



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Evaluación de un sistema para la previsión de Riesgos de
Incendios Forestales Catastróficos basado en Inteligencia
Artificial

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Ahmad Mansilla, Lugman Sami

Tutor/a: Mateo Pla, Miguel Ángel

Cotutor/a: Lemus Zúñiga, Lenin Guillermo

CURSO ACADÉMICO: 2021/2022

Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial

Resumen

En el grupo Tecnologías de la Información y Comunicación contra el cambio climático (ICTvsCC) se están desarrollando herramientas de predicción de los riesgos de la peligrosidad de posibles Incendios forestales. Como parte de ese desarrollo se pretende evaluarlas, comparándolas cuando sea posible con otras herramientas existentes. La experiencia en el grupo nos demuestra que esa tarea requiere técnicas de inteligencia artificial para llevar a cabo el desarrollo de estas herramientas de predicción.

En el presente proyecto se va a realizar la especificación, desarrollo, implementación e integración de un predictor de riesgo de incendios forestales mediante un sistema experto que permita tras su ejecución responder con una respuesta que indique el riesgo de producción de un incendio, en base a las métricas establecidas para la interpretación de este riesgo.

Este proyecto tendrá como base para su realización una aplicación ya existente la cual dispone de un módulo de predicción el cual se encuentra ya integrado en esta aplicación desde la que se parte, este módulo de predicción se encuentra desarrollado mediante otra tecnología de desarrollo. Este módulo de predicción es el módulo sobre el que se llevará a cabo el desarrollo del actual proyecto, intentando desarrollar un nuevo predictor mediante la tecnología de inteligencia artificial basada en un sistema experto. Tras su realización debe poder integrarse en esta aplicación actual y generar la posibilidad de sustituir este módulo de predicción si así se considera oportuno.

Palabras clave: CLIPS, sistema experto, regla, hecho, motor de inferencia, condicional.

Abstract

In the research group Information and Communication against Climate Change (ICTvsCC) tools are being developed to predict the risks of the danger of possible forest fires. As part of this development, it is intended to evaluate them, comparing them whenever possible with other existing tools. The experience in the group shows us that this task requires artificial intelligence techniques to carry out the development of these prediction tools.

In this project, the specification, development, implementation, and integration of a forest fire risk predictor will be carried out through an expert system that allows, after its execution, to respond with a response that indicates the risk of producing a fire, based on the metrics established for the interpretation of this risk.

This project will be based on an existing application which has a prediction module which is already integrated in this application from which it starts, this prediction module is developed using another development technology. This prediction module is the module on which the development of the current project will be carried out, trying to develop a new predictor through artificial intelligence technology based on an expert system. After its completion, it must be able to be integrated into this current application and generate the possibility of replacing this prediction module if it is considered appropriate.

Keywords: CLIPS, expert systems, rule, fact, inference engine, conditional.

Índice

Índice de Figuras	7
Índice de tablas.....	8
Agradecimientos.....	9
1 introducción	11
1.1 Presentación	11
1.2 Motivación.....	11
1.3 Objetivos.....	11
Objetivos primarios.....	11
1.4 Metodología.....	12
1.5 Estructura:.....	12
2 Estado del arte.....	15
2.1 Cambio climático e incendios en el mundo.....	15
2.2 Futuro climático.....	16
Incendios Forestales	17
2.3 Medidas al respecto	18
Causas más importantes de incendios forestales:	18
Factores que influyen en los incendios forestales.....	18
2.4 Caso real de Incendios en la Comunidad Valenciana	19
2.5 Predicción	20
3 Sistemas expertos.....	23
3.1 Arquitectura de un sistema experto	23
3.2 Aplicaciones de los sistemas expertos en la industria	25
3.3 Herramientas disponibles.....	26
3.4 Ventajas y limitaciones	27
4 Análisis del problema y planteamiento de la solución.....	29
4. Software Actual	29
4.3 Problema de la implementación actual	30
4.4 Planteamiento de la solución.....	31
5 Diseño y materiales	35
5.1 Material utilizado	35
5.1.1 Subsystem for Linux (WSL)	35
5.1.2 CLIPS	37
Sintaxis de CLIPS	38
Define una regla en CLIPS.	39
5.1.3 Visual studio code.....	39



Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial

5.1.4 Trello.....	39
5.1.5 Python.....	39
5.2.6 yEd - Graph Editor	40
5.2 Diseño del problema	41
5.3.1 Forma de trabajo	41
5.3.2 Diagrama de Flujo	41
5.3.3 Nomenclatura a seguir	42
6 implementación, integración y pruebas.....	45
6.1 Introducción a la sección.....	45
6.2 Implementación.....	45
6.2.1 Partes a implementar	45
6.2.2 Proceso de implementación	46
6.3 Integración.....	51
6.4 Pruebas	53
Comparación de ambos predictores con valores reales	54
Comparación de ambos predictores con valores aleatorios.....	56
6.5 Medición de tiempos.....	58
7 Presupuesto del proyecto	61
Consideraciones de esta sección	61
Materiales empleados	61
Mano de obra	62
Resumen del presupuesto	62
8 Conclusiones	63
Trabajo futuro	63
Relación del TFG con el Grado.....	64
Conclusiones personales	64
9 Bibliografía.....	67
ANEXO	69
OBJETIVOS DE DESARROLLO SOSTENIBLE.....	69
Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.....	70

Índice de Figuras

Figura 1 Incendio forestal.....	17
Figura 2 Evolución de emisiones de dióxido de carbono debido a incendios 1997-2019.	17
Figura 3 Causas de incendio Comunidad Valenciana	19
Figura 4. Incendios en las distintas provincias	20
Figura 5 Arquitectura de un sistema experto	24
Figura 6. Módulo de visualización, provincia de Alicante.	30
Figura 7. Parte de la función de predicción de riesgo.	32
Figura 8 Instalación de WSL2 en Windows 11	36
Figura 9 Sistema de ficheros de WSL 2 Ubuntu.....	37
Figura 10 Terminal WSL 2 Ubuntu 20.04.....	37
Figura 11 Arquitectura de CLIPS	38
Figura 12 Diagrama de flujo de nuestro predictor	42
Figura 13 Ejemplo básico de regla en clips.....	43
Figura 14 Función tormentas, módulo de predicción c.....	46
Figura 15 Archivo con reglas para predecir riesgo de tormenta.....	47
Figura 16 Función riesgo de incendio modulo c.....	48
Figura 17 Archivo que contiene reglas del método riesgo de incendio	49
Figura 18 Llamada a función CLIPS riesgo	51
Figura 19 Función obtener resultado	51
Figura 20 Función execute de CLIPSPredictor	53
Figura 21 Llamada al predictor c, predictor de prueba.	54
Figura 22 Almacenamiento en fichero de ambos predictores.....	54
Figura 23 Comparación de archivos mediante visual studio code	55
Figura 24 Ejecución de comando diff.....	55
Figura 25 Generación de valores aleatorios	56
Figura 26 listas resultado funciones.....	56
Figura 27 Cálculo de la diferencia de las funciones de predicción	57



Índice de tablas

Tabla 1 prueba datos aleatorios 100 datos	57
Tabla 2 prueba datos aleatorios 1000 datos.....	57
Tabla 3 prueba datos aleatorios 10000 datos	57
Tabla 4 prueba datos aleatorios 100000 datos	58
Tabla 5 Comparación de tiempos de ambos predictores para 100	58
Tabla 6 Comparación de tiempos de ambos predictores para 1000	58
Tabla 7 Comparación de tiempos de ambos predictores para 10.000	58
Tabla 8 Comparación de tiempos de ambos predictores para 100.000	59
Tabla 9 Materiales empleados.....	61
Tabla 10 Mano de obra	62
Tabla 11 Resumen del presupuesto	62
Tabla 12 Evaluación de implicación del proyecto en los ODS.....	69

Agradecimientos

Me gustaría agradecer de forma sincera a todas aquellas personas que me han ayudado a lo largo de este trabajo y han colaborado en el desarrollo de este proyecto. En primer lugar, a mis dos tutores Miguel Ángel Mateo Plá y Lenin Guillermo Lemus Zúñiga por guiarme desde el inicio hasta el final de este proyecto. En segundo lugar, a mi familia que han estado dándome ánimos y confiando en mi en todo momento. A mis compañeros y amigos que he conocido en la universidad con quienes he superado distintas asignaturas del grado y también me han apoyado en este trabajo. A el resto de mis amigos que de la misma manera que mi familia me ha apoyado y han creído en mi durante la realización de este proyecto. Por último, lugar, a la UPV y ETSINF que me han permitido aprender y adquirir mi formación académica, de forma muy agradable, gracias esta formación he conseguido adquirir todo el conocimiento necesario para conseguir realizar este proyecto de manera exitosa.

Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial



1 introducción

1.1 Presentación

En la actualidad se está dando un aumento de las temperaturas, causado principalmente por el cambio climático. Al mismo tiempo se está produciendo un prolongamiento de las temporadas cálidas. Ambos hechos sumados provocan un aumento considerable de los incendios forestales. Estos incendios forestales no solamente son extremadamente dañinos para el ecosistema y el medioambiente en general, sino que son un peligro para la población.

De forma paralela se puede apreciar un escenario a nivel tecnológico como nunca se había contemplado, se aprecian grandes desarrollos tanto a nivel de tecnología como de empleabilidad de esta tecnología, podemos ver como esto está generando que se puedan realizar desarrollos software con una complejidad notablemente mayor que hace unos años y que es capaz de mantenerse y seguir aumentando debido al crecimiento conjunto de la tecnología hardware y software.

Tras contemplar estas dos realidades, en el siguiente proyecto se va a intentar desarrollar una solución que intenta ayudar a predecir el riesgo de un incendio mediante una solución software basada en inteligencia artificial, con el objetivo de poder anticiparnos a este con la mayor seguridad y certeza posible para que se puedan tomar medidas con una antelación adecuada para intentar prevenir el desarrollo de estos incendios en la medida de lo posible.

Durante el desarrollo de este proyecto se va a describir la forma de realizar un predictor bajo la tecnología de un sistema experto basado en reglas que se implementara junto con otros proyectos ya existentes y que nos permita calcular el riesgo de que se produzca un incendio forestal.

1.2 Motivación

Desde un punto de vista personal, se ha elegido este trabajo como trabajo de fin de grado debido a la importancia del tema y el interés por participar en ayudar a un problema tan real e importante como son los incendios forestales. Se desea poder contemplar cómo aplicar el conocimiento adquirido, junto con las herramientas de las que disponemos para llevar a cabo un software que realice de la manera más fiel posible a la realidad una estimación de la producción de un incendio. Esto demuestra cómo tras la realización de la formación académica en el grado de ingeniería informática esta puede ser realmente útil, practica, así como factible para el desarrollo de soluciones que automaticen y ayuden a tomar decisiones, así como aplicarse a causas tan necesarias e importantes para todos nosotros, como el caso de este proyecto.

1.3 Objetivos

Objetivos primarios

En este proyecto el principal objetivo que se ha establecido es realizar un predictor con el que se pueda obtener una predicción con la misma exactitud que un predictor ya existente en la aplicación donde se integrará.



Como segundo objetivo primario será la integración de este módulo de predicción sin modificar la estructura de la aplicación ya existente, que ya está terminada y testeada por el desarrollador de este proyecto.

Ambos objetivos se desean conseguir realizando el desarrollo de este predictor con una mayor facilidad para el cambio o modificación, en comparación con el módulo de predicción ya existente del que se dispone.

Objetivos secundarios

Se pretenderá realizar una solución con un coste temporal que sea compatible con los tiempos definidos para la ejecución de este en los cuales se espera recibir una predicción, intentando que sea siempre lo más ágil posible en su ejecución.

1.4 Metodología

La metodología que se llevará a cabo será una planificación por adelantado de las tareas a realizar, propuestas por los tutores del proyecto o con el conocimiento de ellos.

Estas tareas estarán basadas en la adquisición de información para poder tomar decisiones, diseño de la solución o el desarrollo del proyecto.

Paralelamente realizaremos la escritura de la memoria del proyecto para poder plasmar la información más relevante en el momento que esta es adquirida y aplicada.

La metodología práctica para el desarrollo del proyecto software de este trabajo será: realizar un esquema funcional del programa a realizar, estudiar que herramienta será la que utilicemos, con este esquema intentar conseguir la funcionalidad a implementar con la herramienta escogida, realizar una serie de pruebas y finalmente integrarla en el sistema, que para nosotros debe ir acoplada en un proyecto ya existente.

1.5 Estructura:

La memoria de este TFG está organizada en las siguientes secciones:

- 1. Introducción:** En esta sección se describe el problema básico que se desea resolver y los objetivos concretos que se quieren alcanzar con el proyecto.
- 2. Estado del arte:** En esta sección se describirán los problemas y soluciones a estos problemas con los que nos encontramos en la actualidad.
- 3. Sistemas expertos:** Se describirá que es un sistema experto, como funcionan y algunas aplicaciones de estos.
- 4. Análisis del problema:** Se planteará la implementación actual, analizando las causas por las que se quiere plantear una nueva solución y la posible solución que se va a desarrollar.
- 5. Diseño y materiales:** En esta sección vamos a describir como planteamos nuestro problema para poder darle una solución, así como la forma en que trabajaremos y los materiales (SW) que se usarán en la implementación.
- 6. Implementación, Integración y pruebas:** En esta sección se llevará a cabo el plan establecido en la sección anterior, y se realizará la implementación de esta solución, una vez implementada la integraremos en la aplicación ya existente, la cual hará uso de este módulo desarrollado en el proyecto para obtener las predicciones deseadas y almacenar los resultados de estas predicciones.

Finalmente realizaremos unas pruebas de exactitud del modelo para poder validar la exactitud de el mismo.

7. **Conclusiones:** En esta sección discutiremos los resultados de nuestro trabajo, así como las posibles limitaciones y mejoras de este. También incluimos un apartado de la relación del TFG con el grado cursado *y un conjunto de conclusiones personales*.
8. **Presupuesto:** En esta sección se expondrán los distintos costes de este proyecto.
9. **Bibliografía**

Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial



2 Estado del arte

Esta sección tratará de explicar el contexto actual a nivel climatológico y medioambiental, tanto a nivel internacional, como a nivel nacional de España. Debido a la climatología y el aumento actual de las temperaturas se ilustrará como resulta este cambio en los incendios que se están produciendo a nivel mundial. Se planteará que con la información actual podemos predecir que en ciertos momentos y lugares se puede estimar que se producirá un incendio a nivel estadístico.

El cambio climático está teniendo gran impacto a nivel mundial, con lo que respecta a incendios forestales es sumamente grave, ya que debido al aumento de temperatura las temperaturas que se juntó con los factores como olas de calor, sequías acumuladas y baja humedad unida a una vegetación muy seca y bosques decaídos, está generando incendios mucho más rápidos y de una virulencia nunca vista.[1]

2.1 Cambio climático e incendios en el mundo

El Servicio de Monitoreo Atmosférico Copérnico (CAM5), hizo un seguimiento durante 2019 de la actividad de los incendios y las estimaciones de emisiones a la atmósfera debido a estos fuegos. Cada año arde entre el 3% y el 4% de la superficie terrestre del planeta [1]. Podemos ver parte del estudio en la que se ve el impacto del cambio climático en estas.

Zonas parte del estudio

Círculo polar ártico: Esta zona claramente ha apreciado las condiciones de periodos secos y cálidos, lo que favorece a la virulencia de las llamas, así como a su vez cada vez se aumentan estas condiciones climatológicas. El Ártico se está calentando a una velocidad dos veces más rápida que el resto del planeta, aumentando su temperatura en 10°C con respecto a las obtenidas entre 1981 y 2010, en el mes de julio de 2019 se llegaron a obtener las temperaturas más altas registradas hasta el momento para esta zona. A consecuencia de este aumento, aunque esta zona está adaptada a los incendios, actualmente los ciclos de fuego se están volviendo más intensos, con una virulencia de las llamas en relación con el incremento de las condiciones extremadamente cálidas claramente relacionada.

- **Amazonia:** Amazonia también ha vivido un año particularmente duro, con la ola de incendios de mayor impacto de la última década que afectó principalmente a Brasil y Bolivia. Estos incendios están ligados a procesos de deforestación para la expansión de la actividad agropecuaria, principalmente cultivo de soja o pastos para el ganado. De hecho, las diez ciudades en la Amazonia con más incendios forestales son las mismas con más deforestación. Un dato importante es que las condiciones de la estación seca de 2019 estuvieron dentro de rangos normales. Y, sin embargo, Brasil registró la tasa de deforestación más alta desde 2008, un aumento del 50% en comparación con el promedio de los últimos 10 años y de casi el 150% respecto a 2017. Estos incendios son particularmente graves porque afectan al bosque tropical más grande y diverso del mundo, hábitat del 10% de la biodiversidad del planeta y hogar de más de 34 millones de personas, incluyendo unos 500 pueblos indígenas.

- **Australia:** Australia vivió un año 2019 insólitamente cálido. Nunca el país había tenido temperaturas tan altas en promedio, con una anomalía positiva de +1,52° C. Además, a finales de diciembre se dieron temperaturas récord con máximas de 42°C en todo el país. Todo ello favoreció esos incendios tan devastadores e incontrolables por los dispositivos de extinción.
- **África central:** La zona del planeta que más se quema al año está en el corazón de África. La agencia espacial estadounidense lo define como un “continente en llamas” debido a que concentra el 70% de los focos de incendio del mundo. Solo en Angola y la República Democrática del Congo hubo en apenas unos días más de diez mil focos. Zambia, Mozambique o Madagascar también sufren el mismo proceso. En África se queman unos 300 millones de hectáreas al año, seis veces la superficie de España, sin embargo, suponen graves riesgos, a medio plazo y sin una gestión adecuada aceleran la erosión del suelo. Se calcula que el 10% de los incendios se escapan de control y son los responsables del 90% de la superficie que arde, causando graves daños ambientales y sociales. Aunque la deforestación en África no se deba a estas quemadas, es real, a través principalmente de la tala. La cubierta forestal de la República Democrática del Congo se ha reducido en un 13% en los últimos quince años.
- **Incendios en Europa:** en la actualidad Europa a causa del cambio climático, se está convirtiendo en una zona de riesgo con respecto a los incendios, aumentando su rango de afectación no exclusivamente a los países mediterráneos, por lo que Europa central se está empezando a ver cada vez más afectada. Países como Reino Unido, Irlanda, Finlandia, Letonia, Alemania, Polonia, Suecia o Noruega se vieron sacudidos en 2019, por incendios que se consideran inusuales para estas zonas.

2.2 Futuro climático

Podemos ver como la presencia de mayores temperaturas ya se aprecia por todas partes del planeta, así como lo que ello conlleva; zonas más secas, menor humedad, presencia de olas de calor, sequías, etc. El problema no solo radica en la actualidad, puesto que presente un aumento que tiende a crecer para el futuro, así que debemos tener muy en cuenta el cambio climático y lo que ello conlleva, que para nuestro caso particular será la temperatura, así como el aumento de incendios producidos por esta.

Incendios Forestales



Figura 1 Incendio forestal

Los incendios forestales y el cambio climático constituyen un círculo vicioso. A medida que aumenta el número de incendios también lo hacen las emisiones de gases de efecto invernadero, y se incrementan la temperatura general del planeta y la sucesión de eventos climáticos extremos. Las emisiones debidas a los incendios en 2019 supusieron un repunte a nivel global. En total se liberaron 7.800 millones de toneladas de CO₂, el equivalente a unas 25 veces las emisiones totales de España en un año. Las emisiones de dióxido de carbono provocadas por los incendios han disminuido en los últimos años, pero en 2019 aumentaron en un 26%. Comparado con las emisiones debidas a la quema de combustibles fósiles, los incendios fueron responsables de más de una quinta parte de los 36.800 millones de toneladas de carbono liberadas el pasado año. Puede verse en la Figura 2 Evolución de emisiones de dióxido de carbono debido a incendios 1997-2019..

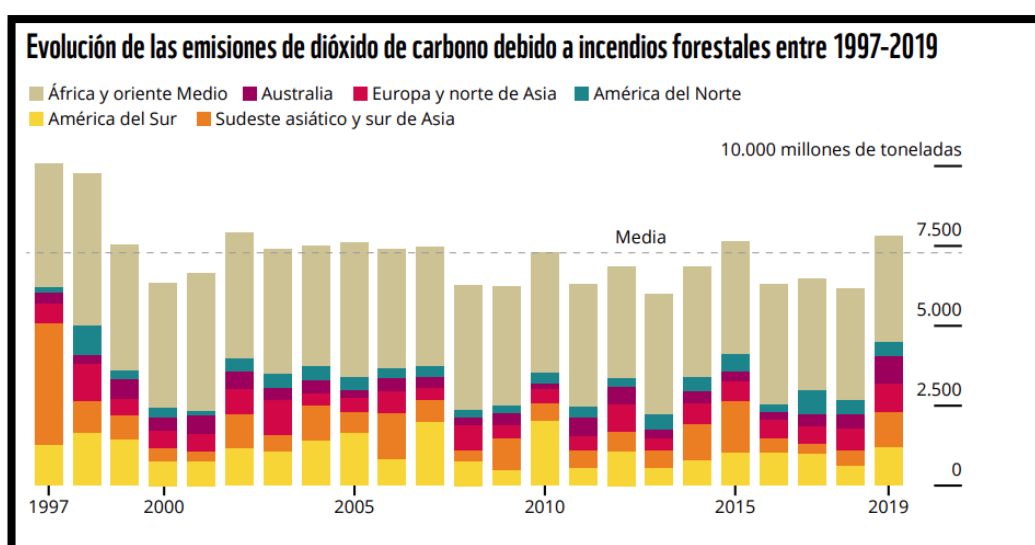


Figura 2 Evolución de emisiones de dióxido de carbono debido a incendios 1997-2019.

2.3 Medidas al respecto

Los incendios forestales tienen un gran impacto, tanto para el ecosistema, como para la economía. Cuando se producen, ejercen un gran impacto negativo en los recursos fundamentales, ya que estos producen desequilibrios en el régimen hídrico, contaminación ambiental, así como a su vez produce que se destruya fauna y vegetación, además de poner en peligro a nuestra población en ciertas circunstancias. Debido a este impacto en los países más desarrollados ya se están tomando medidas para poder predecir estos con la mayor antelación posible, poniendo así un plan de actuación frente a ellos. Para poder predecir de una forma más acertada posible estos incendios, se utilizan distintos sistemas de predicción, todos ellos hacen uso de los datos meteorológicos.

Causas más importantes de incendios forestales:

- Causas naturales
- Causas accidentales
- Causas Intencionadas y Negligencias
- Causadas por incendios anteriores (Reproducción).
- Causas Desconocidas

Factores que influyen en los incendios forestales

La vegetación de una forestación puede ser interpretada como combustible, ya que es el componente activo del que se desarrolla y mantiene el fuego en este tipo de incendios, debido a esta interpretación el clima de este puede determinar el tipo de esta vegetación, por lo tanto, el clima influye de forma directa en la cantidad y tipo de esta vegetación que podemos interpretar como posible combustible para la reproducción del incendio.

Esto nos indica que tomar muy en cuenta la climatología de la zona nos puede dar una ayuda a la hora de realizar una predicción sobre un incendio.

- **Meteorología:** con respecto a la meteorología las componentes que mayor importancia tienen en la producción de un incendio serían: la temperatura, la velocidad del viento, la humedad relativa y la estabilidad atmosférica. Temperaturas muy cálidas hacen que disminuya la humedad relativa del aire, tanto la humedad, como la temperatura influyen en la hidratación de los combustibles muertos. Otro factor de gran importancia sería el viento, cuando se producen vientos fuertes y secos aumenta en gran medida la propagación de estos.
- **Combustibles:** El contenido en humedad está relacionado con la fenología y disponibilidad de agua en el suelo, por lo que se relaciona bien con índices de sequía. En atmósferas estables, la humedad relativa disminuye al aumentar la temperatura, de ahí que el contenido en humedad de estos combustibles sea máximo a primeras horas del día y mínimo al comenzar la tarde. La relación es tal que los efectos más importantes ocurren en las fases primeras de falta de agua en el suelo, Igualmente, el contenido en humedad de la hojarasca depende de las condiciones meteorológicas, de su exposición al sol, así como del contenido en humedad del suelo. Cuanto más desecante sea la atmósfera y menor la humedad

del suelo, más seca estará la hojarasca, lo que aumentará su inflamabilidad y combustibilidad.

- **Topografía:** esta puede influir en el ángulo del fuego, lo que genera un impacto en el desarrollo de este, pendientes empinadas generan una rápida expansión de este comparado con un ángulo de pendiente descendente. Esto no solo puede influir por la disposición del terreno, sino que puede tener un impacto el crecimiento de esta vegetación.

2.4 Caso real de Incendios en la Comunidad Valenciana

Los incendios debidos a negligencias o a accidentes y los incendios intencionados siguen siendo las principales causas de inicio. Durante los años considerados, la causa intencionados supuso el 37% del total de incendios y el 32% de los incendios fueron causados por negligencias y causas accidentales. Ambas causas debidas a acciones o a actividades humanas representan el 69 % del total de incendios, véase la Figura 1.

A causa natural rayos, tiene mucha relevancia en la comunidad Valenciana ya que representa el 25% de las igniciones producidas durante el periodo de estudio. Generalmente, los rayos son producidos por un tipo de nubes de desarrollo vertical llamadas cumulonimbos. La formación de estos cumulonimbos y tormentas es habitual en el interior de la Comunidad Valenciana y se suelen producir habitualmente durante las últimas horas de la tarde y las primeras de la noche, sobre todo en época estival [2].

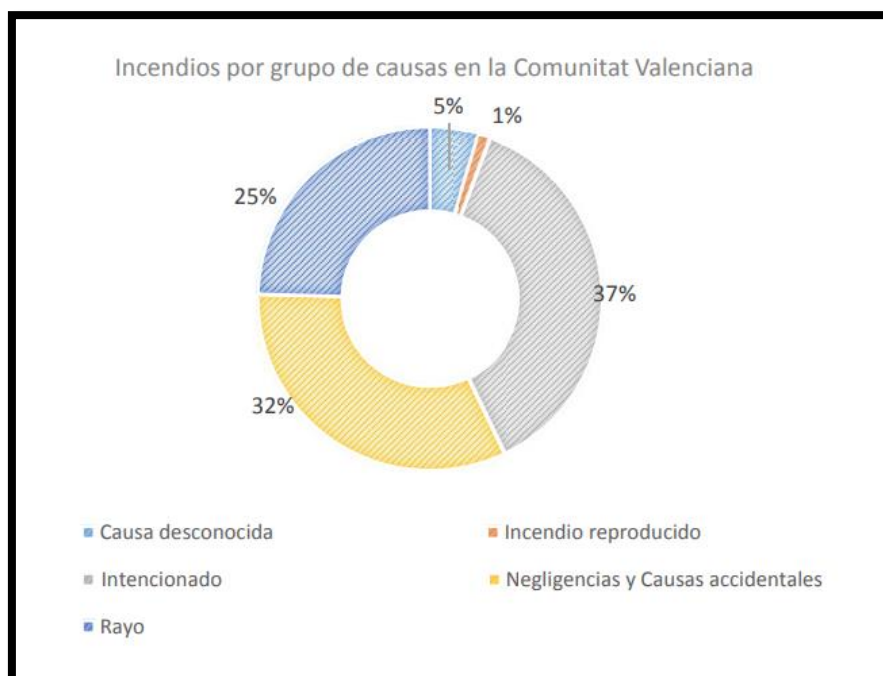


Figura 3 Causas de incendio Comunidad Valenciana

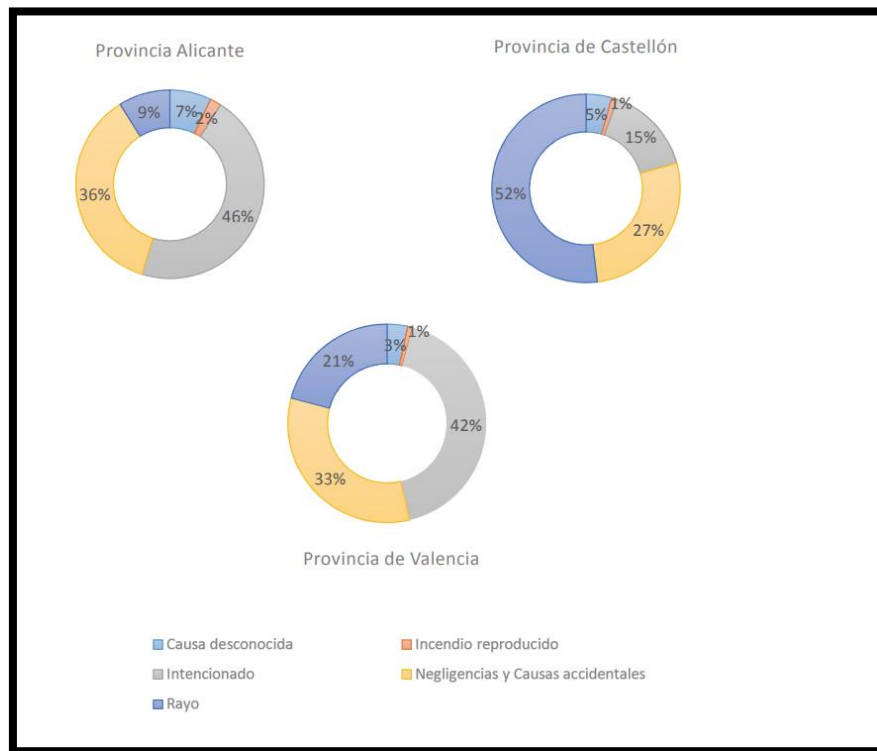


Figura 4 Incendios en las distintas provincias

2.5 Predicción

En base a estos factores climatológicos y ambientales descritos anteriormente, como son la temperatura, la velocidad del viento, la humedad relativa y la estabilidad atmosférica, combustible, la topografía, etc. Se puede apreciar que son valores que se deben tener en cuenta a la hora de intentar tener información sobre la producción de un incendio, que pueden resultar de gran importancia a la hora de realizar una predicción, estos factores pueden estudiarse estadísticamente para poder ver el impacto que estos pueden tener en la producción de un incendio y así como la severidad de este. En base a su importancia se pueden crear predictores que con una mayor o menor fiabilidad intenten predecir la producción de un incendio.

La ocurrencia de un incendio se ha estimado a través de dos estrategias. La primera consiste en estimar indicadores de riesgo de ocurrencia con base en medidas indirectas de atributos climáticos, físicos, ambientales o de las características de los mismos incendios (i. e. las estadísticas de ocurrencia). Esta estrategia se deriva de la ausencia o poca calidad y cantidad de datos de las variables causales de un incendio o del conocimiento limitado del comportamiento del fuego en una región particular. Las estimaciones se hacen a través de indicadores derivados de análisis probabilísticos de la información disponible (distribuciones de tamaños) con refinamientos a través de

modelos de regresión, redes neuronales, programación matemática y modelos. La segunda estrategia es la predicción de la ocurrencia o características de un incendio (tamaño, intensidad y duración), a través de modelos que predicen la probabilidad de observar una superficie afectada (o atributos del incendio) en función de las particularidades del sitio. Las estimaciones se hacen a través de una variedad amplia de técnicas que van desde modelos estadísticos de predicción, hasta redes y métodos bayesianos, entre otras[3].

Como se comentó anteriormente en la actualidad todos los países que cuentan con los recursos suficientes están poniendo en marcha alguna serie de medidas para intentar frenar los incendios.

De esta forma y mediante las distintas estrategias entre las cuales se puede realizar un predictor en la actualidad cada país está optando por la que considera más oportuna, en España se puso en marcha el proyecto Arbaria que consiste en un predictor de riesgo de incendios y extinción de estos y que resulta de especial relevancia para el país pues es el proyecto impulsado por el Ministerio para la Transición Ecológica y el Reto Demográfico, a continuación se describirá este de una forma resumida para tener un poco más sobre este predictor que actualmente se está empleando en España.

Predictor Arbaria

En España el Ministerio para la Transición Ecológica y el Reto Demográfico (MITECO). Ha puesto en marcha un predictor: Arbaria, con la esperanza de ayudar a reducir estos en el país, el proyecto comenzó en 2019, comenzando a utilizarse en verano del 2020.

Este proyecto mediante el uso e interpretación de variables climatológicas y ambientales, intenta tener presente el comportamiento del clima durante el tiempo y así poder ir analizando que factores de riesgo se encuentran que podrían ser causas de un incendio y realizar una predicción de este incendio con la información del momento en el que se está indicando estos factores, para así en caso de producirse estos tener una estimación de la severidad, así como la estimación del esfuerzo que implicaría la extinción del mismo.

Arbaria es un predictor de incendios y extinción de incendios. Este proyecto está basado en una solución en la nube, utilizando Amazon Web Service (AWS) y realiza la predicción por medio de inteligencia artificial. Para llevarlo a cabo han participado entre otras personas Antonio López Santalla, trabajador del Área de Defensa contra Incendios Forestales de la Subdirección General de Política Forestal y Lucha contra la Desertificación (MITECO) y Faustino Sánchez García, del área de Análisis de Datos de la S. G. de Tecnologías de la Información y de las Comunicaciones (MAPA).

El desarrollo de este se basa en librerías de uso libre para Python y que proporciona apoyo al desarrollo en estadística, machine learning y deep learning. Se han desarrollado los modelos de forma que estos se han integrado en aplicaciones informáticas en AWS que permiten la explotación y operatividad con los usuarios que las necesiten.



Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial

3 Sistemas expertos

Un sistema experto es un sistema de información basado en el conocimiento, el cual usa el conocimiento de un área de aplicación compleja y específica a fin de actuar como un consultor experto para los usuarios finales. Los sistemas expertos proporcionan respuestas sobre un área problemática muy específica al hacer inferencias semejantes a las humanas sobre los conocimientos obtenidos en una base de conocimientos especializados.[4]

Para que un sistema actúe como un verdadero experto, es deseable que reúna, en lo posible, lo más importante de las características de un experto humano, esto es:

- Habilidad para adquirir conocimiento.
- Fiabilidad, para poder confiar en sus resultados o apreciaciones.
- Solidez en el dominio de su conocimiento.
- Capacidad para resolver problemas.

Entre los tipos de sistemas expertos más reconocidos y utilizados desde sus inicios hasta la actualidad estarían los siguientes:

- **Basados en reglas:** Aplicando reglas heurísticas apoyadas generalmente en lógica difusa para su evaluación y aplicación. [5]. Hay numerosas formas de representar el conocimiento en IA, sin embargo, los Sistemas Expertos suelen ser llamados sistemas basados en reglas. Reglas “Si...entonces...” Las reglas “sí.. entonces..” son el principal tipo de conocimiento usado en Sistemas Expertos, donde dichas normas se utilizan para capturar razonamiento de expertos que emplean a menudo [6].
- **Basados en casos CBR (Case Based Reasoning):** Aplicando el razonamiento basado en casos, donde la solución a un problema similar planteado con anterioridad se adapta al nuevo problema. [5]. El razonamiento basado en casos generalmente es el proceso de solucionar nuevos problemas basándose en las soluciones de problemas anteriores.[6]
- **Basados en redes:** Aplicando redes bayesianas, basadas en estadística, y el teorema de Baye [5]
- **Sistemas expertos difusos:** los Sistemas Expertos difusos se desarrollan usando el método de lógica difusa, la cual trabaja con incertidumbre. Esta técnica emplea el modelo matemático de conjuntos difusos, simula el proceso del razonamiento normal humano permitiendo a la computadora comportarse menos precisa y más lógicamente que las computadoras convencionales.[6].

3.1 Arquitectura de un sistema experto

En un sistema experto se consideran tres componentes básicos, la base de conocimiento, la base de hechos, y el motor de inferencias. Estos componentes se hayan en todos los sistemas expertos y ante pequeñas modificaciones, siguen teniendo una base estructural basada en estos tres componentes. Véase ilustrado en la Figura 5 Arquitectura de un sistema experto.

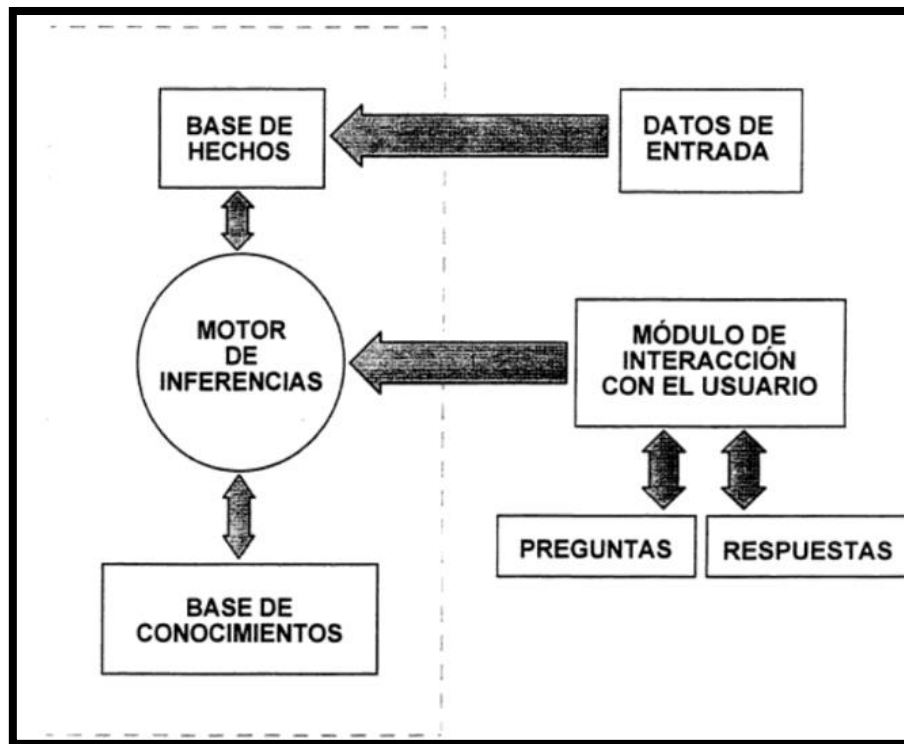


Figura 5 Arquitectura de un sistema experto

La **base de conocimientos** está formada por toda la cantidad de información disponible sobre el campo en el que se desarrolla la aplicación. Esta información debe contener tanto los hechos contrastados como las creencias y los heurísticos adquiridos por los especialistas a través de la experiencia. El orden en que los elementos básicos de información se incluyen en la base de conocimientos no es normalmente determinativo, es decir pueden introducirse en la base tal como se extraen del especialista. Esto es particularmente interesante tanto en el momento de construcción como en el de depuración de la base de conocimientos.

La **base de hechos** está formada por los datos sobre la situación concreta que va a ser considerada. Además, puede utilizarse como memoria auxiliar o de trabajo para almacenar los modos de razonamiento y la forma en que han sido utilizados. De esta forma, bajo los requerimientos del usuario final, el sistema ha de ser capaz de explicar cómo ha conseguido llegar a los resultados finales. Esta clarificación y transparencia en el modo de razonamiento es, la mayoría de las veces, de vital importancia para el usuario final, debido al escepticismo aun existente en una gran parte de las ocasiones sobre el funcionamiento de estos sistemas.

El **motor de inferencias** es un programa que se encarga de gestionar las informaciones existentes en la base de conocimientos y los datos de la base de hechos, confrontándolos para construir una serie de razonamientos que conduzcan a los resultados. Básicamente, puede utilizar dos modos de razonamientos diferentes: el "encadenamiento hacia adelante" ("forward chaining"), y el "encadenamiento hacia atrás" ("backward chaining"). En el primero, una regla puede aplicarse cuando las condiciones indicadas en la parte izquierda (premisas) se cumplen en la situación que se está considerando

indicada en la base de hechos o memoria de trabajo. Si esto ocurre, las condiciones expresadas en la parte derecha de la regla se agregan a la base de hechos y continua el proceso de razonamiento. En el encadenamiento hacia atrás se parte de un conjunto de hipótesis que son contrastadas con las partes derechas de las reglas de la base de conocimientos. Al aplicar una regla se sustituye el objetivo inicial por un conjunto de subobjetivos indicados por las premisas de la regla aplicada y se continua el proceso.

Además de las ya consideradas existe otra parte fundamental incluida en la arquitectura ideal de un sistema experto. Es la formada por los módulos utilizados para la comunicación hombre-máquina. Normalmente el usuario final de un sistema experto es personal no especializado, es decir, no relacionado con su desarrollo e implantación; por esta razón ha de procurarse que posea un elevado grado de accesibilidad y facilidad de manejo. Para ello, se pone especial énfasis en la construcción de módulos que permitan un fácil diálogo entre el usuario y el sistema experto. La tendencia actual es el establecimiento de una conversación en lenguaje natural mediante un subconjunto de palabras utilizadas normalmente en el campo de la aplicación desarrollada. Otra forma alternativa es la presentación en pantalla, de gráficos que permitan una fácil manipulación. La base de conocimiento está compuesta por toda la información disponible de utilidad, en el campo del sistema experto de la aplicación. Esta estará compuesta.[4]

3.2 Aplicaciones de los sistemas expertos en la industria

A lo largo de los años, el uso de los sistemas expertos se ha convertido en una necesidad de primer orden, adaptándose a multitud de sectores y ramas dispares. Estas son sus aplicaciones más destacadas [5]:

- Análisis de préstamos.
- Optimización de almacenes logísticos.
- Toma de decisiones financieras.
- Planificación y programación.
- Gestión de datos.
- Instalaciones sanitarias.
- Evaluación del desempeño de empleados.
- Monitorización y control de procesos.



3.3 Herramientas disponibles

En esta sección se describen brevemente distintas tecnologías disponibles para la construcción de un Sistema Experto[6]:

- **JESS:** El motor de reglas JESS es un proyecto que tuvo su origen en CLIPS pero que fue escrito enteramente en Java. Se desarrolló durante la década de los noventa en los Sandia National Laboratories y comparte con CLIPS varios conceptos de diseño y similitudes con respecto a la sintaxis. Asimismo, implementa la especificación de referencia JSR94.
- **CLIPS:** A mediados de los años ochenta, la NASA requería el apoyo de Sistemas Expertos para el desarrollo de proyectos. Por lo tanto, una serie de prototipos surgen, pero sus resultados no fueron lo suficientemente buenos para cumplir con los requerimientos internos. En consecuencia, se desarrolló un prototipo de un Sistema Experto, denominado CLIPS (C Language Integrated Production System) cuya principal característica era su capacidad para funcionar con otros sistemas existentes. Posteriores mejoras y ampliaciones han convertido CLIPS en un punto de referencia para el desarrollo de otros Sistemas Expertos.
- **PROLOG:** Es un lenguaje de programación lógica de propósito general asociado con la inteligencia artificial y lingüística computacional (Balbin, 1985). Es un lenguaje declarativo basado en reglas. Su nombre deriva del anagrama PROgramación LÓGica. La sintaxis del lenguaje consiste en lo siguiente:
 - Declarar hechos sobre objetos y sus relaciones
 - Hacer preguntas sobre objetos y sus relaciones
 - Definir reglas sobre objetos y sus relaciones
- **Drools:** Al igual que en el caso de los CLIPS y JESS, Drools es la implementación y ampliación del algoritmo Rete diseñado por el Dr. Charles L. Forgy en la Universidad Carnegie Mellon. Básicamente, su algoritmo consiste en una red de nodos interconectados con diferentes características que evalúan las entradas mediante la propagación de los resultados del siguiente nodo cuando hay coincidencias. DROOLS ofrece herramientas de integración con Java, la capacidad de escalabilidad y una división clara entre los datos y la lógica de dominio (Browne, 2009).
- **Jena:** Jena es un framework desarrollado en tecnología Java que incluye un motor de inferencia basado en normas, una API ontológica y un motor de búsqueda (Jena, 2013).
- **JEOps:** JEOPS añade encadenamiento hacia adelante, las normas de producción de primer orden con el fin de facilitar el desarrollo de Sistemas Expertos mediante programación declarativa (Jeops, 2013).
- **OpenCyc:** OpenCyc es la versión de código abierto de la tecnología CyC más completa base de conocimientos generales del mundo y motor de razonamiento de sentido común (Cycorp, 2013).

3.4 Ventajas y limitaciones

Ventajas [6]

Mientras que un experto humano tiene limitaciones y percances propias de su condición humana, es decir: se enferma, envejece, migra a otras empresas, el Sistema Experto, respecto a sus pares humanos, no sufre de estas cuestiones y se convierte en una herramienta estable para su entorno y fiable porque sus actividades son completamente replicables (siempre contesta de la misma manera a menos que se le cambie el diseño). A esto se le suma la velocidad de que es mayor al de un ser humano. Debido a la escasez de expertos humanos en determinadas áreas, los sistemas expertos pueden almacenar su conocimiento para cuando sea necesario poder aplicarlo. Así mismo los SE pueden ser utilizados por personas no especializadas para resolver problemas. Además, si una persona utiliza con frecuencia un se aprenderá de él. Finalmente, si se evalúa el costo total del empleo de esta tecnología, la replicabilidad y estabilidad, asociado a la seguridad que provee, resulta una ecuación favorable, aun considerando que las inversiones iniciales pueden ser relativamente elevadas.

Limitaciones [6]

Es evidente que para actualizar se necesita de reprogramación de estos (tal vez este sea una de sus limitaciones más acentuadas) otra de sus limitaciones puede ser el elevado costo en dinero y tiempo, además que estos programas son poco flexibles a cambios y de difícil acceso a información no estructurada. Los Sistemas Expertos carecen de sentido común, para un sistema experto no hay nada obvio Además no podemos mantener una conversación informal con estos sistemas. Para un sistema experto es muy complicado de aprender de sus errores y de errores ajenos. No son capaces de distinguir cuales son las cuestiones relevantes de un problema y separarlas de cuestiones secundarias. Por otra parte, la inteligencia artificial no ha podido desarrollar sistemas que sean capaces de resolver problemas de manera general o de aplicar el sentido común para resolver situaciones complejas ni de controlar situaciones ambiguas.

Los Sistemas Expertos han sido diseñados para facilitar las tareas en múltiples campos de aplicación y proporcionar equivalentes resultados a los de un experto humano. Se sabe que no son aptos para resolver problemas generales pero su eficiencia aumenta mientras más acotado o específico sea el dominio de aplicación. Actualmente los Sistemas Expertos son empleados en áreas diversas y existe una importante variedad de herramientas y tecnologías para implementarlos. Como todo sistema, poseen fortalezas y debilidades, no obstante, y en términos generales, por su flexibilidad, confiabilidad y escalabilidad se los puede considerar como una tecnología de probada efectividad y lo suficientemente madura para confiar decisiones de considerable criticidad.[6]

Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial

4 Análisis del problema y planteamiento de la solución

En la sección actual se describirá lo más relevante de la aplicación sobre la que se implementará e integrará el proyecto actual y se tratará de mostrar de forma clara como se encuentra esta y que cambios se desean hacer en esta aplicación tras el desarrollo del proyecto actual. El proyecto actual será resultado del análisis y resolución de los cambios que se planteen realizar en esta sección.

4. Software Actual

Actualmente se dispone de una aplicación software la cual mediante distintos módulos consigue realizar el siguiente proceso:

1. Obtener información de los sensores medioambientales de distintas zonas forestales de la comunidad Valenciana y Portugal. En las zonas forestales sobre las que trabaja la aplicación se encuentran algunos sensores los cuales permiten conocer el estado de la temperatura, dirección del viento, racha máxima de viento, humedad relativa, punto de rocío, lifted y temperatura en punto de rocío. Actualmente hay un módulo que recupera la información almacenada de la lectura de estos por otra aplicación externa a nuestra aplicación actual, la cual se encarga del monitoreo y almacenamiento de esta en sus sistemas y la introduce en el sistema sobre el que se trabajará.
2. Realizar una predicción en base a estos factores medioambientales. Mediante el conocimiento de un profesional experto en la temática de la climatología y los incendios forestales se ha llegado a unas conclusiones que ellos aplican para tener una predicción de estos incendios y por consiguiente dar una respuesta del riesgo de que se produzca este. En este módulo con esa información del experto de forma automatizada mediante el módulo actual se realiza una predicción tomando estas variables procedentes de los sensores las cuales se obtiene mediante el módulo anterior y devuelve una respuesta correspondiente a la predicción.
3. Almacenamiento de la información. Estas predicciones que ha realizado el módulo anterior se deben almacenar junto con una serie de información que las acompaña, como es la información con la que se ha llegado a estas conclusiones, en nuestro caso las variables de entrada correspondientes de las lecturas de los sensores, coordenadas y fechas de obtención de estas predicciones.
4. Visualización. Los resultados de las predicciones finalmente serán representados en un mapa el cual ilustre estos resultados de forma visual para el usuario que la consume. Ver Figura 6. Módulo de visualización, provincia de Alicante

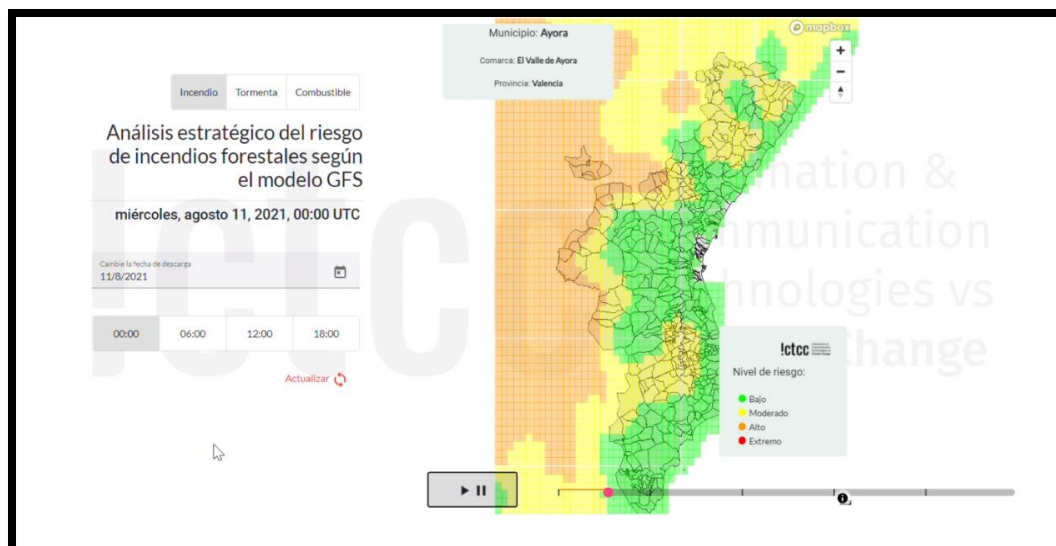


Figura 6. Módulo de visualización, provincia de Alicante

En la anterior imagen Figura 6. Módulo de visualización, provincia de Alicante, se puede apreciar la interfaz gráfica que ofrece este módulo. Aquí se mostrarán los resultados de las predicciones para una determinada fecha, con una hora determinada. La aplicación permite ilustrar las predicciones que se encuentran realizadas para la hora desde la que se ha seleccionado de inicio, hasta la hora más reciente de la cual se dispone de una predicción. El resultado es mostrado en el mapa de la Comunidad Valenciana conforme a los colores correspondientes al valor de predicción de esta zona. Estos son: cero y uno: bajo; dos: moderado; tres alto; cuatro extremo. Se mostrarán en el mapa como verde para el nivel de riesgo bajo, amarillo para el nivel de riesgo moderado, naranja para el nivel de riesgo alto y rojo para el nivel de riesgo alto.

En este proyecto se trabajará únicamente sobre el segundo modulo, la realización de una predicción, en el cual se encuentra implementado mediante un programa realizado en el lenguaje de programación C, el cual permite realizar esta predicción con exactitud y se planteará su remplazo por un sistema experto que se encargue de la resolución de esta predicción de incendio.

4.3 Problema de la implementación actual

Este módulo dedicado a la predicción esta realizado en el lenguaje de programación c. Este módulo es un programa que recibe el valor de los distintos sensores como valores de entrada(variables) y las trata para poder aplicarle el tratamiento de los datos que se ha establecido para que sean variables útiles para realizar una predicción con la lógica que es experto ha especificado y finalmente y como parte fundamental de este módulo, mediante una serie de funciones retorna unas predicciones de riesgo de incendio, riesgo de tormenta, tipología y disponibilidad de combustible, las cuales son valores de predicción que se esperan para saber cómo se encuentran estas zonas y que tienen una relevancia en la producción de un incendio. Estas funciones reproducen la información que el experto ha indicado sobre los casos en los que se reciben estos valores, para poder devolver el resultado acorde a esta configuración de las variables de entrada. Estas funciones están implementadas mediante condicionales encadenados en su totalidad. La

implementación permite realizar una predicción exacta según la lógica que el experto ha indicado. Esta razón indica que este módulo actualmente implementado es viable y actualmente se encuentra en funcionamiento.

El planteamiento por el que se desea comparar este módulo con el planteado en este proyecto final es que es una implementación que se encuentra acoplada en la aplicación. Se dice que esta acoplada por los siguientes motivos: primero se encuentra compilada y cualquier cambio que se produjera tendría que volverse a compilar así como ser sustituida en el servidor que se encuentra por una nueva versión de este módulo compilado, segundo motivo es que es bastante compleja de comprender una vez implementada, puesto que se tratan de bastantes condiciones lo que genera una gran dificultad para poder visualizar correctamente el lugar donde se desearía realizar el cambio, corrección o la implementación de una nueva condición a tener en cuenta. Por otro lado, es propensa a errores por parte de un desarrollador que desee realizar un cambio, debido a la gran complejidad y difícil comprensión de estas generaría una gran posibilidad de realizar un error sobre este cambio.

4.4 Planteamiento de la solución

Una vez visto las causas por las que se piensa que se podría obtener una mejora mediante una tecnología de desarrollo distinta, se va a plantear una solución a estos problemas que intente mejorar la implementación actual, se planteará una posible mejora a este con las indicaciones indicadas por los tutores de este proyecto, junto con los conocimientos adquiridos en el grado de ingeniería informática tras el aprendizaje general de esta carrera y de forma más específica teniendo en cuenta algunas materias más cercanas a la resolución de este problemas como serían las asignatura de ISW la cual aporta una visión de cómo enfrentar un problema software basado en la lectura o interpretación de este, SIN la cual plantea técnicas de inteligencia artificial para resolver una serie de problemas entre los cuales se encuentran los sistemas expertos y TIA en la cual de la misma forma que en SIN se plantea la forma de solucionar problemas mediante inteligencia artificial y de igual manera también se ven los sistemas expertos sobre los que va a tratar el proyecto.

En base a la lectura del problema se puede apreciar como existe una tecnología enfocada a este tipo de problemas, los sistemas expertos, los cuales son una rama de la inteligencia artificial que tratan problemas de esta índole, la automatización del conocimiento de un experto humano con un claro conocimiento del tema y estrategia definida.

Una definición de sistemas expertos sería la siguiente: Un sistema experto es un sistema informático capaz de razonar y actuar al nivel de una persona experta en un campo o actividad en específico[7].

Se aprecia que la definición contiene ciertas similitudes con el módulo que se desea sustituir. De forma paralela de igual manera que sucede en el módulo actual, en caso de querer saber porque se ha producido un valor de predicción, es posible gracias a que con algunos sistemas expertos se puede averiguar la traza que nos ha permitido llegar a este resultado. Algo que es relevante para nuestro problema debido a la posibilidad de averiguar futuramente la consecuencia de esta traza en una predicción, ya que en el caso de este proyecto el resultado de una predicción es un resultado exacto en base a la lectura de las distintas variables de entrada y se quiere estar seguro en todo momento que ante



Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial

un error se pueda obtener la procedencia de este. Por esta razón para el proyecto actual se han descartado otras técnicas de inteligencia artificial que no permitan obtener de forma sencilla y segura el porqué de este resultado. En base a la experiencia y criterio que se tiene sobre este problema, tanto por parte de los tutores como propio, se ha decidido que un sistema experto sería una implementación que podría solventar algunos de los inconvenientes planteados en la implementación actual, así como ser más acertada para el caso que se plantea, debido a la similitud con los problemas que trata de resolver un sistema experto.

Se ha elegido el sistema experto CLIPS, el cual es un lenguaje de programación propio, con un motor de inferencia incluido que permite realizar el desarrollo de un sistema experto de una forma sencilla y completo. La elección de este sistema será porque cumple con los requerimientos de este proyecto y se tiene conocimiento sobre el uso de este.

La solución que se va a plantear será generar un sistema experto en CLIPS que tendrá la posibilidad de sustituir al módulo anterior de predicción. Este se encargará de migrar el conocimiento del módulo anterior que actualmente es un programa escrito en el lenguaje de programación c, cuyo funcionamiento radica en la ejecución de funciones con una lógica de respuesta basada en condicionales las cuales representan la toma de decisiones basadas en las variables de entrada. Puede verse a continuación en la Figura 7 Parte de la función de predicción de riesgo un ejemplo de una función de riesgo de las cuatro de ellas (riesgo, tormentas, tipología y disponibilidad de combustible) la cual se encarga de devolvernos una predicción del riesgo de incendio.

```
int riesgo(int A, double B, double D, double E){
    if(A >= 4){
        if(B < 30){
            // A 4 0 5; B 1 0 2; D 1 2 3 4 0 5; E 4 5 0 6
            if(E >= 30){
                return 0;
            } else {
                // A 4 0 5; B 1 0 2; D 3 4 0 5; E 1 2 0 3
                if(D >= 25){
                    return 1;
                }
                // A 4 0 5; B 1 0 2 D 1 0 2; E 1 2 0 3
                else {
                    return 0;
                }
            }
        }
    } else if(B >= 30 && B < 33){
        // A 4 0 5; B 3; D 1 0 2
        if(D < 25){
            return 0;
        } else {
            // A 4 0 5; B 3; D 3 4 0 5; E 4 5 0 6
            if(E >= 30){
                return 1;
            }
            // A 4 0 5; B 3; D 3 4 0 5; E 1 2 0 3
            else{
                return 2;
            }
        }
    }
    } else if(D >= 33 && B < 42){
        // A 4 0 5; B 4 0 5; D 1 0 2
        if(D < 25){
            return 1;
        } else {
            // A 4 0 5; B 4 0 5; D 3 4 0 5; E 4 5 0 6
            if(E >= 30){
                return 1;
            }
            // A 4 0 5; B 4 0 5; D 3 4 0 5; E 1 2 0 3
            else{
                return 2;
            }
        }
    }
    } else {
        // A 4 0 5; B 6; D 3 4 0 5
        if(D >= 25){
            return 3;
        }
        // A 4 0 5; B 6; D 1 0 2
        else {
            return 1;
        }
    }
}
```

Figura 7 Parte de la función de predicción de riesgo

En la Figura 7 Parte de la función de predicción de riesgo, se encuentra un trozo muy reducido de esta función de predicción del riesgo de producción de un incendio, puede apreciarse como se encuentra realizada mediante condicionales encadenados los cuales ante las distintas configuraciones generadas por las variables de entrada correspondiente a los valores de lectura de los sensores, que nos proporciona el primer módulo explicado de esta aplicación se generan un resultado correspondiente a la predicción.

Se plantea realizar el traslado del conocimiento que representan estas funciones implementadas en el lenguaje de programación c, a un programa en CLIPS que permita representar este conocimiento mediante reglas y hechos y que permita realizar esta predicción mediante la información del experto de una forma más clara y fiel a la realidad.

5 Diseño y materiales

En esta sección se planteará la forma de enfrentar el problema, se diseñará un diagrama de este y se planteará la forma en que se enfrentará. También se describirán los materiales, herramientas y programas que se utilizarán para poder abordar el problema.

5.1 Material utilizado

Para el desarrollo de todo el proyecto se trabajará con un ordenador con 16gb de RAM y un procesador i7 de 10^a generación, este será el único elemento hardware que requeriremos para realizar el desarrollo del proyecto en su totalidad. El sistema operativo sobre el que trabajaremos será un sistema operativo Windows 11, aunque podría utilizarse un sistema operativo Windows 10 sin ningún tipo de restricción en la compatibilidad de programas y herramientas de desarrollo que emplearemos, ya que todos ellos serán de Windows 10, que son compatibles con Windows 11. Como segundo sistema utilizaremos una distribución de Linux (Ubuntu 20.4), pero de forma virtualizada mediante la instalación y uso de WSL2 que se describirá más adelante en esta sección, para poder simular una instalación de Ubuntu, debido a que el resto de los módulos donde será implementada la solución requieren ser ejecutados en un sistema Linux.

Para el desarrollo del siguiente proyecto se ha empleado las siguientes herramientas y software que se muestra a continuación:

Software:

- 5.2.1 Subsystem for Linux 2 (WSL 2)
- 5.2.2 Visual Studio Code
- 5.2.3 CLIPS
- 5.2.4 Trello
- 5.2.5 Python
- 5.2.6 yEd - Graph Editor

A continuación, se describirán estos de forma reducida y se explicará para que se han empleado en este proyecto.

5.1.1 Subsystem for Linux (WSL)

WSL son las siglas de para Windows Subsystem for Linux, es una capa de compatibilidad para entornos nativos Linux dentro de Windows. Microsoft prevé a WSL como una “herramienta enfocada principalmente a desarrolladores que trabajen con proyectos de código abierto” [8]. WSL usa menos recursos que una máquina virtual, es una de las formas más directas de ejecutar Linux en un entorno Windows, al tiempo de permitir a los usuarios trabajar con las aplicaciones de Windows y Linux sobre el mismo conjunto de archivos. La Arquitectura Fue creada para la versión de Ubuntu 16.04, anunciada en 2016.



Lamentablemente WSL, aunque sea muy útil, tenía varias carencias: todas las ejecuciones eran interpretadas, no teníamos acceso a todo el hardware de la máquina, el rendimiento general era bastante malo, etc. Con esto aparece wsl2 en 2019 que ya no sería interpretado, tendría todo el kernel de Linux nativo y corriendo en Windows a todo rendimiento. WSL2 fue recientemente liberado para todos los usuarios de Windows 10 en la versión 2004.[9]

Durante todo el tiempo vamos a estar hablando de WSL en la versión 2 (WSL 2).

Para instalarlo la forma por excelencia es desde la PowerShell. Para que funcione correctamente deberemos asegurarnos de que tenemos una versión de Windows 10 versión 2004 o posterior (en nuestro caso Windows 11). La forma de realizarlo será desde la terminal ejecutar el comando:

A screenshot of a PowerShell terminal window. The title bar reads "PowerShell". The command prompt shows the text "wsl --install" entered on a single line.

Figura 8 Instalación de WSL2 en Windows 11

Este comando ejecutara las acciones necesarias para descargar el kernel de Linux más reciente, establecerá wsl2 como predeterminado e instalara una distribución desde Linux automáticamente (Ubuntu, en nuestro caso 20.04). Una vez termine el proceso se requiere un reinicio, con lo que ya tendremos este sistema listo para su utilización.

Las partes más relevantes de este sistema virtualizado sería el propio sistema de archivos de Linux, que se encontrará bajo el sistema gestor de archivos de Windows Figura 9 Sistema de ficheros de WSL 2 Ubuntu y una terminal con la que podemos interactuar directamente como si en un sistema operativo Linux Ubuntu se tratara, podemos ver como se muestra este en la Figura 10 Terminal WSL 2 Ubuntu 20.04. Con este sistema de archivos y terminal se podrá realizar todo el desarrollo que se requiere para poder desarrollar nuestro proyecto sobre Python 3 así como las herramientas que utiliza del propio sistema operativo Ubuntu.

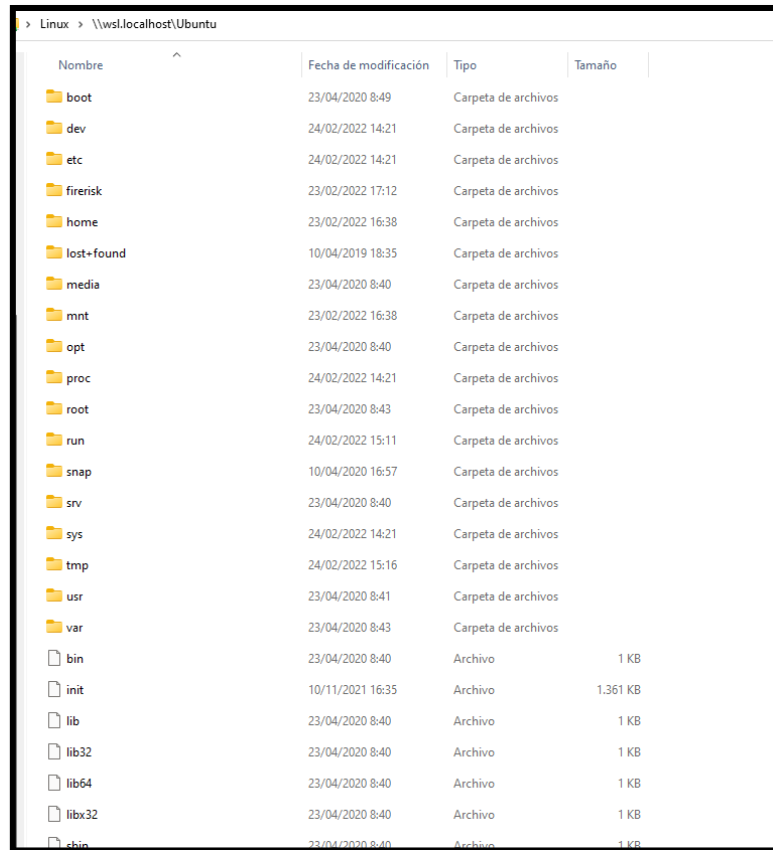


Figura 9 Sistema de ficheros de WSL 2 Ubuntu

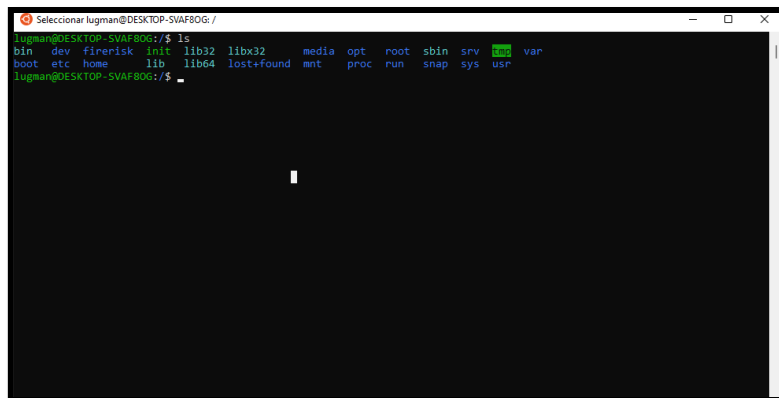


Figura 10 Terminal WSL 2 Ubuntu 20.04

5.1.2 CLIPS

Clips es una herramienta de desarrollo para sistemas basados en reglas [10], con esta herramienta tenemos un acceso a un entorno de desarrollo y producción de sistemas expertos [11].

Fue creado a partir de 1984, en el Lyndon B. Johnson Space Center de la NASA.

CLIPS es un Shell para el desarrollo de sistemas expertos que utiliza una estrategia de control irrevocable con encadenamiento hacia adelante. Este ofrece compatibilidad para la integración con c/c++ (en ambas direcciones).

Arquitectura de CLIPS

CLIPS contiene una base de hechos, que representa el conocimiento del dominio de problema sobre el que se trabaja, estos serían los datos, hechos establecidos y metas a alcanzar, también dispone de un motor de inferencia que decide como se aplican las reglas en base al conocimiento del problema y su solución, así como una base de reglas que representan la forma en la que se desea conseguir el objetivo. Como se ilustra en la siguiente imagen Figura 11 Arquitectura de CLIPS, puede apreciarse como el motor de inferencia está constantemente revisando tanto la base de hechos, como la base de reglas para poder realizar la toma de decisiones y así conseguir llegar a la solución en caso de existir.

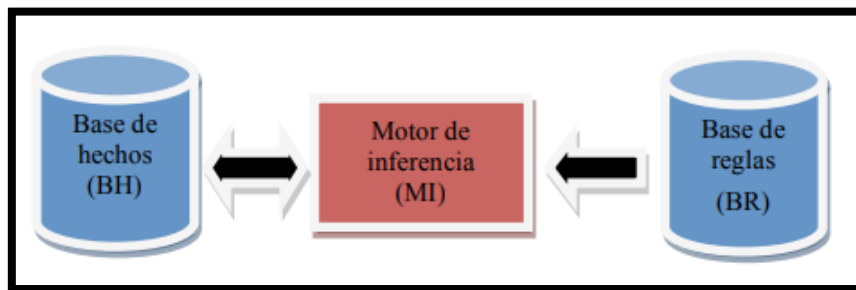


Figura 11 Arquitectura de CLIPS

Sintaxis de CLIPS

Para trabajar con él se describirán los comandos y sintaxis más relevante que se han empleado en el desarrollo de este proyecto.

Comandos:

- **reset:** inicialización. Elimina todas las activaciones de la agenda, todos los hechos de la lista de hechos y todas las instancias de clases definidas por el usuario
- **run:** ejecución. Pone en marcha la ejecución de las reglas ejecutándose las instancias y reglas establecidas en el programa.
- **clear:** limpiar la base de conocimiento.

Reglas: La forma empleada para la creación de reglas en CLIPS.

```
(defrule <nombre-regla>
  [<comentario>]
  [<propiedades>]
  <elemento-condicional>*
  =>
  <acción>*)
```

Define una regla en CLIPS.

Hechos: La forma empleada para insertar hechos en CLIPS.

```
(assert <RHS-pattern>+)
```

Agrega un hecho a la lista de hechos.

Test: *La forma empleada para la evaluación de las condiciones en CLIPS.*

Las restricciones de campo utilizadas dentro de los CE de patrones permiten aplicar restricciones muy descriptivas a la coincidencia de patrones. Se proporciona capacidad adicional con el elemento condicional de prueba.

CLIPS será el software basado en reglas que utilizaremos para nuestro sistema experto, debido a que nos proporciona lo necesario para poder realizar un sistema experto completo y se conoce su utilización debido a que ha sido estudiado y utilizado durante el grado de ingeniería informática.

5.1.3 Visual studio code

Editor de código fuente independiente que se ejecuta en Windows, macOS y Linux. Principalmente utilizado para el desarrollo web, pero con extensiones admite su uso con casi cualquier lenguaje de programación.

Se utilizará el editor de código de Microsoft el cual nos permitirá realizar el desarrollo de nuestras reglas en CLIPS, así como visualizar y trabajar con toda la parte de código de este proyecto.

5.1.4 Trello

Se utilizará Trello como herramienta de gestión de tareas, la cual es una herramienta online fácil de usar que nos permitirá gestionar las tareas que realizaremos.

5.1.5 Python

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se



puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

La aplicación sobre la que se trabajará está realizada en este lenguaje de programación por lo que se tendrá que hacer uso de este lenguaje para la integración de nuestro módulo en clips en esta aplicación.

En la integración y las pruebas de este proyecto además de las librerías empleadas por el desarrollo anterior sobre el que está se implementará se emplearán las siguientes librerías:

- ***clipsy***

Es una librería que hace de intermediaria sobre Python y el api de CLIPS nativa escrita en c. Mediante está conseguiremos implementar las reglas e insertar los hechos que se desarrollarán en este proyecto, así como replicar el motor de inferencia de CLIPS que permita resolver la predicción con las reglas y hechos insertados.

Esta librería será una parte muy importante del proyecto por lo que la elección de esta es crucial, se ha escogido esta librería puesto que es de la más empleadas en la actualidad la cual ha recibido soporte recientemente.

Esta librería parte de una clase denominada Environment, la cual integra todas las funcionalidades que ofrece la librería, dentro de éstas se encuentran las siguientes funciones que se emplearán en este proyecto.

- **build:** equivalente al comando build de CLIPS. Permite que mediante una cadena de texto establezcamos código de CLIPS con su sintaxis.
- **clear:** realiza la funcionalidad de la función clear de CLIPS.
- **load:** permite leer e interpretar código de CLIPS que será recibido desde un archivo.
- **reset:** realiza la funcionalidad de el comando reset de CLIPS.
- **facts:** Permite recuperar una lista iterable de los hechos que se encuentran en el entorno.
- **run:** Ejecuta el entorno, de la misma forma que el comando run de CLIPS.

Esta librería se podrá llevar a cabo el despliegue de las reglas en CLIPS, sin la necesidad de usar el propio entorno de CLIPS.

- ***random:***

Esta librería de Python genera valores aleatorios, la emplearemos para la generación de números aleatorios en la parte de pruebas.

- **time:** Nos permitirá obtener información de tiempo e instante, será utilizada para medir tiempos.
- **difflib:** librería de Python que permite la comparación de archivos y conteo de sus diferencias.

5.2.6 yEd - Graph Editor

Software multiplataforma de código abierto compatible con Windows, Gnu/Linux y Mac OS. Permite el desarrollo de diagramas de propósito general.

En este proyecto se ha empleado para el desarrollo del diagrama de flujo con el que se trabajará en el proyecto.

5.2 Diseño del problema

Como se comentó en la sección anterior, nuestro proyecto se basa en la predicción del riesgo de incendio en base a unas variables de entrada con las que se predice el resultado del problema, estas variables de entrada tienen una configuración que cuando se produce se produce un resultado que se interpreta como la solución. Esta configuración viene dada por el experto en la materia quien ha proporcionado los valores de estas configuraciones. Para plasmar este resultado de forma visual e intuitiva se optará por un diagrama de flujo, el cual representará el flujo que debe seguir nuestro predictor para obtener este resultado. Por razones de privacidad a petición del experto se mostrará el diagrama de flujo del experto censurado.

Una vez plasmada la solución de este y con la información que se sabe de este, se planteará una solución basada en un sistema experto bajo la tecnología de CLIPS, el cual llevaremos a cabo mediante reglas con una nomenclatura acorde a lo que realizan las reglas para poder entender está y que pueda ser modificada o ampliada de forma futura.

Como este módulo será empleado por otro escrito en el lenguaje de programación Python, llevaremos a cabo la importación de una librería que nos permita simular el motor de inferencia de CLIPS, para así poder ejecutar las reglas desarrolladas en este lenguaje.

5.3.1 Forma de trabajo

Para comenzar el desarrollo de la solución mediante el gestor de tareas trello se establecerán las tareas a realizar, separándolo en distintas tareas a las que nos vamos a enfrentar. Se repartirá de forma equitativa la cantidad de condiciones que deseamos plantear en nuestro diagrama para cada tarea y así poder llevarlo a cabo de forma correcta y con menor riesgo de equivocación. Una vez establecido se llevarán a cabo las tareas hasta realizar este diagrama. Para el desarrollo de toda la solución se ha seguido la misma estrategia que ha sido dividir el trabajo en tareas y subtareas para así ir realizando estas tareas de forma continua y clara.

5.3.2 Diagrama de Flujo

El resultado del diagrama como así comentamos será censurado por motivos de privacidad, pero podemos apreciar que este resultará con la siguiente estructura:

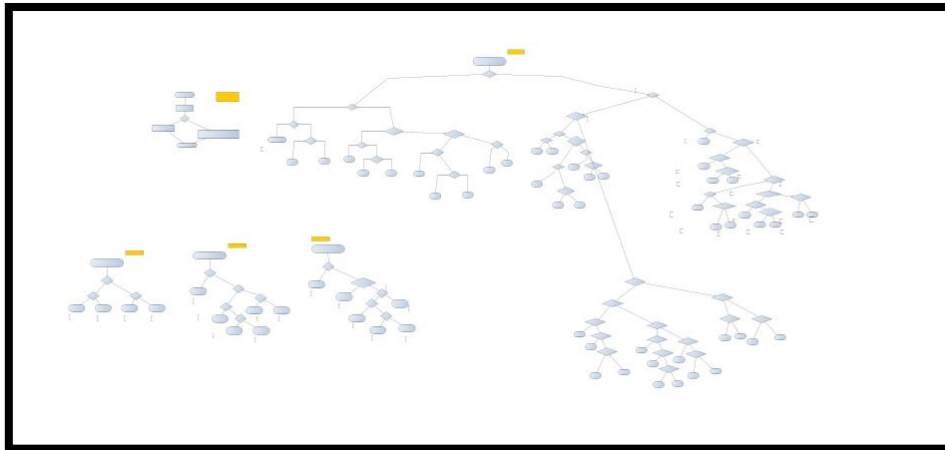


Figura 12 Diagrama de flujo de nuestro predictor

En el diagrama de la Figura 12 Diagrama de flujo de nuestro predictor, puede apreciarse que contiene varios diagramas, el de la izquierda superior muestra la entrada al módulo que realiza la llamada al predictor, el módulo que realiza la llamada al predictor, extrae las variables de entrada y toma la decisión de la llamada a nuestro módulo de predicción o la respuesta directa sin la realización de la predicción en base a un cierto valor de la variable carácter de la precipitación. Cuando no se produce la respuesta directa en base a la variable de precipitación se procede a llamar el módulo de predicción el cual incluye las siguientes funciones: función de riesgo de incendio, cuyo diagrama es el de mayor tamaño y en la Figura 12 Diagrama de flujo de nuestro predictor, se encuentra en la parte superior derecha, en la parte inferior y en disposición de izquierda a derecha de forma consecutiva puede apreciarse la representación del diagrama correspondiente a las funciones de predicción de tormenta, tipología y de disponibilidad de combustible.

Este diagrama será empleado para poder apreciar de forma clara la estructura que tiene nuestro predictor, del cual se parte y así poder llevar a cabo el desarrollo del predictor en clips.

5.3.3 Nomenclatura a seguir

Una vez plasmado en el diagrama la estructura lógica del predictor, se procederá a plantear la forma de realizar las reglas de nuestro sistema experto, para ello y en base a la información recibida en la formación académica se optará por reglas del estilo simples y de longitud no superior a las 10 líneas, de forma que no se incluya un exceso de condicionales de éstas en ellas y puedan separarse en otras más sencillas, lo que favorecerá la corrección y modificación de éstas, también se usarán para las funciones más extensas un nombre coherente de la regla que identifique a ésta.

Podemos apreciar un ejemplo de estas en la Figura 13 Ejemplo básico de regla en clips

```
(defrule Baja
  (CaracterPrecipitacion ?CaracterPrecipitacion)
  (test (< ?CaracterPrecipitacion 3) )
  =>
  (assert (tree CP_Baja))
)
(defrule CP_Bajo_T_Alto
  (tree ?tree)
  (Temperature ?Temperature)
  (test (eq ?tree CP_Baja))
  (test (>= ?Temperature 42))
  =>
  (assert (result 3))
)
```

Figura 13 Ejemplo básico de regla en clips

Como puede apreciarse en la Figura 13 Ejemplo básico de regla en clips, podemos ver cómo está primera regla hace referencia de forma clara a su estado dentro de la lógica de predicción, debido su interpretación que en caso de tener un carácter de precipitación bajo entraremos en esta regla y de allí resultará en la inserción de una nueva variable **tree** que será la que sea condición necesaria para entrar en la segunda regla mostrada en la Figura 13 Ejemplo básico de regla en clips, está segunda en caso de proceder de la anterior que inserta la variable CP_BAJA y adicionalmente cumplirse la condición de que haya una temperatura mayor o igual a 43 grados pcedera a insertar la variable resultado que dará finalización al programa con dicho valor de respuesta.

Como hemos mencionado con anterioridad el diseño de estas reglas se plantea cumpliendo el patrón de ser breves y con un número reducido de condiciones, así como con un nombre descriptivo para poder entender dónde nos encontramos con facilidad.



6 implementación, integración y pruebas

6.1 Introducción a la sección

En esta sección se llevará a cabo el desarrollo del predictor, para el cual se implementará de forma incremental las reglas que permitan realizar un predictor en CLIPS. Una vez realizadas estas reglas, que originalmente serán desarrolladas y probadas en la propia interfaz de clips se trasladarán estas a un archivo que será leído y ejecutado por un programa en el lenguaje Python. Se integrará este programa en Python en el módulo que utiliza el predictor, por lo que sustituirá al anterior predictor. Por última instancia se llevarán a cabo una serie de prueba que permitan comprobar la exactitud de este.

6.2 Implementación

6.2.1 Partes a implementar

Cómo se mencionó en la sección anterior se van a implementar las reglas que permitan realizar la funcionalidad que actualmente está realizando el predictor, este lo está realizando mediante funciones, las cuales incluyen en su totalidad condicionales que permiten devolver el resultado de la predicción de la función en cuestión. Para realizar esto se va a separar de forma clara las funciones las cuales tiene el predictor. Funciones de predicción del predictor actual:

- **Riesgo:** Indica el riesgo de incendio en base la información del experto, este método es el de mayor tamaño y por lo tanto será el que mayor esfuerzo implica a la hora de decidir cómo se llevarán a cabo las reglas que implementen esta funcionalidad, entre las especificaciones de estas reglas se tendrá en cuenta los nombres de las reglas para que así se obtenga un nombre intuitivo de esta regla, así como la inserción de hechos que indiquen desde que regla procedemos para poder avanzar hacia la siguiente regla a inferir por el motor de inferencia.
- **Tormenta:** Indica el riesgo de tormenta, este método es de tamaño reducido y no plantea un reto mayor que el traslado de condiciones en el lenguaje c, a unas reglas de CLIPS que tras su ejecución proporcionen un resultado equivalente.
- **Tipología:** Indica el tipo de tipología en base a la información proporcionada por el experto, esta al igual que la función anterior únicamente implica el traslado de condicionales en c, a unas reglas de CLIPS que tras su ejecución proporcionen un resultado equivalente.
- **Disponibilidad de combustible:** Indica la disponibilidad de combustible, este método al igual que los dos anteriores es de tamaño y complejidad reducida e implica el traslado de reglas en c, a unas reglas de CLIPS que tras su ejecución proporcionen un resultado equivalente.

6.2.2 Proceso de implementación

La forma en la que se han realizado estas reglas, ha sido la creación de estas en archivos con la extensión “.clp” que es el formato de archivo empleado por CLIPS, para ello se ha empleado un archivo con el nombre de la función que representa y que contiene todas las reglas que realizan la predicción que dicha función realizaba.

Se han desarrollado y probado de forma breve y reducida en el entorno CLIPS con dicho motor de inferencia para posteriormente ser trasladadas al módulo que emplee estas y así poder asegurar la exactitud de estas con una mayor cantidad de datos.

Se han creado las reglas que realizan la funcionalidad del diagrama de clase creado y puesto casos básicos de forma analítica que permitan obtener un resultado para cada valor de configuración. Como hemos comentado antes esto se ha realizado en primera instancia en el entorno de CLIPS, en el cual se ha cargado cada uno de estos archivos que contienen las reglas y comprobado que todos los casos posibles se cumplieran antes de dar por finalizada la primera instancia de la creación de estas reglas correspondientes a las funciones de predicción.

Para mostrar el desarrollo de estas funciones a reglas en clips que representen su funcionalidad, separaremos el desarrollo de las funciones con una estructura sencilla que se han mencionado anteriormente (tormenta, tipología y disponibilidad de combustible) y la predicción de riesgo de incendio que implica mayor complejidad.

Desarrollo de la implementación

Se comenzará por las funciones sencillas. Se ilustrará una de ellas la cual es la más reducida, esta sería la de riesgo de tormenta, se ha escogido esta para poder mostrarla en su totalidad, tanto la implementación actual como las reglas que llevarán a cabo la implementación de esta función, que también por su tamaño se podrá ilustrar en su totalidad. Las otras dos restantes que también son sencillas y de tamaño reducido serían: tipología y disponibilidad de combustible. El procedimiento que se realizará para implementar estas será igual que la del riesgo de tormenta, pero con la lógica propia de esta función y ambas con un tamaño algo mayor que esta función de riesgo de tormenta.

Podemos ver esta función en la siguiente imagen, véase la Figura 14 Función tormentas, módulo de predicción c.

```
int tormentas(double G){
    if(G < -2){
        // G 1
        if(G < -5){
            return 3;
        }
        // G 2
        else{
            return 2;
        }
    } else {
        // G 3
        if(G < 0){
            return 1;
        }
        // G 4
        else {
            return 0;
        }
    }
}
```

Figura 14 Función tormentas, módulo de predicción c

Como puede apreciarse está compuesta de condicionales if y else, los cuales tendremos que trasladarlos a reglas de CLIPS. A continuación, mostraremos la representación de esta función en reglas de CLIPS, véase la siguiente imagen, Figura 15 Archivo con reglas para predecir riesgo de tormenta.

```
(defrule storm1
  (Lifted ?Lifted)
  (test (and (< ?Lifted -2) (< ?Lifted -5)))

  =>
  (assert (result 3))
)

(defrule storm2
  (Lifted ?Lifted)
  (test (and (< ?Lifted -2) (>= ?Lifted -5)))

  =>
  (assert (result 2))
)

;Normal
(defrule storm3
  (Lifted ?Lifted)
  (test (and (>= ?Lifted -2) (< ?Lifted 0)))

  =>
  (assert (result 1))
)

(defrule storm4
  (Lifted ?Lifted)
  (test (and (>= ?Lifted -2) (>= ?Lifted 0)))

  =>
  (assert (result 0))
)
```

Figura 15 Archivo con reglas para predecir riesgo de tormenta

Como se puede apreciar en la Figura 15 Archivo con reglas para predecir riesgo de tormenta, en lugar de condicionales (if y else), se han creado reglas compuestas por operadores los cuales representan las condiciones necesarias para que esta regla cumpla con la condición de entrada de los condicionales if y else correspondiente y permitan al motor de inferencia averiguar si esta regla debe ser activada, y por consecuencia realizar la acción correspondiente a la entrada del condicional if y else que representan. La activación de una regla correspondiente a un resultado final implica la inserción de un hecho **resultado** con el valor correspondiente, el cual se leerá para poder averiguar cuál ha sido el resultado de la predicción una vez ya no hayan más reglas que ejecutar, esto es posible debido a que el predictor siempre obtendrá un valor de variables de entrada (que en el caso de las reglas de CLIPS serán llamada hechos) que permitan obtener un único resultado sin ambigüedad y con seguridad, por lo que el hecho **resultado** una vez insertado podrá ser leído de forma correcta y ser tomada como un resultado inequívoco.

Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial

A continuación, se mostrará como se va a implementar la función riesgo de incendio que en el caso de este proyecto será la que mayor implicación a nivel de desarrollo nos ha planteado. Se mostrará a continuación un trozo parcial de este método, el cual será traducido en reglas con el diseño de estas de la forma mencionada anteriormente. En la siguiente imagen podemos apreciar una parte de esta función riesgo de incendio.

```
int riesgo(int A, double B, double D, double E){
    if(A >= 4){
        if(B < 30){
            // A 4 o 5; B 1 o 2; D 1 2 3 4 o 5; E 4 5 o 6
            if(E >= 30){
                return 0;
            } else {
                // A 4 o 5; B 1 o 2; D 3 4 o 5; E 1 2 o 3
                if(D >= 25){
                    return 1;
                }
                // A 4 o 5; B 1 o 2 D 1 o 2; E 1 2 o 3
                else {
                    return 0;
                }
            }
        } else if(B >= 30 && B < 33){
            // A 4 o 5; B 3; D 1 o 2
            if(D < 25){
                return 0;
            } else {
                // A 4 o 5; B 3; D 3 4 o 5; E 4 5 o 6
                if(E >= 30){
                    return 1;
                }
                // A 4 o 5; B 3; D 3 4 o 5; E 1 2 o 3
                else{
                    return 2;
                }
            }
        } else if(B >= 33 && B < 42){
            // A 4 o 5; B 4 o 5; D 1 o 2
            if(D < 25){
                return 1;
            } else {
                // A 4 o 5; B 4 o 5; D 3 4 o 5; E 4 5 o 6
                if(E >= 30){
                    return 1;
                }
                // A 4 o 5; B 4 o 5; D 3 4 o 5; E 1 2 o 3
                else{
                    return 2;
                }
            }
        } else {
            // A 4 o 5; B 6; D 3 4 o 5
            if(D >= 25){
                return 3;
            }
            // A 4 o 5; B 6; D 1 o 2
            else {
                return 1;
            }
        }
    }
}
```

Figura 16 Función riesgo de incendio modulo c

En la anterior imagen Figura 16 Función riesgo de incendio modulo c, se puede apreciar como aun siendo una reducida parte de la función está ya implica un mayor número de condiciones y que de ser trasladadas de la misma manera que en las dos funciones anteriores nos encontraríamos ante reglas extremadamente complejas tanto a nivel de comprensión de estas, cómo a nivel de ejecución debido a que el motor de inferencia deberá testear en todo momento que estás no se están cumpliendo con todos los condicionales que hemos mencionado antes, por lo que se ha optado por realizarlas de forma reducida insertando un hecho llamado **tree** que indique en que regla nos

encontramos y para poder entender con mayor sencillez estás les pondremos un nombre descriptivo que haga razón a estas. A continuación, se ilustrará la imagen de un trozo del archivo que contiene estas reglas que traducen el método riesgo de incendio, véase la siguiente imagen.

```
1 (defrule Baja
2
3   (CaracterPrecipitacion ?CaracterPrecipitacion)
4   (test (< ?CaracterPrecipitacion 3) )
5   =>
6   (assert (tree CP_Baja))
7 )
8 (defrule CP_Bajo_T_Alto
9
10  (tree ?tree)
11  (Temperature ?Temperature)
12  (test (eq ?tree CP_Baja))
13  (test (>= ?Temperature 42))
14  =>
15  (assert (result 3))
16 )
17 ;174
18 (defrule CP_Bajo_T_Medio
19
20  (tree ?tree)
21  (Temperature ?Temperature)
22  (test (eq ?tree CP_Baja))
23  (test (< ?Temperature 42))
24  (test (>= ?Temperature 25))
25  =>
26  (assert (tree CPBajoTMedio))
27 )
28 ;176
29 (defrule CP_Bajo_T_Medio_V_Alto
30
31  (tree ?tree)
32  (WindGustMax ?WindGustMax)
33  (test (eq ?tree CPBajoTMedio))
34  (test (>= ?WindGustMax 35))
35  =>
36  (assert (result 3))
37 )
38 ;180
39 (defrule CP_Bajo_T_Medio_V_Medio
40
41  (tree ?tree)
42  (WindGustMax ?WindGustMax)
43  (test (eq ?tree CPBajoTMedio))
44  (test (>= ?WindGustMax 25))
45  (test (< ?WindGustMax 35))
46  =>
47  (assert (result 2))
48 )
49 ;184
50 (defrule CP_Bajo_T_Medio_V_Bajo
51
52  (tree ?tree)
53  (WindGustMax ?WindGustMax)
54  (test (eq ?tree CPBajoTMedio))
55  (test (< ?WindGustMax 25))
56  (test (< ?WindGustMax 35))
57  =>
58  (assert (result 1))
59 )
```

Figura 17 Archivo que contiene reglas del método riesgo de incendio

Como puede apreciarse en la Figura 17 Archivo que contiene reglas del método riesgo de incendio, la primera regla se llama Baja e inserta un hecho que se llama **tree** con valor **CP_Baja**, este valor del hecho **tree** hace referencia a **carácter de la precipitación baja**. Este intervalo que indica que el carácter de la precipitación menor que 3 es un valor de carácter de la precipitación bajo, se ha establecido en base a la información

proporcionada por el experto, por lo que en este caso es un valor que hace referencia al enunciado del problema real. Este criterio de nomenclatura ha sido aplicado a todas las reglas aquí implementadas.

Se continuará explicando la siguiente regla que será **CP_Bajo_T_Alto**, que hace referencia a **Carácter de la precipitación baja y temperatura alta**, requiere para su entrada en la regla, que se encuentre insertado el hecho **tree** con valor **CP_Baja** correspondiente a la ejecución de la regla anterior, de esta forma hemos gestionado los condicionales del método riesgo de incendio, en el caso de esta regla mencionada anteriormente **CP_Bajo_T_Alto**, es una regla que directamente inserta un resultado final, en este caso inserta el hecho **resultado** con valor 3 que es una predicción válida de este predictor. Este método contiene muchos condicionales encadenados por lo que la siguiente regla de la Figura 17 Archivo que contiene reglas del método riesgo de incendio, no es una regla final que obtiene un resultado, sino que inserta un nuevo hecho **tree** con valor **CPBajoTMedio**, de forma que otra regla sea la que se encargue de ser activada en presencia de este hecho, junto con el resto de las condiciones a cumplir.

Esta es la forma de trabajar que hemos implementado en la creación de las reglas en CLIPS, por lo que de forma similar se irá generando el árbol de dependencias que lleve a un resultado de predicción.

Implementación en de CLIPS en Python

Una vez realizado todo el desarrollo correspondiente a las reglas de estas 4 funciones se ha pasado a realizar una serie de pruebas simples desde el entorno CLIPS para comprobar el correcto funcionamiento de este predictor, así como para detectar y corregirlos a la hora de crear estas reglas, esta serie de pruebas no tienen mayor complejidad que valores analíticamente extraídos del diagrama de flujo, de esta forma y a falta de realizar más pruebas se ha dado por válidas estas reglas.

Tras la realización de estas reglas se han procedido a implementar en un programa realizado en Python, el cual se encargará de realizar la llamada a estos métodos, como no son métodos de Python sino de clips hemos empleado la librería **clipsy**, anteriormente mencionada en la sección 4 Diseño y materiales, que permite simular un motor de inferencia de clips en Python.

A continuación, se describirá como se ha empleado esta librería Python para poder realizar el traslado de las reglas en clips a nuestro programa en Python, así como realizar la inserción del hecho correspondiente a las variables de entrada y leer el resultado correspondiente a el hecho **resultado** que se corresponde con el valor de retorno de la función.

Para utilizar esta biblioteca debemos importarla en nuestro programa. Una vez realizada la importación de esta librería realizaremos la instanciación de un variable con el valor del resultado de la ejecución del método **Environment**, de esta librería y así poder tener un entorno es cual será el motor de inferencia y proporcionará las funciones necesarias para realizar la ejecución de nuestro sistema experto en CLIPS.

Para cada uno de los archivos correspondientes a las funciones del predictor (riesgo de incendio, riesgo de tormenta, disponibilidad de combustible y tipología) se creará una

función y en ella contendrá las siguientes configuraciones: un entorno sobre el cual insertaremos las variables de entrada como hechos, esto realizaremos como una cadena de texto que contendrá los hechos sobre los que queremos obtener el resultado. A continuación, mostraremos una imagen que representa el método para la predicción de riesgo.

```
def risk(A,B,D,E):
    env = clips.Environment()
    rule="""(defrule startup
=>
(assert
(CharacterPrecipitacion {A})
(Temperature {B:.2f})
(WindGustMax {D:.2f})
(RH {E:.2f}))
)""".format(A=A,B=B,D=D,E=E)

    env.build(rule)
    env.load(path="Risk.clp")
    env.reset()

    res = env.run()

    return get_result(env)
```

Figura 18 Llamada a función CLIPS riesgo

Como se puede apreciar en la variable **env** instanciamos el entorno, a continuación, se insertan los hechos con el valor de las variables de entrada que recibirá este método como hechos de clips, éstos serán truncados a 2 decimales ya que así se decidió en el módulo anterior y se mantendrá esta decisión. A continuación, se introducirán estas reglas en el entorno y cargamos el archivo con las reglas que se han desarrollado y que se corresponden con las reglas que generan la predicción del riesgo de incendio. Reiniciamos el entorno y ejecutamos este, una vez termine la ejecución de este pasaremos este entorno a un método creado que se encargará de leer el hecho resultado, este proceso se realizará mediante los métodos ofrecidos por esta librería.

```
def get_result(env):
    for fact in env.facts():
        if fact.template.name == 'result':
            return fact[0]
    return None
```

Figura 19 Función obtener resultado

Este método recorre todos los hechos obtiene el valor del que contiene el nombre result, por lo que obtenemos el resultado de la predicción.

6.3 Integración



Para la integración de este proyecto se ha realizado un módulo que permite la utilización de este nuevo predictor en CLIPS, de forma que pueda elegirse entre este o el anterior predictor.

En el momento de ejecutar la aplicación que hace uso de este predictor la ejecución de esta aplicación se realiza mediante la línea de comandos y esta aplicación recibe ciertos parámetros de configuración, entre ellos se indica un parámetro en específico que indica el módulo de predicción al que se hace referencia, mediante este parámetro de configuración se realizará la integración del predictor de este proyecto y la aplicación que lo utiliza, que pondrá en funcionamiento tanto el módulo de predicción como el resto de módulos que utiliza para la correcta ejecución de esta. La forma de trabajar que tiene es indicar el nombre del predictor en el caso anterior este es “CPredictor”, que se encuentra contenido en una carpeta e incluye un programa en Python con el mismo nombre “CPredictor.py”, este programa contiene un método **execute** el cual se encarga de hacer la llamada al predictor haciéndole llegar los parámetros de entrada y recuperando los resultados obtenidos.

Lo que se va a realizar para la integración de este será el almacenamiento de este en una carpeta, que a su vez tenga el mismo nombre del archivo Python que se quiere ejecutar y el cual implementa la llamada a las reglas y la utilización de la librería `clipsy` para la ejecución de estas reglas en Python.

El nombre decidido para seguir la nomenclatura escogida en este programa será “CLIPSPredictor” y este será tanto para la carpeta como para el predictor.

En nuestro predictor “CLIPSPredictor.py” seguirá la estructura del programa “CPredictor.py”, en lo que respecta a llamadas y respuestas. Por lo tanto, este contendrá un método **execute** que será el que comience el proceso de predicción. El método **execute** recibirá unos parámetros de entrada que serán las variables sobre las que se quiere realizar la predicción de forma exacta al programa “CPredictor.py” y devolverá un resultado para la predicción con la misma estructura de respuesta que el programa “CPredictor.py”, estos resultados de salida serán almacenado en una matriz multidimensional que contendrá la predicción de nuestros métodos (riesgo de incendio, riesgo de tormenta, tipología y disponibilidad de combustible), almacenado en la posición correspondiente a las coordenadas de está predicción. En la siguiente imagen se puede apreciar la estructura de datos que contendrá la respuesta.

Cómo puede apreciarse recibe los parámetros de entrada, los almacena en variables para su uso, llama a los métodos correspondientes a la predicción en nuestro caso separadas en las funciones anteriormente descritas y devuelve el resultado de la predicción. Puede apreciarse que la llamada de los métodos solo se realiza si la variable **pre**, correspondiente a **carácter de la precipitación** es menor que 10, es una especificación del experto y ha sido implementada en esta parte para reducir la dificultad de las reglas del predictor.

De la forma en que se ha realizado se ha podido integrar el proyecto sin modificar en ningún caso la aplicación en la que se va a emplear este módulo de predicción y dando la posibilidad de elegir entre este o el anterior sin ninguna restricción.

```

def execute(data_input):
    data_output = [{}, {}, {}, {}]
    for (x,y) in data_input:
        pre = data_input[(x, y)]['pre']
        tmp = data_input[(x, y)]['tmp']
        gus = data_input[(x, y)]['gus']
        hum = data_input[(x, y)]['hum']
        ltd = data_input[(x, y)]['ltd']

        data_output[0][x] = data_output[0].get(x, {})
        data_output[1][x] = data_output[1].get(x, {})
        data_output[2][x] = data_output[2].get(x, {})
        data_output[3][x] = data_output[3].get(x, {})

        risk_res = (-1 if pre >= 10 else risk(pre,tmp,gus,hum)) + 1
        storm_res = (-1 if pre >= 10 else storm(ltd)) + 1
        typology_res = (-1 if pre >= 10 else typology(tmp,gus)) + 1
        dis_comb_res = (-1 if pre >= 10 else dis_comb(tmp,hum)) + 1

        # Riesgo, disponibilidad, tormentas, tipologia
        data_output[0][x][y] = risk_res
        data_output[1][x][y] = storm_res
        data_output[2][x][y] = typology_res
        data_output[3][x][y] = dis_comb_res

    return data_output

```

Figura 20 Función execute de CLIPSPredictor

6.4 Pruebas

Una vez realizado esta implementación e integración, se procederá a realizar una serie de pruebas que verifiquen la exactitud de la respuesta del predictor desarrollado.

Las pruebas se realizarán haciendo uso del módulo anteriormente implementado que nos permitirá obtener un resultado de predicción correcto. Puesto que nuestro predictor se ha basado en la estructura de este. Por esta razón las pruebas que se realizarán serán la comparación de un resultado obtenido por el módulo de predicción de la aplicación y el resultado obtenido por el predictor desarrollado.

Para realizar esto en primera instancia se han realizado dos tipos de prueba: un tipo de prueba de las que realizaremos será la comparación de ambos predictores con unos valores a predecir en los rangos que se esperan recibir las variables de entrada del módulo de predicción, para este proceso lo que se ha realizado es la generación de valores aleatorios en este rango de estas variables de entrada, el otro tipo de prueba que se ha realizado consistirá en predecir los valores reales de la aplicación haciendo uso de está con una ligera modificación para que se permita llamar a ambos predictores y guardando los resultados de ambas predicciones en dos archivos y comparándolos.

Comenzaremos explicando este segundo, la comparación de ambos predictores con datos reales.

Comparación de ambos predictores con valores reales

En la parte de integración se ha visto como se podía integrar este en la aplicación, ahora se va a modificar este programa “CLIPSPredictor.py” de forma que realice la llamada del predictor en CLIPS y también llame al predictor en C, tras obtener ambos valores de predicción, estos se almacenarán en dos ficheros que posteriormente se procederá a comparar.

Utilizando la implementación ya existente procederemos en nuestro programa CLIPSPredictor, a realizar la modificación comentada, se procederá en su método **execute** a llamar al predictor en C y realizar la predicción en CLIPS posteriormente.

```
process = subprocess.Popen(["./CPredictor/incendios"], stdin=subprocess.PIPE, stdout=subprocess.PIPE, universal_newlines=True)
for (x,y) in data_input:
    pre = data_input[(x, y)]['pre']
    tmp = data_input[(x, y)]['tmp']
    gus = data_input[(x, y)]['gus']
    hum = data_input[(x, y)]['hum']
    ltd = data_input[(x, y)]['ltd']
    process.stdin.write(":{:2f} {:.2f} {:.2f} {:.2f} {:.2f} {:.2f}\n".format(x,y,pre,tmp,gus,hum,ltd))
    process.stdin.flush()
    outputLine = process.stdout.readline()
    f = outputLine.split()
```

Figura 21 Llamada al predictor c, predictor de prueba.

Como puede apreciarse en la Figura 21 Llamada al predictor c, predictor de prueba., esto se realiza mediante un proceso el cual se le envían los parámetros por el flujo de entrada y salida de este. Y se recogen los valores correspondientes a la predicción leyendo este flujo de en entrada y salida.

Por otra parte, lo que realizaremos será llamar a él predictor en CLIPS, de forma conjunta para así poder tener ambos resultados de la predicción.

Una vez finalizado este, se almacenarán ambos en archivos de texto.

```
my = open("myResults.txt", "a")
correct = open("correctResults.txt", "a")

my.write("A: {A} B: {B:1f} D: {D:1f} E: {E:1f} G: {G:1} Result: {res_risk} {res_storm} {res_tip} {res_dis}\n"
        .format(A=tr(A),B=tmp,D=gus,E=hum,G=ltd,res_risk=risk_res,res_storm=storm_res,res_tip=typology_res,res_dis=dis_comb_res))
correct.write("A: {A} B: {B:1f} D: {D:1f} E: {E:1f} G: {G:1} Result: {res_risk} {res_storm} {res_tip} {res_dis}\n"
             .format(A=tr(A),B=tmp,D=gus,E=hum,G=ltd,res_risk=data_output[0][x][y],res_storm=data_output[2][x][y],res_tip=data_output[3][x][y],res_dis=data_output[1][x][y]))
```

Figura 22 Almacenamiento en fichero de ambos predictores

Tras la ejecución de esta prueba se obtendrán dos archivos con los datos correspondientes a las predicciones, que se podrán comparar de forma rápida con el comparador de visual studio code, como primera visualización de estos datos.

```

correctResults.txt -- myResult.txt
Version 1.1 > TFG > F myResult.txt
1 A: 2 B: 19.6 D: 7.2 E: 54.0 G: 1.3489 Result: 1 1 4 1
2 A: 3 B: 20.7 D: 8.3 E: 53.4 G: 0.3489 Result: 1 1 4 1
3 A: 3 B: 21.7 D: 7.2 E: 54.4 G: -0.4511 Result: 1 2 4 1
4 A: 3 B: 20.9 D: 6.5 E: 58.9 G: -1.2511 Result: 1 2 4 1
5 A: 3 B: 20.9 D: 4.4 E: 59.8 G: -1.9511 Result: 1 2 4 1
6 A: 3 B: 22.7 D: 3.6 E: 56.2 G: -1.7511 Result: 1 2 4 1
7 A: 3 B: 23.5 D: 3.6 E: 62.3 G: -3.2511 Result: 1 3 4 1
8 A: 1 B: 24.2 D: 6.9 E: 65.9 G: -3.4511 Result: 1 3 4 1
9 A: 2 B: 23.4 D: 7.2 E: 68.2 G: -2.5511 Result: 1 3 4 1
10 A: 2 B: 21.7 D: 16.2 E: 84.7 G: -2.9511 Result: 1 3 4 1
11 A: 2 B: 22.3 D: 11.9 E: 89.3 G: -3.4511 Result: 1 3 4 1
12 A: 2 B: 23.2 D: 10.1 E: 89.3 G: -4.4511 Result: 1 3 4 1
13 A: 2 B: 25.8 D: 10.5 E: 81.4 G: -5.6511 Result: 2 4 4 1
14 A: 2 B: 18.6 D: 5.8 E: 56.3 G: 2.0489 Result: 1 1 4 1
15 A: 3 B: 19.2 D: 5.4 E: 58.6 G: 0.1489 Result: 1 1 4 1
16 A: 3 B: 19.7 D: 9.4 E: 67.5 G: -1.3511 Result: 1 2 4 1
17 A: 3 B: 19.8 D: 17.3 E: 76.2 G: -3.4511 Result: 1 3 4 1
18 A: 3 B: 19.8 D: 7.6 E: 73.1 G: -4.1511 Result: 1 3 4 1
19 A: 2 B: 20.4 D: 8.7 E: 69.6 G: -3.8511 Result: 1 3 4 1
20 A: 3 B: 21.7 D: 8.3 E: 65.5 G: -3.7511 Result: 1 3 4 1
21 A: 2 B: 22.6 D: 8.0 E: 68.7 G: -3.7511 Result: 1 3 4 1
22 A: 2 B: 20.5 D: 10.1 E: 79.9 G: -3.0511 Result: 1 3 4 1
23 A: 3 B: 20.3 D: 7.2 E: 85.3 G: -2.5511 Result: 1 3 4 1
24 A: 3 B: 23.1 D: 5.4 E: 83.3 G: -3.0511 Result: 1 3 4 1
25 A: 3 B: 25.0 D: 16.2 E: 81.5 G: -5.1511 Result: 1 4 4 1
26 A: -1 B: 26.8 D: 14.4 E: 78.4 G: -5.4511 Result: 0 0 0 0
27 A: 2 B: 18.2 D: 18.4 E: 64.7 G: 0.1489 Result: 1 1 4 1
28 A: 2 B: 17.3 D: 7.6 E: 63.4 G: 0.2489 Result: 1 1 4 1
29 A: 3 B: 18.6 D: 9.0 E: 60.7 G: -0.1511 Result: 1 2 4 1
30 A: 3 B: 19.0 D: 15.9 E: 81.5 G: -2.8511 Result: 1 3 4 1
31 A: 4 B: 19.0 D: 9.0 E: 80.2 G: -4.3511 Result: 1 3 4 1
32 A: 3 B: 18.3 D: 10.8 E: 75.3 G: -3.9511 Result: 1 3 4 1
33 A: 3 B: 18.3 D: 8.0 E: 71.3 G: -3.1511 Result: 1 3 4 1
34 A: 3 B: 20.4 D: 8.0 E: 71.3 G: -3.1511 Result: 1 3 4 1
35 A: 4 B: 20.0 D: 6.2 E: 85.7 G: -3.3511 Result: 1 3 4 1
36 A: 3 B: 22.5 D: 7.2 E: 79.9 G: -2.7511 Result: 1 3 4 1
37 A: 3 B: 25.6 D: 15.5 E: 77.5 G: -4.2511 Result: 1 3 4 1
38 A: -1 B: 26.9 D: 18.4 E: 78.2 G: -5.5511 Result: 0 0 0 0

```

Figura 23 Comparación de archivos mediante visual studio code

En la anterior imagen Figura 23 Comparación de archivos mediante visual studio code, se muestra el inicio de ambos archivos con el resultado de las predicciones junto con las variables con las que se han obtenido las predicciones. Como se aprecia en la imagen no se encuentran diferencias en los datos predecidos por ambos predictores.

Tras el almacenamiento de estos se procederá a compararlos. La forma de compararlos ha sido mediante el comando **diff** de la terminal de Ubuntu, que nos muestra las líneas en la que se encuentran diferencia.

```

lugman@DESKTOP-SVAF80G:~/TFG-Actual/Version1.1/TFG$ diff myResult.txt correctResults.txt
lugman@DESKTOP-SVAF80G:~/TFG-Actual/Version1.1/TFG$

```

Figura 24 Ejecución de comando diff

Como se muestra en la Figura 24 Ejecución de comando diff no ha devuelto ningún resultado, puesto que este comando solo muestra resultado en caso de encontrar diferencia, dando información sobre estas diferencias.

Este proceso se realizará con los datos reales correspondientes a los valores de las variables proporcionadas por los sensores medioambientales que proporciona el primer módulo explicado. Permite obtener datos desde la fecha de un máximo de 9 días, con predicciones tomadas cada 6 horas. Los resultados a predecir serán los correspondientes a la lectura de los sensores de cada zona indicando la ubicación de estos con latitud y longitud con un total de 5002 registros por periodo de fecha. El número de predicciones que han realizado ambos predictores de forma equivalente por lo tanto sería la multiplicación de los **5002**(puntos) multiplicado **4** (4 veces correspondiente a las tomas de datos cada 6 horas, de un día) multiplicado por los **9** (días en la cual sigue estando la información disponible para obtener), se han realizado **180072** comprobaciones. Esta serie de pruebas con datos reales se realizó durante el proyecto un total de 2 veces, dando



un total de comprobaciones de **360144** valores predecidos. Comparando mediante el comando diff que no se encuentren diferencias tras la ejecución de estas pruebas. No se ha encontrado ningún error en esta parte de pruebas.

Tras realizar estas pruebas de equivalencia de predictores realizaremos una serie de pruebas más con datos aleatorios.

Comparación de ambos predictores con valores aleatorios

Estas pruebas consistirán en generar valores aleatorios de variables de entrada que serán empleados como variables de entrada para ambos predictores y almacenados en archivos para así poder comparar y comprobar su similitud. Para ello se empleará la librería random, la cual nos permitirá generar valores aleatorios y generaremos las variables de entrada que simulen todas las combinaciones.

```
def generateCombination():  
    A = random.randint(2,4)  
    B = random.randint(10,60)  
    D = random.randint(10,60)  
    E = random.randint(1,60)  
    G = random.randint(-10,3)  
  
    return (A, B, D, E, G)
```

Figura 25 Generación de valores aleatorios

Con los valores aleatorios generados de la forma que se muestra en la imagen Figura 25 Generación de valores aleatorios se generarán una cantidad de valores los cuales elegiremos nosotros para su posterior uso. Estos valores los emplearemos como valores de entrada de ambos predictores para así poder comprobar la correctitud de nuestro predictor.

Para realizar esta comprobación se creará una lista donde almacenar los resultados de cada una de las funciones de predicción (riesgo de incendio, riesgo de tormenta, disponibilidad de combustible y tipología). Esto lo realizaremos en nuestro programa de pruebas de la siguiente manera y estableciendo los siguientes nombres a estas listas, mostradas en la siguiente imagen Figura 26 Listas resultado funciones:

```
riskCLIPS = list()  
riskStormCLIPS = list()  
riskDispCLIPS = list()  
riskTypologyCLIPS = list()  
  
riskC = list()  
riskStormC = list()  
riskDispC = list()  
riskTypologyC = list()
```

Figura 26 Listas resultado funciones

En estas listas iremos almacenando el resultado de la ejecución de nuestro predictor a medida que este realiza las predicciones. Ambos predictores almacenarán los resultados en el mismo orden, por lo que su comparación será sencilla. Tras finalizar la ejecución de los dos predictores, procederemos a comparar el resultado de ambos.

Para calcular la diferencia se realizará un sumatorio de las restas en valores absolutos de cada uno de los resultados obtenidos para cada una de las funciones y se mostrará el error de cada función.

```

differRisk = 0
differStorm = 0
differDisp = 0
differTipology = 0

for i in range(0, len(testData)):
    differRisk += abs(riskCLIPS[i]) - abs(riskC[i])
    differStorm += abs(riskStormCLIPS[i]) - abs(riskStormC[i])
    differDisp += abs(riskDispCLIPS[i]) - abs(riskDispC[i])
    differTipology += abs(riskTypologyCLIPS[i]) - abs(riskTypologyC[i])

```

Figura 27 Cálculo de la diferencia de las funciones de predicción

En la Figura 27 Cálculo de la diferencia de las funciones de predicción, se muestra el código empleado para llevar a cabo este sumatorio.

A continuación, se muestran los resultados en las siguientes tablas, obtenidos tras la ejecución de las pruebas con diferentes datos.

Con 100 datos

<i>Función</i>	<i>Diferencia</i>	<i>% error</i>
<i>Riesgo de incendio</i>	<i>0</i>	<i>0</i>
<i>Riesgo de tormenta</i>	<i>0</i>	<i>0</i>
<i>Disponibilidad de combustible</i>	<i>0</i>	<i>0</i>
<i>Tipología</i>	<i>0</i>	<i>0</i>

Tabla 1 prueba datos aleatorios 100 datos

Con 1.000 datos

<i>Función</i>	<i>Diferencia</i>	<i>% error</i>
<i>Riesgo de incendio</i>	<i>0</i>	<i>0</i>
<i>Riesgo de tormenta</i>	<i>0</i>	<i>0</i>
<i>Disponibilidad de combustible</i>	<i>0</i>	<i>0</i>
<i>Tipología</i>	<i>0</i>	<i>0</i>

Tabla 2 prueba datos aleatorios 1000 datos

Con 10.000 datos

<i>Función</i>	<i>Diferencia</i>	<i>% error</i>
<i>Riesgo de incendio</i>	<i>0</i>	<i>0</i>
<i>Riesgo de tormenta</i>	<i>0</i>	<i>0</i>
<i>Disponibilidad de combustible</i>	<i>0</i>	<i>0</i>
<i>Tipología</i>	<i>0</i>	<i>0</i>

Tabla 3 prueba datos aleatorios 10000 datos

Con 100.000 datos

<i>Función</i>	<i>Diferencia</i>	<i>% error</i>
<i>Riesgo de incendio</i>	<i>0</i>	<i>0</i>
<i>Riesgo de tormenta</i>	<i>0</i>	<i>0</i>
<i>Disponibilidad de combustible</i>	<i>0</i>	<i>0</i>
<i>Tipología</i>	<i>0</i>	<i>0</i>

Tabla 4 prueba datos aleatorios 100000 datos

Se aprecia tras la ejecución de estas pruebas que no se hayan diferencias, por lo que ambos predictores realizan predicciones equivalentes ante los mismos datos de entrada.

6.5 Medición de tiempos

Adicionalmente a estas pruebas de exactitud realizadas anteriormente, se empleará este programa de prueba, en la cual generamos valores aleatorios con una cantidad de datos que se indicarán para comprobar el tiempo de ejecución es estos predictores, por lo que mediante la librería time, tomaremos tiempos antes y después de la ejecución de estos y compararemos estos tiempos para saber cuál de los dos es mayor. Para cada cantidad de datos se realizará un total de 10 ejecuciones, mostrando en las siguientes tablas su media.

Tiempo medio con 100 datos aleatorios

	Predictor c	Predictor CLIPS
Tiempo en segundos	0.006937	0.199398
Diferencia predictor CLIPS/ predictor c	28	

Tabla 5 Comparación de tiempos de ambos predictores para 100

Tiempo medio con 1.000 Datos aleatorios

	Predictor c	Predictor CLIPS
Tiempo en segundos	0.049429	2.032774
Diferencia predictor CLIPS/ predictor c	41	

Tabla 6 Comparación de tiempos de ambos predictores para 1000

Tiempo medio con 10.000 datos aleatorios

	Predictor c	Predictor CLIPS
Tiempo en segundos	0.5092	21.2472
Diferencia predictor CLIPS/ predictor c	36	

Tabla 7 Comparación de tiempos de ambos predictores para 10.000

Tiempo medio con 100.000 datos aleatorios

	Predictor c	Predictor CLIPS
--	-------------	-----------------

Tiempo en segundos	5.051465	193.888134
Diferencia predictor CLIPS/ predictor c	38	

Tabla 8 Comparación de tiempos de ambos predictores para 100.000

Como se puede apreciar el predictor realizado en CLIPS tarda más tiempo en ejecutarse que el módulo anterior desarrollado en C , habiendo una diferencia entre estos en las pruebas de velocidad de las tablas anteriores de un mínimo de 28 de diferencia de velocidad a favor del predictor realizado en C, para 100 datos y un máximo de 41 de diferencia para datos 1.000 datos.

Evaluación de un sistema para la previsión de Riesgos de Incendios Forestales Catastróficos basado en Inteligencia Artificial



7 Presupuesto del proyecto

En esta sección se indicarán los costes que supondría llevar a cabo este proyecto. Se especificarán todos los elementos de este que tienen un coste directo sobre el desarrollo de este proyecto.

Consideraciones de esta sección

- Se entiende este trabajo como un proyecto de fin de grado, con el objetivo de aprender y mostrar el conocimiento, por lo que no se contempla que haya beneficio económico.
- Puesto que no se tiene información del futuro ámbito de este proyecto en una futura puesta en marcha, no se podrá especificar el coste del despliegue de este software, así como los recursos que consumirá en otro entorno de ejecución.
- Existen elementos que no se han comprado exclusivamente para la realización del presente trabajo, por lo que para calcular el aporte de estos al presupuesto se calculará su amortización
- No se considera la presencia de costes indirectos.

Materiales empleados

Tipo de Material	Cantidad [ud]	Coste unitario [€]	Coste total [€]	Tiempo de uso en el proyecto [horas]	Tiempo de vida [horas]	Coste repercutido [€]
Ordenador sobremesa HP Victus 15L	1	1252	1252	230	14600	19.72
Monitor MSI PRO MP241X	2	123	246	230	17520	3.23
Teclado Newskill Serike	1	51	51	230	11680	1.0
Ratón Tempest X8 Keeper	1	25	25	230	11680	0.49
Windows 10 Pro 64 Bits	1	109	109	230	14600	1.72
Importe total [€]						26.17

Tabla 9 Materiales empleados

Mano de obra

Tipo de mano de obra	Cantidad [horas]	Coste unitario[€/h]	Coste total [€]
Graduado en ingeniería informática	230	13.71 [12]	3153.3
Tutor	12	15.62 [13]	187.44
Segundo tutor	8	15.62 [13]	124.96
Importe total [€]			3465.7

Tabla 10 Mano de obra

Resumen del presupuesto

Tipo de mano de obra	Cantidad [horas]
Coste Materiales	26.17
Coste Mano de obra	3465.7
Importe total [€]	
3491,87	

Tabla 11 Resumen del presupuesto

Cómo se ha mostrado en la tabla de costes de materiales, para la realización de este proyecto se tiene un gasto en materiales de **veintiséis euros con diecisiete céntimos**. El gasto en mano de obra presupone un gasto para este proyecto de **tres mil cuatrocientos sesentaicinco euros y setenta céntimos**. El coste total de este proyecto el cual consta de los gastos de materiales y los gastos en mano de obra será de **tres mil cuatrocientos noventa y un euros, con ochenta y siete centimos**.

8 Conclusiones

Tras realizar este proyecto se ha podido comprobar a lo largo de la memoria como se ha logrado implementar un predictor de incendios en un sistema experto bajo la tecnología de CLIPS. Se ha conseguido realizar de forma exitosa la predicción conforme el conocimiento que el experto ha proporcionado sobre la predicción de riesgo de incendio, tormenta, tipología y disponibilidad de combustible.

La forma en que se ha integrado este módulo en la aplicación sobre la que lo emplearía ha sido una forma bastante acertada, pudiéndose esta implementar sin modificar dicha aplicación y ejecutando mediante una librería el motor de inferencia y demás funciones necesarias de CLIPS para poder ejecutarse las reglas y hechos que dan lugar a un resultado de predicción.

En el caso de esta aplicación la información del experto se encontraba ya implementada en otro módulo el cual se ha estudiado y extraído de este módulo la información necesaria para poder crear el predictor de este proyecto. Una vez realizado el predictor se han llevado a cabo una serie de pruebas que han permitido comprobar que el resultado obtenido por este predictor es equivalente al del predictor ya implementado, por lo que ante la visualización del mapa mostrado en el módulo de visualización Figura 6. Módulo de visualización, provincia de Alicante, que representa dichas predicciones de forma gráfica, no se apreciarían diferencias a nivel visual. Por esta equivalencia entre ambos predictores se considera que se ha cumplido de forma exitosa este desarrollo y la forma en la que está implementado además de ser más fiel a la naturaleza de la especificación del problema, es más manipulable a la hora de realizar algún cambio puesto que únicamente habría que modificar o añadir las reglas en un archivo donde se encuentran, en comparación con el predictor anterior el cual se encontraba todo en una misma función de forma muy compacta, lo cual no permitía una modificación simple sin un previo análisis de esta solución y se encontraba compilada por lo que debía llevarse a cabo el proceso de modificación compilación y sustitución. Por otro lugar el predictor anterior disponía de una mayor velocidad de ejecución, alrededor de 28-41 veces más rápido, esta diferencia se interpreta como algo de esperar puesto que uno está compilado y su lógica estaba basado en condicionales y el predictor desarrollado está realizado mediante reglas y hechos y éstos deben ser ejecutados en un motor de inferencia el cual realiza la comprobación de estas en tiempo de ejecución. No obstante, y ante la pérdida de agilidad de este no se considera un problema puesto que este se ejecutará en un intervalo de tiempo en el cual está lejos de ser un problema la velocidad del actual predictor.

Por último, se concluirán las conclusiones de esta sección dando una opinión de satisfacción de este proyecto. Tras la realización de este se considera satisfactorio el desarrollo de este proyecto, puesto que tras el estudio de cómo llevar a cabo la implementación e integración de este se ha conseguido realizar una implementación sencilla y manipulable, la cual está abierta a futuros cambios o ampliaciones.

Trabajo futuro

Puesto que se ha notado que la velocidad del predictor es inferior a la del predictor anterior, se podría intentar mejorar esta velocidad. Actualmente la aplicación ya hace

uso de concurrencia para la ejecución de este módulo, por lo que sin un previo análisis no se podría decir que se pueda mejorar la velocidad paralelizando este módulo de forma distinta, pero si se cree que se podría mejorar la velocidad de este haciendo uso de algún sistema que permita el almacenamiento del resultado en memoria cache. Quedaría para estudio la implementación de algún sistema que permita el almacenamiento en cache de este módulo como posible mejora en el rendimiento.

Relación del TFG con el Grado

Tras la realización de este proyecto se puede apreciar como la formación adquirida durante la realización del grado en ingeniería informática brinda de un conocimiento general y unas herramientas útiles que con cierto estudio se pueden llevar a la práctica y por consiguiente obtener una solución práctica que resulte válida para la solución de problemas como el actual que son directamente útiles para problemas de la actualidad.

De esta formación y con relación a este proyecto se podrían destacar las siguientes asignaturas:

- **Sistemas inteligentes (SIN):** en la cual se estudia los problemas que resuelve la inteligencia artificial a grandes rasgos y se ven técnicas con las cuales resolver estos problemas, entre las cuales se encuentran los sistemas expertos y el sistema experto CLIPS, que es el que se ha empleado en este proyecto.
- **Técnicas, entornos y aplicaciones de inteligencia artificial (TIA):** en esta asignatura se parte de que se ha cursado la asignatura anteriormente descrita SIN y se amplía esta asignatura con una mayor información sobre cómo enfrentar problemas mediante inteligencia artificial, así como se explican y practican algunas de ellas, también se enseña y practica en el sistema experto CLIPS, de la misma forma que en SIN, pero enfocada a resolver problemas más complejos y utilizando mayor cantidad de funcionalidades que ofrece este sistema experto.
- **Ingeniería del software (ISW):** Esta asignatura brinda un conocimiento útil para el desarrollo del software, presentando formas de interpretar los problemas de una solución a resolver, así como gestionarlas y desarrollarlas de una forma controlada y eficiente.

De estas asignaturas mencionadas podemos destacar que las más relevantes serían SIN y TIA, ambas han proporcionado una visión sobre este problema y el conocimiento sobre sistemas expertos y en concreto el sistema experto CLIPS, con el que se ha desarrollado la solución. La asignatura de ISW también ha sido útil para entender como diseñar el planteamiento de la solución y realizar un diagrama de flujo en el cual ha sido de gran utilidad para desarrollar el predictor.

Conclusiones personales

Tras el desarrollo de este proyecto en opinión se considera que ha sido un reto el cual planteaba una serie de dudas sobre si la realización podría salir de la forma correcta y planteada. La opinión que se tiene sobre este es que ha sido un proyecto eficiente a nivel de desarrollo sin haber sufrido de complicaciones importantes. De este proyecto se ha sacado un conocimiento que se considera muy apropiado con lo que respecta la unión de tecnologías e integración de estas, puesto que no se tenía conocimiento práctico de cómo se podrían unir estas y si hubiera formas de que está fueran útiles exceptuando por uso individual. Adicionalmente a esto se considera relevante la experiencia que es realizar un

proyecto de mayor tamaño que los realizados durante la formación académica y de forma individual en su gran mayoría, así como la documentación redactada en la memoria sobre esta, que ha supuesto un reto el poder explicar de la mejor manera la forma en que se ha realizado. También se han apreciado algunos aspectos sobre este desarrollo que se podrían mejorar, estos serían:

- La realización de pruebas de gran tamaño, se han realizado pruebas sobre partes de este desarrollo bastante desarrolladas, estas permitían en algunas ocasiones el testeado parcial y no se ha realizado, generando después un mayor tiempo para detectar el error.
- La falta de estudio acerca de la aplicación en general, debido a que genera una mayor comprensión sobre el contexto en general y así una mayor facilidad para comprender este proyecto y llevar a cabo la solución.

Debido a las conclusiones obtenidas sobre este proyecto se considera que ha sido un proyecto de gran utilidad como parte final de este grado y nos ha acercado al mundo profesional a nivel práctico.

9 Bibliografía

- [1] 'El planeta en llamas'. https://wwfes.awsassets.panda.org/downloads/wwf_informe_incendios_2020_el_planeta_en_llamas_ok.pdf?54921/Informe-incendios-forestales-2020-El-planeta-en-llamas (accessed Apr. 11, 2022).
- [2] 'E INCENDIOS FORESTALES EN LA COMUNITAT VALENCIANA'. https://agroambient.gva.es/documents/20551003/167181181/07_An%C3%A1lisis_causas_3_12_2018_v4/b67c5ee5-3455-40e5-b426-065e97db8ccc (accessed Apr. 11, 2022).
- [3] J. M. Torres-Rojo and J. M. Torres-Rojo, 'Índice para la estimación de ocurrencia de incendios forestales en superficies extensas', *Revista Chapingo serie ciencias forestales y del ambiente*, vol. 26, no. 3, pp. 433-449, Dec. 2020, doi: 10.5154/R.RCHSCFA.2019.11.082.
- [4] R. Pino Díez, *Introducción a la inteligencia artificial : sistemas expertos, redes neuronales artificiales y computación evolutiva*. Oviedo: Universidad de Oviedo, 2001.
- [5] 'Qué son los sistemas expertos - Agencia B12'. <https://agenciab12.com/noticia/que-son-sistemas-expertos> (accessed Sep. 06, 2022).
- [6] S. Badaró, L. Javier Ibañez, and M. J. Agüero, 'Sistemas Expertos: Fundamentos, Metodologías y Aplicaciones'.
- [7] 'Sistemas Expertos: Definición, Aplicaciones y Ejemplos'. <https://www.tecnologias-informacion.com/sistemas-expertos.html> (accessed Aug. 23, 2022).
- [8] 'Instalación de WSL | Microsoft Docs'. <https://docs.microsoft.com/es-es/windows/wsl/install> (accessed Feb. 24, 2022).
- [9] 'Qué es WSL2, cómo instalarlo y por qué es una gran noticia'. <https://pandorafms.com/blog/es/wsl2/> (accessed Feb. 24, 2022).
- [10] 'IEEE Xplore Full-Text PDF': <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=940977> (accessed Apr. 06, 2022).
- [11] 'CLIPS - Wikipedia, la enciclopedia libre'. <https://es.wikipedia.org/wiki/CLIPS> (accessed Apr. 06, 2022).
- [12] 'BOE.es - BOE-A-2019-14977 Resolución de 7 de octubre de 2019, de la Dirección General de Trabajo, por la que se registra y publica el XIX Convenio colectivo del sector de empresas de ingeniería y oficinas de estudios técnicos.' https://www.boe.es/diario_boe/txt.php?id=BOE-A-2019-14977 (accessed Sep. 06, 2022).

- [13] 'BOE.es - BOE-A-2019-13119 Resolución de 27 de agosto de 2019, de la Dirección General de Trabajo, por la que se registra y publica el VIII Convenio colectivo nacional de universidades privadas, centros universitarios privados y centros de formación de postgraduados.' https://www.boe.es/diario_boe/txt.php?id=BOE-A-2019-13119 (accessed Sep. 05, 2022).
- [14] 'Agua y saneamiento - Desarrollo Sostenible'. <https://www.un.org/sustainabledevelopment/es/water-and-sanitation/> (accessed Sep. 03, 2022).
- [15] 'Ciudades - Desarrollo Sostenible'. <https://www.un.org/sustainabledevelopment/es/cities/> (accessed Sep. 03, 2022).
- [16] 'Cambio climático - Desarrollo Sostenible'. <https://www.un.org/sustainabledevelopment/es/climate-change-2/> (accessed Sep. 03, 2022).
- [17] 'Bosques, desertificación y diversidad biológica - Desarrollo Sostenible'. <https://www.un.org/sustainabledevelopment/es/biodiversity/> (accessed Sep. 03, 2022).

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.				X
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.			X	
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.			X	
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.		X		
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.	X			
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.				X

Tabla 12 Evaluación de implicación del proyecto en los ODS

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Este proyecto final de grado en ingeniería informática se puede relacionar especialmente con los objetivos más próximos a la climatología, contaminación y medio ambiente. Se puede apreciar que para cada uno de estos objetivos se encuentra una relación de mayor o menor implicación. Se procederá a describir estos especificando en cada uno de ellos en base a la relación que se ha hallado, se describirán los que no están marcados en la tabla con el valor no procede y con el orden que se ha establecido anteriormente en la tabla.

El objetivo 6 Agua limpia y saneamiento [14]. Se aprecia como no está directamente relacionado con este proyecto, no obstante, el objetivo de este proyecto es la reducción de los incendios mediante la prevención de estos, por lo que si se consigue satisfacer este objetivo podría reducirse la contaminación que estos generan y que de alguna manera implica la generación de restos y generación de contaminación de algunas zonas potables, que podrían dejar de serlo por un periodo de tiempo. Los incendios pueden afectar a los ecosistemas relacionados con el ciclo de agua, que produce un efecto negativo en este sistema, por lo que podría ayudar la prevención de estos a proteger el ciclo del ecosistema relacionado con el agua.

Se puede apreciar una relación de nuestro proyecto con el objetivo 11 Industria, innovación e infraestructuras [15], a consecuencia que este objetivo pretende generar y sostener una relación sostenible entre la población y el medio ambiente con lo que respecta a la población de ciudades y comunidades. Este objetivo 11 intenta que se tenga un mayor acceso a zonas rurales, patrimonios culturales y naturales del mundo. Como este proyecto intenta reducir los incendios forestales, se puede apreciar que estaríamos ayudando a que este objetivo se consiga en menor medida, manteniendo una mayor cantidad de bosques en buen estado, así como el mantenimiento de áreas próximas a bosques y demás zonas forestales.

El objetivo 13 acción por el clima [16] tiene una relación de cierta importancia con este proyecto a consecuencia de la relación entre la reducción de incendios y la reducción del dióxido de carbono a la atmosfera, el objetivo de este proyecto sería que a consecuencia de una reducción de incendios se produzca una menor cantidad de emisión de estos gases. La relación es directa pero no se considera que este fuertemente relacionado puesto que la emisión de gases que contienen CO₂ no son procedentes como fuente primaria de los incendios forestales, pero si se deben tener en cuenta pues estos, puesto que se añaden al resto de factores que generan contaminación, por lo que se considera que es un objetivo en el cual se estaría ayudado a su logro con el proyecto actual.

Este proyecto se considera fuertemente relacionado con el objetivo número 15 Vida de ecosistemas terrestres [17], ya que es el que tiene una relación clara y evidente con nuestro objetivo principal. En este proyecto se intenta reducir el número de incendios forestales, lo que ayudaría a cumplir este objetivo de la siguiente forma: ayudar a que con una mayor cantidad de bosques las personas que dependen de estos tengan un mayor acceso a estos y con mayor calidad de estos, ya que tras un incendio forestal se pierda gran cantidad de vegetación de estos y reduce el tamaño del bosque que ha sufrido el

incendio, de forma temporal en el mejor de los casos; ayudar a mantener las especies que habitan en estos bosques, puesto que ante un incendio sufren las especies que habitan en estas zonas; frenar la degradación del suelo, tanto de estos bosques como de la propagación de estos a otras zonas terrestres con suelo agrícola; ayudar a que el ecosistema terrestre sea más próspero pues la vegetación y árboles que se pierden en un incendio forestal es nefasto para estos y tiene un tiempo de recuperación muy elevado.