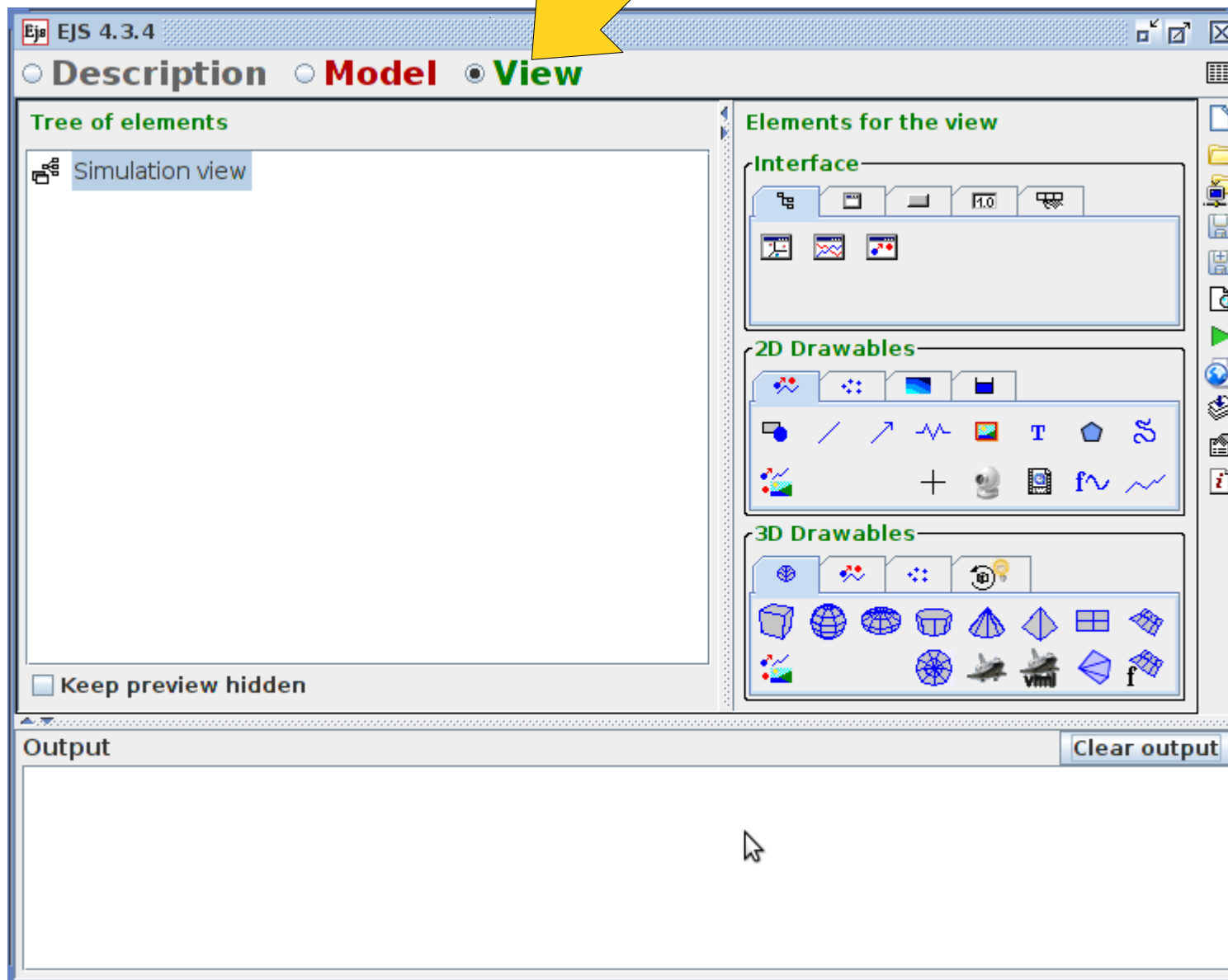




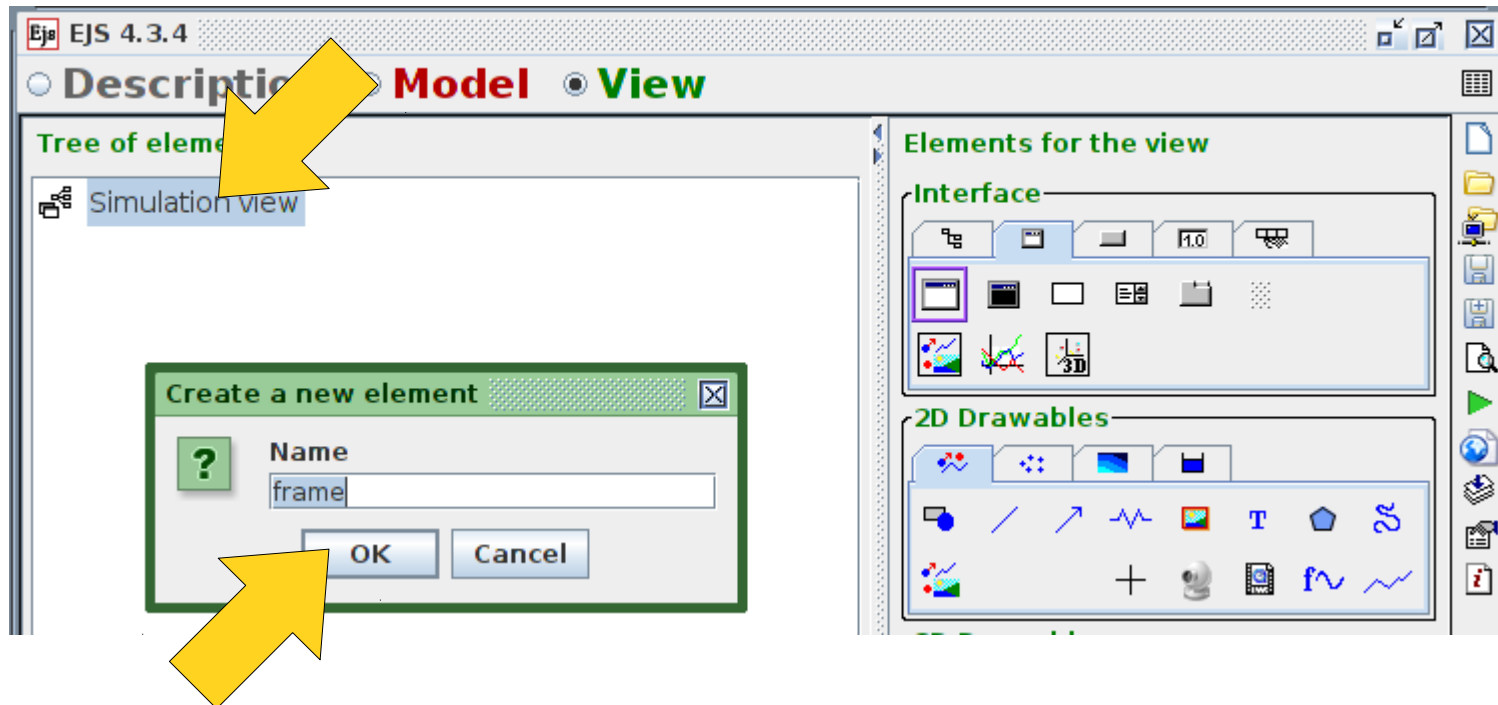
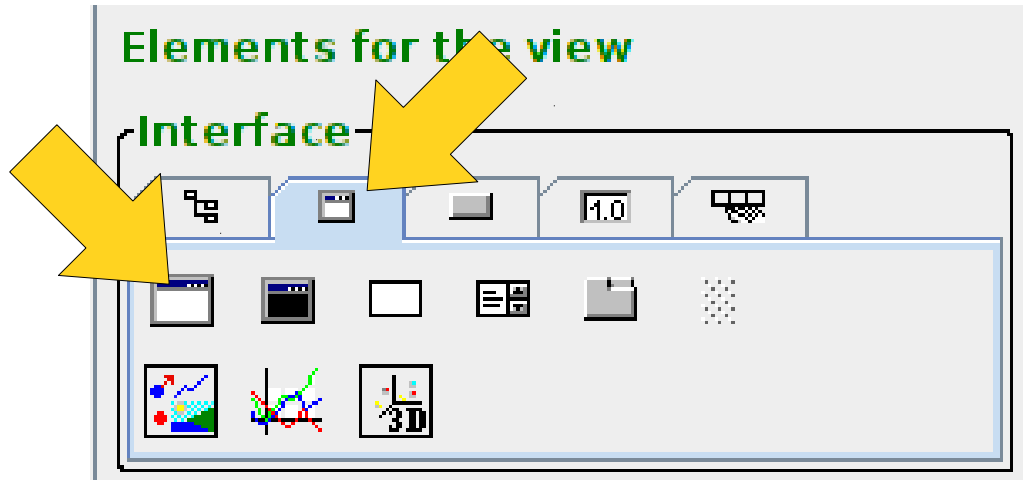
CAOS I

Nuestro primer programa en Easy Java

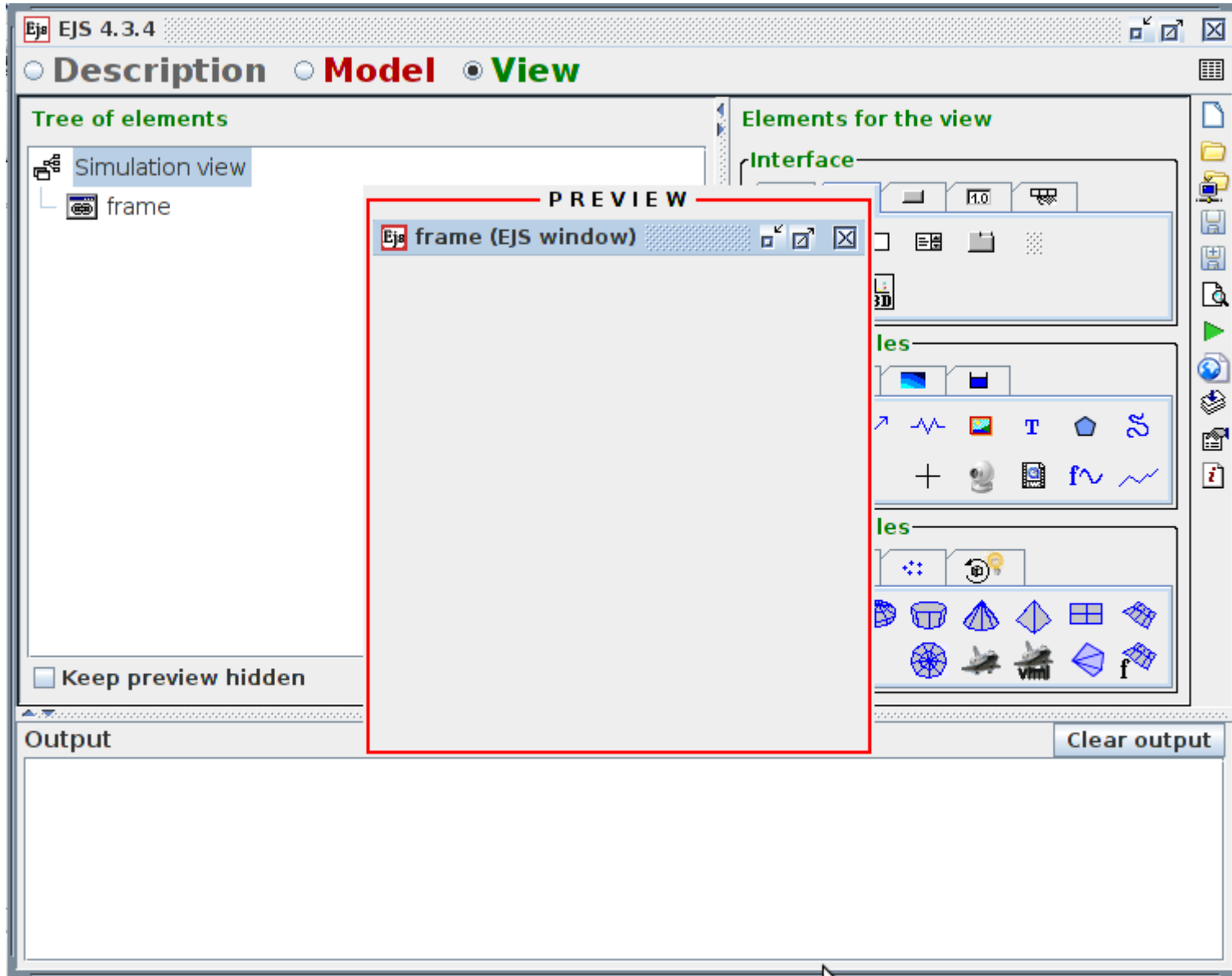
Primero diseñemos nuestro Programa:

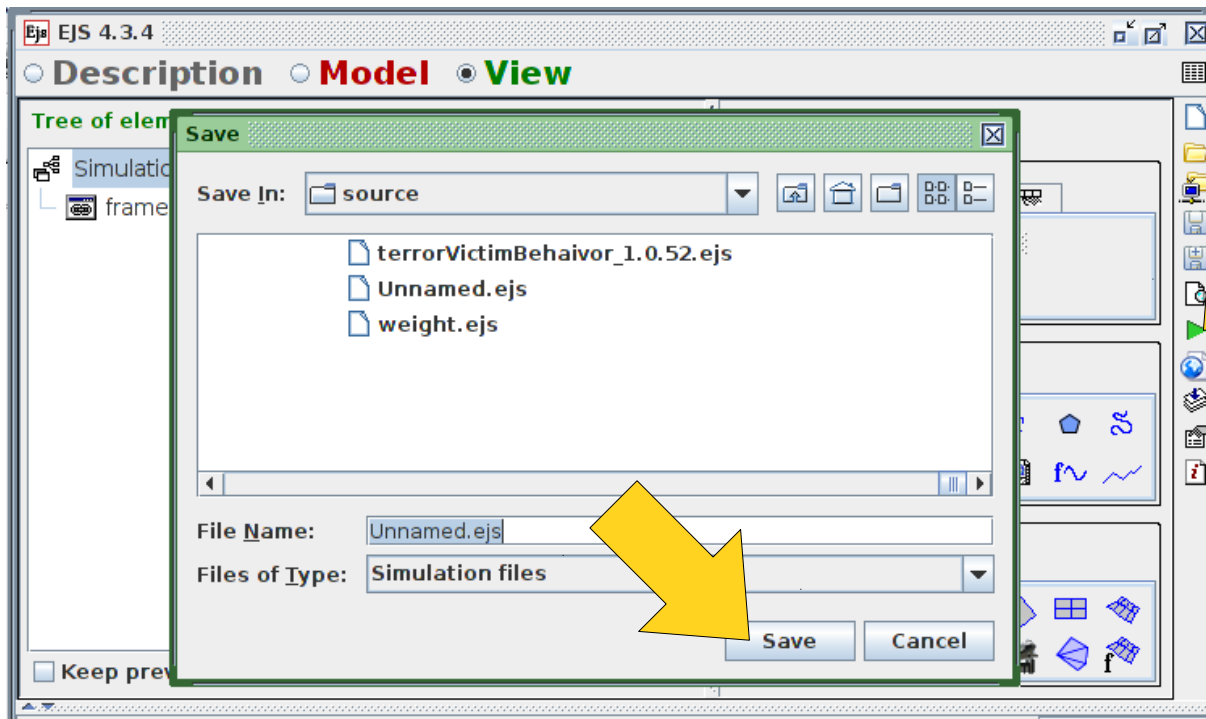


Necesitamos una mesa de trabajo: solo 4 clicks



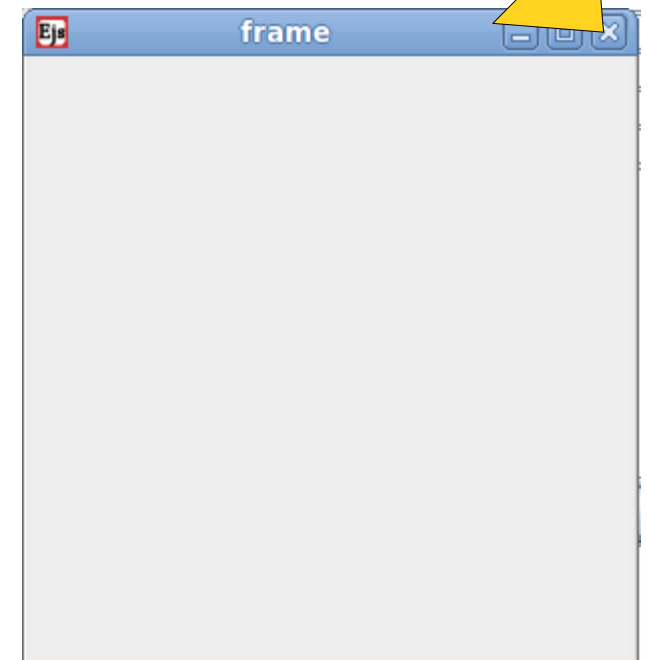
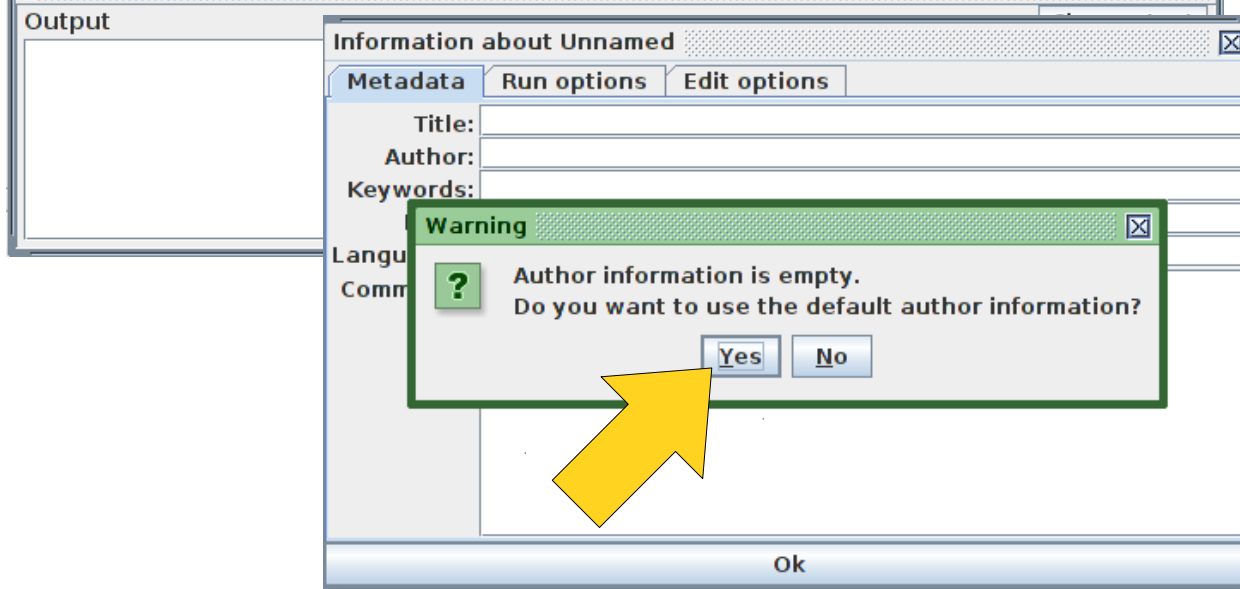
Ahora tenemos nuestro modelo “vacio” ... que ya corre!





Si lo ejecutamos nos pedira que lo guardemos ademas de especificar quien lo diseña.

El resultado es nuestro "meson" vacio que se puede cerrar en la forma tradicional.



El problema... queremos estudiar el comportamiento de la ecuación de Michael Feigenbaum que se denomina “Mapa Logístico”. Esto es, una secuencia que se define con la relación:

$$x_{n+1} = rx_n(1 - x_n)$$

¿Como se aplica? Supongamos que

$$r = 1.8 \quad x_0 = 0.2$$

Entonces

$$x_1 = rx_0(1 - x_0) = 1.8 \times 0.2 \times (1 - 0.2) = 0.288$$

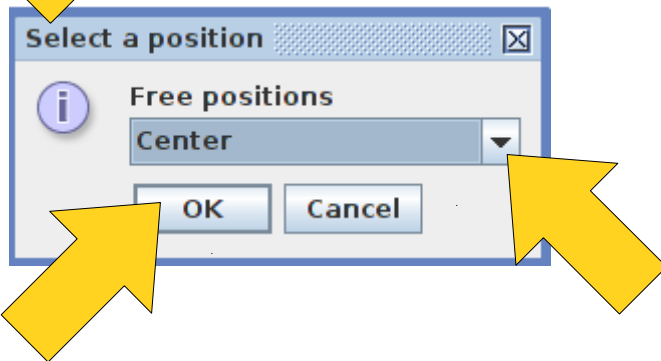
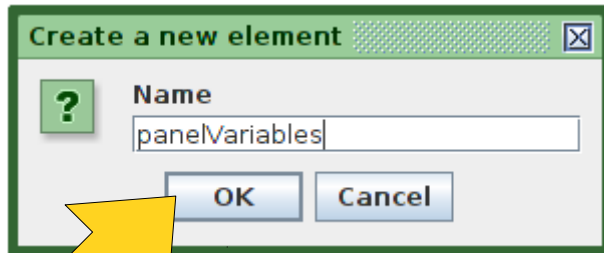
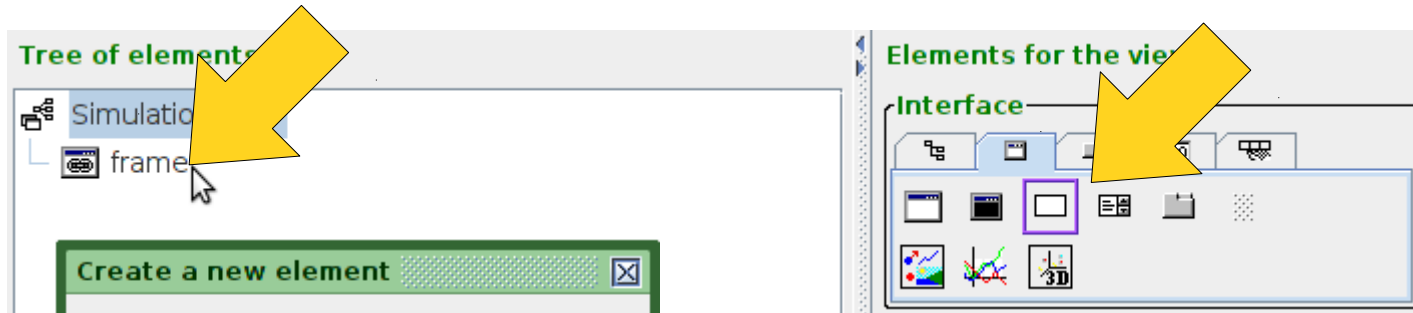
$$x_2 = rx_1(1 - x_1) = 1.8 \times 0.288 \times (1 - 0.288) = 0.369$$

$$x_3 = rx_2(1 - x_2) = 1.8 \times 0.369 \times (1 - 0.369) = 0.419$$

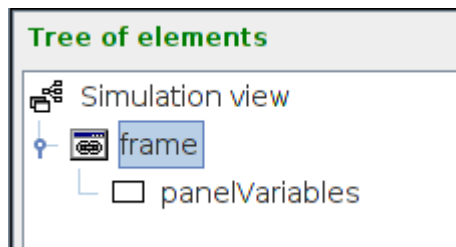
Etc.

Necesitamos poder indicarle 2 parametros al programa.

Para mantener el orden vamos a poner los campos en un panel.

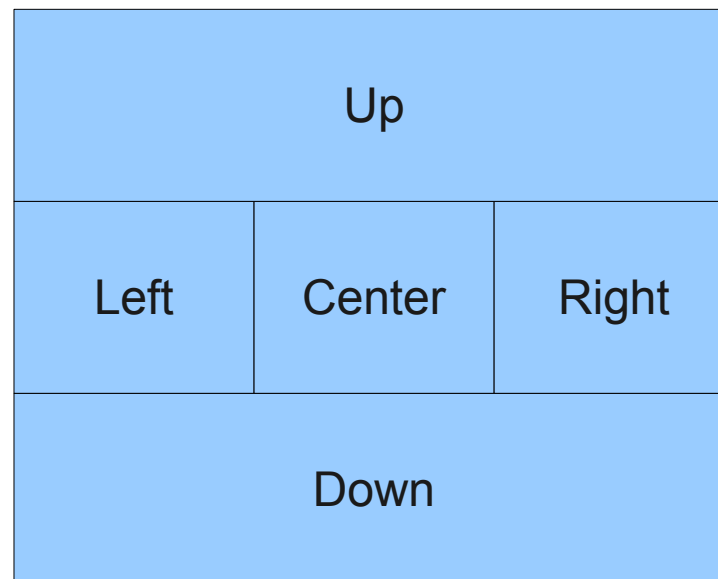


Aqui podemos usar "down" o "abajo".



Le podemos poner un nombre al panel. Recomendación: dejen la raiz del nombre para recordar de que tipo de elemento se trata.

El sistema les preguntara donde ubicar el elemento. Usa el siguiente esquema:



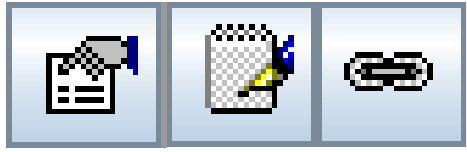
Tree of elements

- Simulation view
 - frame
 - panelVariables

Enter value

Border Title

Ok Cancel



Make your choice

True False

Ok Cancel

Make your choice

Width 0 Height 0

Ok Cancel

Make your choice

Top 0 Bottom 0
Left 0 Right 0

Ok Cancel

Properties for panelVariables (Panel)

Main		Graphical Aspect	
Layout	border	Print Target	
Visible		Background	
Size		Foreground	
Border		Font	
Border Size		Tooltip	
Border Type			
Border Color			
Border Title			
Title Position			
Title Justification			

Choose a font

Name Monospaced

Style Plain

Size (10)

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789- = \ ' ! @ # \$ % ^ & * () _ + | ~ , . / < > ?

Ok Cancel

Make your choice

Grid of 10 squares, one highlighted.

Ok Cancel

Make your choice

Grid of 6 squares with 'x' in corners.

Ok Cancel

Make your choice

Left Center Right

Ok Cancel

Make your choice

Basic colors

Show more colors...

Last colors used

Selection

Red 0
Green 0
Blue 0
Transp. 255

Ok Cancel

Properties for panelVariables (Panel)

Main		Graphical Aspect	
Layout	border	Print Target	
Visible		Background	
Size		Foreground	
		Font	
		Tooltip	

Make your choice

BorderLayout

Horizontal Box

Vertical Box

GridLayout

Rows: 0

Columns: 1

FlowLayout

Left

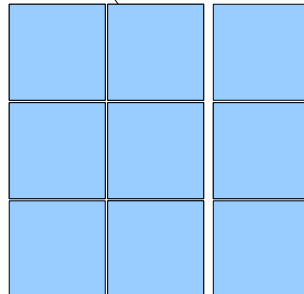
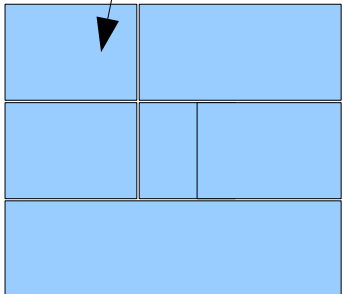
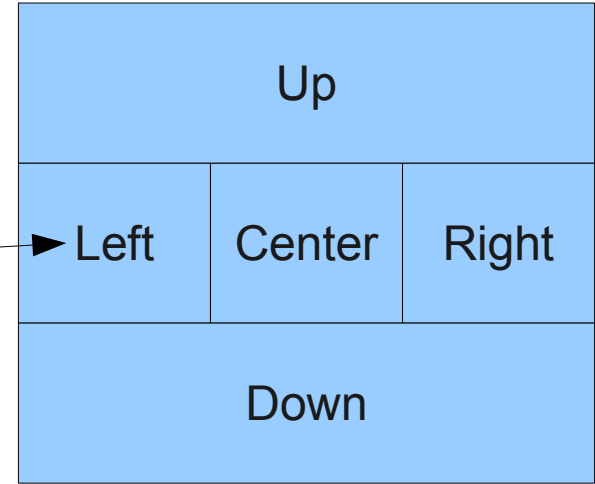
Center

Right

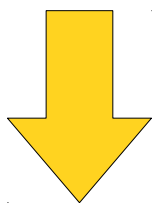
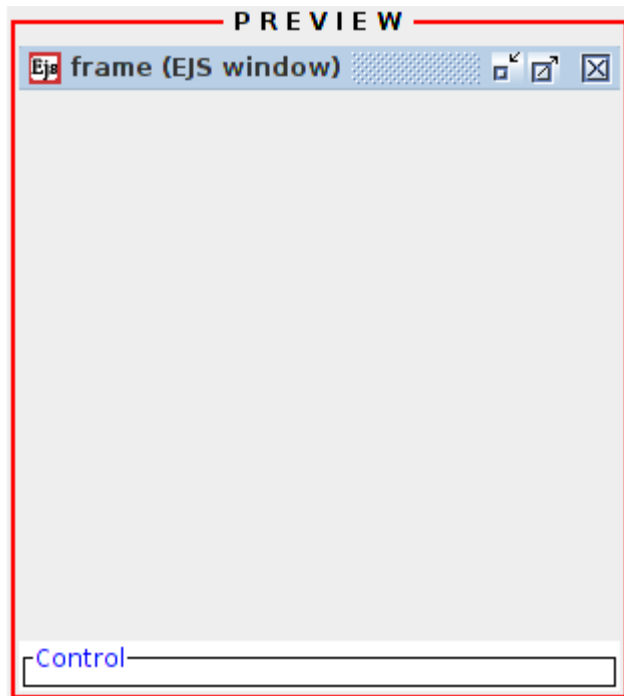
Horizontal G...: 0

Vertical G...: 0

Ok Cancel



Aqui esta nuestra box horizontal con titulo en azul, campo blanco y borde negro.

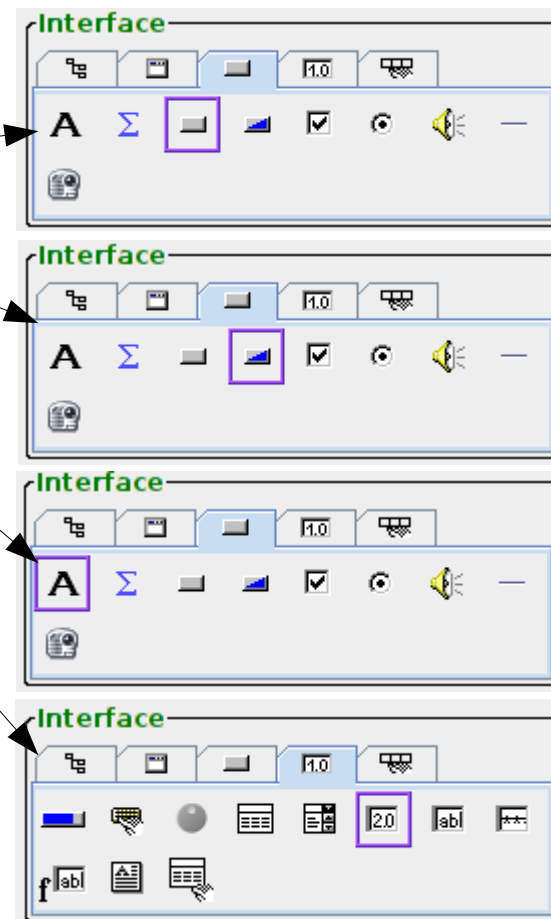


Ahora queremos agregar:

- Un boton de reinicio (RESET)
- Un boton de ejecución (PLAY) y detencion (PAUSE)
- Un cartel que diga "Constante"
- Un campo para ingresar una Constante
- Un cartel que diga "Valor inicial"
- Un campo para ingresar el Valor inicial

Nombres propuestos

- buttonReset
- twoStateButtonGo
- labelConstante
- fieldConstante
- labelValorInicial
- fieldValorInicial



Tree of elements

- Simulation view
 - frame
 - panelVariables
 - buttonReset
 - twoStateButtonGo
 - labelConstante
 - fieldConstante
 - labelValorInicio

Creemos espacio con iconos: borremos textos y seleccionemos iconos.

Properties for buttonReset (Button)

Main		Graphical Aspect	
Text	"buttonReset"	Visible	
Image		Size	
Mnemonic		Background	
Alignment		Foreground	
		Font	
		Tooltip	

Graphic images

Look In: source

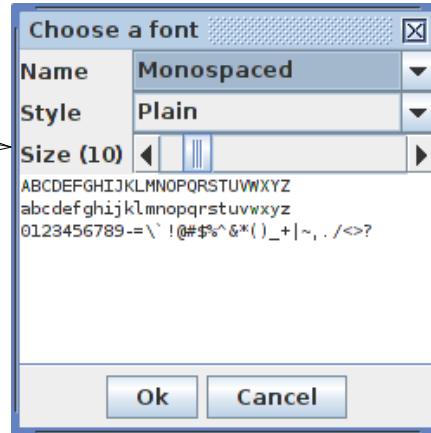
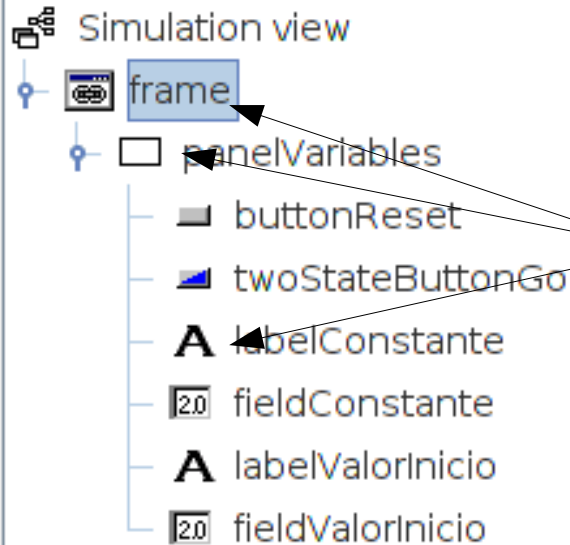
climate	UFRO
enterprise	walker
eq	Alta-UACH.jpg
G4Heart	Alta-UACH2.jpg
G4Test	show01.jpg
model	show02.jpg
sales	
soccer	

Control

Properties for twoStateButtonGo (TwoStateButton)

Main		On Properties		Off Properties	
Variable	_isPaused	Text On	"Play"	Text Off	"Pause"
Enabled		Image On		Image Off	
Alignment		Mnemonic On		Mnemonic Off	
Graphical Aspect		Action On	_play()	Action Off	_pause()
Visible		Foreground On		Foreground Off	
Size		Background On		Background Off	
Font					
Tooltip					

Tree of elements

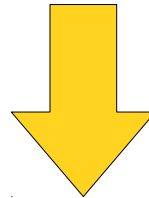


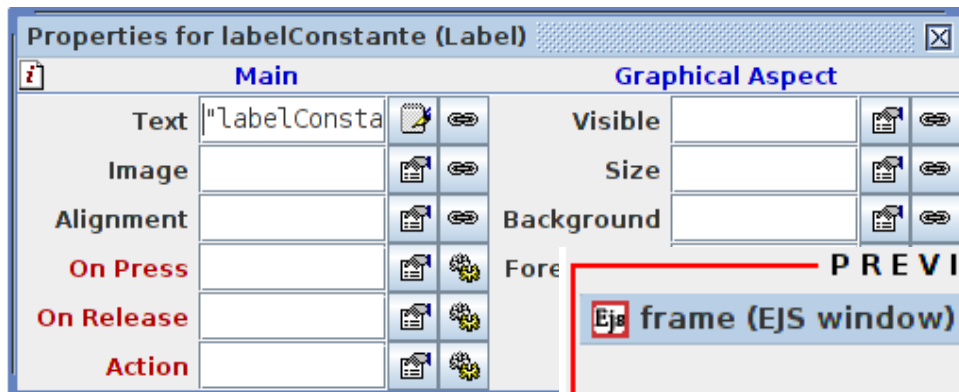
Las propiedades se heredan a los elementos dependientes... ejemplo los Fonts o colores

Si colocamos

Arial,Plain,12

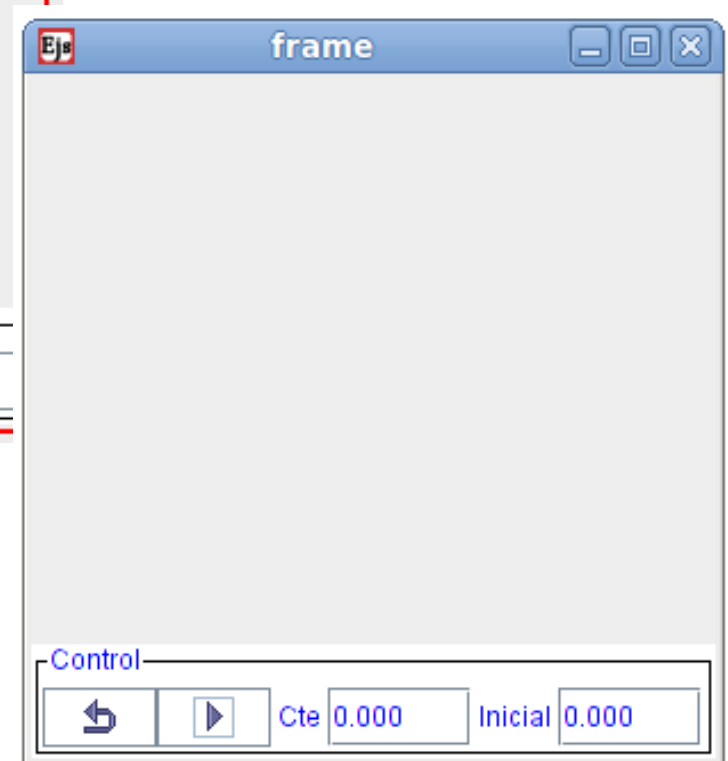
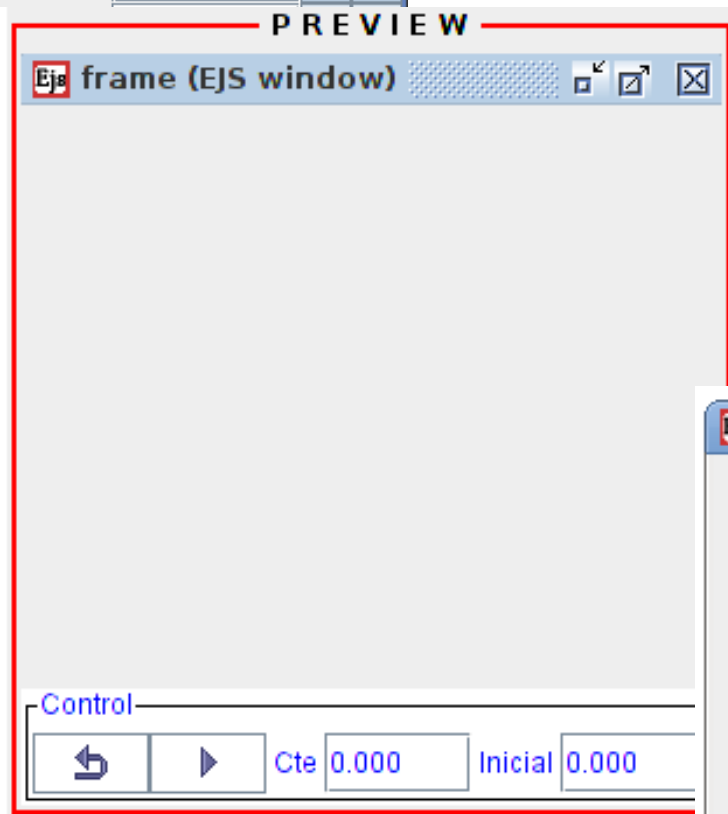
En el frame



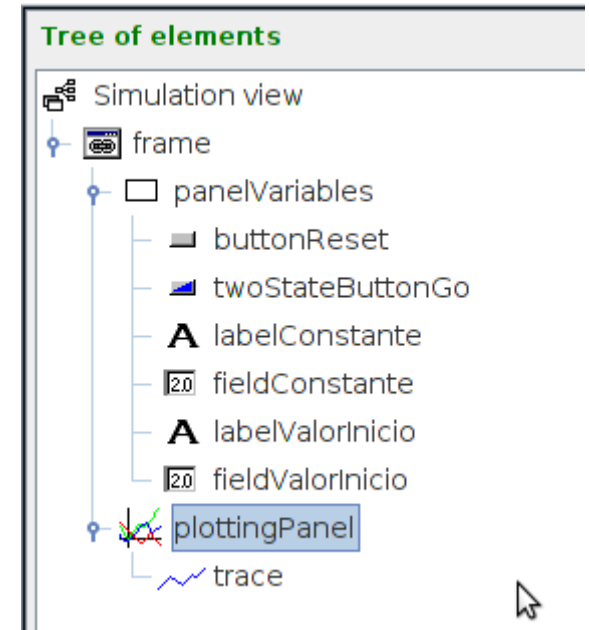
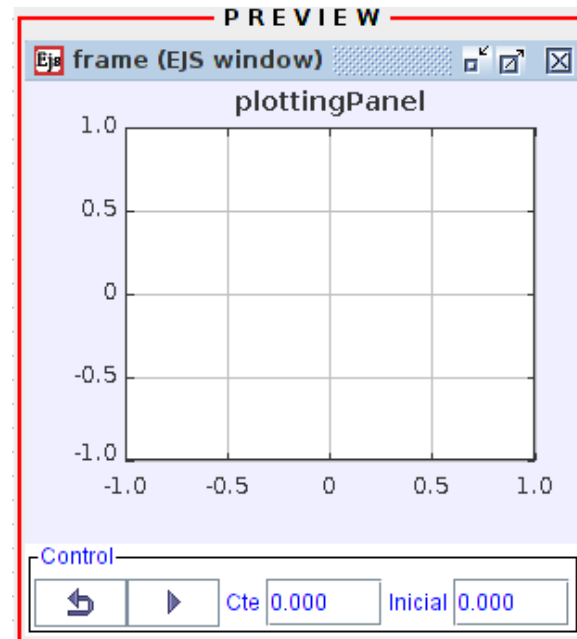
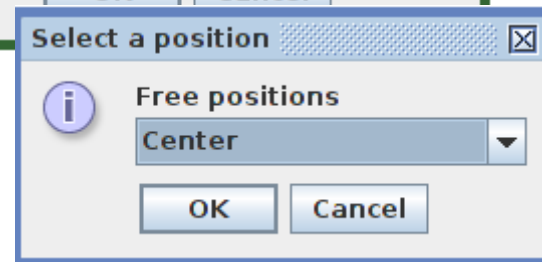
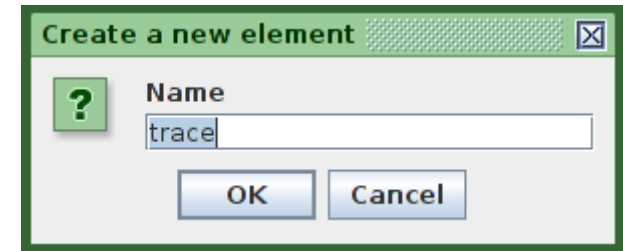
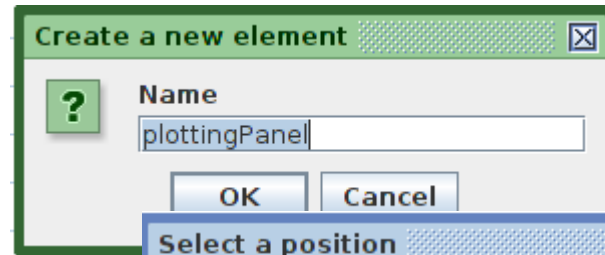
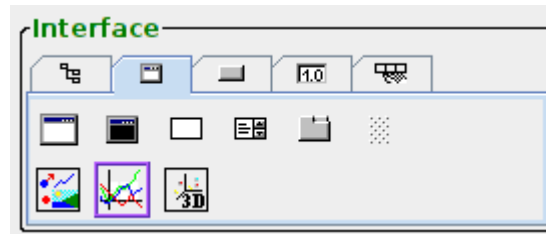


Si ahora editamos las propiedades de los textos (labels) podemos ingresar los nombres "Cte" e "Inicial".

Si se ejecuta el programa no solo se ve el resultado grafico, tambien estan activo los botones de ejecución.



Ahora agregamos una sistema de Coordenadas (ejes) y una curva:



Podemos embellecer el grafico: sin titulo, eje Y en manual, minimo 0, maximo 1.

Scales		Decoration		Configuration	
Autoscale X	true	Title		Anti Aliasing	
Autoscale Y	false	Title Font		Fixed Gutters	
Minimum X		Show Axes		Gutters	
Maximum X		Axes Type		Coordinates	
Minimum Y	0	Title X		X Format	
Maximum Y	1	X Axis Pos		Y Format	
Square		X Axis Type		Messages	
X Margin (%)		Grid X		Expression	
Y Margin (%)		Title Y		Expr Format	
Visibility and Interaction		Y Axis Pos		TL Message	
Menu		Y Axis Type		TR Message	
Menu Entry		Grid Y		BL Message	
Pos X		Delta R		BR Message	
Pos Y		Delta Theta		Graphical Aspect	
On Press				Visible	
On Drag				Size	
On Release				Interior	
On Enter				Background	
On Exit				Foreground	
Key Action				Font	
Key Pressed				Tooltip	
Print Target					

Tambien podemos embellecer la curva: ej. color rojo

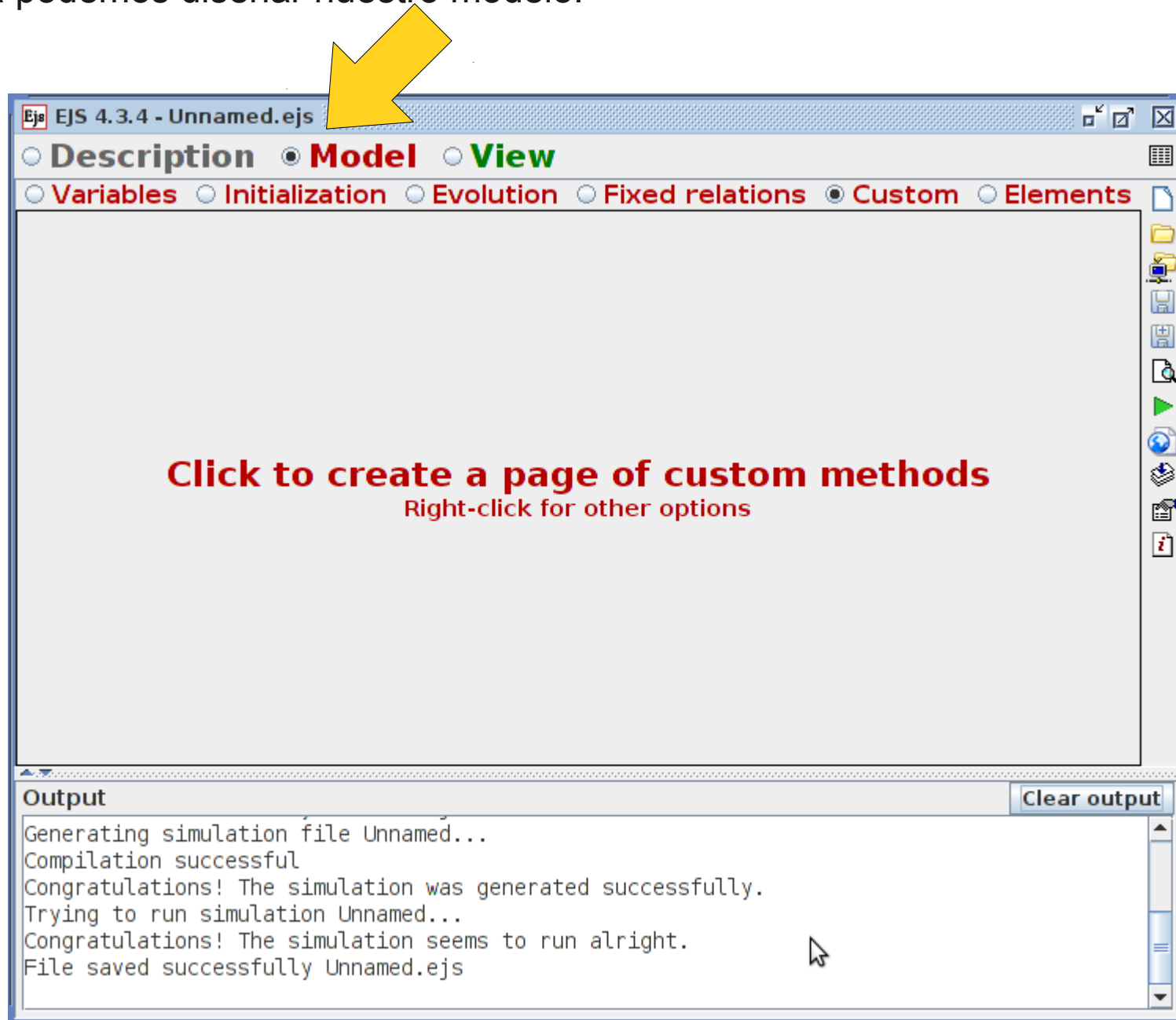
The image displays the 'Properties for trace (Trace)' dialog box in EJS software, which is used to configure the appearance and behavior of a trace. The dialog is divided into three main sections:

- Input:** Contains fields for 'Input X', 'Input Y', 'Position and Size' (Pos X, Pos Y, Position Array, Size X, Size Y, Size Array), and 'Memory' (Memory, Mem Display, Mem Color).
- Visibility and Interaction:** Contains fields for 'Menu Entry', 'Visible', 'Measured', 'Draggable', 'Sensitivity', and several event-driven actions: 'On Press', 'On Drag', 'On Release', 'On Enter', and 'On Exit'.
- Graphical Aspect:** Contains fields for 'Max Points', 'Clear at Input', 'Skip', 'Active', 'No Repeat' (set to true), 'Connected' (set to true), and 'Line Color' (set to red).

An arrow points from the text 'ej. color rojo' to the 'Line Color' field in the 'Graphical Aspect' section.

The 'PREVIEW' window shows a coordinate plane with axes ranging from -1.0 to 1.0. Below the graph is a 'Control' panel with a play button, a 'Cte' field set to 0.000, and an 'Inicial' field set to 0.000.

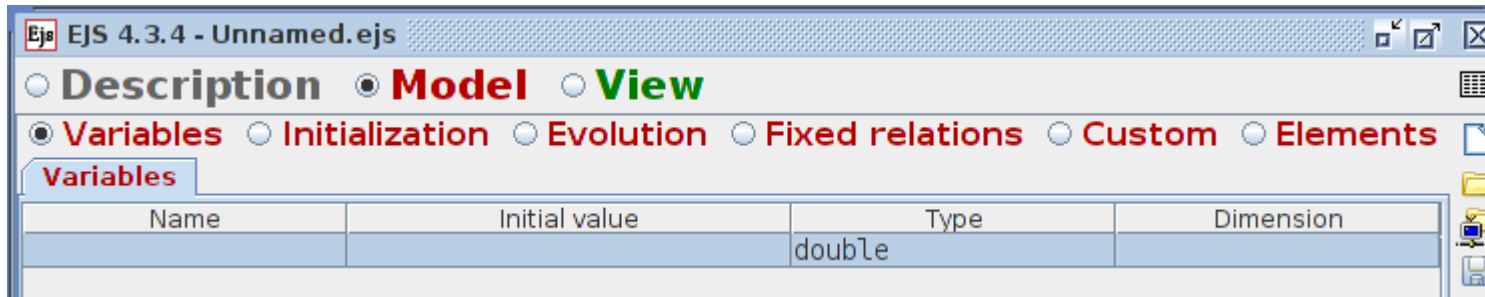
Ahora podemos diseñar nuestro modelo:



Primero definamos las variables

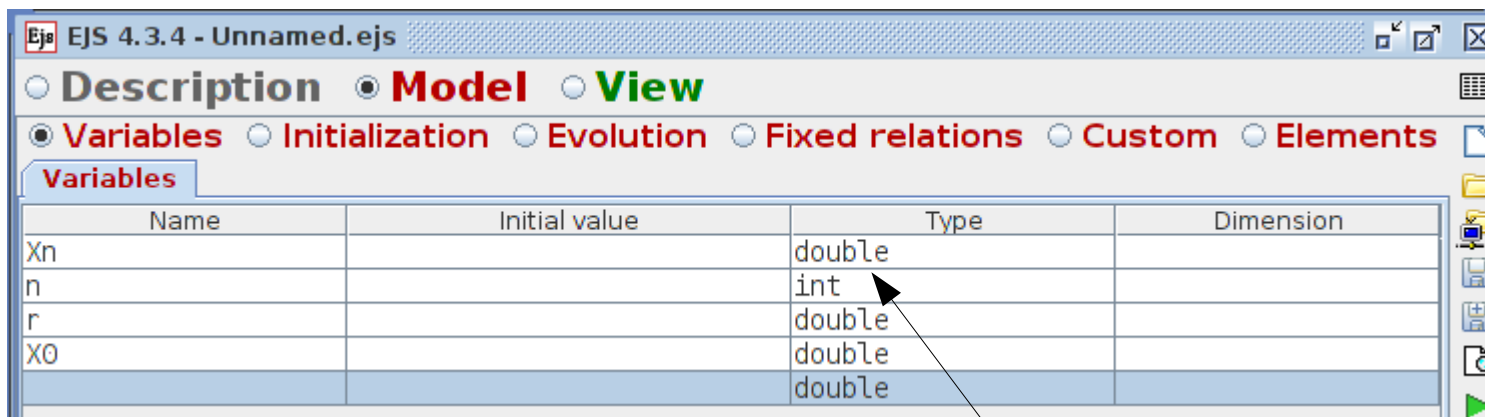
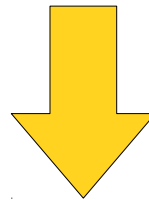
The screenshot shows the EJS 4.3.4 interface. The 'Model' tab is selected, and the 'Variables' sub-tab is active. A dialog box titled 'Enter the new name' is open, with 'Variables' entered in the 'Name' field. A yellow arrow points to the 'OK' button. Below the dialog, the 'Output' window shows the text 'Generating simulation file Unnamed'. At the bottom, a table displays the variable definition:

Name	Initial value	Type	Dimension
		double	



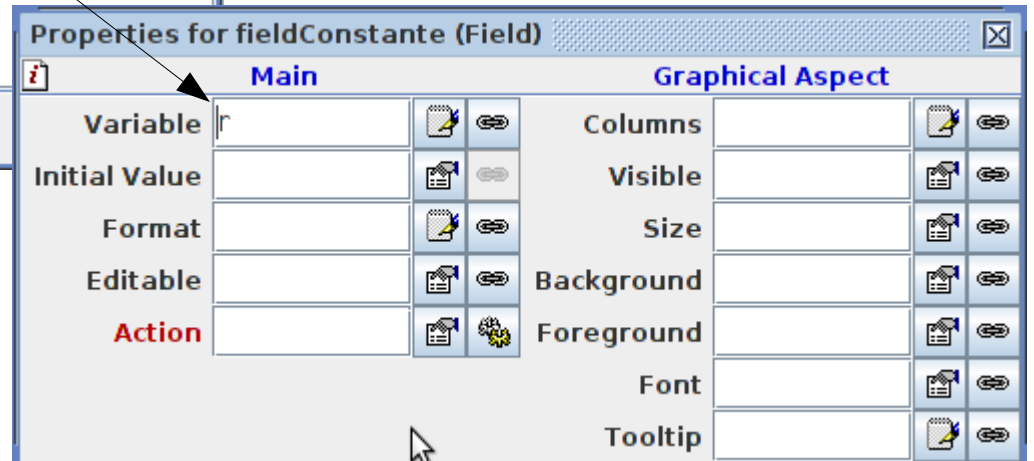
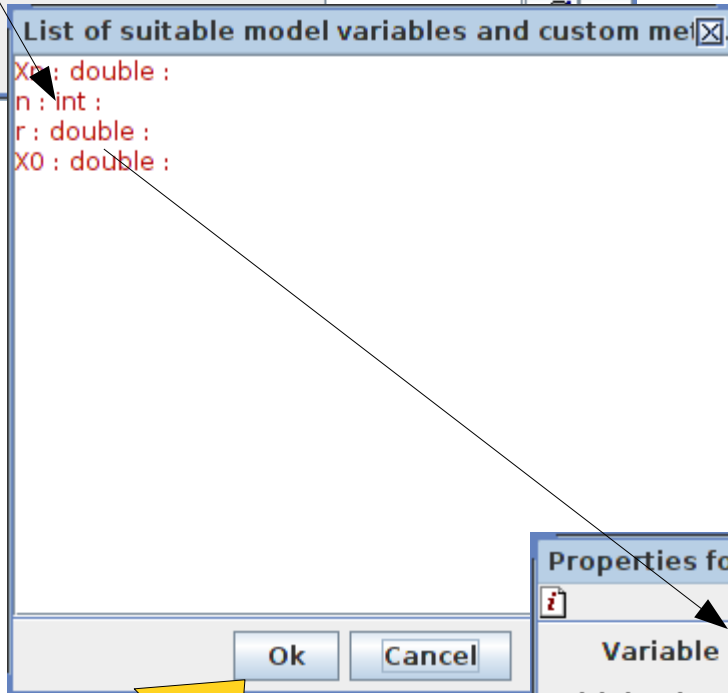
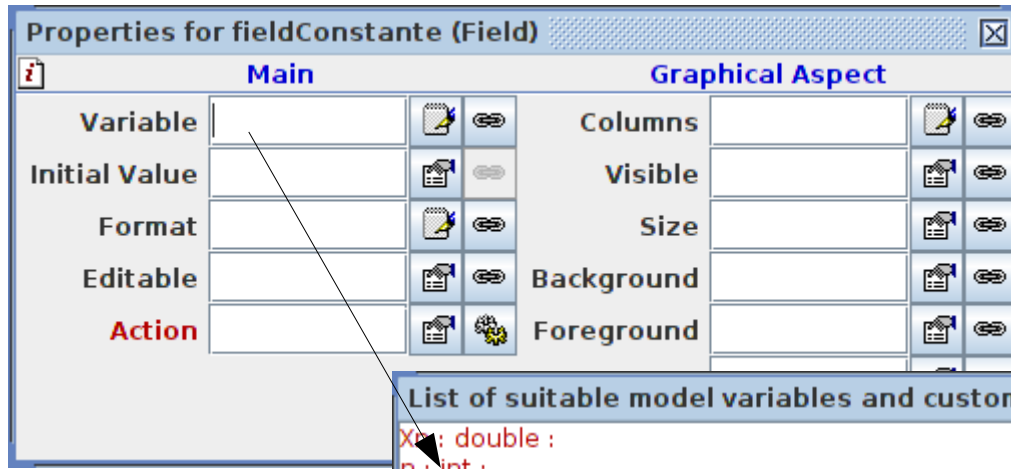
Necesitamos

nuestra variable (X_n),
el contador (n),
la constante (r) y
el valor inicial (X_0)

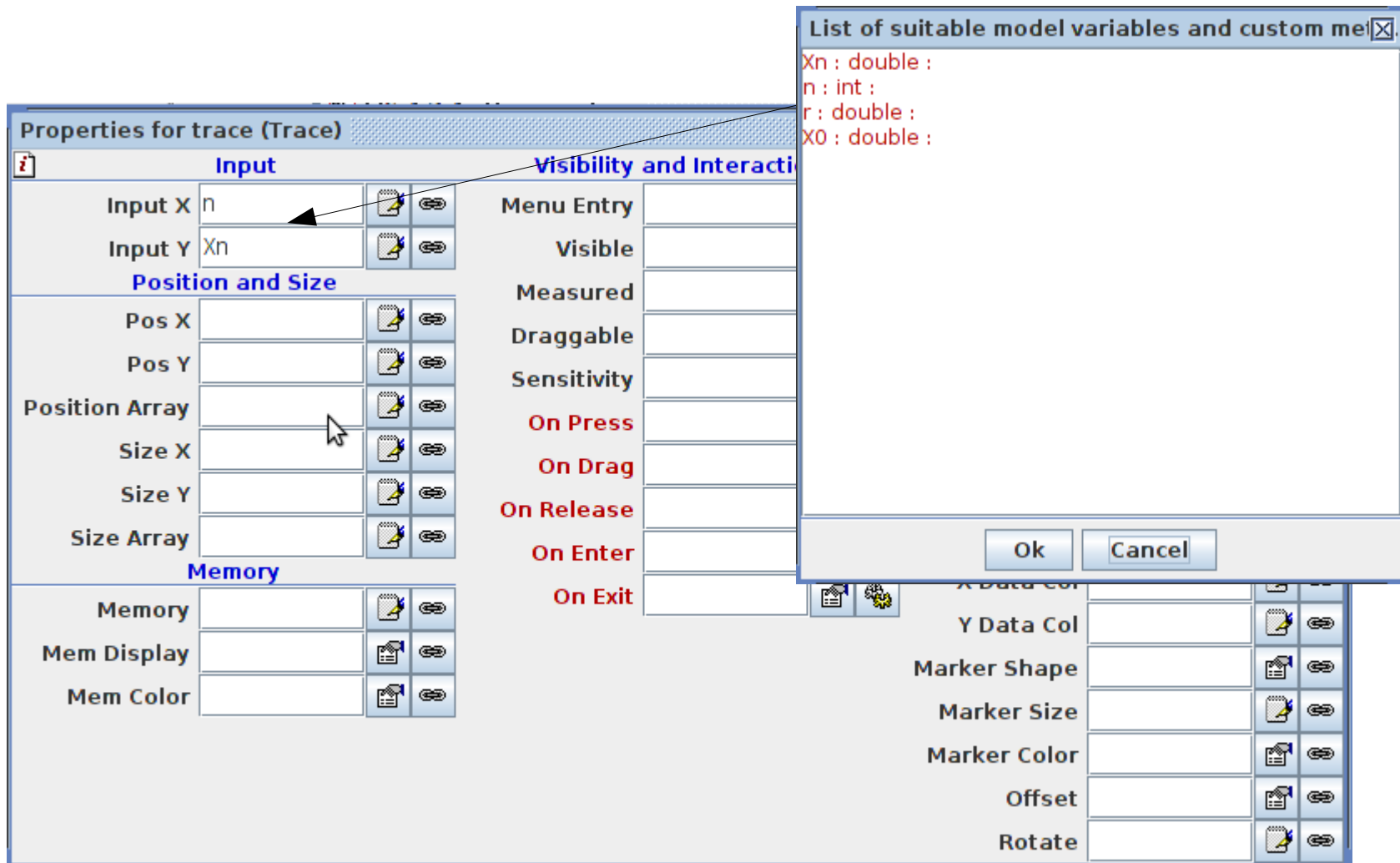


Las variables pueden ser reales (double = doble precision), enteros (int = integers) textos (String) o objetos (objects).

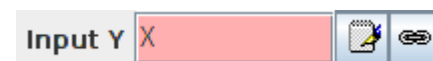
Ahora podemos asociar los campos de la constante y valor inicial a las variables r y $X0$ que definimos.



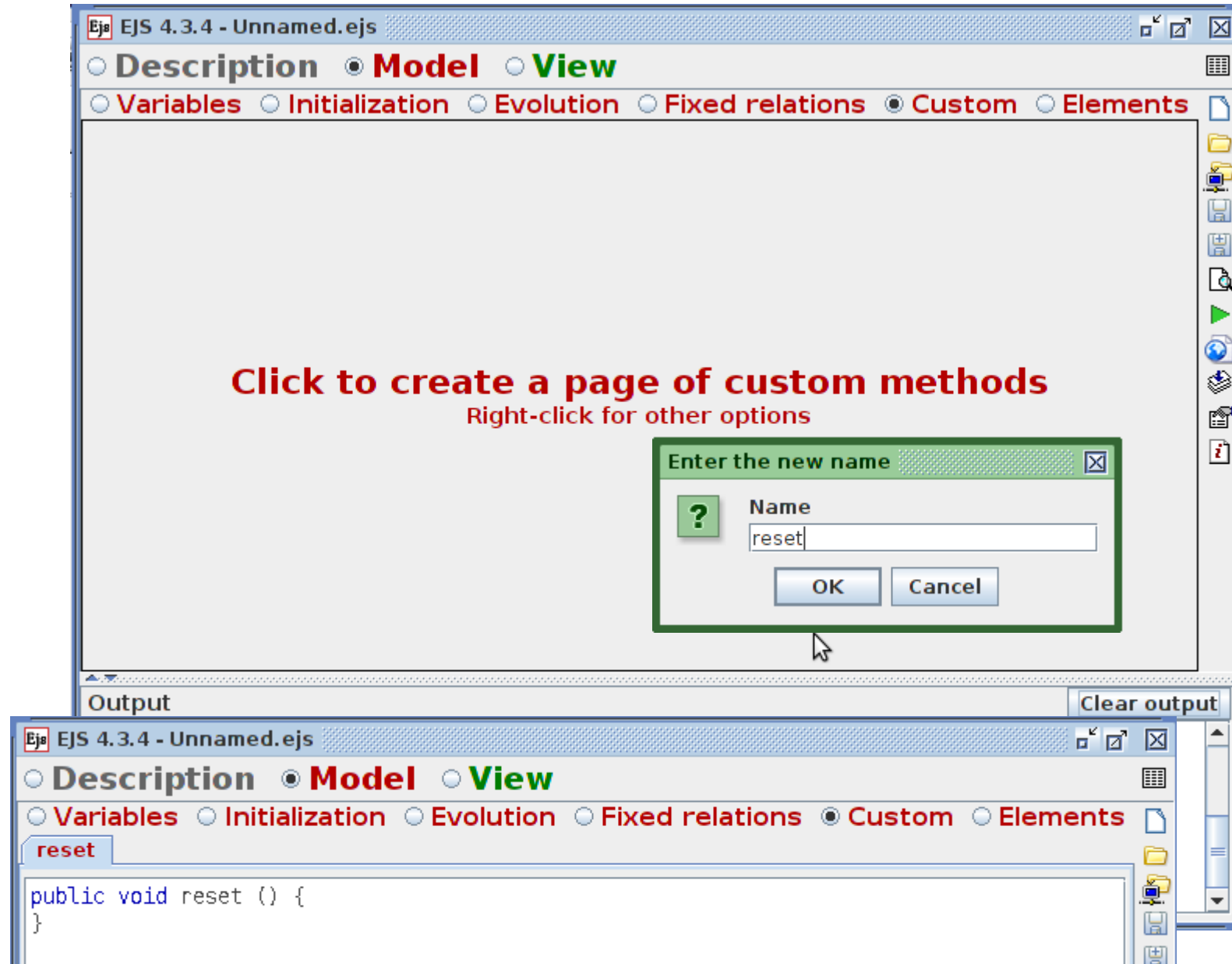
De igual forma podemos asociar las variables contador (n) y valor calculado (Xn) a la curva



Nota sobre editar: amarillo aun "no ingresado", rojo "error"



Antes de ingresar la ecuacion de comportamiento debemos todavia definir una funcion que resetea la simulacion. Esto se hace bajo "Custom" (Propio)



Ahora debemos aprender JAVA ya que la rutina se escribe en dicho lenguaje. En este caso lo unico que hay que hacer es definir los valores iniciales:

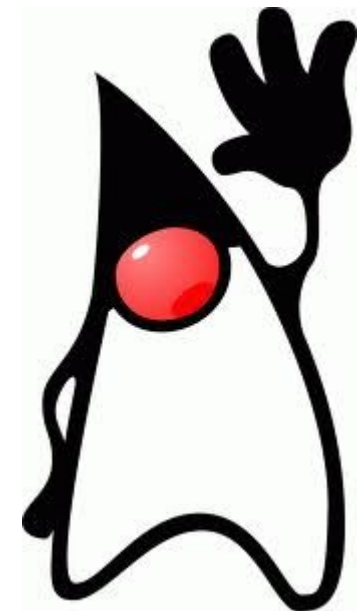
```
n = 0;  
Xn = X0;
```

Donde el “;” marca el final del comando. Como la función no retorna un valor (al inicio se declaro void o sea vacia) no incluye un comando return.

```
public void reset () {  
  
    n = 0;  
    Xn = X0;  
  
    _view.trace.clear();  
  
}
```



Ademas debemos incluir una orden para borrar la curva. Los elementos se llaman con _view. y el sistema muestra los elementos y comandos.



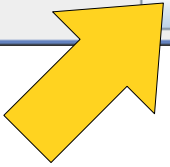
Ahora podemos asociar la función al boton de reseteo.

Ademas es util asociarlo a los campos de la constante y del valor Inicial de modo que si se modifican resetean.

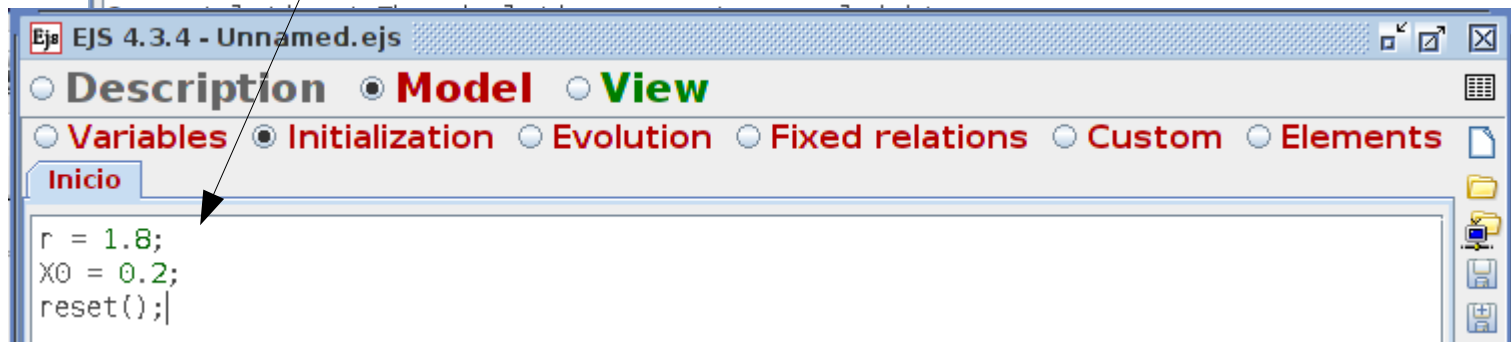
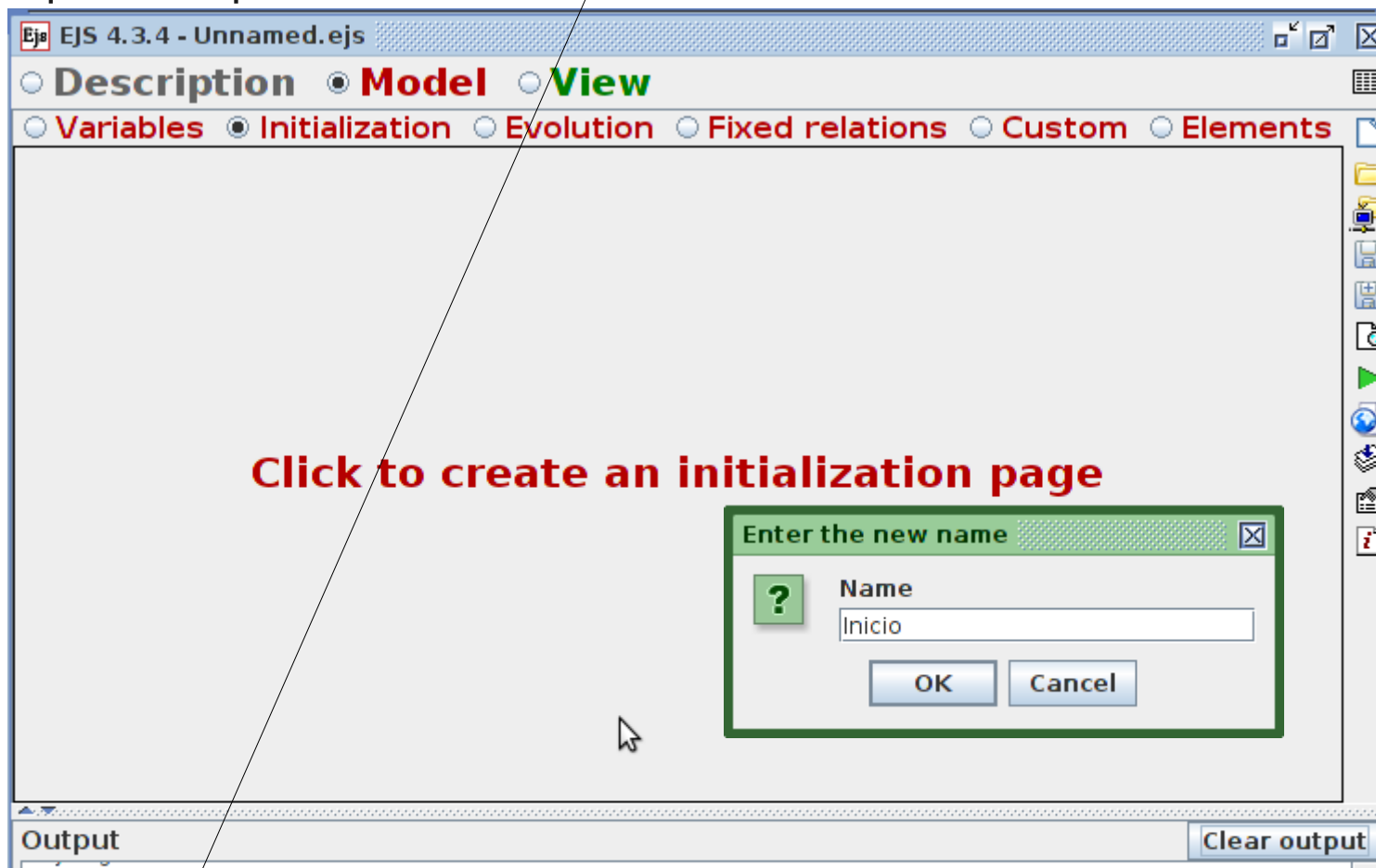
Main		Graphical Aspect	
Text		Visible	
Image	"/org/openso	Size	
Mnemonic		Background	
Alignment		Foreground	
Enabled		Font	
Action		Tooltip	
Immediate			

reset () :
_play() : Play the simulation
_pause() : Pause the simulation
_step() : Run one step of the evolution
_setFPS(int) : Set the (approximate) number of frames
_setDelay(int) : Set the delay (in milliseconds) for the e
_getDelay() : Return the current delay (in milliseconds)
_setSPD(int) : Set the number of model steps per evolu
_reset() : Completely reset the simulation
_initialize() : Read view values and initialize
_isApplet() : Whether the simulation runs as an applet
_isPlaying() : Whether the simulation is playing
_isPaused() : Whether the simulation is paused
_setPageEnabled("pageName"
_resetSolvers() : Synchronizes
_getStringProperty("propertyN
_getArguments() : Return the
_getParameter("parameterN

Main		Graphical Aspect	
Text		Visible	
Image	"/org/openso	Size	
Mnemonic		Background	
Alignment		Foreground	
Enabled		Font	
Action	reset ()	Tooltip	
Immediate			



Para que el programa comience bien se recomienda incluir en el Inicio los vares iniciales que las variables que se editan y llamar la funcion reset para que todo quede seteado.



Ahora podemos finalmente ingresar lo que seria la ecuación de movimiento. Conviene ademas desactivar la ejecución automatica al inicio (Autoplay).

The screenshot shows the EJS 4.3.4 software interface. The main window is titled "Ejs 4.3.4 - Unnamed.ejs". It has a menu bar with "Description", "Model", and "View". Below the menu bar are radio buttons for "Variables", "Initialization", "Evolution", "Fixed relations", "Custom", and "Elements". On the left side, there is a vertical slider for "Frames per second" ranging from 1 to 100, with a current value of 20. Below the slider are input fields for "FPS" (20) and "SPD" (1), and a checked "Autoplay" checkbox. The main workspace is divided into two sections. The top section contains the text "Click to create a page of code" in red. The bottom section contains the text "Click to create a page of ODEs" in red. A mouse cursor is positioned over the top section. A dialog box titled "Enter the new name" is open, showing a text input field with "MapaLogistico" and "OK" and "Cancel" buttons. The bottom of the window features an "Output" window with a "Clear output" button. The output text reads: "Congratulations! The simulation seems to run alright. File saved successfully Unnamed.ejs Warning. Syntax error or custom page reset: Parse error at line 8, column 3. Encountered: } Warning. Syntax error or custom page reset: Parse error at line 8, column 3. Encountered: }".

En este ejercicio trabajaremos con código y no ecuaciones diferenciales

Incrementar contador

EJS 4.3.4 - Unnamed.ejs

Description **Model** View

Variables Initialization Evolution Fixed

Frames per second

100

20

MapaLogistico

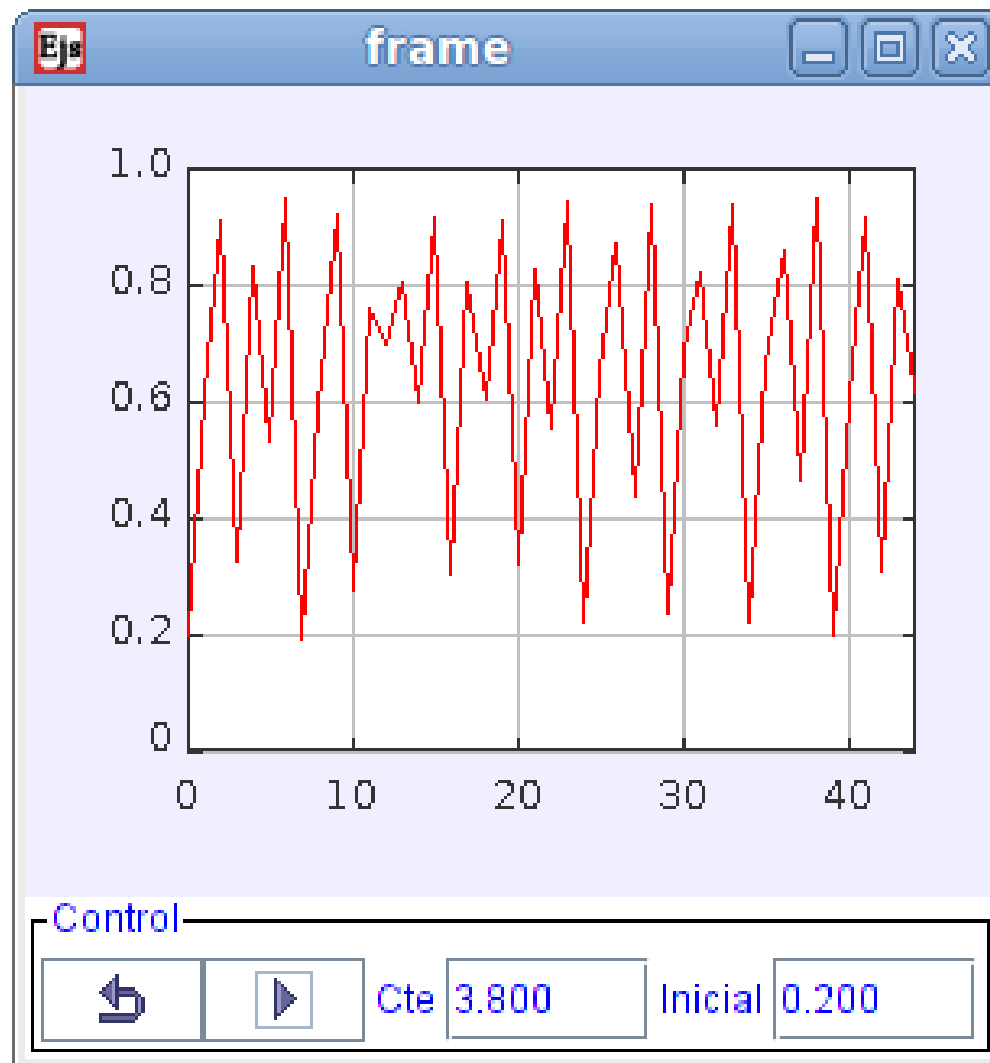
```
n++;  
Xn=r*Xn*(1-Xn);
```

Valor "nuevo"

Asignación

Valor "viejo"

... y listo!!!



¿Se verifica el comportamiento indicado?

