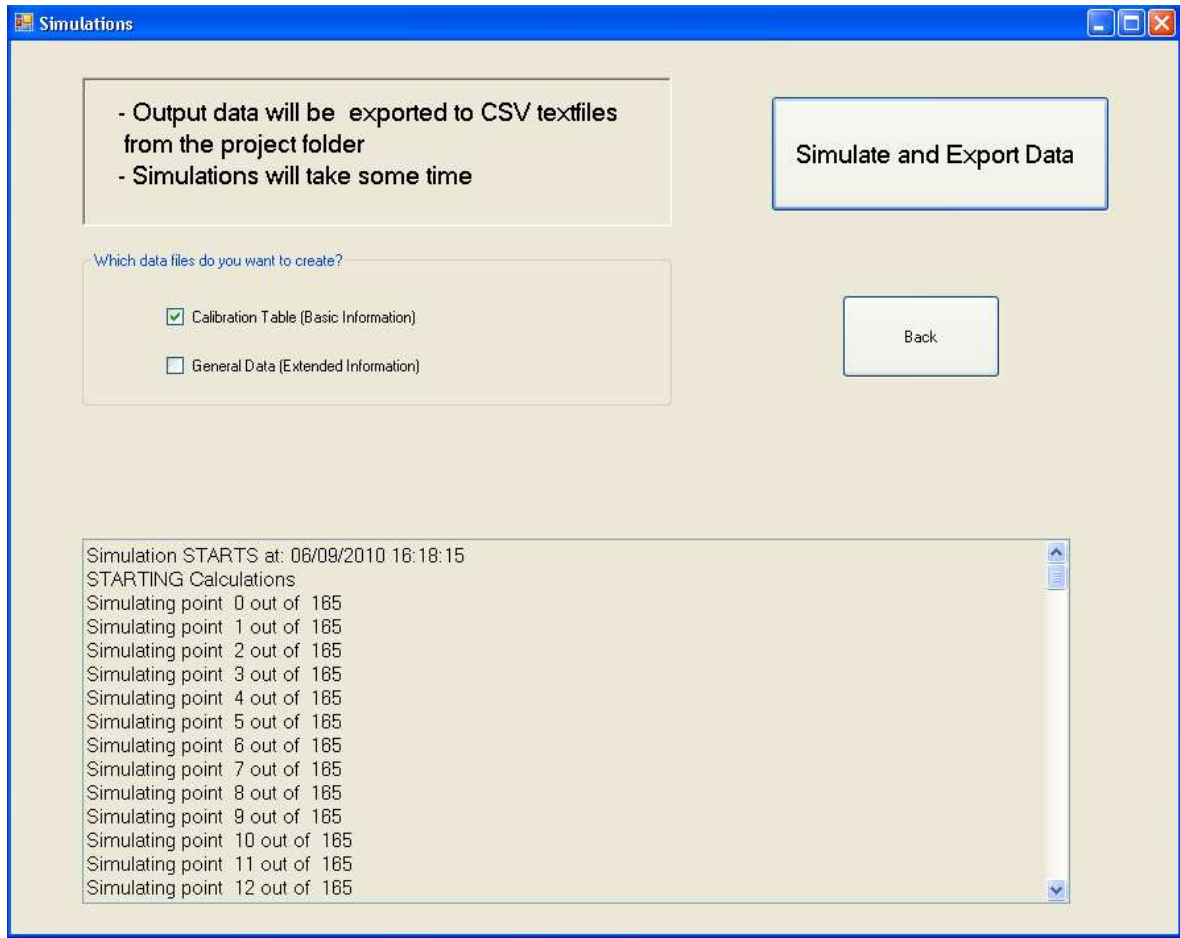
Private Sub frmPatches_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

```

```
    Call DefaultSimulationPoints()  
End Sub
```

```
End Class
```

## 8. Simulación y Resultados: "frmSimPoints.vb"



```
Imports System.Math
Imports System.IO
```

```
Public Class FrmSimulations
```

```
    'Generate a random number between 1001 and 10000
    Dim Random As New Random()
    Dim rn As Integer
    Dim rand As Integer

    Private Sub btCSVSimulations_click(ByVal sender As System.Object,
    ByVal e As System.EventArgs) Handles btCSVSimulations.Click

        'The simulation process starts
        Dim StartSim As DateTime = DateTime.Now 'time when simulation
starts
        txtSimulation.AppendText("Simulation STARTS at: " +
CStr(startSim)) : Cr()

        'First the calculations are carried out and stored in the array
memory
        txtSimulation.AppendText("STARTING Calculations") : Cr()
```

```

Call Module2.Simulate()
txtSimulation.AppendText("ENDING Calculations") : Cr()
txtSimulation.AppendText(" ") : Cr()
txtSimulation.AppendText("-----")
-----") : Cr()

'Now we copy the Input Data Files to the Input subdirectory which
is inside the Project directory
txtSimulation.AppendText("Retrieving Input Data") : Cr()
txtSimulation.AppendText(" ") : Cr()
txtSimulation.AppendText("-----")
-----") : Cr()
Directory.CreateDirectory(ProjAddress + "/InputData")
Call CopyInputFiles(FrmSoilParam.SoilFileName) 'we
call the CopyInputFiles subroutine for each of the input files
Call CopyInputFiles(FrmVegetParam.VegetFileName)
Call CopyInputFiles(FrmHydromet.DailyClimaticFileName)
Call CopyInputFiles(FrmRatingCurves.RatingCurveFileName)
Call CopyInputFiles(frmSimPoints.PatchesFileName)

'The user decides which type of output data to be written on csv
text files
If chkCalibrationTable.Checked = True Then
txtSimulation.AppendText("STARTING to Print General Output
Table ") : Cr()
Call BuildTable()
txtSimulation.AppendText("Printing of General Output Table
FINISHED ") : Cr()
txtSimulation.AppendText(" ") : Cr()
txtSimulation.AppendText("-----")
-----") : Cr()
End If

If chkGeneralData.Checked = True Then
txtSimulation.AppendText("STARTING to Print of Detailed Data
Files ") : Cr()
Call GeneralData()
txtSimulation.AppendText("Printing of Detailed Data Files
FINISHED ") : Cr()
txtSimulation.AppendText(" ") : Cr()
txtSimulation.AppendText("-----")
-----") : Cr()
End If

'We inform about the time it took to carry out the simulation
Dim EndSim As DateTime = DateTime.Now 'time when siluation ends
txtSimulation.AppendText("Simulation ENDS at: " + CStr(EndSim)) :
Cr()
Dim SimulationDuration As TimeSpan = EndSim - StartSim
txtSimulation.AppendText("Simulation ENDS at: " +
CStr(SimulationDuration.ToString)) : Cr()

'Lastly we open the containing folder
txtSimulation.AppendText("Output Folder Opens") : Cr()
Process.Start("explorer.exe", Module1.OutputAddress)

End Sub

```



```

Private Sub BuildTable()

    'This subroutine builds the ETindex Summary Table used for
    calibration
    Dim ObsVegTot(nptos - 1) As String      'Observed Vegetation
    Acronym (ex: "RH+RJ")
    Dim ColSeparator As String = ";"
    'Symbol which determines the separation between columns
    Dim nveg As Integer =
    FrmVegetParam.VegetationDataGridView.RowCount
    Dim FILE_NAME As String = (Module1.OutputAddress + "\Calibration
    Matrix " + frmMDIParent1.txtProyName.Text + ".csv")

    For i = 0 To nptos - 1
        ObsVegTot(i) =
    frmSimPoints.PatchTable.Rows.Item(i).Cells.Item("colObservedVegAcronym").
    Value()
    Next

    Using tw As TextWriter = New StreamWriter(FILE_NAME)
        'writing the header
        tw.WriteLine("*****")
        tw.WriteLine("CALIBRATION MATRIX " +
    frmMDIParent1.txtProyName.Text)
        tw.WriteLine("*****")
        tw.WriteLine()
        tw.WriteLine()
        tw.WriteLine("Point" + ColSeparator + "ObsVeg" + ColSeparator
    + "ETindex_1" + ColSeparator + "ETindex_2" + ColSeparator + "ETindex_3" +
    ColSeparator + "ETindex_4")

        With tw
            For q = 0 To nptos - 1
                .WriteLine((CStr(q + 1) + ColSeparator _
                    + ObsVegTot(q)) + ColSeparator _
                    + Str(ETindexAVGCol(q, 0)) + ColSeparator
    -
                    + Str(ETindexAVGCol(q, 1)) + ColSeparator _
                    + Str(ETindexAVGCol(q, 2)) + ColSeparator _
                    + Str(ETindexAVGCol(q, 3)) + ColSeparator)
            Next q
        End With
        tw.Close()
    End Using
End Sub

Private Sub GeneralData()

    Dim ColSeparator As String = ";"
    'Symbol which determines the separation between columns
    Dim GenSummary As String = (Module1.OutputAddress + "\Summary
    from " + frmMDIParent1.txtProyName.Text + ".csv")
    Dim PatchAddress As String
    Dim nveg As Integer =
    FrmVegetParam.VegetationDataGridView.RowCount

```

```

    Dim BlankSpacesHeader As String = ColSeparator + "          "
'Blank Spaces between column header in the output textfile
    Dim BlankSpacesValues As String = ColSeparator + " "
'Blank Spaces between column header in the output textfile

    Dim Writer1_GenSummary As New System.IO.StreamWriter(GenSummary)

    'Now we write a tittle
    With Writer1_GenSummary
        .WriteLine("*****")
        .WriteLine("SUMMARY FROM " + frmMDIParent1.txtProyName.Text)
        .WriteLine("*****")
        .WriteLine()
        .WriteLine()
    End With

    'For each patch we create a table
    For r = 0 To nptos - 1
        With Writer1_GenSummary
            .WriteLine("PATCH " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colPatch").Value)
            .WriteLine("VegType " + ColSeparator + "ETRtot_Index")

                'In each patch table we write the values of
                "ETRtot_Index" for each "Vegetation Type"
                For u = 0 To
FrmVegetParam.VegetationDataGridView.RowCount - 2

                .WriteLine(FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("
colAcronim").Value + (ColSeparator) + Str(ETindexAVGCol(r, u)))
                Next u

                'Lastly we write some characteristics from the Patch
                .WriteLine("Observed Vegetation: " + ColSeparator +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colObservedVegAcronym").
Value)

                .WriteLine("Patch Elevation " + ColSeparator +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colElevation").Value)
                .WriteLine("Soil Type " + ColSeparator +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value)
                .WriteLine("Transect " + ColSeparator +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colTransect").Value)
                .WriteLine()
            End With
        End With
    Next r

'*****
'*****
    'With the "Writer2_PatchSummary" Object we create a file that
summarizes the data from each Patch
    For r = 0 To nptos - 1
        rand = Random.Next(1001, 10000)

        txtSimulation.AppendText("Printing Files for Simulation
Point: " + CStr(r + 1) + " out of " + CStr(nptos)) : Cr()
        txtSimulation.AppendText(" ") : Cr()
    End For

```

```

PatchAddress = Module1.OutputAddress + "\Results from Patch_"
+ frmSimPoints.PatchTable.Rows.Item(r).Cells.Item(0).Value
Directory.CreateDirectory(PatchAddress)

Dim PatchSummary As String = (PatchAddress + "\Summary from
Patch_" + frmSimPoints.PatchTable.Rows.Item(r).Cells.Item(0).Value +
".csv")
Dim Writer2_PatchSummary As New
System.IO.StreamWriter(PatchSummary)

'Dim s As Integer =
frmPatches.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value - 1

Dim n As String
Dim j As Integer
n =
CStr(frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value
())

Transect =
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colTransect").Value()

For b = 0 To FrmSoilParam.SoilDataGridView.RowCount - 2
If n =
FrmSoilParam.SoilDataGridView.Rows.Item(b).Cells.Item("colSoilCode").Value
() Then
j = b
GoTo break
End If
Next
break:

With Writer2_PatchSummary
.WriteLine("*****")
.WriteLine("SUMMARY FROM PATCH " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colPatch").Value)
.WriteLine("*****")

'Exporting General Characteristics from the Patch
.WriteLine("Patch Elevation: " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colElevation").Value + "
meters")
.WriteLine("Soil Type: " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value)
.WriteLine("Pertaining Transect: " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colTransect").Value)
.WriteLine("Observed Vegetation: " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colObservedVegetationDes
cription").Value)

'Exporting Soil Parameters from the Patch
.WriteLine("")
.WriteLine("")
.WriteLine("-----")
.WriteLine("Soil Parameters " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value)
.WriteLine("-----")
.WriteLine("Soil Type: " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value)

```

```

        .WriteLine("Porosity: " +
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colPst").Value + "
[]")
        .WriteLine("Porosity Index: " +
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colIp").Value)
        .WriteLine("Bubble Pressure: " +
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colPb").Value +
"Kpa")
        .WriteLine("Saturated Conductivity: " +
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colKs").Value + "
mm/hr")
        .WriteLine("Field Capacity Moisture: " +
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colMfc").Value + "
[]")

'Evaporation Index Results and Vegetation Parameters for
all Vegetation Types: ETind=ETtot/ETP
        .WriteLine("")
        .WriteLine("")
        .WriteLine("-----")
-----")
        .WriteLine("Evaporation Index Results and Vegetation
Input Parameters")
        .WriteLine("-----")
-----")
        .WriteLine("Simulated Vegetation Type /VegType")
        .WriteLine("Simulated Evapotranspiration Index [ ]
ETindex")
        .WriteLine("Maximum Root Depth [m] Zr")
        .WriteLine("Effective Root Depth [m] Ze")
        .WriteLine("Saturation Extinction Depth [m] Zsat")
        .WriteLine("Transpiration Factor from Saturated Zone [ ]
Rj ")
        .WriteLine("Maximum Soil-Root Water Conductance [mmMpa-
1h-1] CRT")
        .WriteLine("Transpiration Factor from Unsaturated Zone [
] Ri")
        .WriteLine("Plant Cover Month 1 [ ] Cov1")
        .WriteLine("Plant Cover Month 2 [ ] Cov2")
        .WriteLine("Plant Cover Month 3 [ ] Cov3")
        .WriteLine("Plant Cover Month 4 [ ] Cov4")
        .WriteLine("Plant Cover Month 5 [ ] Cov5")
        .WriteLine("Plant Cover Month 6 [ ] Cov6")
        .WriteLine("Plant Cover Month 7 [ ] Cov7")
        .WriteLine("Plant Cover Month 8 [ ] Cov8")
        .WriteLine("Plant Cover Month 9 [ ] Cov9")
        .WriteLine("Plant Cover Month 10 [ ] Cov10")
        .WriteLine("Plant Cover Month 11 [ ] Cov11")
        .WriteLine("Plant Cover Month 12 [ ] Cov12")
        .WriteLine("")

        .WriteLine("VegType" + ColSeparator _
+ "ETind" + ColSeparator _
+ "Zr " + ColSeparator _
+ "Ze " + ColSeparator _
+ "Zsat " + ColSeparator _
+ "Ri " + ColSeparator _
+ "Rj " + ColSeparator _
+ "CRT " + ColSeparator _
+ "Description " + ColSeparator _

```

```

        + "Cov1 " + ColSeparator _
    + "Cov2 " + ColSeparator _
    + "Cov3 " + ColSeparator _
    + "Cov4 " + ColSeparator _
    + "Cov5 " + ColSeparator _
    + "Cov6 " + ColSeparator _
    + "Cov7 " + ColSeparator _
    + "Cov8 " + ColSeparator _
    + "Cov9 " + ColSeparator _
    + "Cov10 " + ColSeparator _
    + "Cov11 " + ColSeparator _
    + "Cov12 " + ColSeparator)

    'All the vegetation values are printed
    For u = 0 To
FrmVegetParam.VegetationDataGridView.RowCount - 2

.WriteLine(FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("
colVegCode").Value _
        + ColSeparator + " " + Str(Round(ETindexAVGCol(r,
u), 3)) _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colZr").Val
ue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colZe").Val
ue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colZsat").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colRi").Val
ue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colRj").Val
ue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCrt").Va
lue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colVegDescr
iption").Value _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov0").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov1").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov2").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov3").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov4").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov5").V
alue _

```

```

        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov6").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov7").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov8").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov9").V
alue _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov10").
Value _
        + ColSeparator + " " +
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov11").
Value)
    Next u

    'The Rating Curve from the transect that the the patch
pertains is printed
    .WriteLine("")
    .WriteLine("")
    .WriteLine("-----")
    .WriteLine("Rating Curve")
    .WriteLine("-----")
    .WriteLine("Flow [m3/s] Qcurve")
    .WriteLine("Water Level [m] Hcurve")
    .WriteLine("")
    'The headers are now printed
    .WriteLine("Qcurve " + ColSeparator + "Hcurve ")
    'The values from the rating curve are printed

    Dim f As Integer =
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colTransect").Value()
    For d = 0 To
FrmRatingCurves.RatingCurvesDatagrid.RowCount - 2
        .WriteLine(Str(Qcurve(d, f - 1)) + " " +
ColSeparator + Str(Hcurve(d, f - 1)))
    Next d
    End With 'We close the Writer2_PatchSummary Object that
write the summary for each Patch

'*****
*****
    'With the Writer3_DailyData we create a group of files for
each "Patch" that show the "Daily Calculations" for each "Vegetation
Type"
    For u = 0 To FrmVegetParam.VegetationDataGridView.RowCount -
2
        Dim DailyData As String = (PatchAddress + "\Vegetation_"
+ Str(u + 1) + " in Patch " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item(0).Value + ".csv")
        Dim Writer3_DailyData As New
System.IO.StreamWriter(DailyData)
        With Writer3_DailyData
            .WriteLine("*****")

```

```

        .WriteLine("RESULTS FROM PATCH " +
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item(0).Value + " OF
VEGETATION_" + Str(u + 1))
        .WriteLine("*****")

        .WriteLine("Date ")
        .WriteLine("Daily Precipitation [mm] PP")
        .WriteLine("Daily Potential Evapotranspiration [mm]
ETP ")

        .WriteLine("River Flow [m3/s] Qday")
        .WriteLine("Daily Plant Cover [] DailyCov")
        .WriteLine("Water Table Elevation (i) [m] WTE")
        .WriteLine("End of Day Moisture [mm] Hfin")
        .WriteLine("Matrix Potential [Kpa] Pmat")
        .WriteLine("Root Water Uptake [mm] Rwu")
        .WriteLine("Hydraulic Conductivity [mm/hr] Kh")
        .WriteLine("Capilarity Water Uptake [mm] Cwu")
        .WriteLine("Excess Water [mm] Exc")
        .WriteLine("Water Income to the Soil [mm] Win")
        .WriteLine("Initial Moisture [mm] Hini")
        .WriteLine("Daily Relative Moisture [ ]Hrel ")
        .WriteLine("Actual Unsaturated Zone
Evapotranspiration [mm] ETRuns")
        .WriteLine("Actual Saturated Zone Evapotranspiration
[mm] ETRsat")

        .WriteLine("Actual Total Evapotranspiration [mm]
ETRtot")

        .WriteLine("Evapotranspiration Index [mm] ETindex")
        .WriteLine("")

        .WriteLine("Date " _
+ BlankSpacesHeader + "PP" _
+ BlankSpacesHeader + "ETP" _
+ BlankSpacesHeader + "Qday" _
+ BlankSpacesHeader + "DailyCov" _
+ BlankSpacesHeader + "WTE" _
+ BlankSpacesHeader + "Hfin" _
+ BlankSpacesHeader + "Pmat" _
+ BlankSpacesHeader + "Rwu" _
+ BlankSpacesHeader + "Kh" _
+ BlankSpacesHeader + "Cwu" _
+ BlankSpacesHeader + "Exc" _
+ BlankSpacesHeader + "Win" _
+ BlankSpacesHeader + "Hini" _
+ BlankSpacesHeader + "Hrel" _
+ BlankSpacesHeader + "ETRuns" _
+ BlankSpacesHeader + "ETRsat" _
+ BlankSpacesHeader + "ETRtot" _
+ BlankSpacesHeader + "ETindex")

For v = 0 To FrmHydromet.DailyInputsTable.RowCount -
2
        .WriteLine((TimeD(v)) + ColSeparator _
+ Str(Round(PPd(v), 4)) + ColSeparator
_
+ Str(Round(ETPd(v), 4)) +
ColSeparator _
+ Str(Round(Qday(v), 4)) + ColSeparator _
+ Str(Round(Cov(u, MonthNumber(v) - 1), 4))
+ ColSeparator _

```

```

+ Str(Round(Hdaily(Transect, v), 4)) +
ColSeparator _
+ Str(Round(HsoilCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(PmatCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(RwuCol(r, u, v), 4)) +
ColSeparator _
+ Str(KhCol(r, u, v)) + ColSeparator _
+ Str(Round(CwuCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(ExcCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(WinCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(HiniCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(HrelCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(ETRunsCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(ETRsatsCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(ETRtotCol(r, u, v), 4)) +
ColSeparator _
+ Str(Round(ETindexDAYCol(r, u, v),
4)))
Next v
End With
Writer3_DailyData.Close()

Next u
Writer2_PatchSummary.Close()
Next r
Writer1_GenSummary.Close()

'Lastly we open the containing folder
Process.Start("explorer.exe", Module1.OutputAddress)

End Sub

Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
frmSimPoints.Show()
End Sub

End Class

```



## 9. Modulo 1: "Module1"

```
Imports System.IO
```

```
Module Module1
```

```
    Friend ProjAddress As String    'Name of the new project's directory
    Friend InputAddress As String   'Name of the Input Data Subdirectory,
    (inside the project directory)
    Friend OutputAddress As String  'Name of the Output Data Subdirectory
    (inside the project directory)
```

```
    Friend rowcount_daily As Integer
    Friend rowcount As Integer
```

```
    'CLIMATOLOGICAL DATA AND RIVER LEVEL FORECAST CALCULATIONS
```

```
    Friend Qday() As Single         'Daily Flow [m3/s]
    Friend Qcurve(,) As Single     'Rating Curve Flow [m3/s]
    Friend Hcurve(,) As Single     'Rating Curve Water Table Elevation (m)

    Friend PPd() As Single         'Precipitation (mm/day)
    Friend ETPd() As Single        'Potential Evapotranspiration (mm/day)
    Friend TimeD() As String       'Date (dd/mm/yyyy)
```

```
    'Parameters from the soil type that the user has selected in the
    "lbTextureType" listbox and that will be used inthe Daily State Variable
    Form
```

```
    Friend Cov(,) As Single        'Plant Cover Cv_yearly
    Friend Texture As String       'Texture Class
    Friend Pst() As Single         'Φ Porosity [ ]
    Friend Hres As Single          'θr Residual Water Content [ ]
    Friend Pb() As Single          'Bubbling capillary pressure (mm)
    Friend Ip() As Single          'λ=Pore -size index
    Friend Mfc() As Single         'Field Capacity Moisture [ ]
    Friend Ks() As Single          'Fully Saturated conductivity (mm/hr)
    Friend Zmin() As Single        'Minimum Capillary Depth to be
    Considered [m]
```

```
    'VEGETATION DATA(form2)
```

```
    Friend Zr() As Single          'Root Depth Zr max [m]
    Friend Ze() As Single          'Effective Root Depth Ze [m]
    Friend Zsat() As Single        'Extinction Saturation Psat [m]
    Friend Ri() As Single          'Transpiration Factor from Unsaturated
    Zone [ ]
    Friend Rj() As Single          'Transpiration Factor from Saturated
    Zone [ ]
```

```
    Friend Crt() As Single         'Maximum Soil-Root Water Conductance
    CRT [mmMpa-lh-1]
    Friend Pwp() As Single         'Moisture at Wilting Point [ ]
    Friend Pcrit() As Single       'Critical Moisture [ ]
    Friend Veg As String           'River Vegetation Type [1 to 5]
```

```

'GENERAL CALCULATIONS(form4)
Friend Mcrit() As Single      'θCrit Moisture at Critical Point [ ]
Friend Mwp() As Single       'θwp Moisture at Wilting Point [ ]
Friend CZr() As Single       'Root Depth Elevation (m)
Friend CZe() As Single       'Effective Root Depth Elevation (m)
Friend CPsat() As Single     'Extinction Saturation Elevation (m)
Friend Hfc() As Single       'Field Capacity Depth (mm)
Friend Hwp() As Single       'Permanent Wilting Point Depth (mm)
Friend Hcrit() As Single     'Critical Moisture Depth (mm)
Friend Pm50() As Single      'Medium Saturation Moisture Pressure
[Kpa]
Friend Pfc() As Single       'Field Capacity Pressure [Kpa]
Friend CZmin() As Single     'Elevation to be considered in CWU [m]

Friend Sub CopyInputFiles(ByVal FilePath As String)

    'this sub permits copying the input files to the project's folder
    Directory.CreateDirectory(ProjAddress + "/InputData")
    Dim NewCopy As String
    ' the next instruccion extracts the file name from its path
    Dim Filename As String = IO.Path.GetFileName(FilePath)
    NewCopy = ProjAddress + "InputData\" + Filename
    If System.IO.File.Exists(FilePath) = True Then
        System.IO.File.Copy(FilePath, NewCopy)
    End If

End Sub

Friend Sub Automatic()
    'this sub permits automatic simulations
    FrmSoilParam.btNext.PerformClick()
    FrmVegetParam.btNext.PerformClick()
    FrmHydromet.btNext.PerformClick()
    FrmRatingCurves.btCalculateDailyFlows.PerformClick()
    FrmRiverLevels.btNext.PerformClick()
    frmSimPoints.btNext.PerformClick()
    FrmSimulations.Show()
    FrmSoilParam.Hide()
End Sub

Friend Sub Cr()
    FrmSimulations.txtSimulation.Text =
FrmSimulations.txtSimulation.Text + Environment.NewLine
End Sub

End Module

```

## 10. Modulo 2: "Module2"

```
Imports System.Math
Imports System.IO

Module Module2

    Friend Es() As Single           'Patch Elevation (m above sea
level)
    Friend ClimaticCount As Integer
    Friend FlowCount As Integer

    Friend MonthNumber() As Integer 'stores the month number for each
daily interval

    Friend nptos As Integer
    Friend nveg As Integer

    Friend Transect As Integer
    Friend Hsoil() As Single       'Daily Soil Moisture Depth (mm)
    Dim Psoil() As Single          'Daily Soil Moisture Relation [ ]

    Friend Pmat() As Single        'Matrix Potential [mm]
    Friend Rwu() As Single         'Root Water Uptake Ahr [mm]
    Friend Kh() As Single          'Hydraulic Conductivity [mm/hr]
    Friend Cwu() As Single         'Capilarity Water Uptake [mm]
    Friend Exc() As Single         'Excess of Water [mm]
    Friend Win() As Single         'Water Income [mm]
    Friend Hini() As Single        'Initial Moisture [mm]
    Friend Hrel() As Single        'Daily Relative Moisture [ ]
    Friend ETRnul() As Single      'Casos ETR=0 (simplificacion de
calculos)
    Friend ETRuns() As Single      'Real Evapotranspiration of the
Unsaturated Zone [mm]
    Friend ETRsat() As Single      'Real Evapotranspiration of the
Saturated Zone [mm]
    Friend ETRtot() As Single      'Real Total Evapotranspiration [mm]
    Friend Bal() As Single         'Water Balance [mm]
    Friend ETindexDay() As Single  'Daily ETindex

    Public Es_ini As Single        'Initial Elevation (m)
    Public delta_cota As Single    'Delta Elevation (m)
    Public ncotas As Integer       'Number of Elevations (m)

    'Single Array(Transect,Time)
    Friend Hdaily(,) As Single     'Daily Water Table Elevation per each
Transect(m)

    'Triple Arrays (Patch,Time, Vegetation)
    Friend HfcCol(,,) As Single
    Friend HdailyCol(,,) As Single
    Friend HsoilCol(,,) As Single
    Friend PmatCol(,,) As Single
    Friend RwuCol(,,) As Single
    Friend KhCol(,,) As Single
```

```

Friend CwuCol(,,) As Single
Friend ExcCol(,,) As Single
Friend WinCol(,,) As Single
Friend HiniCol(,,) As Single
Friend HrelCol(,,) As Single
Friend ETRsatCol(,,) As Single
Friend ETRunsCol(,,) As Single
Friend ETRtotCol(,,) As Single
Friend BalCol(,,) As Single
Friend ETindexDAYCol(,,) As Single

Friend ETindexAVGCol(,) As Single
Friend ArrayOut(,) As Single ' ETindexCol array sorted with keys by
minimum to maximum vegetation type ETindex values
'number of dimensions of the square calibration array (includes
vegetation functional types "Single mixtures")

Friend Sub Simulate()

    rowcount = FrmHydromet.DailyInputsTable.RowCount

    Dim m As Integer 'counter which returns month number (0 to 11)
of each year

    Dim SumETRtot As Single
    Dim sumETPd As Single
    Dim ETindexTot As Single

    ClimaticCount = FrmHydromet.DailyInputsTable.RowCount
    nptos = frmSimPoints.PatchTable.RowCount - 1 'Number of
simulation points to be calculated
    nveg = FrmVegetParam.VegetationDataGridView.RowCount - 2

    Dim numTrans As Integer 'Number of Transects in the stretch
FlowCount = FrmRiverLevels.tableRiverLevels.RowCount - 2
    numTrans = FrmRiverLevels.tableRiverLevels.ColumnCount - 1
    ReDim Preserve Hdaily(numTrans, ClimaticCount)
    For d = 1 To numTrans
        For v = 0 To FlowCount
            Hdaily(d, v) =
FrmRiverLevels.tableRiverLevels.Rows.Item(v).Cells.Item(d).Value()
        Next v
    Next d

    ReDim Cov(nveg, 11) 'Plant Cover Cv_yearly
    ReDim Preserve Es(nptos)

    ReDim Preserve Hsoil(ClimaticCount) 'Daily Soil Moisture
Depth (mm)
    ReDim Preserve Psoil(ClimaticCount) 'Daily Soil Moisture
Relation (cm3/cm3)
    ReDim Preserve Pmat(ClimaticCount) 'Matrix Potential [mm]

    ReDim Preserve Rwu(ClimaticCount) 'Root Water Uptake Ahr
[mm]
    ReDim Preserve Kh(ClimaticCount) 'Hydraulic
Conductivity [mm/hr]
    ReDim Preserve Cwu(ClimaticCount) 'Capilarity Water
Uptake [mm]

```

	ReDim Preserve	Exc(ClimaticCount)	'Excess of Water [mm]
	ReDim Preserve	Win(ClimaticCount)	'Water Income [mm]
	ReDim Preserve	Hini(ClimaticCount)	'Initial Moisture [mm]
	ReDim Preserve	Hrel(ClimaticCount)	'Daily Relative
Moisture (cm3/cm3)			
	ReDim Preserve	ETRnul(ClimaticCount)	'Casos ETR=0
(simplificacion de calculos)			
	ReDim Preserve	ETRuns(ClimaticCount)	'Real
Evapotranspiration of the Unsaturated Zone [mm]			
	ReDim Preserve	ETRsat(ClimaticCount)	'Real
Evapotranspiration of the Saturated Zone [mm]			
	ReDim Preserve	ETRtot(ClimaticCount)	'Real Total
Evapotranspiration [mm]			
	ReDim Preserve	Bal(ClimaticCount)	'Water Balance [mm]
	ReDim Preserve	ETindexDay(ClimaticCount)	'Daily ETindex
	ReDim	Zr(nveg)	'Root Depth Zr max [m]
	ReDim	Ze(nveg)	'Effective Root Depth Ze [m]
	ReDim	Zsat(nveg)	'Extinction Saturation Psat [m]
	ReDim	Ri(nveg)	'Efficiency of Static Water Purchase
(Fest)			
	ReDim	Rj(nveg)	'Fraction of Active Roots below Ze (Rj)
	ReDim	Crt(nveg)	'Maximum Soil-Root Water Conductance CRT
[mmMpa-1h-1]			
	ReDim	Pwp(nveg)	'Wilting Point Pressure [Kpa]
	ReDim	Pcrit(nveg)	'Critical Point Pressure [Kpa]
	ReDim	Pst(nptos)	
	ReDim	Ip(nptos)	
	ReDim	Pb(nptos)	
	ReDim	Mfc(nptos)	
	ReDim	Ks(nptos)	
	ReDim	Zmin(nptos)	
	ReDim	Hfc(nptos)	'Field Capacity Depth (mm)
	ReDim	Mcrit(nveg)	
	ReDim	Mwp(nveg)	
	ReDim	Hwp(nveg)	'Permanent Wilting Equivalent water (mm)
	ReDim	Hcrit(nveg)	'Critical Moisture Equivalent water (mm)
	ReDim	Pm50(nveg)	'Medium Saturation Moisture Pressure
[Kpa]			
	ReDim	Pfc(nveg)	'Field Capacity Pressure [Kpa]
	ReDim	CZr(Es.Length - 1)	
	ReDim	CZe(Es.Length - 1)	
	ReDim	CPsat(Es.Length - 1)	
	ReDim	CZmin(Es.Length - 1)	
	ReDim	HfcCol(nptos, nveg, ClimaticCount)	
	ReDim	HdailyCol(nptos, nveg, ClimaticCount)	
	ReDim	HsoilCol(nptos, nveg, ClimaticCount)	
	ReDim	PmatCol(nptos, nveg, ClimaticCount)	
	ReDim	RwuCol(nptos, nveg, ClimaticCount)	
	ReDim	KhCol(nptos, nveg, ClimaticCount)	
	ReDim	CwuCol(nptos, nveg, ClimaticCount)	

```

ReDim ExcCol(nptos, nveg, ClimaticCount)
ReDim WinCol(nptos, nveg, ClimaticCount)
ReDim HiniCol(nptos, nveg, ClimaticCount)
ReDim HrelCol(nptos, nveg, ClimaticCount)
ReDim ETRunsCol(nptos, nveg, ClimaticCount)
ReDim ETRsatCol(nptos, nveg, ClimaticCount)
ReDim ETRtotCol(nptos, nveg, ClimaticCount)
ReDim BalCol(nptos, nveg, ClimaticCount)
ReDim ETindexDAYCol(nptos, nveg, ClimaticCount)

ReDim ETindexAVGCol(nptos, nveg)

For r = 0 To nptos '- 1
    FrmSimulations.txtSimulation.AppendText("Simulating point " +
Str(r) + " out of " + Str(nptos)) : Cr()
    Es(r) =
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colElevation").Value()
    Dim n As String
    Dim j As Integer
    n =
CStr(frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colSoilType").Value
())

    Transect =
frmSimPoints.PatchTable.Rows.Item(r).Cells.Item("colTransect").Value()

    For b = 0 To FrmSoilParam.SoilDataGridView.RowCount - 2
        If n =
FrmSoilParam.SoilDataGridView.Rows.Item(b).Cells.Item("colSoilCode").Valu
e() Then
            j = b
            GoTo break
        End If
    Next
break:

    Pst(r) =
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colPst").Value()
'Max. Root Depth Zr [m]
    Ip(r) =
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colIp").Value()
'Effective Root Depth Ze [m]
    Pb(r) =
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colPb").Value()
'Extinction Saturation Psat [m]
    Ks(r) =
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colKs").Value()
'Moisture at Wilting Point [ ]
    Mfc(r) =
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colMfc").Value()
'Fraction of Active Roots below Effective Root Depth(Rj) [mm/mm]
    Zmin(r) =
FrmSoilParam.SoilDataGridView.Rows.Item(j).Cells.Item("colZmin").Value()

For u = 0 To nveg '- 1
    'We put the following variables=0 so the sum is done
correctly

```

```

SumETRtot = 0
sumETPd = 0
ETindexTot = 0

Zr(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colZr").Value()
Ze(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colZe").Value()
Zsat(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colZsat").Value()
Ri(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colRi").Value()
Rj(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colRj").Value()
Crt(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCrt").Value()
Pwp(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colPwp").Value()
Pcrit(u) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colPcrit").Value()

'Determin m for each day inside v loop
For m = 0 To 11 'There are 12 months being January
month=0
Cov(u, m) =
FrmVegetParam.VegetationDataGridView.Rows.Item(u).Cells.Item("colCov" +
CStr(m)).Value()
Next

Mcrit(u) = Pst(r) * (Pb(r) / Pcrit(u)) ^ Ip(r)
Mwp(u) = Pst(r) * (Pb(r) / Pwp(u)) ^ Ip(r)
CZr(r) = Es(r) - Zr(u)
CZe(r) = Es(r) - Ze(u)
Cpsat(r) = Es(r) - Zsat(u)
Hfc(r) = Mfc(r) * Ze(u) * 1000

Hwp(u) = Mwp(u) * Ze(u) * 1000
Hcrit(u) = Mcrit(u) * Ze(u) * 1000
Pm50(u) = Pb(r) / (0.5 * Pst(r)) ^ (1 / Ip(r))
Pfc(u) = Pb(r) / (Mfc(r) / Pst(r)) ^ (1 / Ip(r))
CZmin(r) = Es(r) - Zmin(r)

Hsoil(0) = Hcrit(u)
Psoil(0) = Hsoil(0) / (1000 * Ze(u))

For v = 1 To ClimaticCount - 2

m = MonthNumber(v) - 1 'returns the month number
of each daily interval to calculate Cov

Hsoil(v) = Hsoil(v - 1)

```

```

Psoil(v) = Hsoil(v) / (1000 * Ze(u))

If Hdaily(Transect, v) >= Es(r) Then
    Pmat(v) = Pb(r)
ElseIf Hdaily(Transect, v) < Es(r) Then
    If Psoil(v) = 0 Then
        Pmat(v) = Pwp(u)
    Else
        Pmat(v) = Pb(r) / (Psoil(v) / Pst(r)) ^ (1 /
Ip(r))
        End If
    End If

Kh(v) = Ks(r) * (Pb(r) / Pmat(v)) ^ (3 * Ip(r) + 2)

If Hdaily(Transect, v - 1) >= CZe(r) Or
Hdaily(Transect, v - 1) < CZr(r) Then
    Rwu(v) = 0
ElseIf Hdaily(Transect, v - 1) < CZe(r) And
Hdaily(Transect, v - 1) > CZr(r) Then
    Rwu(v) = Max(0, (-Crt(u) * 10 / 1000) * (Pfc(u) -
Pmat(v)) * (1 / (1 + ((Pmat(v) / Pm50(u)) ^ 3.22)) * Rj(u))
    End If

If Hdaily(Transect, v) <= CZmin(r) Then
    Cwu(v) = 0
ElseIf Hdaily(Transect, v) > CZmin(r) And
Hdaily(Transect, v) >= CZe(r) Then
    Cwu(v) = Hfc(r) - Hsoil(v - 1) - Rwu(v)
ElseIf Hdaily(Transect, v) > CZmin(r) And
Hdaily(Transect, v) < CZe(r) Then
    Cwu(v) = Max(0, Min(Hfc(r) - Hsoil(v - 1) -
Rwu(v), ((-0.102 * Pmat(v) / (Hdaily(Transect, v) - CZe(r))) - 1) * Kh(v)
* 24))
    End If

Exc(v) = Max(0, PPd(v) + Cwu(v) + Rwu(v) - Hfc(r) +
Hsoil(v - 1))
Win(v) = PPd(v) + Cwu(v) + Rwu(v) - Exc(v)
Hini(v) = Win(v) + Hsoil(v - 1)

If Hini(v) < Hwp(u) Or Hcrit(u) < Hwp(u) Then
    Hrel(v) = 0
Else
    Hrel(v) = Min(1, (Hini(v) - Hwp(u)) / (Hcrit(u) -
Hwp(u)))
    End If

If Hdaily(Transect, v) >= CPsat(r) Then
    ETRuns(v) = 0
Else
    If Hdaily(Transect, v) >= CZe(r) Then
        ETRuns(v) = Ri(u) * Cov(u, m) * ETPd(v) * (1
- ((Hdaily(Transect, v) - CZe(r)) / (CPsat(r) - CZe(r))))
    Else

```



```

                                ETRuns(v) = Min(Hini(v) - Hwp(u), Ri(u) *
Hrel(v) * Cov(u, m) * ETPd(v))
                                End If
                                End If

                                'Codigo despues de mayo de 2010
                                If Hdaily(Transect, v) >= CPsat(r) Or
Hdaily(Transect, v) < CZr(r) Then
                                    ETRsat(v) = 0
                                ElseIf Hdaily(Transect, v) >= CZe(r) Then
                                    ETRsat(v) = Min(Cov(u, m) * ETPd(v) - ETRuns(v),
Cov(u, m) * ETPd(v) * Rj(u) * (1 - (Hdaily(Transect, v) - CZe(r)) /
(CPsat(r) - CZe(r))))
                                Else
                                    ETRsat(v) = Min(Cov(u, m) * ETPd(v) - ETRuns(v),
Cov(u, m) * ETPd(v) * Rj(u) * (Hdaily(Transect, v) - CZr(r)) / (CZe(r) -
CZr(r)))
                                End If

                                Hsoil(v) = Max(Hini(v) - ETRuns(v), Hwp(u))
                                ETRtot(v) = ETRsat(v) + ETRuns(v)
                                Bal(v) = Hsoil(v) - (Hsoil(v - 1) + Win(v) -
ETRuns(v))

                                ETindexDay(v) = ETRtot(v) / (ETPd(v) * Cov(u, m))

                                SumETRtot = SumETRtot + ETRtot(v)
                                sumETPd = sumETPd + ETPd(v)
                                ETindexTot = SumETRtot / (sumETPd * Cov(u, m))

                                HfcCol(r, u, v) = Hfc(r)
                                HsoilCol(r, u, v) = Hsoil(v)
                                PmatCol(r, u, v) = Pmat(v)
                                RwuCol(r, u, v) = Rwu(v)
                                KhCol(r, u, v) = Kh(v)
                                CwuCol(r, u, v) = Cwu(v)
                                ExcCol(r, u, v) = Exc(v)
                                WinCol(r, u, v) = Win(v)
                                HiniCol(r, u, v) = Hini(v)
                                HrelCol(r, u, v) = Hrel(v)
                                ETRunsCol(r, u, v) = ETRuns(v)
                                ETRsatCol(r, u, v) = ETRsat(v)
                                ETRtotCol(r, u, v) = ETRtot(v)
                                BalCol(r, u, v) = Bal(v)
                                ETindexDAYCol(r, u, v) = ETindexDay(v)
                                ETindexAVGCol(r, u) = ETindexTot
                                Next v
                                Next u
                                Next r
                                End Sub

                                End Module

```

## **ANEXO VIII: CÓDIGO FUENTE DEL PROGRAMA RIBAV-2**

En este anexo se presenta el código fuente del software de Ribav 2D. Al igual que en Ribav1D, el código fuente se presenta distribuido en dos tipos de archivo: los archivos de modulo y los archivos "Windows-Form". Como el objetivo de este programa era ser integrado en el software de Ripflow como librería DLL, no se ha desarrollado demasiado la interfaz gráfica de usuario debido a que se utilizará la interfaz propia de Ripflow. Los archivos que contienen el código fuente de Ribav 2D son los siguientes:

1. Datos de Entrada Generales: *"frmInputs.vb"*
2. Datos de Entrada Relativos a la Vegetación Dinámica: *"frmVegDynamics.vb"*
3. Módulo de Subrutinas de Mapas: *"modMaps.vb"*
4. Modulo de Subrutinas de Hidrometeorología: *"modHydromet.vb"*
5. Módulo General: *"modMain.vb"*

El código fuente de Ribav2D se muestra a continuación distribuido entre sus distintos archivos y formularios:

## 1. Datos de Entrada Generales: "frmInputs.vb"

The screenshot shows the 'General Input Data' dialog box. It contains several sections for file selection and parameter setting:

- ASCII Raster Files:**
  - Soil Types Map: D:\Archivos Joaquin\Pruebas Programas\Rib (Change Soil Type Map)
  - Initial Vegetation Types Map: D:\Archivos Joaquin\Pruebas Programas\Rib (Change Initial Vegetation Type Map)
  - Digital Elevation Model Map: D:\Archivos Joaquin\Pruebas Programas\Rib (Change DEM Map)
- CSV Parameters Files:**
  - Soil Parameters File: D:\Archivos Joaquin\Pruebas Programas\Rib (Change Soil Parameters File)
  - Vegetation Parameters File: D:\Archivos Joaquin\Pruebas Programas\Rib (Change Vegetation Parameters File)
- Hydrometeorological Files:**
  - Daily Hydrometeorological Data: D:\Archivos Joaquin\Pruebas Progra (Change Hydrometeorological Data)
  - River Level Maps: D:\Archivos Joaquin\Pruebas Progra (Change River Level Maps)

A table titled 'River Flow (m3/s) | Map File' is displayed, showing a list of river flow values and their corresponding map files:

River Flow (m3/s)	Map File
0	\Data\INPUT\Hydromet\Wte
0.25	\Data\INPUT\Hydromet\Wte
0.47	\Data\INPUT\Hydromet\Wte
1	\Data\INPUT\Hydromet\Wte
2	\Data\INPUT\Hydromet\Wte
5	\Data\INPUT\Hydromet\Wte
10	\Data\INPUT\Hydromet\Wte

At the bottom of the dialog, there are three buttons: 'Load Default Values', 'Simulate', and 'Change ET Limits'.

```
Public Class FrmMain
```

```
    Private Sub btLoadDefault_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btLoadDefault.Click
```

```
        Dim Lines() As String
        Dim DefaultRelativeFilePaths As String =
System.Environment.CurrentDirectory +
"\Data\DefaultRelativeFilePaths.csv"
        Lines = IO.File.ReadAllLines(DefaultRelativeFilePaths)

        txtSoilMap.Text = System.Environment.CurrentDirectory + Lines(0)
        "\Data\INPUT\Maps\SoilMap.asc"
        txtInitialVegMap.Text = System.Environment.CurrentDirectory +
Lines(1) " "\Data\INPUT\Maps\InitialVegetationMap.asc"
        txtDEMMap.Text = System.Environment.CurrentDirectory + Lines(2)
        "\Data\INPUT\Maps\DemMap.asc"
        txtSoilParam.Text = System.Environment.CurrentDirectory +
Lines(3) " "\Data\INPUT\Parameters\SoilParameters.csv"
        txtVegParam.Text = System.Environment.CurrentDirectory + Lines(4)
        "\Data\INPUT\Parameters\VegetationParameters.csv"
        txtHydrometData.Text = System.Environment.CurrentDirectory +
Lines(5) " "\Data\INPUT\Hydromet\Hydromet.csv"
        txtWTEMaps.Text = System.Environment.CurrentDirectory + Lines(6)
        "\Data\INPUT\Hydromet\WteMaps.csv"
        frmVegDynamics.txtYearlyETindexDirectory.Text =
System.Environment.CurrentDirectory + Lines(7) '+
\Data\OUTPUT\ETindexMaps\
        frmVegDynamics.txtDynamicVegDirectory.Text =
System.Environment.CurrentDirectory + Lines(8) '+ \Data\OUTPUT\VegMaps\
```

```
        Call LoadWTEMapsTable(txtWTEMaps.Text)
```

```

End Sub

Private Sub btSimulate_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
    MainHydromet()
    Simulate()
End Sub

Private Function ChangeFilePath(ByVal InitialDirectory As String,
ByVal filter As String)

    Dim openFileDialog1 As New OpenFileDialog
    Dim FilePath As String
    openFileDialog1.ShowDialog()
    openFileDialog1.InitialDirectory =
(System.Environment.CurrentDirectory + InitialDirectory)
    openFileDialog1.Filter = filter & "CSV Files (*.csv)|*.csv|All
Files (*.*)|*.*"
    FilePath = openFileDialog1.FileName
    Return FilePath

End Function

Private Sub btChangeSoilMap_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btChangeSoilMap.Click
    txtSoilMap.Text = ChangeFilePath("\Data", "ASC Files
(*.asc)|*.asc|All Files (*.*)|*.*")
End Sub

Private Sub btChangeVegMap_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btChangeVegMap.Click
    txtInitialVegMap.Text = ChangeFilePath("\Data\", "ASC Files
(*.asc)|*.asc|All Files (*.*)|*.*")
End Sub

Private Sub btChangeDEMMap_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btChangeDEMMap.Click
    txtDEMMap.Text = ChangeFilePath("\Data\", "ASC Files
(*.asc)|*.asc|All Files (*.*)|*.*")
End Sub

Private Sub btChangeSoilParam_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btChangeSoilParam.Click
    txtSoilParam.Text = ChangeFilePath("\Data\", "CSV Files
(*.csv)|*.csv|All Files (*.*)|*.*")
End Sub

Private Sub btChangeVegParam_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btChangeVegParam.Click
    txtVegParam.Text = ChangeFilePath("\Data\", "CSV Files
(*.csv)|*.csv|All Files (*.*)|*.*")
End Sub

Private Sub btChangeHydrometData_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btChangeHydrometData.Click
    txtHydrometData.Text = ChangeFilePath("\Data\", "CSV Files
(*.csv)|*.csv|All Files (*.*)|*.*")
End Sub

```

```

Private Sub btChangeWTEMaps_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btChangeWTEMaps.Click
    txtWTEMaps.Text = ChangeFilePath("\Data\", "ASC Files
(*.asc)|*.asc|All Files (*.*)|*.*")

End Sub

Private Sub LoadWTEMapsTable(ByVal InputFile As String)

    Dim FilePath As String = InputFile

    'A datatable is declared to retrieve the data from the CSV file
Dim WteMapsTable As DataTable
WteMapsTable = New DataTable

    WteMapsTable.Columns.Add("River Flow (m3/s)")
    WteMapsTable.Columns.Add("Map File")

    'The data from the CSV file is passed to the datatable and its
datacolumns
    'with the help of the string.split method to separate the columns
Dim TextLine As String = ""
Dim SplitLine() As String
Dim i As Integer = 0 'loop counter
If System.IO.File.Exists(FilePath) = True Then
    Dim objReader As New System.IO.StreamReader(FilePath)
    Dim a As Integer = WteMapsTable.Columns.Count
    Do While objReader.Peek() <> -1
        TextLine = objReader.ReadLine()
        SplitLine = Split(TextLine, ";")
        WteMapsTable.Rows.Add(SplitLine)
    Loop
Else
    MsgBox("WTE File Does Not Exist")
End If

    'The datatable could be transfered easily to a datagridview
WTEMapsDatagridView.DataSource = WteMapsTable.DefaultView

WTEMapsDatagridView.AutoSizeColumns(DataGridViewAutoSizeColumnsMode.All
Cells)

End Sub

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    btLoadDefault.PerformClick()
    frmVegDynamics.ETLowerLimit =
(frmVegDynamics.txtETLowerLimit.Text)
    frmVegDynamics.ETUpperLimit =
(frmVegDynamics.txtETUpperLimit.Text)
End Sub

```

```

Private Sub btChangeETlimits_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btChangeETlimits.Click
    frmVegDynamics.Show()
    Me.Hide()
End Sub

Private Sub btSimulate_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btSimulate.Click

    'The following variables are used to determin the simulation
process duration
    Dim StartingTime As DateTime
    Dim EndingTime As DateTime
    Dim SimulationDuration As TimeSpan

    'The simulation process starts
    StartingTime = DateTime.Now
    frmSimulationInformation.txtSimulation.AppendText("Simulation
STARTS at: " + CStr(StartingTime)) : Cr()

    frmSimulationInformation.Show()
    MainHydromet()
    Simulate() 'Further messages are added during the simulations

    frmSimulationInformation.txtSimulation.AppendText(" ") : Cr()
    EndingTime = DateTime.Now
    frmSimulationInformation.txtSimulation.AppendText("Simulation
ENDS at: " + CStr(EndingTime)) : Cr()
    SimulationDuration = EndingTime - StartingTime
    frmSimulationInformation.txtSimulation.AppendText(" ") : Cr()
    frmSimulationInformation.txtSimulation.AppendText("The simulation
lasted: " + SimulationDuration.ToString) : Cr()
    frmSimulationInformation.txtSimulation.AppendText("-----
-----") : Cr()

End Sub

Friend Sub Cr()
    frmSimulationInformation.txtSimulation.Text =
frmSimulationInformation.txtSimulation.Text + Environment.NewLine
End Sub

Private Sub GroupBox3_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox3.Enter

End Sub

End Class

```

## 2. Datos de Entrada Relativos a la Vegetación Dinámica: "frmVegDynamics.vb"

The screenshot shows a Windows application window titled "Vegetation Dynamics". Inside, there is a dialog box titled "Actual Evapotranspiration Index Ranges". The dialog contains a vertical axis labeled "ET Index" ranging from 0 at the bottom to 1 at the top. Three stages are marked on the axis: "Succession" (top), "Maintains Stage" (middle), and "Retrogression" (bottom). A "Succession Threshold" is indicated at the value 1, and a "Retrogression Threshold" is indicated at the value 0. To the right of the axis are two input fields for these thresholds, with the first containing "1" and the second containing "0". Below the axis are two buttons: "Back" and "Load Default". At the bottom of the dialog, there are two text boxes for specifying directories to save maps, both containing the same path: "D:\Archivos Joaquin\Pruebas Programas\Ribav2D\Ribav2D (14-6-10)\_last standalone\Ribav2D ultimo\Ribav2D\bin\D".

```
Public Class VegDynamics

    Friend ETLowerLimit As Single 'Lower ETindex limit /below
retrogression process
    Friend ETUpperLimit As Single 'Upper ETindex limit /above succession
process

    Private Sub btSimulate_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btSimulate.Click
        FrmMain.btSimulate.PerformClick()
    End Sub

    Private Sub btBack_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btBack.Click
        FrmMain.Show()
        ETLowerLimit = (txtETLowerLimit.Text)
        ETUpperLimit = (txtETUpperLimit.Text)
    End Sub
End Class
```



```

        Me.Hide()
    End Sub

    Private Sub VegDynamics_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        btLoadDefault.PerformClick()
    End Sub

    Friend Sub VegDynamicsCalc(ByVal year As Integer)

        Dim VegIndex(VegetationParameterArray.GetLength(0) - 1) As
Integer 'vectors of indexes or keys of the vegetation type
        Dim MinVegIndex As Integer 'minimum index or key of the
vegetation type
        Dim MaxVegIndex As Integer 'maximum index or key of the
vegetation type

        For z = 0 To VegetationParameterArray.GetLength(0) - 1
            VegIndex(z) = VegetationParameterArray(z, 0)
        Next

        MinVegIndex = VegIndex.Min
        MaxVegIndex = VegIndex.Max

        For i = 0 To ASCIIDataSt.nrows - 1 'nptos '- 1
            'Dim i As Integer = 39
            For j = 0 To ASCIIDataSt.ncols - 1
                'Dim j As Integer = 342

                'The succession/retrogression of the vegetation functional
types is taken into account
                'The MaximumVegIndex(TV) and MaximumVegIndex(RH)are also
taken into account
                If ETindexYearAvg(year, i, j) = ASCIIDataSt.NODATA_value
Then
                    VegetationMapArray(i, j) = ASCIIDataSt.NODATA_value
                Else
                    If ETindexYearAvg(year, i, j) >= ETUpperLimit Then
                        If VegetationMapArray(i, j) <> MaxVegIndex Then
                            VegetationMapArray(i, j) =
VegetationMapArray(i, j) + 1
                        Else
                            VegetationMapArray(i, j) =
VegetationMapArray(i, j)
                        End If
                    ElseIf ETindexYearAvg(year, i, j) <= ETLowerLimit
Then
                        If VegetationMapArray(i, j) <> MinVegIndex Then
                            VegetationMapArray(i, j) =
VegetationMapArray(i, j) - 1
                        Else
                            VegetationMapArray(i, j) =
VegetationMapArray(i, j)
                        End If
                    End If
                End If
            Next
        Next
    End Sub

```

```

        OutAsciiMap(VegetationMapArray, txtDynamicVegDirectory.Text +
"VegetationEndYear_" + Str(year + 1) + ".asc")

    End Sub

    Friend Sub ETindexMap(ByVal year)

        Dim CurrentYearETindex(,) As Single 'ETindex for a given year
        ReDim CurrentYearETindex(ETindexYearAvg.GetLength(1) - 1,
ETindexYearAvg.GetLength(2) - 1)

        For i = 0 To CurrentYearETindex.GetLength(0) - 1
            For j = 0 To CurrentYearETindex.GetLength(1) - 1
                CurrentYearETindex(i, j) = ETindexYearAvg(year, i, j)
            Next
        Next

        OutAsciiMap(CurrentYearETindex, txtYearlyETindexDirectory.Text +
"ETindexEndYear_" + Str(year + 1) + ".asc")

    End Sub

    Friend Sub OutAsciiMap(ByVal InputMatrix(,) As Single, ByVal
FileToSave As String)

        Dim objWriter As New System.IO.StreamWriter(FileToSave)

        'We write the first 6 lines which describe the mesh>>changing ","
to "."
        objWriter.WriteLine("ncols" + "          " +
ASCIIDataSt.ncols.ToString)
        objWriter.WriteLine("nrows" + "          " +
ASCIIDataSt.nrows.ToString)
        objWriter.WriteLine("xllcorner" + "        " +
ASCIIDataSt.xllcorner.ToString)
        objWriter.WriteLine("yllcorner" + "        " +
ASCIIDataSt.yllcorner.ToString)
        objWriter.WriteLine("cellsize" + "         " +
ASCIIDataSt.cellsize.ToString)
        objWriter.WriteLine("NODATA_value" + "      " +
ASCIIDataSt.NODATA_value.ToString)

        'We now pass to the file the data from the double array
        Dim record(InputMatrix.GetLength(0) - 1) As String

        For i = 0 To InputMatrix.GetLength(0) - 1
            For j = 0 To InputMatrix.GetLength(1) - 1
                If j = 0 Then
                    record(i) = InputMatrix(i, j).ToString
                Else
                    record(i) = record(i) + " " + InputMatrix(i,
j).ToString
                End If
            Next j
            ' record(i) = record(i).Replace(",", ".") 'We replace "." by
", "
            objWriter.WriteLine(record(i))
        End Sub

```

```

        Next i

        objWriter.Close()

    End Sub

    Private Sub btLoadDefault_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btLoadDefault.Click

        Dim Lines() As String
        Dim DefaultRelativeFilePaths As String =
System.Environment.CurrentDirectory +
"\Data\DefaultRelativeFilePaths.csv"
        Lines = IO.File.ReadAllLines(DefaultRelativeFilePaths)

        txtYearlyETIndexDirectory.Text =
System.Environment.CurrentDirectory + Lines(7) '+
\Data\OUTPUT\ETIndexMaps\
        txtDynamicVegDirectory.Text = System.Environment.CurrentDirectory
+ Lines(8) '+ \Data\OUTPUT\VegMaps\

        ETLowerLimit = (txtETLowerLimit.Text)
        ETUpperLimit = (txtETUpperLimit.Text)

    End Sub

End Class

```

### 3. Módulo de Subrutinas de Mapas: "modMaps.vb"

```
Imports System.IO
Imports System.Data.OleDb

Structure OutputStructure
    Dim arrayOut(,) As String
    Dim nRows As Integer
    Dim nColumns As Integer
End Structure

Module ModMaps

    Dim SoilAsciiSt As AsciiData
    Public Output As New OutputStructure

    Friend Function LoadSingleAscii(ByVal file As String, ByVal
    AsciiStartline As Integer)

        Dim Lines() As String
        Lines = IO.File.ReadAllLines(file)

        '*****Extracts the number of columns from the ASCII file
        AsciiDataSt.ncols = ExtractNumber(Lines(0))
        AsciiDataSt.nrows = ExtractNumber(Lines(1))
        AsciiDataSt.xllcorner = ExtractNumber(Lines(2))
        AsciiDataSt.yllcorner = ExtractNumber(Lines(3))
        AsciiDataSt.cellsize = ExtractNumber(Lines(4))
        AsciiDataSt.NODATA_value = ExtractNumber(Lines(5))

        Dim AsciiColumnSeparator As String = " " '*****ASCII file column
separator
        Dim dataArray(Lines.Length - AsciiStartline - 1,
        AsciiDataSt.ncols - 1) As String

        'In the following array the string values are stored as doubles
        ReDim NumberArray(Lines.Length - 1 - AsciiStartline,
        AsciiDataSt.ncols - 1)

        'The values of the lines of the ASCII are split by the column
separator and stored in the matrix
        For i = AsciiStartline To Lines.Length - 1
            Dim aux As String
            Dim cells() As String
            aux = Lines(i)
            cells = Split(aux, AsciiColumnSeparator)
            For j = 0 To AsciiDataSt.ncols - 1
                dataArray(i - AsciiStartline, j) = cells(j)
            Next
        Next

        'In the following loop the string values are converted to doubles
        For i = 0 To Lines.Length - 1 - AsciiStartline
```

```

        For j = 0 To ASCIIDataSt.ncols - 1
            NumberArray(i, j) = CSng(DataArray(i, j))
        Next
    Next

    Return NumberArray

End Function

Private Function ExtractCompleteNumber(ByVal inputString As String)

    '***** This function extracts the numerical values (number of
columns)
'from the first line of the ASCII file *****

    ' Convert it to an array of characters
    Dim myChars() As Char = inputString.ToCharArray()

    Dim AuxString As String = "" 'auxiliary string to be used in the
loop
    Dim ExtractedNumber As Single 'outputnumber

    ' Loop through the array testing if each is a digit or if it is a
minus sign
    For Each ch As Char In myChars
        If Char.IsDigit(ch) Or InStr(1, ch, "-") Or InStr(1, ch, ".")
Then
            AuxString = AuxString + ch
        End If
    Next

    Try
        ExtractedNumber = CDb1(AuxString)
    Catch ex As Exception
        'An error message would appear if the Extracted Number could
not be converted to integer
        MsgBox("The format of the ASCII file is not Valid",
MessageBoxIcon.Exclamation)
    End Try

    Return ExtractedNumber

End Function

Friend Function LoadTextFile(ByVal File As String, ByVal
ASCIIStartline As Integer, ByVal ColumnSeparator As String) As
OutputStructure

    'This subroutines is used to load soil and vegetation parameters
from CSV textfile
    Dim Lines() As String
    Lines = IO.File.ReadAllLines(File)

    Dim ColumnCount(Lines.Length - 1) As Integer
    Dim count As Integer 'column separator counter

```

```

For i = 0 To Lines.Length - 1
    count = 0
    For Each ch As Char In Lines(i)
        If InStr(1, ch, ";") Then
            count = count + 1
        End If
    Next
    ColumnCount(i) = count + 1
Next

Output.nColumns = ColumnCount.Max
Output.nRows = Lines.Length - ASCIIStartline
Dim DataArray(Lines.Length - ASCIIStartline - 1, Output.nColumns
- 1) As String
ReDim Output.arrayOut(Lines.Length - ASCIIStartline - 1,
Output.nColumns - 1)

'The values of the lines of the ASCII are split by the column
separator and stored in the matrix
For i = ASCIIStartline To Lines.Length - 1
    Dim aux As String
    Dim cells() As String
    aux = Lines(i)
    cells = Split(aux, ColumnSeparator)
    For j = 0 To Output.nColumns - 1
        Output.arrayOut(i - ASCIIStartline, j) = cells(j)
    Next
Next

Return Output

End Function

Friend Function ASCIIMapCharacteristics(ByVal file As String, ByVal
GridStartLine As Integer) As ASCIIData
    Dim Lines(GridStartLine - 1) As String
    IO.File.OpenRead(file)
    Dim objReader As New System.IO.StreamReader(file)

    For i = 0 To GridStartLine - 1
        Lines(i) = objReader.ReadLine
    Next

    ASCIIDataSt.ncols = ExtractCompleteNumber(Lines(0))
    ASCIIDataSt.nrows = ExtractCompleteNumber(Lines(1))
    ASCIIDataSt.NODATA_value = ExtractCompleteNumber(Lines(5))

    Return ASCIIDataSt
End Function

End Module

```

#### 4. Modulo de Subrutinas de Hidrometeorología: “modHydromet.vb”

```
Module ModHydromet

    Structure HydrometStructure
        Friend DailyDate(,) As Date
        Friend Years() As Integer
        Friend PP(,) As Single
        Friend ETP(,) As Single
        Friend Qdaily(,) As Single
    End Structure

    Public FlowFilePaths() As String
    Public Qcurve() As Single
    Public Hcurve(,,) As Single
    Public WteDay() As Single
    Public Hydromet As New HydrometStructure
    Public NumberArray(,) As Single 'output map matrix
    Friend OutputASCII(,) As Single

    Friend Sub MainHydromet()
        Call WteMaps(FrmMain.txtWTEMaps.Text)
        MultipleASCIIToArray(FlowFilePaths)
        OutputASCII = ASCIIDataSt.NumberArray
        Call LoadHydrometeorological(FrmMain.txtHydrometData.Text)
    End Sub

    Friend Function MultipleASCIIToArray(ByVal FlowFilePaths() As String)

        '**** This subroutine stores multiple ASCII grids like the Wte-
        grids from Hcruvethat vary through River Flows**
        For m = 0 To FlowFilePaths.Length - 1
            Dim Lines() As String
            Lines =
            IO.File.ReadAllLines(System.Environment.CurrentDirectory +
            FlowFilePaths(m))

            '*****Extracts the general info from the first rows of the
            ASCII file
            ASCIIDataSt.ncols = ExtractNumber(Lines(0))
            ASCIIDataSt.nrows = ExtractNumber(Lines(1))
            ASCIIDataSt.xllcorner = ExtractNumber(Lines(2))
            ASCIIDataSt.yllcorner = ExtractNumber(Lines(3))
            ASCIIDataSt.cellsize = ExtractNumber(Lines(4))
            ASCIIDataSt.NODATA_value = ExtractNumber(Lines(5))

            Dim ASCIIStartline As Integer = 6 '****Line where the
            numerical values of the grid start
            Dim ASCIIColumnSeparator As String = " " '*****ASCII file
            column separator
            Dim dataArray(Lines.Length - 1 - ASCIIStartline,
            ASCIIDataSt.ncols - 1) As String
            'In the following array the string values are stored as
            doubles
        End For
    End Function
End Module
```

```

        ReDim NumberArray(Lines.Length - 1 - ASCIIStartline,
ASCIIIDataSt.ncols - 1)
        ReDim Preserve Hcurve(FlowFilePaths.Length - 1, Lines.Length
- ASCIIStartline - 1, ASCIIIDataSt.ncols - 1)

        'The values of the lines of the ASCII are split by the column
separator and stored in the matrix
        For i = ASCIIStartline To Lines.Length - 1
            Dim aux As String
            Dim cells() As String
            aux = Lines(i)
            cells = Split(aux, ASCIIIColumnSeparator)
            For j = 0 To ASCIIIDataSt.ncols - 1
                dataArray(i - ASCIIStartline, j) = cells(j)
                'Hcurve(m, i, j) = dataArray(i, j)
            Next
        Next

        'In the following loop the string values are converted to
doubles
        For i = 0 To Lines.Length - 1 - ASCIIStartline
            For j = 0 To ASCIIIDataSt.ncols - 1
                NumberArray(i, j) = CDBl(dataArray(i, j))
                Hcurve(m, i, j) = NumberArray(i, j)
            Next
        Next
    Next

    Return Hcurve

End Function

Friend Function ExtractNumber(ByVal inputString As String)

    '***** This function extracts the numerical values from the
first line of the ASCII file *****

    ' Convert it to an array of characters
    Dim myChars() As Char = inputString.ToCharArray()
    Dim AuxString As String = "" 'auxiliary string to be used in the
loop
    Dim ExtractedNumber As Single 'outputnumber

    ' Loop through the array testing if each is a digit or if it is a
minus sign
    For Each ch As Char In myChars
        If Char.IsDigit(ch) Or InStr(1, ch, "-") Or InStr(1, ch, ".")
Then 'change dot to comma if necessary
            AuxString = AuxString + ch
        End If
    Next

    Try
        ExtractedNumber = CDBl(AuxString)
    Catch ex As Exception
        'An error message would appear if the Extracted Number could
not be converted to integer
        MsgBox("The format of the NoData Value from the ASCII file is
not Valid", MessageBoxButtons.Exclamation)
    End Try

```



```

Return ExtractedNumber

End Function

Public Sub WteMaps(ByVal file As String)

    '***** This subroutine reads the paths of the files of the
Wte Maps *****

    'A string is written for the file's path
Dim FilePath As String = file

    'A datatable is declared to retrieve the data from the CSV file
Dim WteMapsTable As DataTable
WteMapsTable = New DataTable

    'The datacolumns are declared and named
Dim FlowValues As DataColumn = New DataColumn("FlowValues")
Dim FlowFiles As DataColumn = New DataColumn("FlowFiles")

    'The datatypes of the datacolumns are defined
FlowValues.DataType = System.Type.GetType("System.Single")
FlowFiles.DataType = System.Type.GetType("System.String")

    'The datacolumns are added to the datatable
WteMapsTable.Columns.Add(FlowValues)
WteMapsTable.Columns.Add(FlowFiles)

    'The data from the CSV file is passed to the datatable and its
datacolumns
    'with the help of the string.split method to separate the columns
Dim TextLine As String = ""
Dim SplitLine() As String
If System.IO.File.Exists(FilePath) = True Then
    Dim objReader As New System.IO.StreamReader(FilePath)
    Dim a As Integer = WteMapsTable.Columns.Count
    Do While objReader.Peek() <> -1
        TextLine = objReader.ReadLine()
        SplitLine = Split(TextLine, ";")
        WteMapsTable.Rows.Add(SplitLine)
    Loop
Else
    MsgBox("Hydrometeorological File Does Not Exist")
End If

    'The datatable could be transfered easily to a datagridview
'Form1.DataGridView2.DataSource = WteMapsTable.DefaultView

    'The flow values intervals of each map are loaded to a vector
array
ReDim Qcurve(WteMapsTable.Rows.Count - 1)
For i = 0 To WteMapsTable.Rows.Count - 1
    Qcurve(i) = WteMapsTable.Rows(i).Item("FlowValues")
Next

    'The flowfiles paths are loaded to a vector array

```

```

ReDim FlowFilePaths(WteMapsTable.Rows.Count - 1)

For f = 0 To WteMapsTable.Rows.Count - 1
    FlowFilePaths(f) = WteMapsTable.Rows(f).Item("FlowFiles")
Next

End Sub

Friend Sub LoadHydrometeorological(ByVal Hydrometfile As String)

    'This subroutine is used to read the Hydromet Data(PP,ETP,Q) and
    divide them by years

    'A string is written for the file's relative path
    ' Hydrometfile = "\Data\Clima.csv"
    Dim FilePath As String = Hydrometfile

    'A datatable is declared to retrieve the data from the CSV file
    Dim table1 As DataTable
    table1 = New DataTable("Hydromet")

    'The datacolumns are declared and named
    Dim DateMeteo As DataColumn = New DataColumn("DateMeteo")
    Dim PPcol As DataColumn = New DataColumn("PPcol")
    Dim ETPcol As DataColumn = New DataColumn("ETPcol")
    Dim Qdailycol As DataColumn = New DataColumn("Qdailycol")

    'The datatypes of the datacolumns are defined
    DateMeteo.DataType = System.Type.GetType("System.DateTime")
    PPcol.DataType = System.Type.GetType("System.Single")
    ETPcol.DataType = System.Type.GetType("System.Single")
    Qdailycol.DataType = System.Type.GetType("System.Single")

    'The datacolumns are added to the datatable
    table1.Columns.Add(DateMeteo)
    table1.Columns.Add(PPcol)
    table1.Columns.Add(ETPcol)
    table1.Columns.Add(Qdailycol)

    'The data from the CSV file is passed to the datatable and its
    datacolumns
    'with the help of the string.split method to separate the columns
    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(FilePath) = True Then
        Dim objReader As New System.IO.StreamReader(FilePath)
        Dim a As Integer = table1.Columns.Count
        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, ";")
            table1.Rows.Add(SplitLine)
        Loop
    Else
        MsgBox("Hydrometeorological File Does Not Exist")
    End If

    'The datatable could be transfered easily to a datagridview
    'Form1.DataGridView1.DataSource = table1.DefaultView

```

```

    'The following loop calculates the number of years of
hydrometeorological data
    Dim b As Integer = table1.Rows.Count 'this variable count the
total number of days from all the years
    Dim n As Integer = 0 'counts the total number of years

    For i = 0 To b - 1
        Dim a As Integer 'represents every NEW year of every 31Dic-
1Jan daily interval
        Dim c As Integer 'represents every OLD year of every 31Dic-
1Jan daily interval
        a = (Year(table1.Rows(i).Item(0)))
        If i > 0 Then
            c = (Year(table1.Rows(i - 1).Item(0)))
            If c <> a Then
                n = n + 1
            End If
        End If
    Next i

    'The fields/columns of the Hydrometeorological Structure are
redimensioned
    'by number of years and by numbers of days from each year

    ReDim Hydromet.DailyDate(n, b / (n + 1))
    ReDim Hydromet.PP(n, b / (n + 1))
    ReDim Hydromet.ETP(n, b / (n + 1))
    ReDim Hydromet.Qdaily(n, b / (n + 1))

    'The data from the datatable is separated by year in the
following loop
    Dim y As Integer = 0 'counts number of years of data
    Dim d As Integer = 0 'counts number of days for each year of
data

    'The years are recorded in the following vector
    ReDim Hydromet.Years(n - 1)

    'The following double-loop separates the hydrometeorological data
by years
    For i = 0 To b - 1
        Dim a As Integer 'represents every OLD year of every 31Dic-
1Jan daily interval
        Dim c As Integer 'represents every NEW year of every 31Dic-
1Jan daily interval

        ' The data from the datatable is passed to the structure
elements
        Hydromet.DailyDate(y, d) = table1.Rows(i).Item("DateMeteo")
        Hydromet.PP(y, d) = table1.Rows(i).Item("PPcol")
        Hydromet.ETP(y, d) = table1.Rows(i).Item("ETPcol")
        Hydromet.Qdaily(y, d) = table1.Rows(i).Item("Qdailycol")

        ' The daily interval for each year is incremented by 1
        d = d + 1

        'This control strucutre DETECTS the year changes

```

```

        'it also prevents errors in the last 2 days of the last year
of the data
    If i < b - 1 Then
        a = (Year(table1.Rows(i).Item(0)))
        c = (Year(table1.Rows(i + 1).Item(0)))
    Else
        a = (Year(table1.Rows(i).Item(0)))
        c = a
    End If

    'This control structure ACTIVATES when the year changes take
place
    'It increments the array column j=j+1 because j represents
the year
    'If the year isn't a leap year it leaves a blank value at the
end which is eliminated in the general module's methods
    Dim j As Integer
    If c <> a Then
        'For j = 0 To n - 1
        If y = 0 Then 'in the first year we put the initial value
            Hydromet.Years(j) = a
            j = j + 1
        Else 'for the following years
            Hydromet.Years(j) = c
            j = j + 1
        End If
        y = y + 1
        d = 0
    End If
Next

End Sub

Friend Sub WteValues(ByVal InputFile As String, ByVal StartRow As
Integer) 'As OutputStructure

    'A string is written for the file's relative path
    ' file = "\Data\WteMaps.csv"
    Dim FilePath As String = System.Environment.CurrentDirectory +
InputFile

    'A datatable is declared to retrieve the data from the CSV file
    Dim WteMapsTable As DataTable
    WteMapsTable = New DataTable

    'The data from the CSV file is passed to the datatable and its
datacolumns
    'with the help of the string.split method to separate the columns
    Dim TextLine As String = ""
    Dim SplitLine() As String
    If System.IO.File.Exists(FilePath) = True Then
        Dim objReader As New System.IO.StreamReader(FilePath)
        Dim a As Integer = WteMapsTable.Columns.Count
        Do While objReader.Peek() <> -1
            TextLine = objReader.ReadLine()
            SplitLine = Split(TextLine, " ")
            ' Dim FlowValues As DataColumn = New DataColumn

```

```

        For i = 0 To SplitLine.Length
            WteMapsTable.Columns.Add()
        Next
        WteMapsTable.Rows.Add(SplitLine)
    Loop
Else
    MsgBox("Hydrometeorological File Does Not Exist")
End If

'The datatable could be transfered easily to a datagridview
' Form1.DataGridView2.DataSource = WteMapsTable.DefaultView

End Sub

Friend Function WteCalculations(ByVal y As Integer, ByVal i As
Integer, ByVal j As Integer, ByVal d As Integer)

    ReDim WteDay(d)

    For d = 0 To Hydromet.Qdaily.GetLength(1) - 1 '(y,
d)rowcount_daily - 2
        For q = 0 To Qcurve.Length - 1 'rowcount_curve - 2
            If Hcurve(q, i, j) = ASCIIDataSt.NODATA_value Then 'we
check if cell has a notadatavalue
                WteDay(d) = ASCIIDataSt.NODATA_value
            Else
                If Math.Abs(Hydromet.Qdaily(y, d) - Qcurve(q)) <
0.0000001 Then
                    WteDay(d) = Hcurve(q, i, j)
                    GoTo break
                Else
                    If Hydromet.Qdaily(y, d) < Qcurve(q) Then
                        WteDay(d) = Hcurve(q - 1, i, j) +
(Hydromet.Qdaily(y, d) - Qcurve(q - 1)) / (Qcurve(q) - Qcurve(q - 1)) *
(Hcurve(q, i, j) - Hcurve(q - 1, i, j))
                        GoTo break
                    End If
                End If
            End If
        Next q
    Next d

break:

    Return WteDay
End Function

End Module

```

## 5. Módulo General: "modMain.vb"

```
Structure ASCIIIData
    Friend ncols As Integer
    Friend nrows As Integer
    Friend xllcorner As Single
    Friend yllcorner As Single
    Friend cellsize As Single
    Friend NODATA_value As Single
    Friend NumberArray(,) As Single
End Structure

Module Module1

    'Input files and Maps
    Friend SoilMap As String           'Soil ASCII Map file path
    Friend DEMMap As String           'DEM ASCII Map file path
    Friend SoilParam As String        'Soil Parameters CSV file path
    Friend VegParam As String         'Vegetation Parameters qCSV file path
    Friend HydrometData As String     'Hydrometeorology data file path
    Friend WTEMaps As String          'WTE Map file path

    Friend ASCIIIDataSt As New ASCIIIData
    Public outASCII As New ASCIIIData
    Public VegetationParameterArray(,) As String
    Public VegetationMapArray(,) As Single 'ver cambio anual

    Friend ETindexYearAvg(,,) As Single
    Friend nDays As Integer          'Number of days in each year
    Friend nRows As Integer          'Number of rows in the grid of cells
    Friend nColumns As Integer       'Number of columns in the grid of cells
    Friend nYears As Integer         'Number of years of simulation

    Friend Sub Simulate()

        Dim VegMap As String          'Vegetation ASCIIMap file path
        //Depend on the year

        'The paths of the MAP inputfiles is read from the textboxes
        SoilMap = FrmMain.txtSoilMap.Text

        DEMMap = FrmMain.txtDEMMap.Text
        SoilParam = FrmMain.txtSoilParam.Text
        VegParam = FrmMain.txtVegParam.Text
        HydrometData = FrmMain.txtHydrometData.Text
        WTEMaps = FrmMain.txtWTEMaps.Text
        nYears = Hydromet.Qdaily.GetLength(0)
        nDays = Hydromet.Qdaily.GetLength(1)

        Dim SoilnColumns As Integer = ASCIIIMapCharacteristics(SoilMap,
6).ncols
        Dim ElevationnColumns As Integer =
ASCIIIMapCharacteristics(DEMMap, 6).ncols

        'The following block calculates the maximum number of ROWS of
each map grid using ASCIIIMapCharacteristics function
```

```

        Dim SoilnRows As Integer = ASCIIMapCharacteristics(SoilMap,
6).nrows
        Dim ElevationnRows As Integer = ASCIIMapCharacteristics(DEMMap,
6).nrows

        'Matrix where the input maps data is stored
        Dim SoilMapArray(,) As Single
        Dim ElevationMapArray(,) As Single
        SoilMapArray = LoadSingleAscii(SoilMap, 6)
'poner direccion Ascii mapa suelos
        ElevationMapArray = LoadSingleAscii(DEMMap, 6)
'poner direccion Ascii mapa elevaciones

        Dim y As Integer = 1

        frmSimulationInformation.txtSimulation.AppendText(" ") :
FrmMain.Cr()
        frmSimulationInformation.txtSimulation.AppendText("Simulating
YEAR " + Str(y + 1) + " out of " +
Str(ModHydromet.Hydromet.Qdaily.GetLength(0) - 1)) : FrmMain.Cr()

        'WE LOAD THE DATA FROM THE INITIAL VEGETATION MAP OR FROM THE
DYNAMIC VEGETATION MAP
        VegMap = FrmMain.txtInitialVegMap.Text

        'The VegetationTypeArray is loaded from the vegetation map file
        'We also define the starting line
        VegetationMapArray = LoadSingleAscii(VegMap, 6)

        'Columns are counted from the vegetation map for each year
        Dim VegetationnColumns As Integer =
ASCIIMapCharacteristics(VegMap, 6).ncols
        'Rows are counted from the vegetation map
        Dim VegetationnRows As Integer = ASCIIMapCharacteristics(VegMap,
6).nrows

        'The coincident number of ROWS of the three maps is calculated
(Soil, DEM, Vegetation)
        'Remember vegetation map can vary for each year
        Dim nColumnsProv As Integer 'Auxiliary variable used in the
math.max function
        nColumnsProv = Math.Max(SoilnColumns, VegetationnColumns)
        nColumns = Math.Max(nColumnsProv, ElevationnColumns)

        'The coincident number of COLUMNS of the three maps is calculated
(Soil, DEM, Vegetation)
        'Remember vegetation map can vary for each year
        Dim nRowsProv As Integer 'Auxiliary variable used in the math.max
function
        nRowsProv = Math.Max(SoilnRows, VegetationnRows)
        nRows = Math.Max(nRowsProv, ElevationnRows)

        'We call the general module
        Call GeneralSub(y, VegMap, SoilMapArray, ElevationMapArray)

        'We print the yearly map of ETindex
        Call frmVegDynamics.ETindexMap(y)

```

```

        'We call the vegetation succession module
        frmVegDynamics.VegDynamicsCalc(y)

End Sub

Private Sub GeneralSub(ByVal y As Integer, ByVal VegMap As String,
ByVal SoilMapArray(,) As Single, ByVal ElevationMapArray(,) As Single)

    Dim SumETRtot As Single        'Sum of year ETRtot
    Dim sumETPd As Single          'Sum of year ETP
    Dim ETindex As Single          'ETindex for a given year

    'We redimension the variables whose dimensions depend on nRows
    and nColumns
    ReDim Preserve ETindexYearAvg(nYears, nRows - 1, nColumns - 1)

    'Leap years are taken into account in this conditional structure
    to determin ndays
    If Date.IsLeapYear(Hydromet.Years(y)) = True Then
        nDays = Hydromet.Qdaily.GetLength(1)
    Else
        nDays = Hydromet.Qdaily.GetLength(1) - 1
    End If

    'We use PPy and ETPy auxiliary vectors to prevent errors during
    leap years
    Dim PPy() As Single            'Daily Precipitation for a
    given year [mm]
    Dim ETPy() As Single          'Daily Evapotranspiration for
    a given year [mm]
    ReDim Preserve PPy(nDays)    'Daily Precipitation for a
    given year [mm]
    ReDim Preserve ETPy(nDays)  'Daily Evapotranspiration for
    a given year [mm]

    For d = 0 To Hydromet.DailyDate.GetLength(1) - 1
        PPy(d) = Hydromet.PP(y, d)
        ETPy(d) = Hydromet.ETP(y, d)
    Next

    'The paths of the PARAMETERS inputfiles are also read from the
    textboxes
    SoilParam = FrmMain.txtSoilParam.Text
    VegParam = FrmMain.txtVegParam.Text

    Dim SoilParameterArray(,) As String
    Dim nRowsSoilParameters As Integer
    SoilParameterArray = LoadTextFile(SoilParam, 0, ";").arrayOut
    nRowsSoilParameters = LoadTextFile(SoilParam, 0, ";").nRows

    'Dim VegetationParameterArray(,) As String
    Dim nRowsVegetationParameters As Integer
    VegetationParameterArray = LoadTextFile(VegParam, 0,
";").arrayOut
    nRowsVegetationParameters = LoadTextFile(VegParam, 0, ";").nRows

```



```

For i = 0 To ASCIIDataSt.nrows - 1 'nptos '- 1
    For j = 0 To ASCIIDataSt.ncols - 1
        If SoilMapArray(i, j) = ASCIIDataSt.NODATA_value Or
VegetationMapArray(i, j) = ASCIIDataSt.NODATA_value Or
ElevationMapArray(i, j) = ASCIIDataSt.NODATA_value Then
            'If at least one of the maps has a NODATA_value then
the calculations are cancelled for that cell and the output ETindex has a
NODATA_value also
                ETindexYearAvg(y, i, j) = ASCIIDataSt.NODATA_value
            Else
                'Parameters from the soil type
                Dim Pst As Single           'Φ Porosity [ ]
                Dim Pb As Single           'Bubbling capillary pressure
(mm)
                Dim Ip As Single           'λ=Pore -size index
                Dim Mfc As Single           'Field Capacity Moisture [ ]
                Dim Ks As Single           'Fully Saturated
conductivity (mm/hr)
                Dim Zmin As Single         'Minimum Capillary Depth to
be Considered [m]

                'VEGETATION DATA(form2)
                Dim Zr As Single           'Root Depth Zr max [m]
                Dim Ze As Single           'Effective Root Depth Ze [m]
                Dim Zsat As Single         'Extinction Saturation Psat
[m]
                Dim Ri As Single           'Transpiration Factor from
Unsaturated Zone [ ]
                Dim Rj As Single           'Transpiration Factor from
Saturated Zone [ ]
                Dim Crt As Single         'Maximum Soil-Root Water
Conductance CRT [mmMpa-lh-1]
                Dim Pwp As Single         'Moisture at Wilting Point [
]
                Dim Pcrit As Single       'Critical Moisture [ ]
                ' Dim Veg As String       'River Vegetation Type
[1 to 5]
                Dim Cov As Single         'Plant Cover Cv_yearly

                'GENERAL CALCULATIONS(form4)
                Dim Mcrit As Single        'θCrit Moisture at Critical
Point [ ]
                Dim Mwp As Single         'θwp Moisture at Wilting
Point [ ]
                Dim CZr As Single         'Root Depth Elevation (m)
                Dim CZe As Single         'Effective Root Depth
Elevation (m)
                Dim CPsat As Single       'Extinction Saturation
Elevation (m)
                Dim Hfc As Single         'Field Capacity Depth (mm)
                Dim Hwp As Single         'Permanent Wilting Point
Depth (mm)
                Dim Hcrit As Single       'Critical Moisture Depth
(mm)
                Dim Pm50 As Single        'Medium Saturation Moisture
Pressure [Kpa]

```

[Kpa]	Dim Pfc As Single	'Field Capacity Pressure
in CWU [m]	Dim CZmin As Single	'Elevation to be considered
cell (m above sea level)	Dim Es As Single	'Elevation of the terrain
Elevation [m] (above sea level)	Dim Wte() As Single	'Daily Water Table
Depth (mm)	Dim Hsoil() As Single	'Daily Soil Moisture
Relation [ ]	Dim Psoil() As Single	'Daily Soil Moisture
[mm]	Dim Pmat() As Single	'Matrix Potential [mm]
[mm/hr]	Dim Rwu() As Single	'Root Water Uptake Ahr
[mm]	Dim Kh() As Single	'Hydraulic Conductivity
	Dim Cwu() As Single	'Capilarity Water Uptake
	Dim Exc() As Single	'Excess of Water [mm]
	Dim Win() As Single	'Water Income [mm]
	Dim Hini() As Single	'Initial Moisture [mm]
[ ]	Dim Hrel() As Single	'Daily Relative Moisture
(simplificacion de calculos)	Dim ETRnul() As Single	'Casos ETR=0
of the Unsaturated Zone [mm]	Dim ETRuns() As Single	'Real Evapotranspiration
of the Saturated Zone [mm]	Dim ETRsat() As Single	'Real Evapotranspiration
Evapotranspiration [mm]	Dim ETRtot() As Single	'Real Total
	Dim Bal() As Single	'Water Balance [mm]
Depth (mm)	ReDim Preserve Hsoil(nDays)	'Daily Soil Moisture
Relation (cm3/cm3)	ReDim Preserve Psoil(nDays)	'Daily Soil Moisture
[mm]	ReDim Preserve Pmat(nDays)	'Matrix Potential
Ahr [mm]	ReDim Preserve Rwu(nDays)	'Root Water Uptake
Conductivity [mm/hr]	ReDim Preserve Kh(nDays)	'Hydraulic
Uptake [mm]	ReDim Preserve Cwu(nDays)	'Capilarity Water
[mm]	ReDim Preserve Exc(nDays)	'Excess of Water
	ReDim Preserve Win(nDays)	'Water Income [mm]
[mm]	ReDim Preserve Hini(nDays)	'Initial Moisture
Moisture (cm3/cm3)	ReDim Preserve Hrel(nDays)	'Daily Relative

```

ReDim Preserve ETRnul(nDays) 'Casos ETR=0
(simplificacion de calculos)
ReDim Preserve ETRuns(nDays) 'Real
Evapotranspiration of the Unsaturated Zone [mm]
ReDim Preserve ETRsat(nDays) 'Real
Evapotranspiration of the Saturated Zone [mm]
ReDim Preserve ETRtot(nDays) 'Real Total
Evapotranspiration [mm]
ReDim Preserve Bal(nDays) 'Water Balance [mm]

Dim r As Integer 'Soil Type Index
Dim u As Integer 'Vegetation Type Index
Dim n As String 'Soil OR Vegetation type index of the
cell

'Determining soil type index
n = SoilMapArray(i, j) 'soil type index of the cell
For b = 0 To nRowsSoilParameters - 1 'b is the row
index
    If n = SoilParameterArray(b, 0) Then
        r = b
        GoTo break1
    End If
Next
break1:

'Determining vegetation type index
n = VegetationMapArray(i, j) 'vegetation type index
of the cell
'Determining vegetation type index
For b = 0 To nRowsVegetationParameters - 1 'b is the
row index
    If n = VegetationParameterArray(b, 0) Then
        u = b
        GoTo break2
    End If
Next
break2:

r = SoilMapArray(i, j) 'Index for type of soil of
the cell
u = VegetationMapArray(i, j) 'Index for type of
vegetacion of the cell

Es = CSng(ElevationMapArray(i, j)) 'the Elevation is
loaded directly from the DEM Map and array

Pst = CSng(SoilParameterArray(r - 1, 1))
Ip = CSng(SoilParameterArray(r - 1, 2))
Pb = CSng(SoilParameterArray(r - 1, 3))
Ks = CSng(SoilParameterArray(r - 1, 4))
Mfc = CSng(SoilParameterArray(r - 1, 5))
Zmin = CSng(SoilParameterArray(r - 1, 6))

Zr = CSng(VegetationParameterArray(u - 1, 1))
Ze = CSng(VegetationParameterArray(u - 1, 2))
Zsat = CSng(VegetationParameterArray(u - 1, 3))
Ri = CSng(VegetationParameterArray(u - 1, 4))
Rj = CSng(VegetationParameterArray(u - 1, 5))

```

```

Cov = CSng(VegetationParameterArray(u - 1, 6))
Crt = CSng(VegetationParameterArray(u - 1, 7))
Pwp = CSng(VegetationParameterArray(u - 1, 8))
Pcrit = CSng(VegetationParameterArray(u - 1, 9))

'*****
'We put the following variables=0 so the sum is done
correctly ///Very Important
SumETRtot = 0
sumETPD = 0
ETindex = 0

'*****

If Pcrit = 0 Then
    Mcrit = 0
Else
    Mcrit = Pst * (Pb / Pcrit) ^ Ip
End If
Mwp = Pst * (Pb / Pwp) ^ Ip

CZr = Es - Zr
CZe = Es - Ze
Cpsat = Es - Zsat
Hfc = Mfc * Ze * 1000

Hwp = Mwp * Ze * 1000
Hcrit = Mcrit * Ze * 1000
Pm50 = Pb / (0.5 * Pst) ^ (1 / Ip)
Pfc = Pb / (Mfc / Pst) ^ (1 / Ip)
CZmin = Es - Zmin

'The WteCalculations function is called which has as
input the: year, row, column and Qdaily)
Wte = WteCalculations(y, i, j, nDays)

For d = 0 To nDays - 1 '- 2

    If d = 0 Then
        Hsoil(0) = Hcrit
    End If

    Psoil(d) = Hsoil(d) / (1000 * Ze)

    If Wte(d) >= Es Then
        Pmat(d) = Pb
    ElseIf Wte(d) < Es Then
        If Psoil(d) = 0 Then
            Pmat(d) = Pwp
        Else
            Pmat(d) = Pb / (Psoil(d) / Pst) ^ (1 /
Ip)

        End If
    End If

    Kh(d) = Ks * (Pb / Pmat(d)) ^ (3 * Ip + 2)

```

```

'Root water uptake calculations
If d = 0 Then 'for the first day
    If Wte(0) >= CZe Or Wte(0) < CZr Then
        Rwu(d) = 0
    ElseIf Wte(0) < CZe And Wte(0) > CZr Then
        Rwu(d) = Math.Max(0, (-Crt * 10 / 1000) *
(Pfc - Pmat(d)) * (1 / (1 + ((Pmat(d) / Pm50)) ^ 3.22)) * Rj)
    End If
Else 'for the rest of the days
    If Wte(d - 1) >= CZe Or Wte(d - 1) < CZr Then
        Rwu(d) = 0
    ElseIf Wte(d - 1) < CZe And Wte(d - 1) > CZr
Then
        Rwu(d) = Math.Max(0, (-Crt * 10 / 1000) *
(Pfc - Pmat(d)) * (1 / (1 + ((Pmat(d) / Pm50)) ^ 3.22)) * Rj)
    End If
End If

''Capillary water uptake calculations
If Wte(d) <= CZmin Then
    Cwu(d) = 0
ElseIf Wte(d) > CZmin And Wte(d) >= CZe Then
    If d = 0 Then 'for the first day
        Cwu(d) = Hfc - Hcrit - Rwu(d)
    Else 'for the rest of the days
        Cwu(d) = Hfc - Hsoil(d - 1) - Rwu(d)
    End If
ElseIf Wte(d) > CZmin And Wte(d) < CZe Then
    If d = 0 Then 'for the first day
        Cwu(d) = Math.Max(0, Math.Min(Hfc - Hcrit
- Rwu(d), ((-0.102 * Pmat(d) / (Wte(d) - CZe)) - 1) * Kh(d) * 24))
    Else 'for the rest of the days
        Cwu(d) = Math.Max(0, Math.Min(Hfc -
Hsoil(d - 1) - Rwu(d), ((-0.102 * Pmat(d) / (Wte(d) - CZe)) - 1) * Kh(d)
* 24))
    End If
End If

'We calculate the excess of water of the tank
If d = 0 Then 'for the first day
    Exc(d) = Math.Max(0, PPy(d) + Cwu(d) + Rwu(d)
- Hfc + Hcrit)
Else 'for the rest of the days
    Exc(d) = Math.Max(0, PPy(d) + Cwu(d) + Rwu(d)
- Hfc + Hsoil(d - 1))
End If

Win(d) = PPy(d) + Cwu(d) + Rwu(d) - Exc(d)

If d = 0 Then 'for the first day
    Hini(d) = Win(d) + Hcrit
Else 'for the rest of the days
    Hini(d) = Win(d) + Hsoil(d - 1)
End If

If Hini(d) < Hwp Or Hcrit < Hwp Then
    Hrel(d) = 0

```

```

Else
    Hrel(d) = Math.Min(1, (Hini(d) - Hwp) /
(Hcrit - Hwp))
End If

If Wte(d) >= CPsat Then
    ETRuns(d) = 0
Else
    If Wte(d) >= CZe Then
        ETRuns(d) = Ri * Cov * ETPy(d) * (1 -
((Wte(d) - CZe) / (CPsat - CZe)))
    Else
        ETRuns(d) = Math.Min(Hini(d) - Hwp, Ri *
Hrel(d) * Cov * ETPy(d))
    End If
End If

'Codigo ETsat despues de mayo de 2010
If Wte(d) >= CPsat Or Wte(d) < CZr Then
    ETRsat(d) = 0
ElseIf Wte(d) < CPsat And Wte(d) >= CZe Then
    ETRsat(d) = Math.Min(Cov * ETPy(d) -
ETRuns(d), Cov * ETPy(d) * Rj * (1 - (Wte(d) - CZe) / (CPsat - CZe)))
ElseIf Wte(d) < CZe And Wte(d) >= CZr Then
    ETRsat(d) = Math.Min(Cov * ETPy(d) -
ETRuns(d), Cov * ETPy(d) * Rj * (Wte(d) - CZr) / (CZe - CZr))
End If

If d <> 0 Then
    Hsoil(d) = Math.Max(Hini(d) - ETRuns(d), Hwp)
End If

ETRtot(d) = ETRsat(d) + ETRuns(d)

If d = 0 Then 'for the first day
    Bal(d) = Hsoil(d) - (Hcrit + Win(d) -
ETRuns(d))
Else 'for the rest of the days
    Bal(d) = Hsoil(d) - (Hsoil(d - 1) + Win(d) -
ETRuns(d))
End If

'The sums of ETRtot and ETP are calculated
If d <> 0 Then
    SumETRtot = SumETRtot + ETRtot(d)
    sumETPd = sumETPd + ETPy(d)
End If
Next d

ETindex = SumETRtot / (sumETPd * Cov)
ETindexYearAvg(y, i, j) = ETindex

End If

Next j
Next i

```

```
End Sub
```

```
Friend Function CommaToPointSeparator(ByVal inputString As String)  
    inputString.Replace(",", ".")  
    Return inputString  
End Function
```

```
End Module
```

**ANEXO IX: CÓDIGO FUENTE PROGRAMA PARA  
EL CALCULO DEL COEFICIENTE KAPPA DE  
COHEN ENTRE LOS MAPAS DE VEGETACIÓN  
OBSERVADA Y VEGETACIÓN SIMULADA**

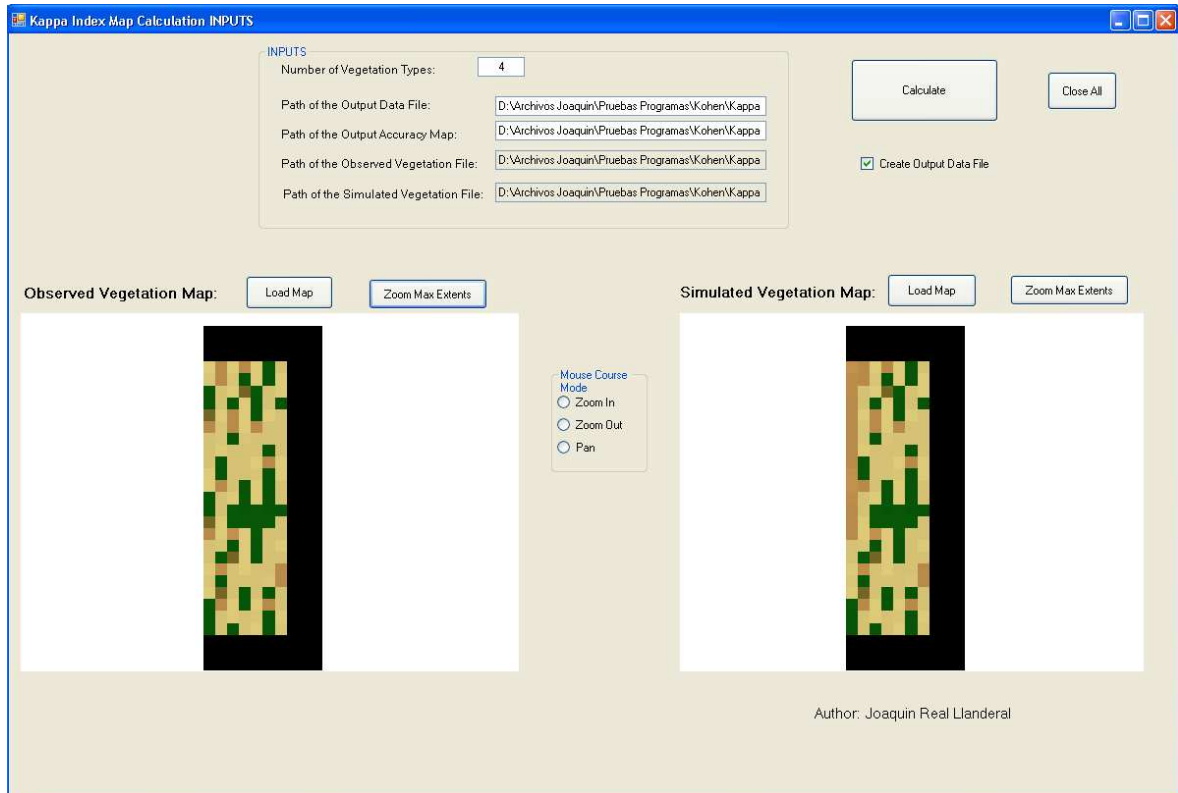


En este anexo se presenta el código fuente del programa que calcula el coeficiente Kappa de Cohen entre los mapas de vegetación observada y vegetación simulada. Este programa tiene 2 archivos de formulario de windows, un archivo de modulo y un archivo de clase. Los archivos del código de este programa son los siguientes:

1. Datos de Entrada Generales: *"frmInputs.vb"*
2. Resultados de Salida: *"frmOuputs.vb"*
3. Módulo General: *"GeneralModule.vb"*
4. Clase de Mapas: *"Map Class"*

El código fuente de este programa se muestra a continuación distribuido entre sus distintos archivos y formularios:

## 1. Datos de Entrada Generales: "frmInputs.vb"



```
Public Class frmInputs
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
        Dim DefaultFilepath As String = System.Environment.CurrentDirectory + "\Data\Default\DefaultInputs.asc"  
        ConfigDirectory = System.Environment.CurrentDirectory
```

```
        Dim Lines() As String  
        Lines = IO.File.ReadAllLines(DefaultFilepath)  
        txtNumVegTypes.Text = Lines(0)  
        txtObsVegPath.Text = System.Environment.CurrentDirectory +  
Lines(1)  
        txtSimVegPath.Text = System.Environment.CurrentDirectory +  
Lines(2)  
        txtOuputFilePath.Text = System.Environment.CurrentDirectory +  
Lines(3)  
        txtOutputAccuracyMap.Text = System.Environment.CurrentDirectory +  
Lines(4)
```

```

        DefaultLoadMap(txtObsVegPath.Text, ObsVegMap)
        DefaultLoadMap(txtSimVegPath.Text, SimVegMap)

    End Sub

    Private Sub btCalculate_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btCalculate.Click
        If frmOutputs.dtgMatrix.RowCount <> 0 Then
            MsgBox("Please Restart the Program to Carry out Further
Calculations")
            Exit Sub
        End If
        frmOutputs.Show()
    End Sub

    Private Sub rdZoomIn_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rdZoomIn.CheckedChanged
        ObsVegMap.CursorMode = MapWinGIS.tkCursorMode.cmZoomIn
        SimVegMap.CursorMode = MapWinGIS.tkCursorMode.cmZoomIn
    End Sub

    Private Sub rdPan_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles rdPan.CheckedChanged
        ObsVegMap.CursorMode = MapWinGIS.tkCursorMode.cmPan
        SimVegMap.CursorMode = MapWinGIS.tkCursorMode.cmPan
    End Sub

    Private Sub rdZoomOut_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rdZoomOut.CheckedChanged
        ObsVegMap.CursorMode = MapWinGIS.tkCursorMode.cmZoomOut
        SimVegMap.CursorMode = MapWinGIS.tkCursorMode.cmZoomOut
    End Sub

    Private Sub dtZoomMaxExtents_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles dtOBSZoomMaxExtents.Click
        ObsVegMap.ZoomToMaxExtents()
    End Sub

    Private Sub btAddData_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btObsMapAdd.Click
        txtObsVegPath.Text = LoadMap(ObsVegMap)
    End Sub

    Private Sub btSimMapAdd_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btSimMapAdd.Click
        txtSimVegPath.Text = LoadMap(SimVegMap)
    End Sub

```

```

Public Sub DefaultLoadMap(ByVal mapPath As String, ByVal map As
AxMapWinGIS.AxMap)

    'Create instances of the possible data objects,
    'to retrieve the dialog filters of supported formats.
    Dim sf As New MapWinGIS.Shapefile
    Dim grd As New MapWinGIS.Grid
    Dim img As New MapWinGIS.Image

    'We want users to be able to open any file type.
    'Merge the dialog filters by placing a pipe character
    'in between each:
    'First, determine what kind it was:
    Dim extension As String = _
    IO.Path.GetExtension(mapPath).ToLower()
    If sf.CdlgFilter.ToLower().Contains(extension) Then
        'It's a shapefile
        sf.Open(mapPath)
        map.AddLayer(sf, True)
        'Zoom to all visible layers
        map.ZoomToMaxExtents()
        'Return dlg.FileName 'Done
    ElseIf grd.CdlgFilter.ToLower().Contains(extension) Then
        'NOTE: a .tif can be a GeoTIFF (a grid)
        'or an image. Check this, if the file
        'is a tif:
        If mapPath.ToLower().EndsWith(".tif") Then
            If Not map.IsTIFFGrid(mapPath) Then
                'It's an image, not a grid.
                'Open it as an image.
                img.Open(mapPath)
                map.AddLayer(img, True)
                'Zoom to all visible layers
                map.ZoomToMaxExtents()
                ' Return dlg.FileName 'Done
            End If
        End If
        'Open the grid:
        grd.Open(mapPath)
        'Define a coloring scheme to color this grid:
        Dim sch As New MapWinGIS.GridColorScheme
        'Use a predefined coloring scheme "Fall Leaves"
        sch.UsePredefined(grd.Minimum, grd.Maximum, _
        MapWinGIS.PredefinedColorScheme.FallLeaves)
        'Convert it to an image that can be displayed:
        Dim u As New MapWinGIS.Utils
        Dim gridimage As MapWinGIS.Image
        gridimage = u.GridToImage(grd, sch)
        'Add the generated image to the map:
        map.AddLayer(gridimage, True)
        'Zoom to all visible layers
        map.ZoomToMaxExtents()
        ' Return dlg.FileName 'Done
    ElseIf img.CdlgFilter.ToLower().Contains(extension) Then
        'It's a plain image

```

```

        img.Open(mapPath)
        map.AddLayer(img, True)
        'Zoom to all visible layers
        map.ZoomToMaxExtents()
        'Return dlg.FileName 'Done
    End If

End Sub

Public Function LoadMap(ByVal map As AxMapWinGIS.AxMap)

    'Create instances of the possible data objects,
    'to retrieve the dialog filters of supported formats.
    Dim sf As New MapWinGIS.Shapefile
    Dim grd As New MapWinGIS.Grid
    Dim img As New MapWinGIS.Image
    'We want users to be able to open any file type.
    'Merge the dialog filters by placing a pipe character
    'in between each:
    Dim dlg As New OpenFileDialog
    dlg.InitialDirectory = ConfigDirectory + "\Data\INPUT\"
    dlg.Filter = "All Files (*.*)|*.*|" & sf.CdlgFilter & _
    "|" & grd.CdlgFilter & "|" & img.CdlgFilter
    If dlg.ShowDialog() = Windows.Forms.DialogResult.OK Then
        'If the user didn't cancel the dialog, try to open the file.
        'First, determine what kind it was:
        Dim extension As String = _
        IO.Path.GetExtension(dlg.FileName).ToLower()
        If sf.CdlgFilter.ToLower().Contains(extension) Then
            'It's a shapefile
            sf.Open(dlg.FileName)
            map.AddLayer(sf, True)
            'Zoom to all visible layers
            map.ZoomToMaxExtents()
            'Return dlg.FileName 'Done
        ElseIf grd.CdlgFilter.ToLower().Contains(extension) Then
            'NOTE: a .tif can be a GeoTIFF (a grid)
            'or an image. Check this, if the file
            'is a tif:
            If dlg.FileName.ToLower().EndsWith(".tif") Then
                If Not map.IsTIFFGrid(dlg.FileName) Then
                    'It's an image, not a grid.
                    'Open it as an image.
                    img.Open(dlg.FileName)
                    map.AddLayer(img, True)
                    'Zoom to all visible layers
                    map.ZoomToMaxExtents()
                    ' Return dlg.FileName 'Done
                End If
            End If
            'Open the grid:
            grd.Open(dlg.FileName)
            'Define a coloring scheme to color this grid:

```

```

    Dim sch As New MapWinGIS.GridColorScheme
    'Use a predefined coloring scheme "Fall Leaves"
    sch.UsePredefined(grd.Minimum, grd.Maximum, _
    MapWinGIS.PredefinedColorScheme.FallLeaves)
    'Convert it to an image that can be displayed:
    Dim u As New MapWinGIS.Utills
    Dim gridimage As MapWinGIS.Image
    gridimage = u.GridToImage(grd, sch)
    'Add the generated image to the map:
    map.AddLayer(gridimage, True)
    'Zoom to all visible layers
    map.ZoomToMaxExtents()
    ' Return dlg.FileName 'Done
ElseIf img.CdlgFilter.ToLower().Contains(extension) Then
    'It's a plain image
    img.Open(dlg.FileName)
    map.AddLayer(img, True)
    'Zoom to all visible layers
    map.ZoomToMaxExtents()
    'Return dlg.FileName 'Done
End If
End If

Return dlg.FileName
End Function

```

```

Private Sub dtSIMZoomMaxExtents_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles dtSIMZoomMaxExtents.Click
    SimVegMap.ZoomToMaxExtents()
End Sub

```

```

Private Sub btCloseAll_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btCloseAll.Click
    Call CloseAll()
End Sub

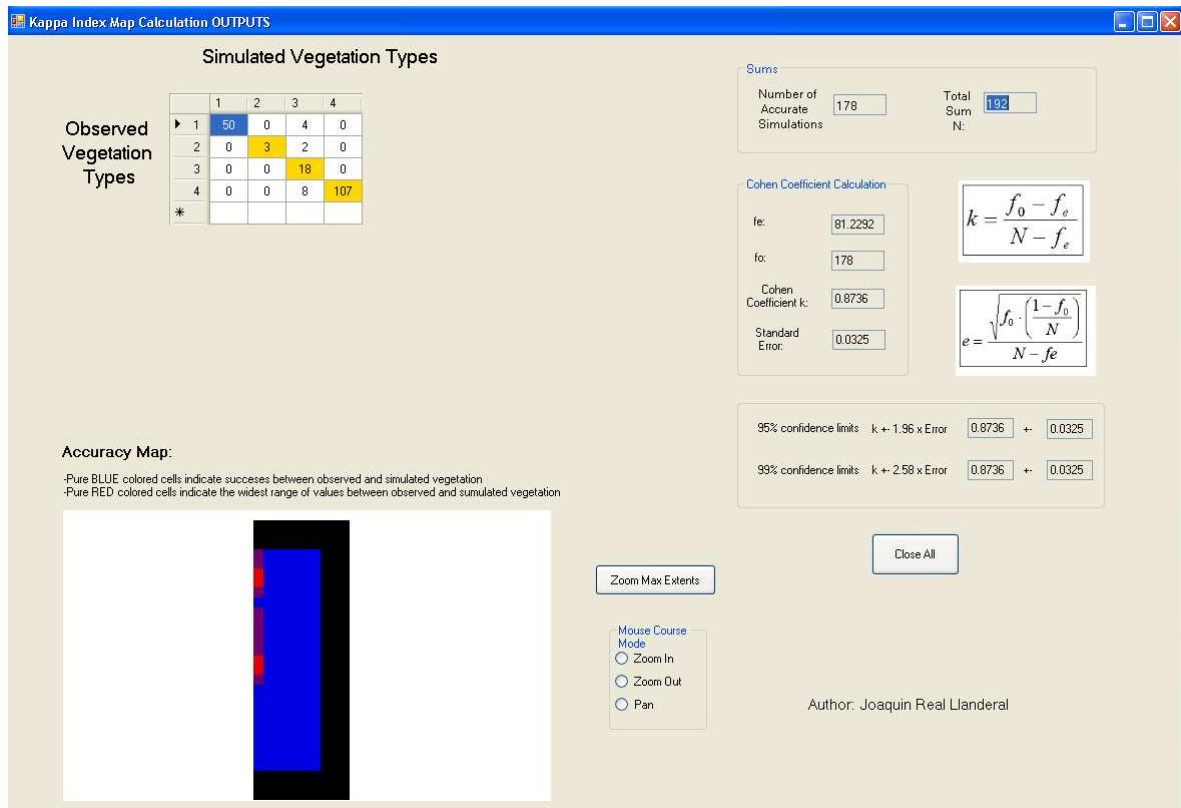
```

```

End Class

```

## 2. Resultados de Salida: "frmOutputs.vb"



```
Public Class frmOutputs
```

```
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
Call MainSub()  
Call CellsSums()  
Call ConfusionTableDraw()  
Call OutputTextboxes()
```

```
AccuracyMapCalculate(ObsVegMap, SimVegMap,  
ASCIIDataSt.NODATA_value)
```

```
If frmInputs.chkOutputFile.Checked = True Then  
Call OutputFile(frmInputs.txtOuputFilePath.Text)  
End If
```

```
End Sub
```

```
Private Sub OutputTextboxes()
```

```

txtDiagonalSum.Text = CStr(Math.Round((fo), 4))
txtTotalSum.Text = CStr(TotalSum)

txtfe.Text = CStr(Math.Round((fe), 4))
txtfo.Text = CStr(Math.Round((fo), 4))
txtK.Text = CStr(Math.Round((k), 4))
txtStdError.Text = CStr(Math.Round((StdError), 4))

txt2K.Text = CStr(Math.Round((k), 4))
txt3k.Text = CStr(Math.Round((k), 4))
txt2e.Text = CStr(Math.Round((StdError), 4))
txt3e.Text = CStr(Math.Round((StdError), 4))

End Sub

Private Sub ConfusionTableDraw()

    With dtgMatrix

        For i = 0 To NumVegTypes - 1
            .Columns.Add("SimVeg" + CStr(i + 1), CStr(i + 1))
            .Columns(i).DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter
        Next

        dtgMatrix.RowCount = NumVegTypes + 1

        For i = 0 To NumVegTypes - 1
            For j = 0 To NumVegTypes - 1
                .Rows.Item(i).Cells.Item(j).Value() =
ConfussionMatrix(i, j)
                'Only for esthetics the cells of the diagonal have a
different font
                If i = j Then
                    .Rows.Item(i).Cells.Item(j).Style.BackColor =
Color.Gold
                End If
            Next
        Next

        .RowHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter
        .ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter

        .AutoSizeColumns(DataGridViewAutoSizeColumnsMode.AllCells)

        For j = 0 To NumVegTypes - 1
            .Rows(j).HeaderCell.Value = CStr(j + 1)
        Next

    End With

```



```

End Sub

Private Sub AccuracyMapCalculate(ByVal ObsVegMap(,) As Single, ByVal
SimVegMap(,) As Single, ByVal NODATA_value As Single)

    Dim MapC As New MapClass 'Class which has the subtract map method
    Dim AccuracyMap(,) As Single 'Accuracy map array

    AccuracyMap = MapC.SubtractMaps(ObsVegMap, SimVegMap,
NODATA_value)

    OutAsciiMap(AccuracyMap, frmInputs.txtOutputAccuracyMap.Text)
    LoadAccuracyMap(AccuracyMapGrid,
frmInputs.txtOutputAccuracyMap.Text)

End Sub

Private Sub LoadAccuracyMap(ByVal MapGrid As AxMapWinGIS.AxMap, ByVal
ASCIIFilePath As String)

    'Create instances of the possible data objects,
    'to retrieve the dialog filters of supported formats.
    Dim sf As New MapWinGIS.Shapefile
    Dim grd As New MapWinGIS.Grid
    Dim img As New MapWinGIS.Image
    'We want users to be able to open any file type.
    ' Merge the dialog filters by placing a pipe character
    ' in between each:

    'First, determine what kind it was:

    Dim extension As String = _
IO.Path.GetExtension(ASCIIFilePath).ToLower()
    If sf.CdlgFilter.ToLower().Contains(extension) Then
        'It's a shapefile
        sf.Open(ASCIIFilePath)
        MapGrid.AddLayer(sf, True)
        'Zoom to all visible layers
        MapGrid.ZoomToMaxExtents()
        'Return dlg.FileName 'Done
    ElseIf grd.CdlgFilter.ToLower().Contains(extension) Then
        'NOTE: a .tif can be a GeoTIFF (a grid)
        'or an image. Check this, if the file
        'is a tif:
        If ASCIIFilePath.EndsWith(".tif") Then
            If Not MapGrid.IsTIFFGrid(ASCIIFilePath) Then
                'It's an image, not a grid.
                'Open it as an image.
                img.Open(ASCIIFilePath)
                MapGrid.AddLayer(img, True)
            End If
        End If
    End If
End Sub

```

```

        'Zoom to all visible layers
        MapGrid.ZoomToMaxExtents()
        ' Return dlg.FileName 'Done
    End If
End If
'Open the grid:
grd.Open(ASCIIFilePath)
'Define a coloring scheme to color this grid:
Dim sch As New MapWinGIS.GridColorScheme
Dim break As New MapWinGIS.GridColorBreak
Dim break2 As New MapWinGIS.GridColorBreak

'Set the color break high value to represent the grid's
maximum value
break.HighValue = grd.Maximum
'Set the color break low value to represent the grid's
minimum value
break.LowValue = grd.Minimum
'Set the color break high color to red
break.HighColor = Convert.ToInt32(RGB(255, 0, 0))
'Set the color break low color to blue
break.LowColor = Convert.ToInt32(RGB(0, 0, 255))
'insert color break into color scheme
sch.InsertBreak(break)

'Convert it to an image that can be displayed:
Dim u As New MapWinGIS.Utils
Dim gridimage As MapWinGIS.Image
gridimage = u.GridToImage(grd, sch)
'Add the generated image to the map:
MapGrid.AddLayer(gridimage, True)
'Zoom to all visible layers

MapGrid.ZoomToMaxExtents()
' Return dlg.FileName 'Done
ElseIf img.CdlgFilter.ToLower().Contains(extension) Then
'It's a plain image
img.Open(ASCIIFilePath)
MapGrid.AddLayer(img, True)
'Zoom to all visible layers
MapGrid.ZoomToMaxExtents()
'Return dlg.FileName 'Done
End If

End Sub

Private Sub GridLowColor()
Dim break As New MapWinGIS.GridColorBreak
Dim low As Double
'Get the low value in the color break
low = break.LowValue
'Set the low value in the color break

```

```

        break.LowValue = 1000
    End Sub

    Private Sub btCloseAll_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles btCloseAll.Click
        Call CloseAll()
    End Sub

    Private Sub dtOBSZoomMaxExtents_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles dtOBSZoomMaxExtents.Click
        AccuracyMapGrid.ZoomToMaxExtents()
    End Sub

    Private Sub rdZoomIn_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rdZoomIn.CheckedChanged
        AccuracyMapGrid.CursorMode = MapWinGIS.tkCursorMode.cmZoomIn
    End Sub

    Private Sub rdPan_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles rdPan.CheckedChanged
        AccuracyMapGrid.CursorMode = MapWinGIS.tkCursorMode.cmPan
    End Sub

    Private Sub rdZoomOut_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles rdZoomOut.CheckedChanged
        AccuracyMapGrid.CursorMode = MapWinGIS.tkCursorMode.cmZoomOut
    End Sub

End Class

```

### 3. Módulo General: "GeneralModule.vb"

```
Imports System.Text

Module GeneralModule

    Structure ASCIIIData
        Friend ncols As Integer
        Friend nrows As Integer
        Friend xllcorner As Single
        Friend yllcorner As Single
        Friend cellsize As Single
        Friend NODATA_value As Single
        Friend NumberArray(,) As Single
    End Structure

    Friend ConfigDirectory As String 'system.environment.currentdirectory
    WHEN the program is loaded
    Friend ASCIIIDataSt As New ASCIIIData
    Dim ExtractedNumber As Single 'number extracted from the ascii file
    header
    Public NumberArray(,) As Single 'matrix returned from the ascii map
    files

    Friend ConfussionMatrix(,) As Single
    Friend NumVegTypes As Integer = CInt(frmInputs.txtNumVegTypes.Text)
    'Number of different vegetation types

    Friend TotalSum As Single 'sum of the values of ALL the TABLE
    Friend fo As Double 'sum of the values of the main diagonal which is
    the number of "aciertos"
    Friend fe As Single 'fe is the sum of the elements of fe(i)
    Friend k As Single 'Cohen Coefficient
    Friend StdError As Single 'Standard error

    Friend ObsVegMap(,) As Single 'Ascii Observed Vegetation File to be
    loaded
    Friend SimVegMap(,) As Single 'Ascii Simulated Vegetation File to be
    loaded

    Friend Sub MainSub()

        Dim ObsVegPath As String = frmInputs.txtObsVegPath.Text
        'System.Environment.CurrentDirectory +
        "/Data/INPUT/ObservedVegetationMap.asc"
        Dim SimVegPath As String = frmInputs.txtSimVegPath.Text
        'System.Environment.CurrentDirectory +
        "/Data/INPUT/SimulatedVegetationMap.asc"

        ObsVegMap = LoadSingleAscii(ObsVegPath, 6)
        SimVegMap = LoadSingleAscii(SimVegPath, 6)
    End Sub
End Module
```

```

        'We check if the Observed Vegetation Map and Simulated Vegetation
Map differ in the Number of ROWS and COLUMNS
        If ObsVegMap.GetLength(0) <> SimVegMap.GetLength(0) Then
            MsgBox("Observed Vegetation Map and Simulated Vegetation Map
differ in the Number of ROWS. Please check input data.")
        ElseIf ObsVegMap.GetLength(1) <> SimVegMap.GetLength(1) Then
            MsgBox("Observed Vegetation Map and Simulated Vegetation Map
differ in the Number of COLUMNS. Please check input data.")
        Else
            ReDim ConfussionMatrix(NumVegTypes - 1, NumVegTypes - 1)
            For i = 0 To ObsVegMap.GetLength(0) - 1
                For j = 0 To ObsVegMap.GetLength(1) - 1
                    'we first test if hte value of the cell is not a
NODATA_value
                    If ObsVegMap(i, j) <> ASCIIDataSt.NODATA_value Then
                        ConfussionMatrix(ObsVegMap(i, j) - 1,
SimVegMap(i, j) - 1) = ConfussionMatrix(ObsVegMap(i, j) - 1, SimVegMap(i,
j) - 1) + 1
                    End If
                Next
            Next
        End If
    End Sub

Friend Sub CellsSums()

    Dim ColumnSum(NumVegTypes - 1) As Single 'sum of the values of
each COLUMN
    Dim RowSum(NumVegTypes - 1) As Single 'sum of the values of each
ROW

    'First we calculate ColumnSum and DiagonalSum
    For j = 0 To NumVegTypes - 1
        For i = 0 To NumVegTypes - 1
            ColumnSum(j) = ColumnSum(j) + ConfussionMatrix(i, j)
            If i = j Then
                fo = fo + ConfussionMatrix(i, j)
            End If
        Next
    Next

    'Then we calculate the RowSum
    For i = 0 To NumVegTypes - 1
        For j = 0 To NumVegTypes - 1
            RowSum(i) = RowSum(i) + ConfussionMatrix(i, j)
        Next
    Next

    'Lastly we calculate the TotalSum
    TotalSum = (ColumnSum.Sum)

    Call CohenCalculation(ColumnSum, RowSum, TotalSum)

```

End Sub

```
Friend Function LoadSingleAscii(ByVal file As String, ByVal
ASCIISartline As Integer)

    Dim Lines() As String
    Lines = IO.File.ReadAllLines(file)

    '*****Extracts the number of columns from the ASCII file
    ASCIIDataSt.ncols = ExtractNumber(Lines(0))
    ASCIIDataSt.nrows = ExtractNumber(Lines(1))
    ASCIIDataSt.xllcorner = ExtractNumber(Lines(2))
    ASCIIDataSt.yllcorner = ExtractNumber(Lines(3))
    ASCIIDataSt.cellsize = ExtractNumber(Lines(4))
    ASCIIDataSt.NODATA_value = ExtractNumber(Lines(5))

    Dim ASCIIColumnSeparator As String = " " '*****ASCII file column
separator
    Dim dataArray(Lines.Length - ASCIIStartline - 1,
ASCIIDataSt.ncols - 1) As String
    'In the following array the string values are stored as doubles
    ReDim NumberArray(Lines.Length - 1 - ASCIIStartline,
ASCIIDataSt.ncols - 1)

    'The values of the lines of the ASCII are split by the column
separator and stored in the matrix
    For i = ASCIIStartline To Lines.Length - 1
        Dim aux As String
        Dim cells() As String
        aux = Lines(i)
        cells = Split(aux, ASCIIColumnSeparator)
        For j = 0 To ASCIIDataSt.ncols - 1
            dataArray(i - ASCIIStartline, j) = cells(j)
        Next
    Next

    'In the following loop the string values are converted to doubles
    For i = 0 To Lines.Length - 1 - ASCIIStartline
        For j = 0 To ASCIIDataSt.ncols - 1
            NumberArray(i, j) = CSng(dataarray(i, j))
        Next
    Next

    Return NumberArray

End Function
```

```

Friend Function ExtractNumber(ByVal inputString As String)

    '***** This function extracts the numerical values from the
first line of the ASCII file *****

    ' Convert it to an array of characters
Dim myChars() As Char = inputString.ToCharArray()

Dim AuxString As String = "" 'auxiliary string to be used in the
loop
Dim ExtractedNumber As Single 'outputnumber

    ' Loop through the array testing if each is a digit or if it is a
minus sign
For Each ch As Char In myChars
    If Char.IsDigit(ch) Or InStr(1, ch, "-") Or InStr(1, ch, ".")
Then 'change dot to comma if necessary
        AuxString = AuxString + ch
    End If
Next

Try
    ExtractedNumber = CDb1(AuxString)
Catch ex As Exception
    'An error message would appear if the Extracted Number could
not be converted to integer
    MsgBox("The format of the NoData Value from the ASCII file is
not Valid", MessageBoxIcon.Exclamation)
End Try

Return ExtractedNumber

End Function

Public Function DefaultNumVegTypes(ByVal InputFile As String)
Dim Lines() As String
Lines = IO.File.ReadAllLines(InputFile)
Return CStr(Lines(0))
End Function

Public Sub CohenCalculation(ByVal ColumnSum() As Single, ByVal
RowSum() As Single, ByVal TotalSum As Single)

    'First we calculate the fe(i) for each vegetation type
Dim fei(NumVegTypes - 1) As Single
For i = 0 To NumVegTypes - 1
    fei(i) = (RowSum(i) * ColumnSum(i)) / TotalSum
Next
fe = fei.Sum

    'Now we calculate the Cohen Coefficient
'fo is the sum of elements of the main diagonal ("aciertos)

```

```

k = (fo - fe) / (TotalSum - fe)
StdError = ((fo * (1 - fo / TotalSum)) ^ (0.5)) / (TotalSum - fe)

End Sub

Friend Sub OutputFile(ByVal OutputFile As String)

    Dim objWriter As New System.IO.StreamWriter(OutputFile)

objWriter.WriteLine("*****")
objWriter.WriteLine("*Confusion Matrix and Cohen Coefficient
Calculation*")

objWriter.WriteLine("*****")
objWriter.WriteLine("")
objWriter.WriteLine("Simulated/Observed Vegetation Types")

Dim ColumnHeader As New StringBuilder

For i = 0 To NumVegTypes - 1
    ColumnHeader.Append("; " + "ObsVeg" + Str(i + 1))
Next
objWriter.WriteLine(ColumnHeader.ToString)

For i = 0 To NumVegTypes - 1
    Dim ConfussionMatrixStrings As New StringBuilder
    For j = 0 To NumVegTypes - 1
        ConfussionMatrixStrings.Append(CStr(ConfussionMatrix(i,
j).ToString + ";"))
    Next
    objWriter.WriteLine("SimVeg" + Str(i + 1) + ";" +
ConfussionMatrixStrings.ToString)
Next

objWriter.WriteLine("")
objWriter.WriteLine("Number of Vegetation Types: " +
NumVegTypes.ToString)
objWriter.WriteLine("Number of Accurate Simulations fe: " +
fe.ToString)
objWriter.WriteLine("Total Sum N:" + ";" + TotalSum.ToString)
objWriter.WriteLine("fo:" + ";" + fo.ToString)

objWriter.WriteLine("Cohen Coefficient k:" + " " +
k.ToString)
objWriter.WriteLine("Standard Error:" + " " +
StdError.ToString)
objWriter.WriteLine("95% confidence limits " + "(k +- 1.96 x
Error)" + k.ToString + " +- 1.96 x " + StdError.ToString)
objWriter.WriteLine("99% confidence limits " + "(k +- 2.58 x
Error)" + k.ToString + " +- 2.58 " + StdError.ToString)

```



```

        objWriter.Close()

        Process.Start("explorer.exe", ConfigDirectory + "\Data\OUTPUT\")

    End Sub

    Public Sub CloseAll()
        frmInputs.Close()
        frmOutputs.Close()
    End Sub

    Friend Sub OutAsciiMap(ByVal InputMatrix(,) As Single, ByVal
FileToSave As String)

        Dim objWriter As New System.IO.StreamWriter(FileToSave)

        'We write the first 6 lines which describe the mesh>>changing ","
to "."
        objWriter.WriteLine("ncols" + "      " +
ASCIIDataSt.ncols.ToString)
        objWriter.WriteLine("nrows" + "      " +
ASCIIDataSt.nrows.ToString)
        objWriter.WriteLine("xllcorner" + "      " +
ASCIIDataSt.xllcorner.ToString)
        objWriter.WriteLine("yllcorner" + "      " +
ASCIIDataSt.yllcorner.ToString)
        objWriter.WriteLine("cellsize" + "      " +
ASCIIDataSt.cellsize.ToString)
        objWriter.WriteLine("NODATA_value" + "      " +
ASCIIDataSt.NODATA_value.ToString)

        'We now pass to the file the data from the double array
Dim record(InputMatrix.GetLength(0) - 1) As String

        For i = 0 To InputMatrix.GetLength(0) - 1
            For j = 0 To InputMatrix.GetLength(1) - 1
                If j = 0 Then
                    record(i) = InputMatrix(i, j).ToString
                Else
                    record(i) = record(i) + " " + InputMatrix(i,
j).ToString
                End If
            Next j
            objWriter.WriteLine(record(i))
        Next i

        objWriter.Close()

    End Sub

End Module

```

#### 4. Clase de Mapas: "Map Class"

```
Public Class MapClass

    Public Shared Function SubtractMaps(ByVal MapA(,) As Single, ByVal
MapB(,) As Single, ByVal NODATA_value As Single)

        Dim MapAccuracy(MapA.GetLength(0) - 1, MapA.GetLength(1) - 1) As
Single

        If MapA.GetLength(0) <> MapB.GetLength(0) Or MapA.GetLength(0) <>
MapB.GetLength(1) Then
            MsgBox("Observed vegetation map and Simulated vegetation map
should have the same dimensions.")
        Else
            For i = 0 To MapA.GetLength(0) - 1
                For j = 0 To MapA.GetLength(1) - 1
                    If MapA(i, j) = NODATA_value Or MapB(i, j) =
NODATA_value Then
                        MapAccuracy(i, j) = NODATA_value
                    Else
                        MapAccuracy(i, j) = Math.Abs(MapA(i, j) - MapB(i,
j))
                    End If
                Next
            Next
        End If

        Return MapAccuracy

    End Function

End Class
```

## **ANEXO X: TABLAS DE RESULTADOS DEL ANÁLISIS DE SENSIBILIDAD**

### Tablas de resultados para régimen natural de caudales (años 1942-1949)

En las zonas con Suelo 4 (limoso-arenoso) se obtienen las siguientes tablas para cada tipo funcional de vegetación:

<b>Vegetación:</b>	<b>RH (Herbáceas de Ribera)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Natural</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.754	0.310	0.00E+00	4.35E-02	0.00E+00	2.87E-02	0.00E+00	0.00E+00	Ze
237.497	7.004	0.311	0.00E+00	4.38E-02	0.00E+00	2.81E-02	0.00E+00	0.00E+00	Ze
236.747	6.254	0.313	0.00E+00	4.42E-02	0.00E+00	2.73E-02	0.00E+00	0.00E+00	Ze
235.997	5.504	0.314	0.00E+00	4.48E-02	0.00E+00	2.63E-02	0.00E+00	0.00E+00	Ze
235.248	4.755	0.317	0.00E+00	4.57E-02	0.00E+00	2.48E-02	0.00E+00	0.00E+00	Ze
234.498	4.005	0.321	0.00E+00	4.73E-02	0.00E+00	2.26E-02	0.00E+00	0.00E+00	Ze
233.747	3.254	0.328	0.00E+00	5.00E-02	0.00E+00	1.83E-02	0.00E+00	0.00E+00	Ze
232.997	2.504	0.343	2.18E-04	5.55E-02	0.00E+00	4.79E-03	0.00E+00	0.00E+00	Ze
232.247	1.754	0.417	1.59E-03	9.50E-02	1.79E-04	1.24E-03	0.00E+00	0.00E+00	Ze
232.116	1.623	0.434	1.34E-03	8.74E-02	1.79E-04	7.41E-03	0.00E+00	0.00E+00	Ze
231.967	1.474	0.454	1.80E-03	1.11E-01	1.79E-04	2.41E-02	0.00E+00	0.00E+00	Ze
231.819	1.326	0.487	2.43E-03	1.49E-01	2.25E-04	4.25E-03	0.00E+00	0.00E+00	Ze
230.952	0.459	0.999	7.45E-02	3.98E-01	0.00E+00	3.26E-01	0.00E+00	0.00E+00	Ze

<b>Vegetación:</b>	<b>RA (Arbórea de Ribera)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Natural</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.754	0.390	7.51E-02	3.13E-02	0.00E+00	8.58E-03	1.77E-05	1.77E-05	Zr
237.497	7.004	0.392	2.97E-01	3.14E-02	0.00E+00	6.85E-03	1.06E-04	7.90E-05	Zr
236.747	6.254	0.397	3.52E-01	3.40E-02	0.00E+00	1.98E-03	1.56E-03	1.52E-03	Zr
235.997	5.504	0.682	4.01E-01	3.72E-02	0.00E+00	1.40E-01	9.98E-02	7.75E-02	Zr
235.248	4.755	0.762	4.39E-01	5.34E-02	0.00E+00	1.11E-01	1.32E-01	7.11E-02	Zr
234.498	4.005	0.832	3.08E-01	6.24E-02	0.00E+00	6.85E-02	1.47E-01	5.86E-02	Zr
233.747	3.254	0.893	1.25E-01	7.61E-02	0.00E+00	2.79E-02	1.49E-01	5.05E-02	Rj
232.997	2.504	0.953	5.42E-02	8.20E-02	6.00E-06	2.81E-03	1.16E-01	3.36E-02	Rj
232.247	1.754	1.000	8.07E-05	2.83E-02	2.11E-05	9.26E-05	2.03E-04	0.00E+00	Ze
232.116	1.623	0.999	1.18E-04	1.71E-02	2.33E-05	5.93E-05	2.92E-04	0.00E+00	Ze
231.967	1.474	0.999	1.30E-04	4.81E-03	2.58E-05	5.03E-05	4.10E-04	0.00E+00	Ze
231.819	1.326	0.999	3.25E-05	0.00E+00	1.45E-05	8.32E-05	3.71E-04	0.00E+00	Rj

230.952	0.459	0.989	1.10E-02	5.75E-04	4.59E-03	3.56E-02	1.10E-01	0.00E+00	Rj
---------	-------	-------	----------	----------	----------	----------	----------	----------	----

<b>Vegetación:</b>	<b>RJ (Juveniles de Ribera)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Natural</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.754	0.310	0.00E+00	4.35E-02	0.00E+00	2.87E-02	0.00E+00	0.00E+00	Ze
237.497	7.004	0.311	0.00E+00	4.38E-02	0.00E+00	2.81E-02	0.00E+00	0.00E+00	Ze
236.747	6.254	0.313	0.00E+00	4.42E-02	0.00E+00	2.73E-02	0.00E+00	0.00E+00	Ze
235.997	5.504	0.314	0.00E+00	4.48E-02	0.00E+00	2.63E-02	0.00E+00	0.00E+00	Ze
235.248	4.755	0.317	0.00E+00	4.57E-02	0.00E+00	2.48E-02	0.00E+00	0.00E+00	Ze
234.498	4.005	0.321	0.00E+00	4.73E-02	0.00E+00	2.25E-02	0.00E+00	0.00E+00	Ze
233.747	3.254	0.328	0.00E+00	5.00E-02	0.00E+00	1.81E-02	0.00E+00	0.00E+00	Ze
232.997	2.504	0.342	3.19E-05	5.60E-02	0.00E+00	4.28E-03	2.99E-05	0.00E+00	Ze
232.247	1.754	0.417	1.46E-01	9.73E-02	0.00E+00	1.45E-02	4.95E-04	3.76E-04	Zr
232.116	1.623	0.442	1.60E-01	7.68E-02	0.00E+00	2.66E-02	4.80E-03	4.52E-03	Zr
231.967	1.474	0.505	1.66E-01	9.40E-02	0.00E+00	1.46E-03	2.22E-02	2.00E-02	Zr
231.819	1.326	0.592	1.75E-01	1.07E-01	0.00E+00	2.49E-02	4.69E-02	3.95E-02	Zr
230.952	0.459	0.975	4.30E-03	3.83E-02	0.00E+00	1.51E-01	1.77E-02	0.00E+00	Ri

<b>Vegetación:</b>	<b>TV (Vegetación Terrestre)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Natural</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.754	0.3896	0.00E+00	3.12E-02	0.00E+00	8.70E-03	0.00E+00	0.00E+00	Ze
237.497	7.004	0.3913	0.00E+00	3.12E-02	0.00E+00	6.92E-03	0.00E+00	0.00E+00	Ze
236.747	6.254	0.3938	0.00E+00	3.13E-02	0.00E+00	4.35E-03	0.00E+00	0.00E+00	Ze
235.997	5.504	0.3975	0.00E+00	3.18E-02	0.00E+00	4.30E-04	0.00E+00	0.00E+00	Ze
235.248	4.755	0.4039	2.78E-06	3.42E-02	0.00E+00	5.08E-03	0.00E+00	0.00E+00	Ze
234.498	4.005	0.4208	2.57E-05	5.43E-02	0.00E+00	1.26E-02	0.00E+00	0.00E+00	Ze
233.747	3.254	0.4546	2.98E-04	1.46E-01	4.95E-06	3.66E-02	6.92E-05	1.20E-05	Ze
232.997	2.504	0.5649	1.97E-02	3.65E-01	8.68E-05	4.95E-04	1.33E-04	7.98E-06	Ze
232.247	1.754	0.9985	4.70E-03	3.49E-01	1.21E-04	2.14E-01	9.97E-04	0.00E+00	Ze
232.116	1.623	0.9978	8.78E-03	2.58E-01	4.37E-04	2.49E-01	1.45E-03	0.00E+00	Ze
231.967	1.474	0.9961	3.15E-02	9.39E-02	3.82E-04	2.31E-01	4.21E-03	0.00E+00	Ri
231.819	1.326	0.9879	1.01E-01	1.93E-03	1.86E-03	2.33E-01	1.97E-02	0.00E+00	Ri

230.952	0.459	0.6142	2.19E-02	3.52E-01	2.89E-02	1.99E-01	1.70E-01	0.00E+00	Ze
---------	-------	--------	----------	----------	----------	----------	----------	----------	----

En las zonas con Suelo 10 (Arenoso con muchas Gravas) se obtienen las siguientes tablas para cada tipo funcional de vegetación:

<b>Vegetación:</b>	<b>RH (Herbáceas de Ribera)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>
<b>Régimen de Q:</b>	<b>Natural</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
230.606	0.113	0.998	7.36E-01	3.89E-02	3.10E-03	3.30E-04	0.00E+00	0.00E+00	Zr
231.631	1.138	0.280	2.60E-03	2.98E-02	2.19E-04	2.19E-02	0.00E+00	0.00E+00	Ze
232.098	1.605	0.279	1.40E-03	3.33E-02	1.79E-04	2.42E-02	0.00E+00	0.00E+00	Ze
232.953	2.46	0.279	2.44E-04	3.53E-02	0.00E+00	2.48E-02	0.00E+00	0.00E+00	Ze
233.532	3.039	0.279	0.00E+00	3.59E-02	0.00E+00	2.50E-02	0.00E+00	0.00E+00	Ze
233.788	3.295	0.279	0.00E+00	3.59E-02	0.00E+00	2.50E-02	0.00E+00	0.00E+00	Ze
234.693	4.2	0.279	0.00E+00	3.59E-02	0.00E+00	2.50E-02	0.00E+00	0.00E+00	Ze

<b>Vegetación:</b>	<b>RA (Arbórea de Ribera)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>
<b>Régimen de Q:</b>	<b>Natural</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
230.606	0.113	0.900	1.08E-01	3.13E-02	3.67E-02	8.99E-02	2.86E-01	0.00E+00	Rj
231.631	1.138	0.999	1.14E-04	1.89E-01	2.25E-05	3.76E-04	1.17E-03	0.00E+00	Ze
232.098	1.605	0.824	2.03E-02	2.49E-01	2.34E-05	3.72E-02	1.86E-01	3.03E-03	Rj
232.953	2.46	0.732	9.88E-02	1.10E-01	6.18E-06	4.19E-02	1.90E-01	5.17E-03	Rj
233.532	3.039	0.675	1.60E-01	1.07E-01	0.00E+00	4.70E-02	1.70E-01	6.41E-03	Rj
233.788	3.295	0.650	1.87E-01	1.05E-01	0.00E+00	4.69E-02	1.59E-01	6.77E-03	Zr
234.693	4.2	0.557	2.70E-01	8.56E-02	0.00E+00	3.54E-02	1.13E-01	7.84E-03	Zr

<b>Vegetación:</b>	<b>RJ (Juveniles de Ribera)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>

Régimen de Q: Natural

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
230.606	0.113	0.667	4.32E-02	3.56E-02	0.00E+00	2.08E-01	1.21E-01	0.00E+00	Ri
231.631	1.138	0.340	9.30E-02	5.14E-02	0.00E+00	6.55E-03	3.61E-02	2.46E-03	Zr
232.098	1.605	0.283	4.54E-02	3.69E-02	0.00E+00	1.06E-02	2.82E-03	1.11E-03	Zr
232.953	2.46	0.279	9.52E-05	3.59E-02	0.00E+00	2.43E-02	3.07E-05	0.00E+00	Ze
233.532	3.039	0.279	0.00E+00	3.59E-02	0.00E+00	2.46E-02	0.00E+00	0.00E+00	Ze
233.788	3.295	0.279	0.00E+00	3.59E-02	0.00E+00	2.49E-02	0.00E+00	0.00E+00	Ze
234.693	4.2	0.279	0.00E+00	3.59E-02	0.00E+00	2.50E-02	0.00E+00	0.00E+00	Ze

Vegetación: TV (Vegetación Terrestre)  
Suelo: 10 (Arenoso y Gravoso)  
Régimen de Q: Natural

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
230.606	0.113	0.3679	5.14E-01	4.93E-01	6.67E-02	5.93E-02	1.61E-01	0.00E+00	Zr
231.631	1.138	0.9501	2.76E-01	5.14E-01	6.42E-03	2.74E-01	6.08E-02	0.00E+00	Ze
232.098	1.605	0.4860	6.74E-02	6.40E-01	4.46E-04	2.69E-02	2.30E-02	9.92E-04	Ze
232.953	2.46	0.3655	7.16E-03	6.82E-02	1.89E-05	3.46E-02	1.57E-04	5.41E-06	Ze
233.532	3.039	0.3608	3.30E-04	6.20E-02	7.76E-06	2.61E-02	5.91E-05	4.47E-06	Ze
233.788	3.295	0.3608	1.75E-04	6.19E-02	3.50E-06	2.61E-02	3.36E-05	4.47E-06	Ze
234.693	4.2	0.3607	0.00E+00	5.10E-02	0.00E+00	2.53E-02	0.00E+00	0.00E+00	Ze

## Tablas de resultados para régimen alterado de caudales (años 2000-2006)

En las zonas con Suelo 4 (limoso-arenoso) se obtienen las siguientes tablas para cada tipo funcional de vegetación:

<b>Vegetación:</b>	<b>RH (Herbáceas de Ribera)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.955	0.374	0.00E+00	5.56E-02	0.00E+00	7.85E-02	0.00E+00	0.00E+00	Ri
237.497	7.205	0.375	0.00E+00	5.58E-02	0.00E+00	7.84E-02	0.00E+00	0.00E+00	Ri
236.747	6.455	0.377	0.00E+00	5.61E-02	0.00E+00	7.82E-02	0.00E+00	0.00E+00	Ri
235.997	5.705	0.379	0.00E+00	5.65E-02	0.00E+00	7.80E-02	0.00E+00	0.00E+00	Ri
235.248	4.956	0.382	0.00E+00	5.72E-02	0.00E+00	7.76E-02	0.00E+00	0.00E+00	Ri
234.498	4.206	0.387	0.00E+00	5.85E-02	0.00E+00	7.70E-02	0.00E+00	0.00E+00	Ri
233.747	3.455	0.395	0.00E+00	6.07E-02	0.00E+00	7.70E-02	0.00E+00	0.00E+00	Ri
232.997	2.705	0.412	0.00E+00	6.44E-02	0.00E+00	8.00E-02	0.00E+00	0.00E+00	Ri
232.247	1.955	0.442	8.23E-05	7.53E-02	0.00E+00	9.57E-02	0.00E+00	0.00E+00	Ri
232.116	1.824	0.452	1.52E-04	8.03E-02	0.00E+00	9.31E-02	0.00E+00	0.00E+00	Ri
231.967	1.675	0.465	2.62E-04	8.47E-02	0.00E+00	8.27E-02	0.00E+00	0.00E+00	Ze
231.819	1.527	0.481	4.52E-04	9.87E-02	0.00E+00	6.18E-02	0.00E+00	0.00E+00	Ze
230.952	0.66	1.000	8.45E-03	1.09E-01	2.79E-04	4.71E-01	0.00E+00	0.00E+00	Ri

<b>Vegetación:</b>	<b>RA (Arbórea de Ribera)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.955	0.504	7.46E-03	9.26E-02	0.00E+00	9.90E-02	0.00E+00	0.00E+00	Ri
237.497	7.205	0.506	2.23E-01	9.37E-02	0.00E+00	9.84E-02	0.00E+00	0.00E+00	Zr
236.747	6.455	0.509	2.66E-01	9.54E-02	0.00E+00	9.66E-02	1.97E-06	1.97E-06	Zr
235.997	5.705	0.723	3.05E-01	5.32E-02	0.00E+00	1.66E-01	7.62E-02	5.99E-02	Zr
235.248	4.956	0.787	3.40E-01	6.03E-02	0.00E+00	1.28E-01	1.08E-01	5.71E-02	Zr
234.498	4.206	0.849	2.36E-01	6.84E-02	0.00E+00	8.24E-02	1.31E-01	5.30E-02	Zr
233.747	3.455	0.905	1.16E-01	7.62E-02	0.00E+00	3.24E-02	1.38E-01	4.56E-02	Rj



232.997	2.705	0.959	4.97E-02	7.83E-02	0.00E+00	4.57E-03	1.11E-01	3.25E-02	Rj
232.247	1.955	1.000	1.20E-05	2.82E-02	2.76E-06	1.45E-05	5.20E-05	0.00E+00	Ze
232.116	1.824	1.000	2.17E-05	1.38E-02	5.82E-06	2.07E-05	8.12E-05	0.00E+00	Ze
231.967	1.675	1.000	2.51E-05	2.71E-03	4.91E-06	3.15E-05	1.07E-04	0.00E+00	Ze
231.819	1.527	1.000	3.72E-05	1.01E-05	9.65E-06	4.75E-05	1.51E-04	0.00E+00	Rj
230.952	0.66	0.998	1.02E-03	1.57E-04	5.14E-04	1.56E-02	6.78E-02	0.00E+00	Rj

<b>Vegetación:</b>	<b>RJ (Juveniles de Ribera)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.955	0.374	0.00E+00	5.56E-02	0.00E+00	7.85E-02	0.00E+00	0.00E+00	Ri
237.497	7.205	0.375	0.00E+00	5.58E-02	0.00E+00	7.84E-02	0.00E+00	0.00E+00	Ri
236.747	6.455	0.377	0.00E+00	5.61E-02	0.00E+00	7.82E-02	0.00E+00	0.00E+00	Ri
235.997	5.705	0.379	0.00E+00	5.65E-02	0.00E+00	7.80E-02	0.00E+00	0.00E+00	Ri
235.248	4.956	0.382	0.00E+00	5.72E-02	0.00E+00	7.76E-02	0.00E+00	0.00E+00	Ri
234.498	4.206	0.387	0.00E+00	5.85E-02	0.00E+00	7.70E-02	0.00E+00	0.00E+00	Ri
233.747	3.455	0.395	0.00E+00	6.07E-02	0.00E+00	7.70E-02	0.00E+00	0.00E+00	Ri
232.997	2.705	0.412	9.25E-07	6.44E-02	0.00E+00	7.98E-02	0.00E+00	0.00E+00	Ri
232.247	1.955	0.442	1.42E-01	7.55E-02	0.00E+00	9.37E-02	1.38E-05	0.00E+00	Zr
232.116	1.824	0.452	1.55E-01	8.06E-02	0.00E+00	8.84E-02	4.66E-05	2.60E-05	Zr
231.967	1.675	0.514	1.64E-01	7.94E-02	0.00E+00	8.04E-02	2.02E-02	1.93E-02	Zr
231.819	1.527	0.609	1.72E-01	7.39E-02	1.25E-04	8.16E-02	5.31E-02	4.55E-02	Zr
230.952	0.66	0.997	3.89E-04	2.90E-03	0.00E+00	1.30E-01	1.39E-03	0.00E+00	Ri

<b>Vegetación:</b>	<b>TV (Vegetación Terrestre)</b>
<b>Suelo:</b>	<b>4 (Limo-Arenoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
238.247	7.955	0.504	0.00E+00	9.26E-02	0.00E+00	9.90E-02	0.00E+00	0.00E+00	Ri
237.497	7.205	0.506	0.00E+00	9.37E-02	0.00E+00	9.85E-02	0.00E+00	0.00E+00	Ri
236.747	6.455	0.509	0.00E+00	9.54E-02	0.00E+00	9.77E-02	0.00E+00	0.00E+00	Ri
235.997	5.705	0.514	0.00E+00	9.82E-02	0.00E+00	9.62E-02	0.00E+00	0.00E+00	Ze
235.248	4.956	0.521	0.00E+00	1.04E-01	0.00E+00	9.36E-02	0.00E+00	0.00E+00	Ze
234.498	4.206	0.534	0.00E+00	1.17E-01	0.00E+00	8.99E-02	0.00E+00	0.00E+00	Ze

233.747	3.455	0.561	0.00E+00	1.61E-01	0.00E+00	8.58E-02	0.00E+00	0.00E+00	Ze
232.997	2.705	0.629	8.80E-03	4.31E-01	2.46E-06	5.77E-02	2.07E-05	0.00E+00	Ze
232.247	1.955	1.000	1.19E-03	3.31E-01	4.15E-05	3.46E-01	2.65E-04	3.86E-05	Ri
232.116	1.824	1.000	1.55E-03	2.48E-01	3.16E-05	2.76E-01	2.58E-04	0.00E+00	Ri
231.967	1.675	0.999	3.47E-03	5.73E-02	6.98E-05	2.35E-01	6.05E-04	0.00E+00	Ri
231.819	1.527	0.999	2.74E-02	6.38E-04	1.64E-04	2.19E-01	1.82E-03	0.00E+00	Ri
230.952	0.66	0.663	9.24E-01	2.77E-01	2.61E-02	2.36E-01	1.62E-01	0.00E+00	Zr

En las zonas con Suelo 10 (Arenoso con muchas Gravas) se obtienen las siguientes tablas para cada tipo funcional de vegetación:

<b>Vegetación:</b>	<b>RH (Herbáceas de Ribera)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado					Parámetro más Sensible	
			Zr	Ze	Zsat	Ri	Rj		CRT
230.606	0.314	0.998	4.64E-01	1.33E+00	5.80E-04	7.09E-03	6.20E-04	7.53E-05	Ze
231.631	1.339	0.337	8.21E-04	5.64E-02	4.72E-05	6.47E-02	2.11E-07	2.11E-07	Ri
232.098	1.806	0.337	1.63E-04	5.77E-02	0.00E+00	6.53E-02	0.00E+00	0.00E+00	Ri
232.953	2.661	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri
233.532	3.24	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri
233.788	3.496	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri
234.693	4.401	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri

<b>Vegetación:</b>	<b>RA (Arbórea de Ribera)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado					Parámetro más Sensible	
			Zr	Ze	Zsat	Ri	Rj		CRT
230.606	0.314	0.951	8.03E-02	8.75E-03	2.36E-02	1.02E-01	2.49E-01	0.00E+00	Rj
231.631	1.339	1.000	5.84E-05	2.04E-01	1.33E-05	7.15E-05	2.39E-04	0.00E+00	Ze
232.098	1.806	0.817	1.83E-02	2.38E-01	4.14E-05	4.10E-02	1.81E-01	2.89E-03	Ze
232.953	2.661	0.752	8.55E-02	1.04E-01	0.00E+00	4.20E-02	1.60E-01	3.79E-03	Rj
233.532	3.24	0.706	1.37E-01	1.01E-01	0.00E+00	4.24E-02	1.41E-01	4.58E-03	Rj
233.788	3.496	0.685	1.61E-01	9.96E-02	0.00E+00	4.17E-02	1.32E-01	4.91E-03	Zr

234.693	4.401	0.606	2.34E-01	9.37E-02	0.00E+00	1.12E-02	9.22E-02	6.20E-03	Zr
---------	-------	-------	----------	----------	----------	----------	----------	----------	----

<b>Vegetación:</b>	<b>RJ (Juveniles de Ribera)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
230.606	0.314	0.822	3.49E-02	1.01E-01	0.00E+00	2.54E-01	1.14E-01	6.04E-05	Ri
231.631	1.339	0.383	8.42E-02	6.87E-02	0.00E+00	1.63E-02	2.75E-02	2.88E-03	Zr
232.098	1.806	0.337	3.50E-02	5.80E-02	0.00E+00	6.09E-02	1.02E-04	7.44E-05	Ri
232.953	2.661	0.336	1.40E-06	5.80E-02	0.00E+00	6.53E-02	0.00E+00	0.00E+00	Ri
233.532	3.24	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri
233.788	3.496	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri
234.693	4.401	0.336	0.00E+00	5.80E-02	0.00E+00	6.55E-02	0.00E+00	0.00E+00	Ri

<b>Vegetación:</b>	<b>TV (Vegetación Terrestre)</b>
<b>Suelo:</b>	<b>10 (Arenoso y Gravoso)</b>
<b>Régimen de Q:</b>	<b>Alterado</b>

Cota (m.s.n.m.)	Elevación sobre el Nivel del Río (m)	Valores Estándar del Índice de Evapotranspiración	Variación del Índice de Evapotranspiración según el Parámetro Modificado						Parámetro más Sensible
			Zr	Ze	Zsat	Ri	Rj	CRT	
230.606	0.314	0.481	1.28E+00	5.92E-01	3.88E-02	8.86E-02	2.00E-01	0.00E+00	Zr
231.631	1.339	0.984	1.81E-01	5.33E-01	3.51E-03	2.51E-01	3.78E-02	8.47E-06	Ze
232.098	1.806	0.482	6.09E-02	5.76E-01	4.40E-05	3.51E-02	2.08E-02	1.37E-03	Ze
232.953	2.661	0.444	1.54E-03	7.42E-02	2.72E-06	9.69E-02	2.22E-05	9.99E-08	Ri
233.532	3.24	0.443	7.01E-06	7.38E-02	0.00E+00	9.74E-02	5.72E-07	0.00E+00	Ri
233.788	3.496	0.443	0.00E+00	7.38E-02	0.00E+00	9.75E-02	0.00E+00	0.00E+00	Ri
234.693	4.401	0.443	0.00E+00	7.37E-02	0.00E+00	9.75E-02	0.00E+00	0.00E+00	Ri

## **ANEXO XI: TABLAS DE PARÁMETROS DE SUELO**

### **Tablas de parámetros de suelo del tramo de Lorcha (río Serpis):**

En la primera tabla de parámetros básicos de suelo, obsérvese que existen dos horizontes para el suelo 5 (5-A y 5-B). Los datos de estos horizontes han sido introducidos en el modelo del "Soil Water Characteristics" de forma separada, pero los resultados obtenidos, es decir los parámetros del modelo resultantes, han sido combinados de forma ponderada en parámetros únicos con respecto al grosor de cada horizonte.

Tabla III.l **Parámetros de suelo básicos del tramo de Lorcha (río Serpis):**

<b>Código Suelo</b>	<b>% Gravas &gt;2mm</b>	<b>% Arenas 0,05&lt;S &lt; 2mm</b>	<b>% Arcillas C&lt; 0,002</b>	<b>% Materia Orgánica</b>
1	1.1	42.2	4	0.44
2	0.1	36.6	9	0.98
3	0.6	47.2	10	1.46
4	0.1	30.6	15.9	1.96
5-A	1.6	46.3	9.3	4.4
5-B	65	78.6	7.5	2.89
6	57.6	51.9	13.9	2.7
7	58.2	73.9	8.4	2.13
8	0	56.6	9.8	1.87
9	56.3	86.7	6.9	0.53
10	59.5	67.2	4.9	3.41

Tabla III.m **Parámetros de suelo del modelo del tramo de Lorcha (río Serpis):**

<b>Codigo del Suelo</b>	<b>Presión de Burbujeo</b>	<b>Porosidad</b>	<b>Índice de Porosidad</b>	<b>Conductividad Hidráulica Saturada del Suelo</b>	<b>Humedad a Capacidad de Campo</b>
	<b>Pb (Kpa)</b>	<b>Pst ( )</b>	<b>Ip ( )</b>	<b>Ks (mm/hr)</b>	<b>Mfc ( )</b>
1	5.384	0.388	0.449	36.0	0.180
2	4.403	0.412	0.578	22.6	0.221
3	2.475	0.407	0.266	32.0	0.205
4	4.343	0.451	0.238	15.8	0.277
5	1.253	0.502	0.233	46.4	0.212
6	1.196	0.458	0.206	14.3	0.230
7	0.187	0.441	0.215	30.5	0.145
8	0.984	0.435	0.238	40.4	0.189
9	0.021	0.412	0.209	38.8	0.088
10	0.516	0.486	0.260	40.0	0.164

### **Tablas de parámetros de suelo del tramo de Tormo (río Mijares):**

Tabla III.n *Parámetros de suelo básicos del tramo de Tormo (río Mijares):*

<b>Código Suelo</b>	<b>% Gravas &gt;2 mm</b>	<b>% Arenas 0,05 &lt;S&lt; 2mm</b>	<b>% Arcillas C &lt; 0,002 mm</b>	<b>% Materia Orgánica</b>
1	4.3	76	6.1	1.25
2	65	18.6	1	1.25
3	65	10	1	4.24

Tabla III.o *Parámetros de suelo del modelo del tramo de Tormo (río Mijares):*

<b>Código del Suelo</b>	<b>Presión de Burbujeo</b>	<b>Porosidad</b>	<b>Índice de Porosidad</b>	<b>Conductividad Hidráulica Saturada del Suelo</b>	<b>Humedad a Capacidad de Campo</b>
	<b>Pb (Kpa)</b>	<b>Pst ( )</b>	<b>Ip ( )</b>	<b>Ks (mm/hr)</b>	<b>Mfc ( )</b>
1	0.427	0.244	0.134	78.4	0.110
2	0.432	0.232	0.042	37.7	0.094
3	0.539	0.393	5.337	28.7	0.110

**Tablas de parámetros de suelo del tramo de Cirat (río Mijares):**

Tabla III.p **Parámetros de suelo básicos del tramo de Cirat (río Mijares):**

Código Suelo	% Gravas >2 mm	% Arenas 0,05 <S< 2mm	% Arcillas C < 0,002 mm	% Materia Orgánica
1	74.4	20.6	1	0.68
2	59.9	32.7	1.6	2.55
3	76.9	19	1.2	0.99
4	62	32.1	1.4	0.21
5	50.5	38.2	1.8	0.82
6	1.1	40.6	7.9	0.52

Tabla XVII.q **Parámetros de suelo del modelo del tramo de Cirat (río Mijares):**

Código del Suelo	Presión de Burbujeo	Porosidad	Índice de Porosidad	Conductividad Hidráulica Saturada del Suelo	Humedad a Capacidad de Campo
	Pb (Kpa)	Pst ( )	Ip ( )	Ks (mm/hr)	Mfc ( )
1	0.42	0.441	0.651	49.2	0.079
2	0.466	0.235	0.078	43.9	0.112
3	0.427	0.288	0.129	39.6	0.089
4	0.41	0.362	0.325	43.9	0.073
5	0.418	0.340	0.482	48.8	0.097
6	0.393	0.339	4.532	24.2	0.199

**Tablas de parámetros de suelo de los tramos de Terde I y Terde II (río Mijares):**

Tabla III.r **Parámetros de suelo básicos de los tramos de Terde I y Terde II (río Mijares):**

Código Suelo	% Gravas >2 mm	% Arenas 0,05 <S< 2mm	% Arcillas C < 0,002 mm	% Materia Orgánica
1	26.7	41.7	1.4	0.49
2	58	28.1	4	1.98
3	3.3	40.5	14.2	1.91
4	5.3	57.7	8	0.77
5	65	7.2	0.6	0.67
6	15.3	58.7	10	2.03
7	7	82.5	4.4	0.27
8	64.2	30.1	2	0.95
9	0.3	48.5	10.2	0.55
10	65	21.1	2	0.39

Tabla III.s **Parámetros de suelo del modelo de los tramos de Terde I y Terde II (río Mijares):**

Código del Suelo	Porosidad	Índice de Porosidad	Presión de Burbujeo	Conductividad Hidráulica Saturada del Suelo	Humedad a capacidad de Campo
	Pst ( $\lambda$ )	Ip ( $\lambda$ )	Pb (Kpa)	Ks (mm/hr)	Mfc ( $\lambda$ )
1	0.397	0.530	3.848	56.2	0.131
2	0.436	0.208	0.342	22.7	0.168
3	0.441	0.226	2.625	19.9	0.247
4	0.406	0.277	1.037	42.8	0.154
5	0.412	0.220	0.057	28.2	0.102
6	0.432	0.184	0.246	37.1	0.174
7	0.414	0.251	0.024	98.0	0.069
8	0.423	0.231	0.056	34.5	0.095
9	0.398	0.278	2.370	26.8	0.19
10	0.403	0.206	0.045	25.1	0.104



**Tablas de parámetros de suelo de los tramos de Rabo del Batán I y II (río Cabriel):**

Tabla III.t *Parámetros de suelo básicos del tramo de Rabo del Batán (río Cabriel):*

Código Suelo	% Gravas >2 mm	% Arenas 0,05 <S< 2mm	% Arcillas C < 0,002 mm	% Materia Orgánica
1	67.8	13.9	4.1	2.08
2	70.8	25.3	1.4	1
3	9.5	46.2	5.8	1.43
4	30.8	55.9	4.4	3.46
5	0	34.8	14	4.05
6	0	25.3	15.9	1.53
7	78.4	8.9	4.2	1.95
8	78	18.5	1.4	2.81
9	0	80.1	11	1.23
10	0	84	6.4	1.53
11	75.2	20.7	2.4	4.54
12	26.1	39.5	3.5	0.87

Tabla III.u *Parámetros de suelo del modelo del tramo de Rabo del Batán (río Cabriel):*

Código del Suelo	Presión de Burbujeo	Porosidad	Índice de Porosidad	Conductividad Hidráulica Saturada del Suelo	Humedad a Capacidad de Campo
	Pb (Kpa)	Pst ( )	Ip ( )	Ks (mm/hr)	Mfc ( )
1	0.482	0.192	0.052	71.0	0.138
2	0.446	0.237	2.291	10.0	0.238
3	0.408	0.371	2.656	37.4	0.161
4	0.415	0.159	0.028	52.4	0.134
5	0.489	0.140	0.017	30.3	0.17
6	0.423	0.342	2.789	41.9	0.178
7	0.427	0.232	0.027	39.1	0.083
8	0.528	0.242	3.824	26.9	0.315
9	0.434	0.207	0.029	89.3	0.102
10	0.465	0.182	0.019	38.3	0.121
11	0.516	0.240	2.872	33.3	0.287
12	0.445	0.195	2.528	5.92	0.2680