



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

ANEXO: Guía funcionamiento EJS

Proyecto Final de Carrera

Ingeniería Informática

Autor: Manuel González Marco

Director: Marina Vallés Miquel

Valencia, 4 de Julio de 2013



Tabla de contenidos

1. Introducción	5
2. Funcionamiento.....	6
2.1. Organización de proyectos.....	6
2.2. Ejecución.....	7
3. Interfaz Gráfica	9
3.1. La consola	9
3.2. La interfaz de usuario.....	9
3.2.1. Panel de trabajo de la descripción.....	11
3.2.2. Panel de trabajo para el modelo	11
3.2.3. Panel de trabajo para la vista	15
4. Bibliografía	16

1. Introducción

En esta guía vamos a tratar de explicar cómo utilizar EJS para realizar tanto pequeñas simulaciones como algún programa para simular algo más complejo.

En primer lugar, explicaremos el funcionamiento del EJS y, después, mostraremos las interfaces y realizaremos la explicación de algunos de los elementos más importantes.

2. Funcionamiento

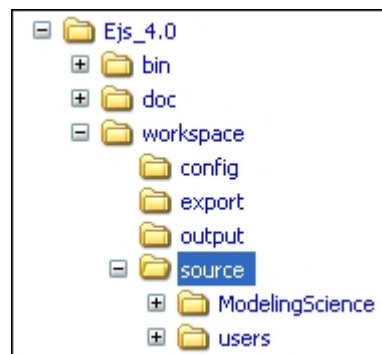
En este apartado vamos a explicar cómo organiza los espacios de trabajo y como realiza su ejecución.

2.1. Organización de proyectos

El EJS organiza los proyectos de la forma conocida como “espacio de trabajo”. Este tipo de organización se basa en un directorio donde EJS almacena todos sus ficheros de simulación de un proyecto. Dentro de un espacio de trabajo podemos tener un número ilimitado de simulaciones.

Dentro de cada espacio de trabajo, se encuentran 4 carpetas:

- **config:** Contiene los ficheros de configuración creados por EJS cada vez que cambie usted sus opciones de ejecución.
- **export:** En esta carpeta se guarda de forma predeterminada los paquetes autoejecutables.
- **output:** Almacena ficheros intermedios generados cuando se realizan simulaciones.
- **source:** Aquí es donde debemos almacenar nuestros ficheros de simulación (ficheros XML). También debemos de almacenar aquí los ficheros auxiliares, como por ejemplo las imágenes GIF o ficheros de datos.



Podemos cambiar de espacio de trabajo de una manera muy sencilla desde dentro de la aplicación, para poder separar los proyectos diferentes, o por tipos de simulación diferentes. Si lo que queremos es crear un nuevo espacio de trabajo, deberemos de crear la carpeta que queremos utilizar como espacio de trabajo y desde dentro de la aplicación, cambiamos el espacio de trabajo, seleccionando dicha carpeta. Para realizar la elección del espacio debemos de ir a la consola de EJS y seleccionar la pestaña de *Opciones Básicas*, y veremos la opción "*Su espacio de trabajo es*", donde podremos seleccionar el directorio deseado.

Cuando ejecutamos por primera vez el EJS, este nos pedirá que definamos nuestro espacio de trabajo. La única condición para que sea un directorio válido, es que se tenga permiso de escritura. También podemos seleccionar el subdirectorio llamado “*workspace*” dentro del directorio “*EJS_X.X*”, aunque es preferible el crear un directorio en nuestra carpeta personal.

Una vez seleccionado el directorio, las siguientes veces que arranquemos el EJS se abrirá dicho directorio de forma predeterminada. Para forzar al EJS a que nos pregunte el directorio al arrancar es eliminando el archivo llamado “*.EjsConsole.txt*”, donde se guardan nuestras opciones.

2.2. Ejecución

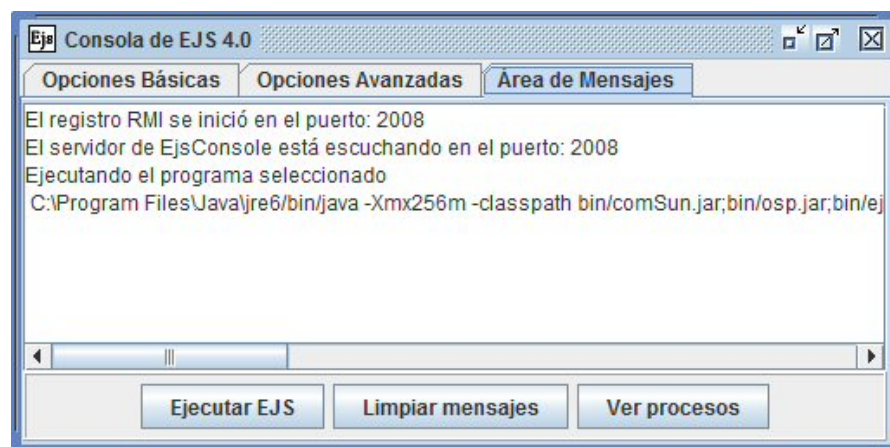
Para arrancar el EJS, tanto en Windows como en Mac OS, debemos darle doble clic al archivo llamado “*EjsConsole.jar*”, el cual está en el directorio “*EJS_X.X*” creado al descomprimir el paquete descargado de la página de EJS.

Para arrancarlo en Linux es distinto, ya que Linux/Unix interpreta el fichero “*EjsConsole.jar*” como un fichero de archivo. Existen 2 formas de arrancarlo:

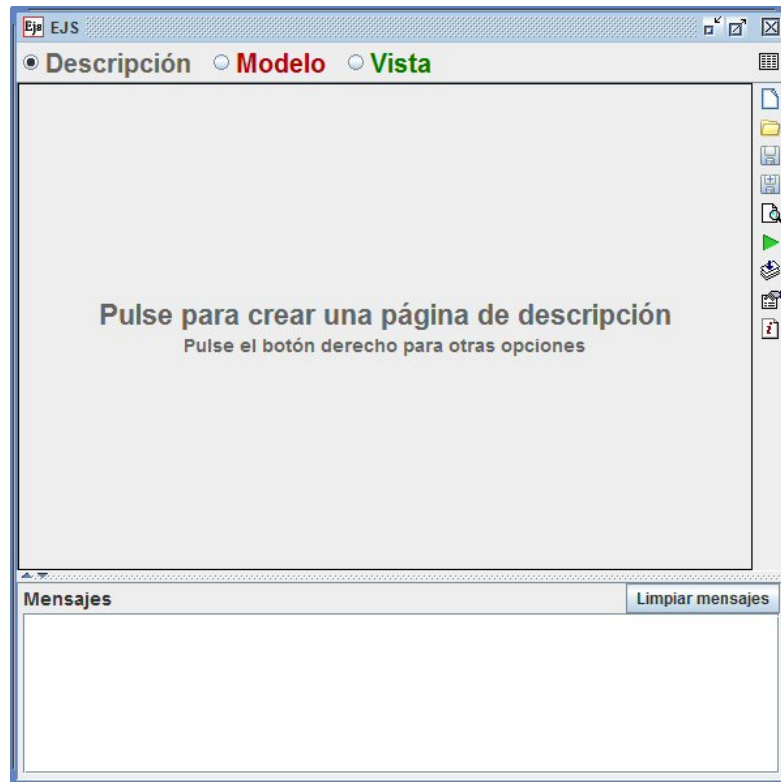
- Desde una ventana de terminal: Vamos al directorio que contiene el fichero “*EjsConsole.jar*”, y escribimos el comando “*java -jar EjsConsole.jar*”.
- Desde un icono de la aplicación: Debemos crear un nuevo “*enlace a la aplicación*”; la aplicación es “*java -jar \$camino/EjsConsole.jar*”, donde \$camino es el camino completo del directorio donde se encuentra el fichero “*EjsConsole.jar*”, y el directorio de trabajo es \$camino.

Al arrancar EJS se nos abrirán, como mínimo, 2 ventanas:

- La consola de EJS, la cual presenta 3 pestañas:



- La ventana de edición de EJS.



La consola abrirá de forma automática la simulación de la sesión anterior. Las ventanas de vista pueden aparecer a la vez que la ventana de edición de EJS. Otra cosa que podremos observar es que durante el arranque, una pequeña ventana de diálogo mostrará la evolución de arranque.

Para ejecutar una segunda copia de EJS a la vez debemos utilizar los botones “Ejecutar Easy Java Simulations” o “Ejecutar EJS” de la consola de EJS. Nos es recomendable intentar la utilización de una segunda copia de EJS con el doble-clic en “EjsConsola.jar”.

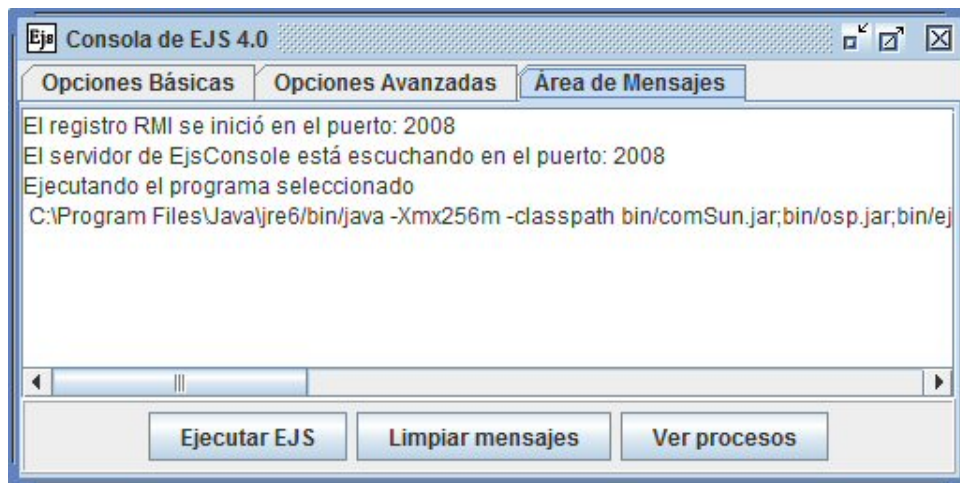
3. Interfaz Gráfica

En este apartado vamos a mostrar y realizar una pequeña explicación de las distintas ventanas que forman el EJS.

3.1. La consola

Es el punto de partida para ejecutar EJS. Se utiliza para arrancar EJS y ejecutar otros procesos. Algo importante es que es la ventana donde se muestran los mensajes de salida y posibles mensajes de error.

La consola es una ventana aparte de la de EJS, la cual no puede cerrarse. No puede ser cerrada directamente, para cerrarla debemos de cerrar la ventana del EJS. Al arrancar ejecuta una primera copia de EJS. Podemos utilizar los distintos botones incluidos en la ventana para realizar las acciones que indican.



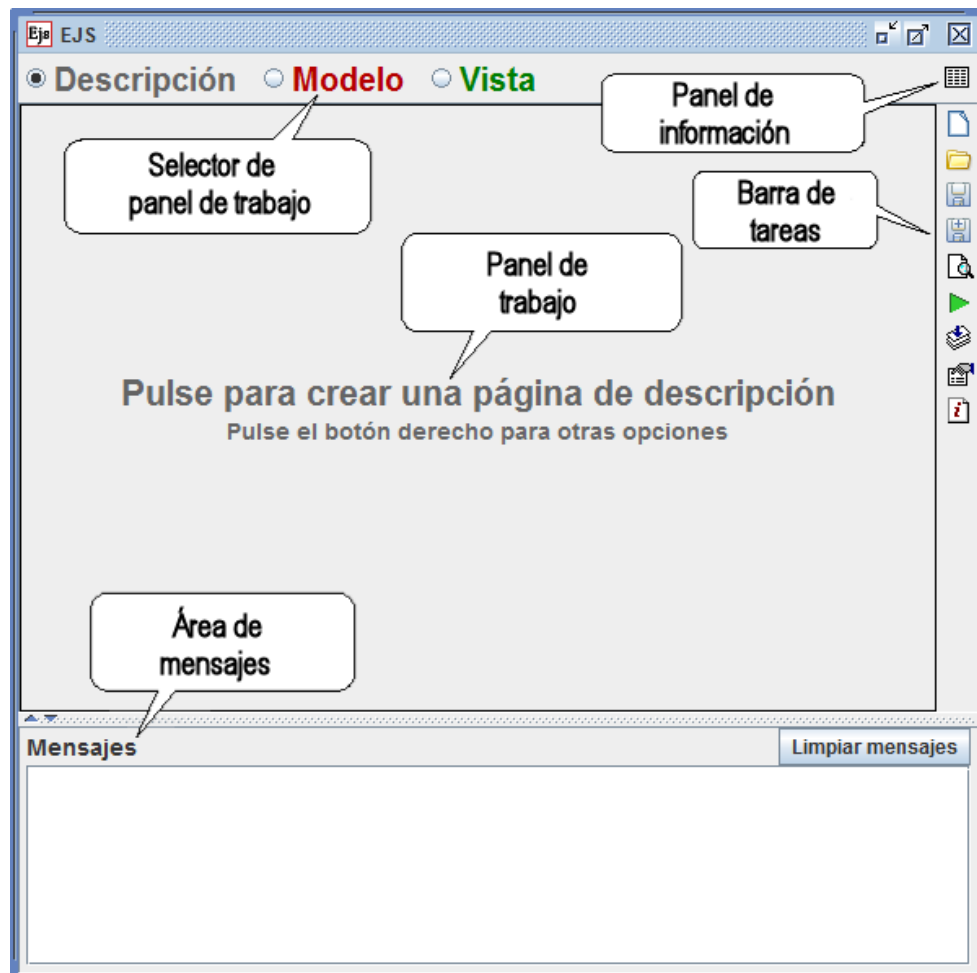
3.2. La interfaz de usuario

La barra de tareas de la derecha proporciona una serie de botones para mostrar un panel de información sobre la simulación, limpiar, abrir y guardar un fichero, buscar en el código, configurar EJS, y mostrar la ayuda de EJS. También proporciona iconos para ejecutar una simulación y para empaquetar simulaciones en un fichero JAR o exportarlas como pallets en páginas web. Al pulsar encima de algunos iconos con el botón secundarios se puede realizar otras acciones alternativas.

La parte inferior de la interfaz contiene un área de mensajes donde EJS muestra mensajes de información. La parte central de la interfaz contiene los paneles de trabajo donde se construye la simulación.

La interfaz está formada por 3 paneles:

- **Descripción:** Nos permite crear y editar en código HTML una descripción de la simulación. Cada página aparece en una pestaña de este panel de trabajo y el usuario puede, haciendo clic con el botón derecho, editar el texto o importar ficheros HTML escritos con otras herramientas.
- **Modelo:** Está dedicado al proceso de modelado. Este panel se usa para crear variables que describen el modelo de nuestra simulación, para inicializar estas variables, y para escribir algoritmos que describen cómo varían estas variables con el tiempo.
- **Vista:** Es donde creamos la interfaz gráfica de usuario de la simulación, que permitirá a los usuarios finales controlar la simulación y mostrar sus salidas. Para crear dicha interfaz, debemos seleccionar diferentes elementos de unas paletas y añadiéndolas al denominado Árbol de elementos de la vista.

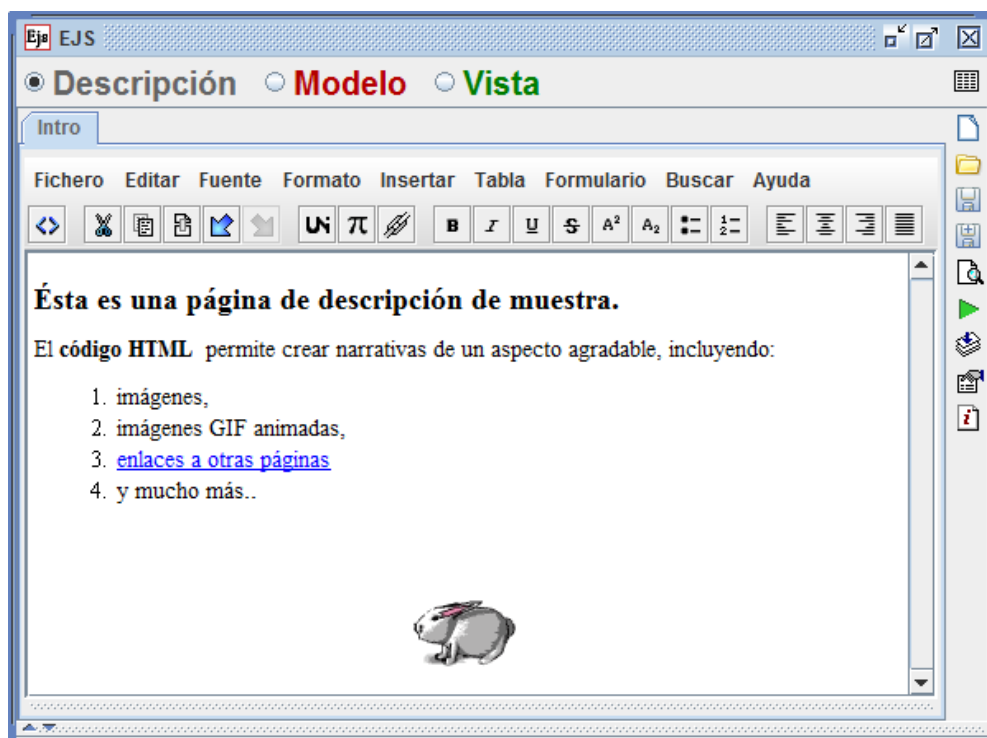


3.2.1. Panel de trabajo de la descripción

El panel de descripción proporciona un lugar donde escribir la introducción para la simulación. Esta información se mostrará al usuario al arrancar la simulación, cuando se ejecuta como una aplicación, o será parte de las páginas HTML que acompañan a la simulación, cuando se ejecuta como un applet.

Hay 3 modos de rellenar esta descripción:

1. Usar el sencillo editor incluido en el EJS.
2. Importar archivo HTML.
3. Enlace a archivo HTML externo.



3.2.2. Panel de trabajo para el modelo

El panel de modelo está subdividido en 5 subpaneles:

- Variables.
- Inicialización.
- Evolución
- Relaciones fijas.
- Propio.

Al pulsar en el subpanel en el que nos encontremos, crearemos una nueva página a la que nos piden que pongamos nombre. Al crearse la página podremos ver arriba una pestaña. Esta pestaña sirve para al hacer doble-clic permitirnos crear una nueva página o en caso de tener más de una, navegar entre ellas al hacer clic sobre la que queramos ir. Otra opción que tenemos al darle al botón secundario sobre la pestaña es la de activar o desactivar esa pestaña, para que de esa forma cuando ejecutemos la simulación no se tenga en cuenta esa página.

A continuación, conoceremos de una manera más profunda los distintos subpaneles.

VARIABLE

En este panel es donde se realiza la declaración de todas las variables que vayamos a utilizar durante la programación de nuestra simulación. Cada variable debe de tener nombre, tipo y, si fuera un vector o matriz, debemos proporcionarle la dimensión del mismo.

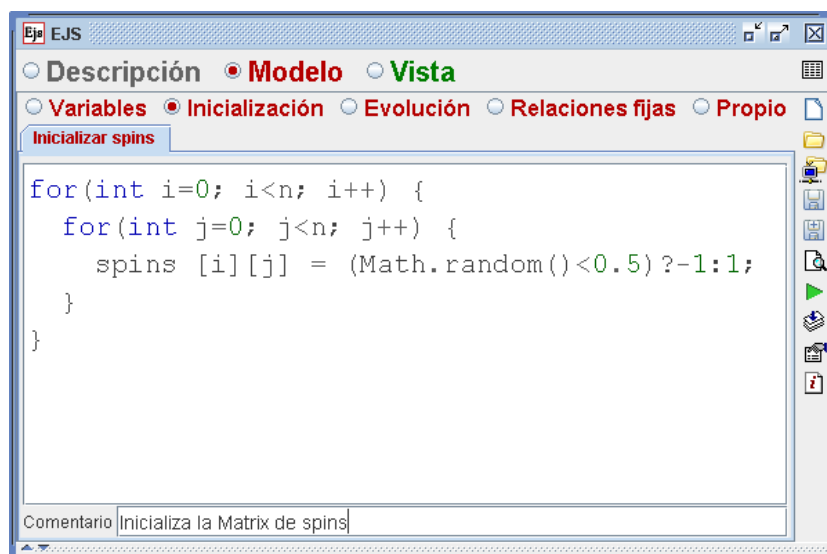
En cuanto al tipo de la variable, disponemos de 5 tipos distintos:

1. **boolean:** para valores de verdadero o falso.
2. **int:** para valores enteros.
3. **double:** para números en coma flotante, es decir, números reales en general.
4. **String:** para caracteres y textos.
5. **Object:** tipo que nos ofrece la programación orientada a objetos de Java y que permite a programadores más avanzados crear sus propios tipos.

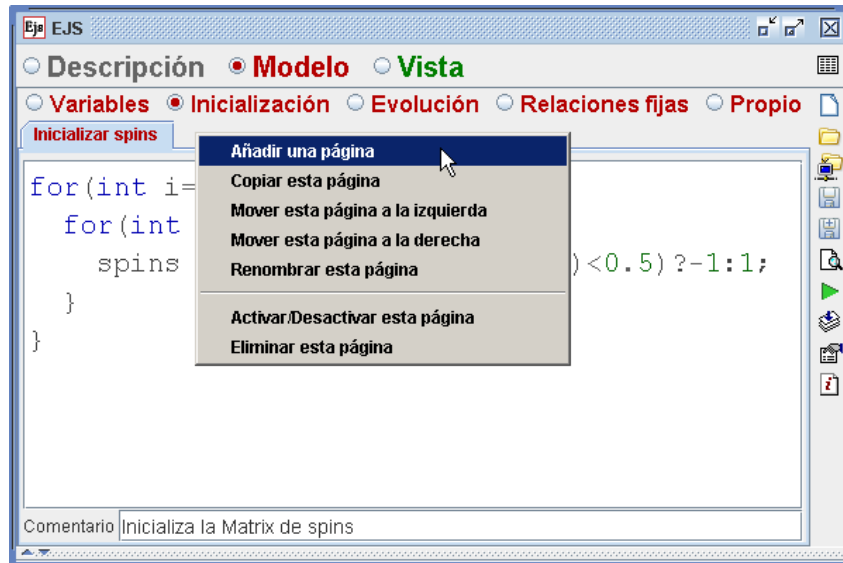
Otra propiedad que disponemos en las variables es la de valor inicial. No es obligatorio darle valor, pero si queremos, por ejemplo, que una variable sea constante podemos hacerlo desde aquí.

INICIALIZACIÓN

En este panel es donde realizaremos la inicialización de las variables oportunas, o bien porque preferimos realizarlo desde aquí, o bien, la opción de valor inicial comentada anteriormente no es suficiente debido a la necesidad de realizar un calcular inicial más complejo.

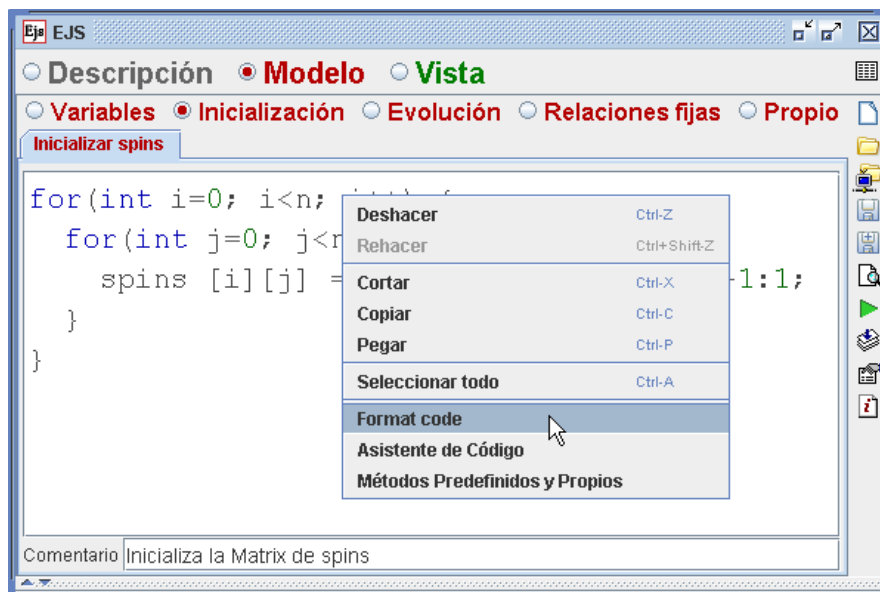


Las distintas páginas de inicialización creadas se ejecutarán de izquierda a derecha. Disponemos de la opción de reordenarlas en el orden que queramos. Para ello hacemos doble clic sobre la pestaña de la página a reorganizar y veremos la opción “Mover esta página a la derecha/izquierda”.



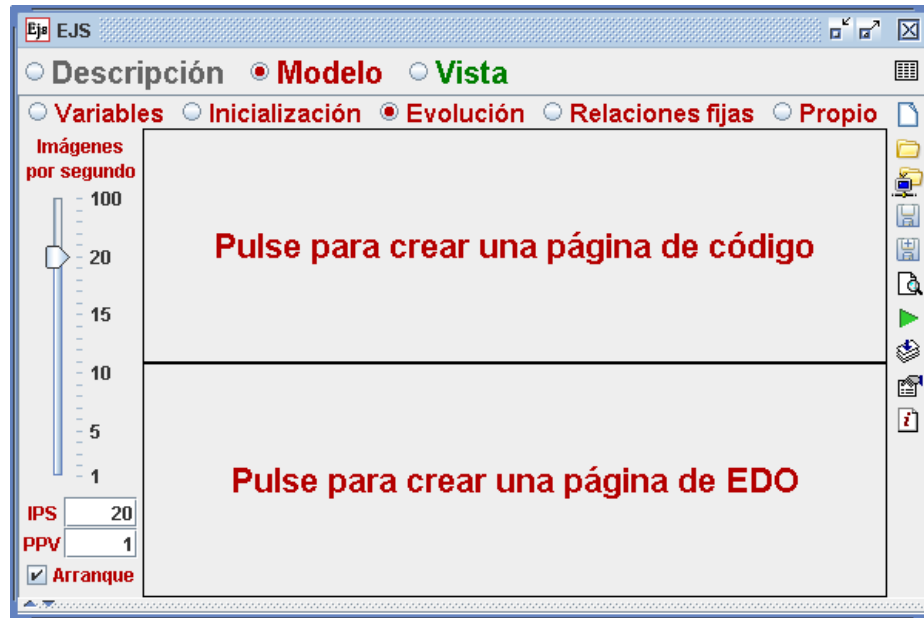
Si hacemos doble clic sobre la pagina, nos saldrá otro cuadro de opciones diferentes, entre los que entre otros destacamos 3:

- **Formar Code:** Esta opción dará un correcto formato de programación al código.
- **Asistente de código:** Proporciona ayuda para la incorporación de estructuras de control de flujo tales como bucles y condicionales.
- **Métodos predefinidos y propios:** Lista de métodos ya implementados en EJS, los cuáles pueden ser utilizados.



EVOLUCIÓN DEL MODELO

En este panel podemos crear páginas de código y de EDO. Las páginas de código son las idóneas para simulaciones discretas, mientras que las de EDO son para aquellos modelos en los que necesitemos la resolución numérica de ecuaciones diferenciales ordinarias facilitándonos la tarea de programar el código en Java necesario.



RELACIONES FIJAS

En las páginas de este subpanel escribiremos la mayor parte de código en java. Debemos de escribir las ecuaciones de relaciones fijas del modelo transcritas a Java.

Para orientarnos para diferenciar entre ecuaciones de evolución y de relaciones fijas debemos fijarnos en que las variables a calcular su valor no deberán de depender de sus valores actuales ya que si es así deberíamos de ponerlas en evolución porque seguramente se trate de una ecuación de evolución.

CÓDIGO PROPIO

En estas páginas deberemos de escribir nuestras funciones propias. Estas funciones a diferencia del código escrito en los anteriores paneles no es añadido de forma automática a la simulación sino que debe de ser invocado desde código. Por lo general, suele ser código que solo queremos que se utilice si se cumple unos requisitos previos.

Otra diferencia que encontramos con el código escrito en los paneles anteriores, es que al ser funciones deberemos de darle un nombre, el tipo de valor de salida y especificar la accesibilidad de la misma. También podemos aceptar parámetros de entrada, para los cuáles deberemos de especificar su tipo en caso de tenerlos.


3.2.3. Panel de trabajo para la vista

En este panel es donde deberemos de crear la interfaz gráfica para usuario y enlazar los distintos elementos de la vista con variables o métodos previamente definidos. A continuación lo trataremos con mayor profundidad.

CONECTAR ELEMENTOS

Para conectar una variable del modelo con el elemento de vista apropiado, simplemente escriba el nombre de la variable en el campo de texto que encontrará en el editor de propiedades del elemento.

La conexión es bidireccional, es decir, tanto si actualizamos la variable como el elemento con el que está enlazado, el otro extremo de la conexión será actualizado con el mismo valor. Dicho de otra forma, los 2 extremos del enlace siempre tienen el último valor asignado a uno de los 2 extremos.

Otra forma es elegir la variable que se relacionará con la propiedad utilizando el segundo icono de la derecha. Este icono, que muestra una cadena de unión, , le mostrará una ventana con una lista de todas las variables del modelo que pueden ser relacionadas con la propiedad.

Como recomendación a la hora de realizar los enlaces, es recomendable escribir las cadenas de tipo String entre comillas, aunque si se nos olvida, normalmente, el EJS lo añadirá solo.

ELEMENTOS CONECTADOS A EXPRESIONES

Las propiedades de los elementos de vista también pueden conectarse a expresiones que involucren variables del modelo (o no), como por ejemplo, `Math.sin(0.5)`, donde EJS evaluará la expresión y asignará el resultado de la misma en el elemento de la vista enlazado.

El único problema es la pérdida de la bidireccionalidad.

REFERENCIA DE LOS ELEMENTOS DE LA VISTA EN EJS

A continuación podremos encontrar una referencias a los distintos elementos que podemos encontrar para crear la interfaz. Están divididos en 3 grupos:

1. [Elementos de la interfaz.](#)
2. [Elementos de dibujo 2D.](#)
3. [Elementos de dibujo 3D.](#)

4. Bibliografía

La información utilizada para apoyarse en la creación de la guía está disponible en la página [Web Oficial de EJS](#).