



ras. Jornadas gvSIG, México  
Geomática y software libre en  
México: Retos y posibilidades

# Scripting en gvSIG

Joaquín José del Cerro  
[jjdelcerro@gvsig.com](mailto:jjdelcerro@gvsig.com)



# Que podemos hacer con scripting

Scripting no es una herramienta GIS.

¿ Para que sirve ?

¿Que podemos hacer con ella ?



# Que podemos hacer con scripting

Scripting no es una herramienta GIS.

¿ Para que sirve ?

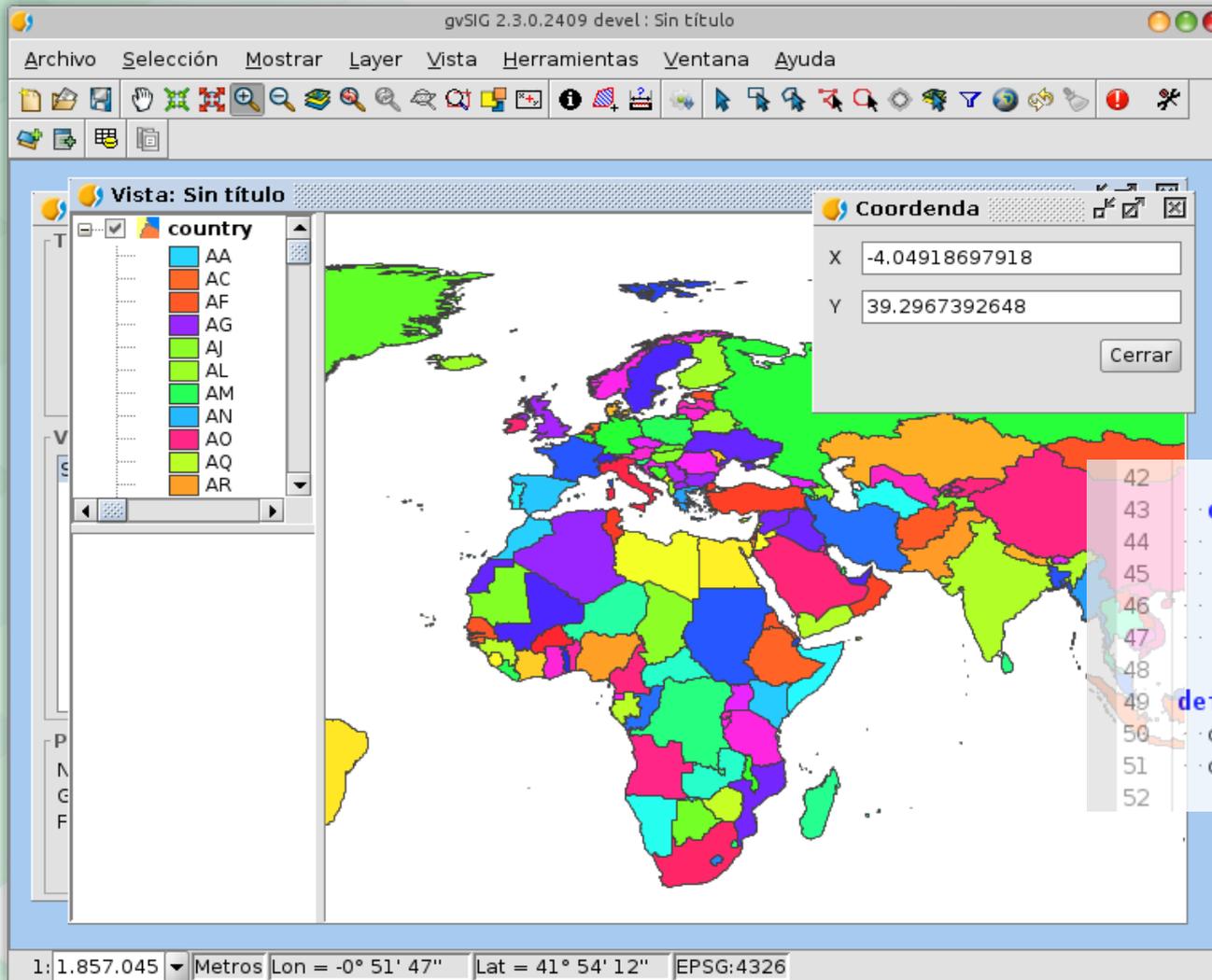
¿Que podemos hacer con ella ?

**Podemos crear nuestras propias herramientas.**



# Que podemos hacer con scripting

## Consultar coordenadas





# Que podemos hacer con scripting

```

37 def extentChanged(self, *args):
38     self.calcular_sumatorio()
39
40 def calcular_sumatorio(self):
41     encuadre = self.mapContext.getViewPort().getEnvelope().getGeometry()
42     lineas = self.layer.getFeatureStore().getFeatureSet().iterator()
43     suma = 0
44     for linea in lineas:
45         g = linea.getDefaultGeometry()
46         if encuadre.intersects(g):
47             suma += int(linea.get(self.fieldname))
48         self.txtSuma.setText(str(suma))
49
50 def main(*args):
51     if currentView() == None:
52         msgbox("Debera tener una vista abierta y activa")
53     return
  
```

monit...  
 pob\_total 57292  
 Cerrar

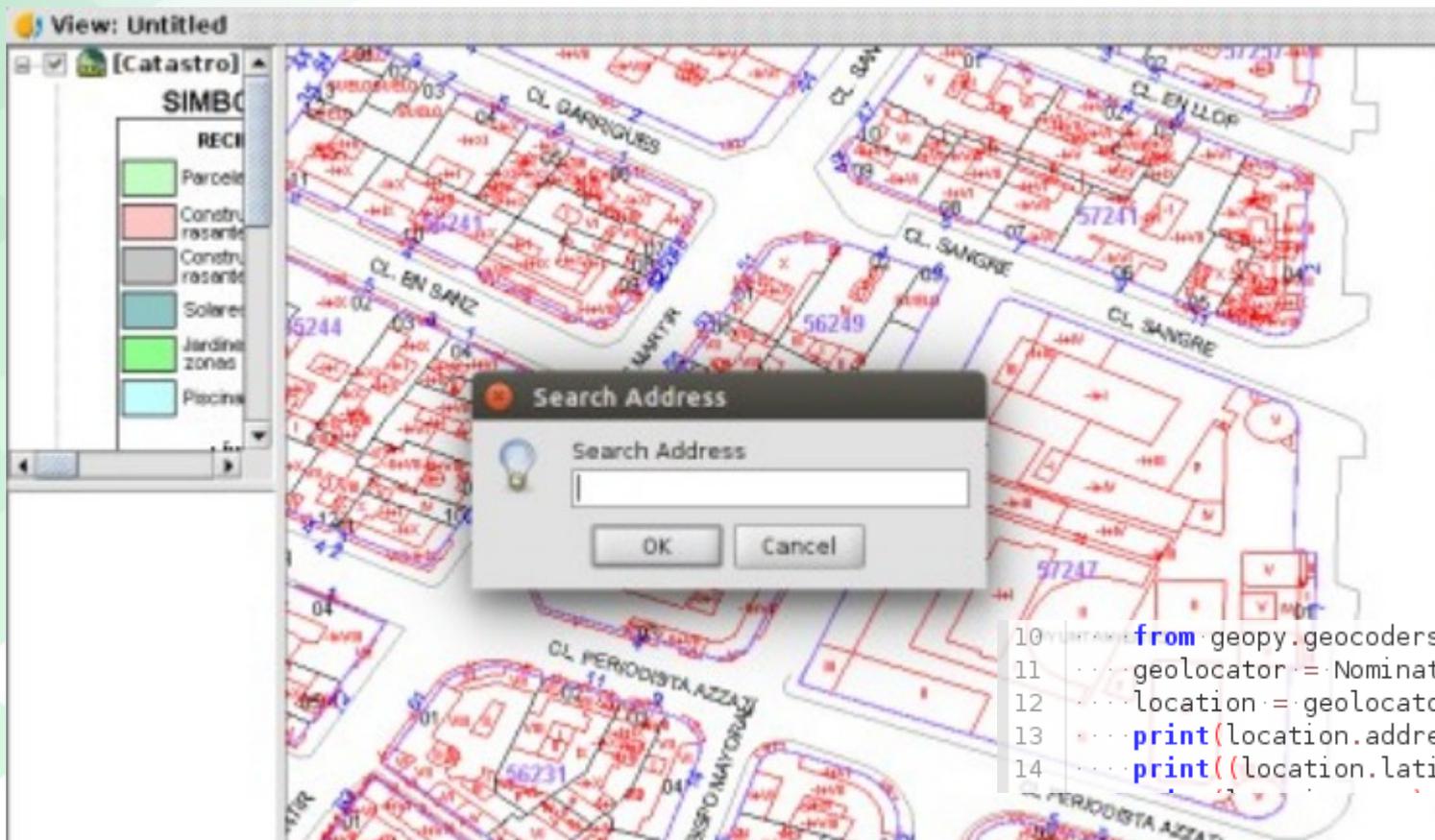
1: 13.734 Metros X = 726.437,68 Y = 4.371.187,05 EPSG:23030

Monitorizar datos



# Que podemos hacer con scripting

Usar servicios web  
especializados  
geopy

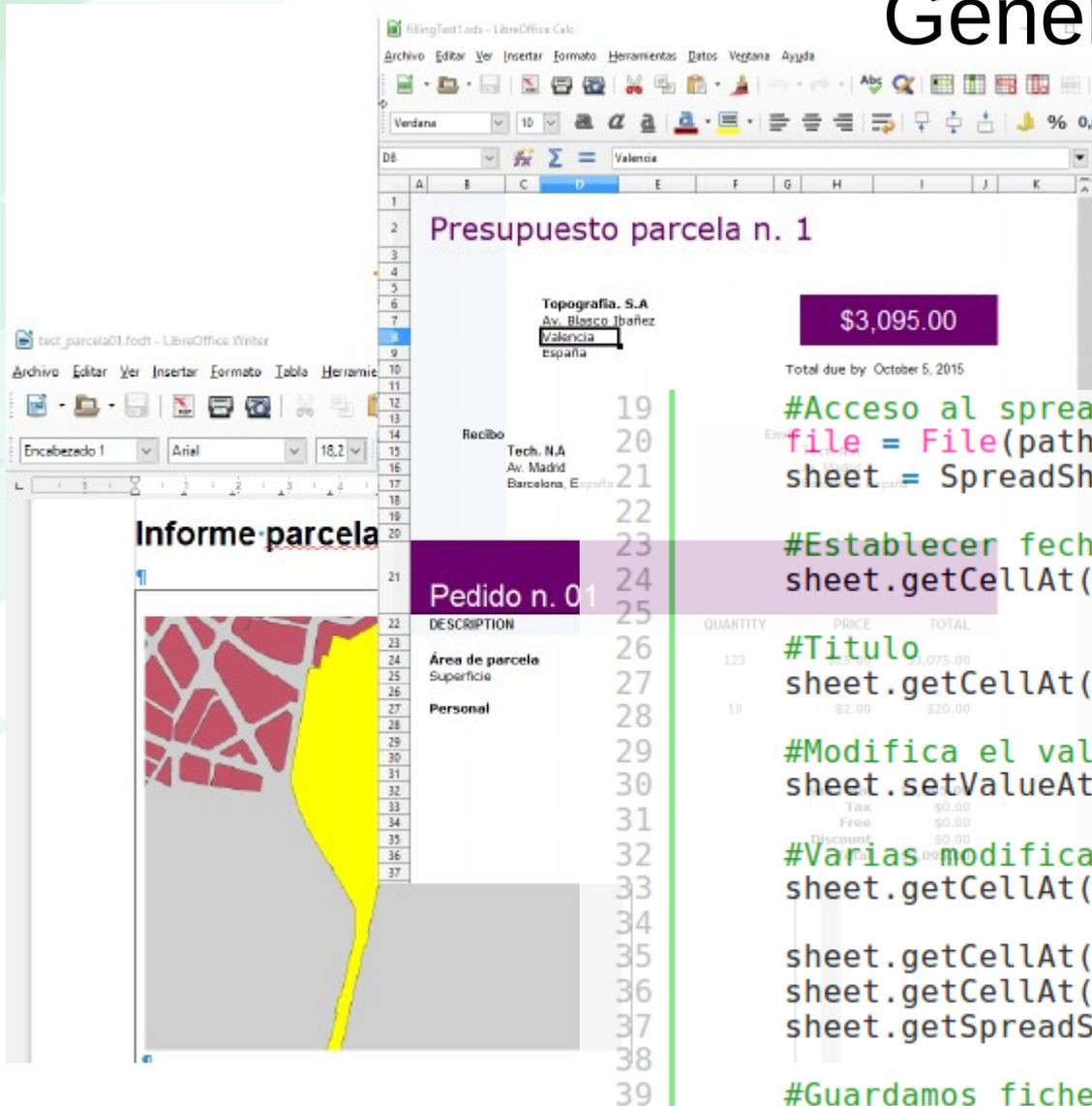


```
10 from geopy.geocoders import Nominatim
11 ... geolocator = Nominatim()
12 ... location = geolocator.geocode("175 5th Avenue NYC")
13 ... print(location.address)
14 ... print((location.latitude, location.longitude))
```



# Que podemos hacer con scripting

## Generación de informes Personalizados jOpenDocument



```

#Acceso al spreadsheet y numero de hoja
file = File(pathTemplate)
sheet = SpreadSheet.createFromFile(file).getSheet(0)

#Establecer fecha actual a la castilla I10
sheet.getCellAt("I10").setValue(Date())

#Titulo
sheet.getCellAt("B21").setValue("Pedido n. 01")

#Modifica el valor a la celda 1,1. Seria B2
sheet.setValueAt("Presupuesto parcela n. 1", 1, 1)

#Varias modificaciones a las celdas
sheet.getCellAt("F24").setValue(123)

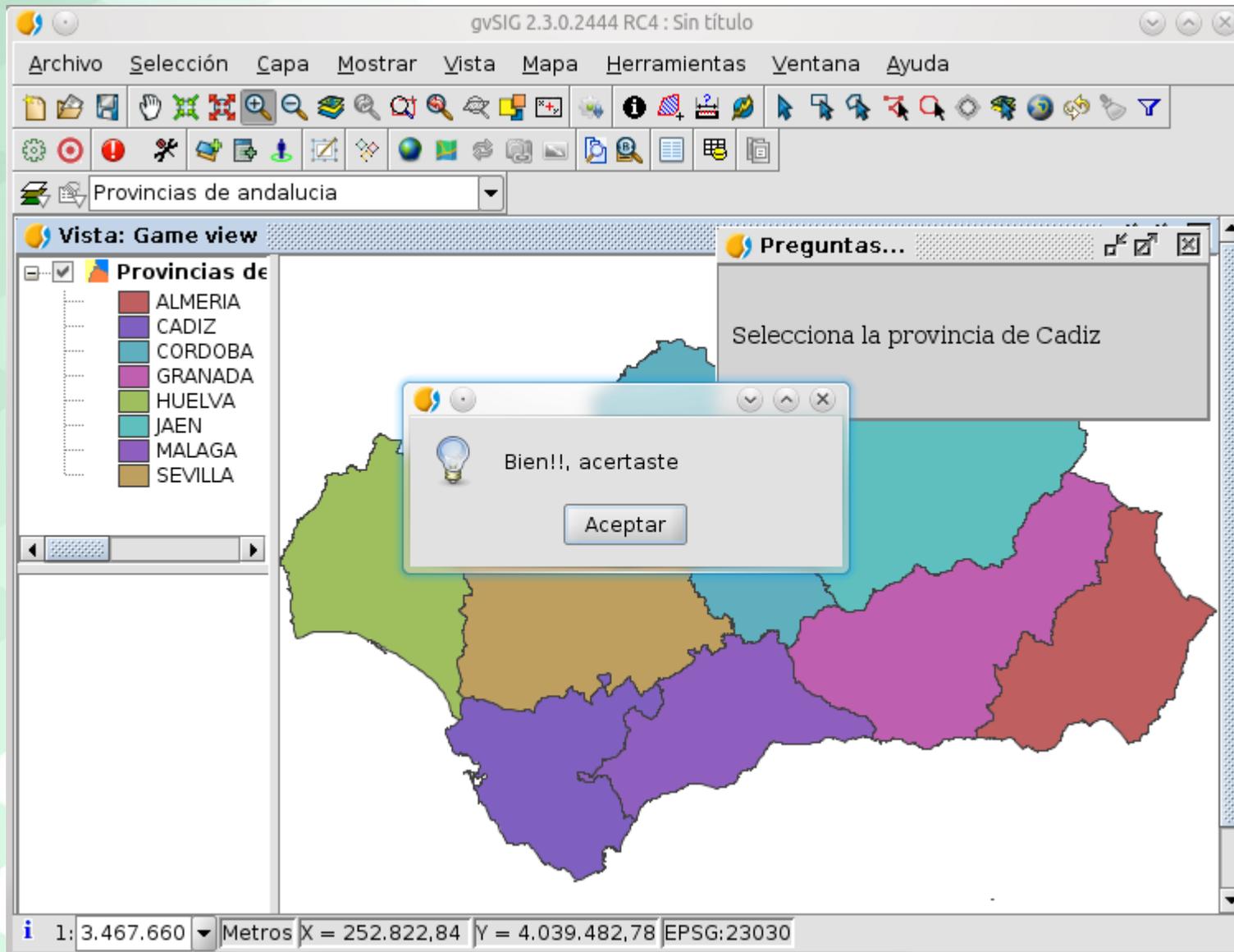
sheet.getCellAt("B27").setValue("Personal")
sheet.getCellAt("H27").setValue(2)
sheet.getSpreadSheet().getTableModel("Products").setVa

#Guardamos fichero
  
```



# Que podemos hacer con scripting

Juegos  
didácticos





# Que podemos hacer con scripting

## Personalizar mapas

The screenshot shows the QGIS 2.3.0.2444 RC4 interface. The main map window displays a map of Spain with a legend containing 'Espana\_comu' and 'esp\_provincias'. A script editor is open over the map, showing Python code for customizing the map layout. The 'Customize layout' dialog box is also open, showing options for selecting a map template, tags for view and title, and expressions for the title and logo path.

```

25 context = mapa.getLayoutContext()
26 # Nos recorremos todos los elementos del mapa buscando los que hemos etiquetado
27 for elemento in context.getAllFFrames():
28     if elemento.getTag() == tagVista:
29         # Ajustamos el encuadre de la vista al del elemento seleccionado
30         mapContext = elemento.getMapContext()
31         mapContext.getViewPort().setEnvelope(encuadre)
32         mapContext.invalidate()
33     elif elemento.getTag() == tagTitulo:
34         # Cambiamos el rotulo del texto que habiamos etiquetado en el mapa
35         elemento.clearText()
36         elemento.addText(titulo)
37     elif elemento.getTag() == tagLogo:
38         # Cambiamos la imagen del logotipo
39         elemento.setImage(logo)
40     context.fullRefresh()
41
42 def btnCerrar_click(self, *args):

```

The 'Customize layout' dialog box shows the following settings:

- Seleccione el mapa a usar como plantilla: **Plantilla**
- Tag para identificar a la vista: **Vista**
- Tag para identificar al titulo: **Titulo**
- Tag para identificar al logotipo: **Titulo**
- Expresion para componer el titulo: "Provincias de..." + NOMBRE99
- Expresion para componer la ruta al logotipo: "data/logos/" + COM[2:4] + ".png"

Buttons: **Actualizar mapa**, **Cerrar**



# El entorno de scripting

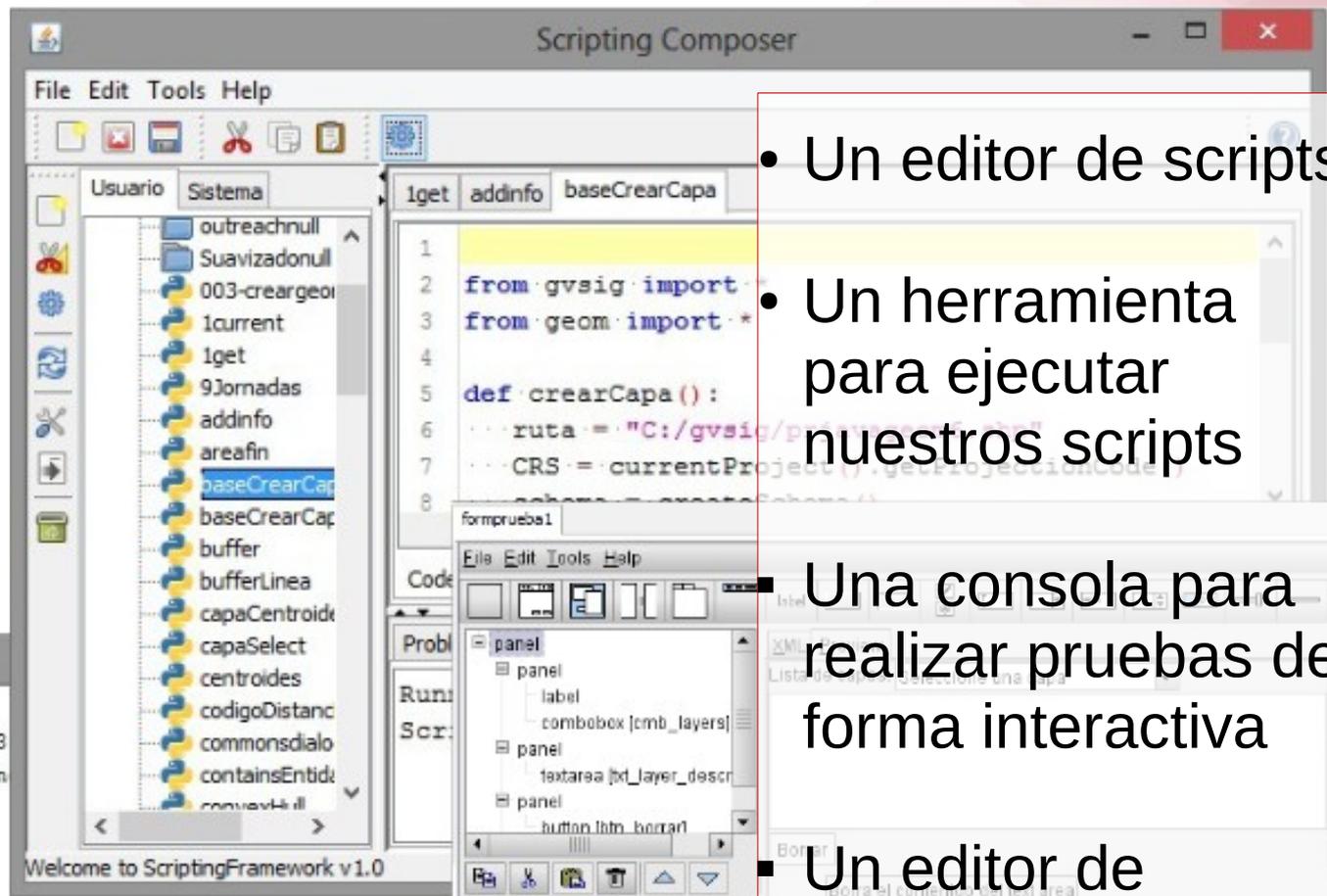
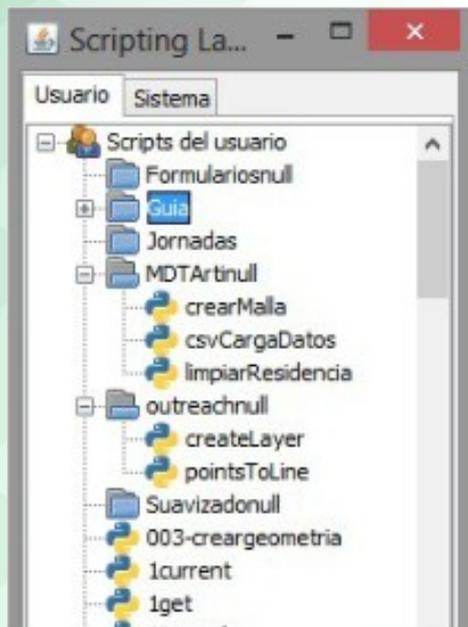
Y...

Para hacer estas cosas...

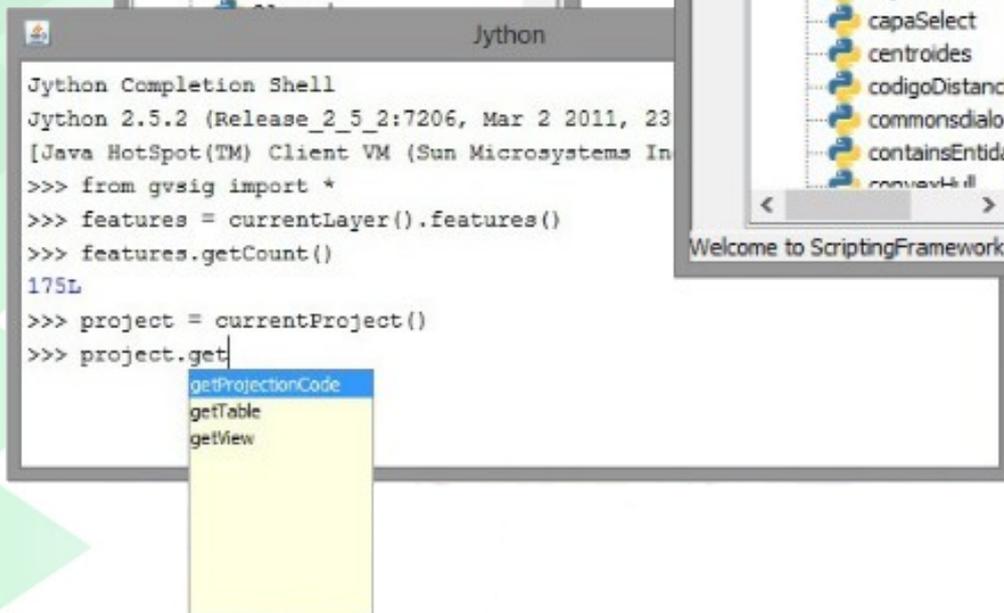
¿ de que entorno disponemos ?



# El entorno de scripting



- Un editor de scripts
- Un herramienta para ejecutar nuestros scripts
- Una consola para realizar pruebas de forma interactiva
- Un editor de formularios





# El entorno de scripting

Todo integrado en gvSIG

Vamos a hacer un repaso de las principales funcionalidades



# Funcionalidades

## Lenguajes disponibles

- Python (Jython 2.7.1b3)
- Groovy (2.4.6)
- javascript
- R (Renjin)
- Acceso (limitado) a R nativo a traves de python
- Scala (experimental)



# El entorno de scripting

## Funcionalidades

Lenguajes disponibles

Desde el proyecto apostamos por **Python**

Python + Java = **Jython**



# El entorno de scripting

## Funcionalidades

### Editor de scripts con...

```
15 class MonitorSuma(ViewPortListener,FormPanel):
16
17     def __init__(self, mapContext, layer, label, fieldname):
18         FormPanel.__init__(self,script.getResource("monitor_suma3.xml"))
19         ....
20         self.mapContext = mapContext
21         self.layer = layer
22         self.fieldname = fieldname
23         self.lblEtiqueta.setText(label)
24         self.calcular_sumatorio()
25         mapContext.getViewPort().addViewPortListener(self)
26
27     def btnCerrar_click(self,*args):
28         self.mapContext.getViewPort().removeViewPortListener(self)
29         self.hide()
30         ....
31     def backColorChanged(self,*args):
32         pass
33
```

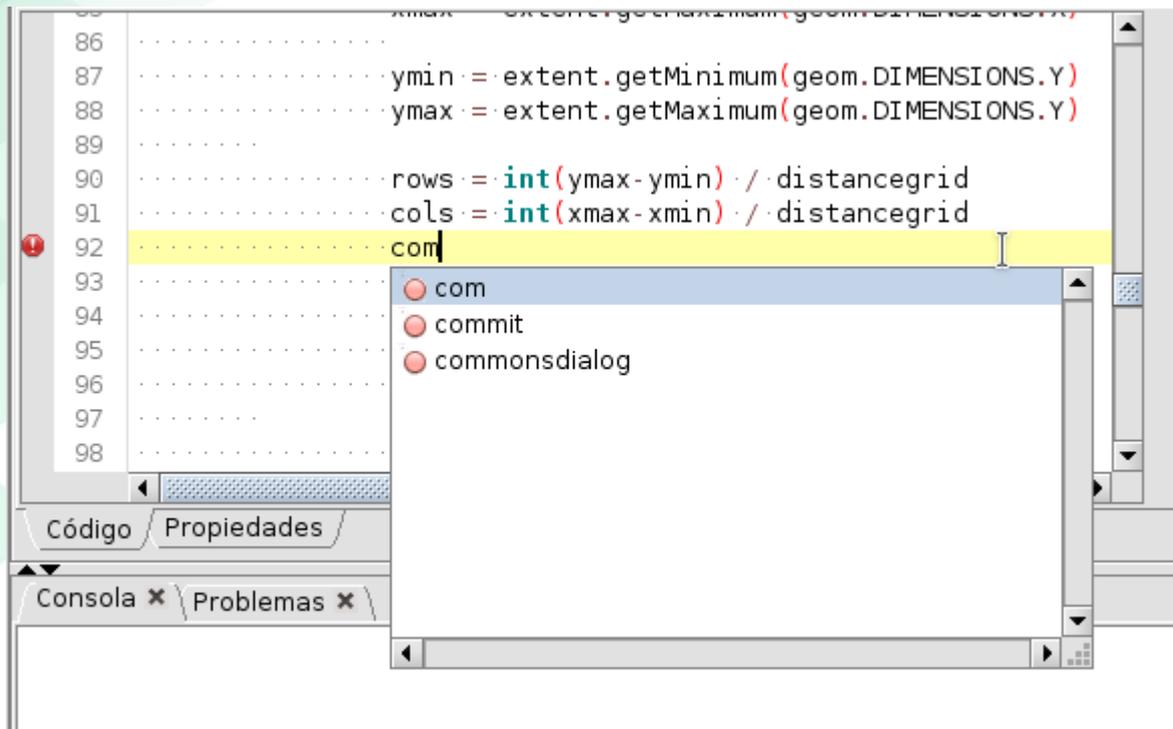
Resultado de sintaxis



# El entorno de scripting

## Funcionalidades

Editor de scripts con...



Autocompletado



# El entorno de scripting

## Funcionalidades

Editor de scripts con...

The screenshot shows the 'Editor de scripts' window with a search dialog box open. The search term is 'fmap'. The dialog includes options for 'Coincidir mayúsculas/minúsculas', 'Palabra completa', 'Expresión', 'Marcar Todo', and 'Dirección' (Arriba/Abajo). The main editor window shows a Python script with several lines of code, including imports from 'org.gvsig.fmap'.

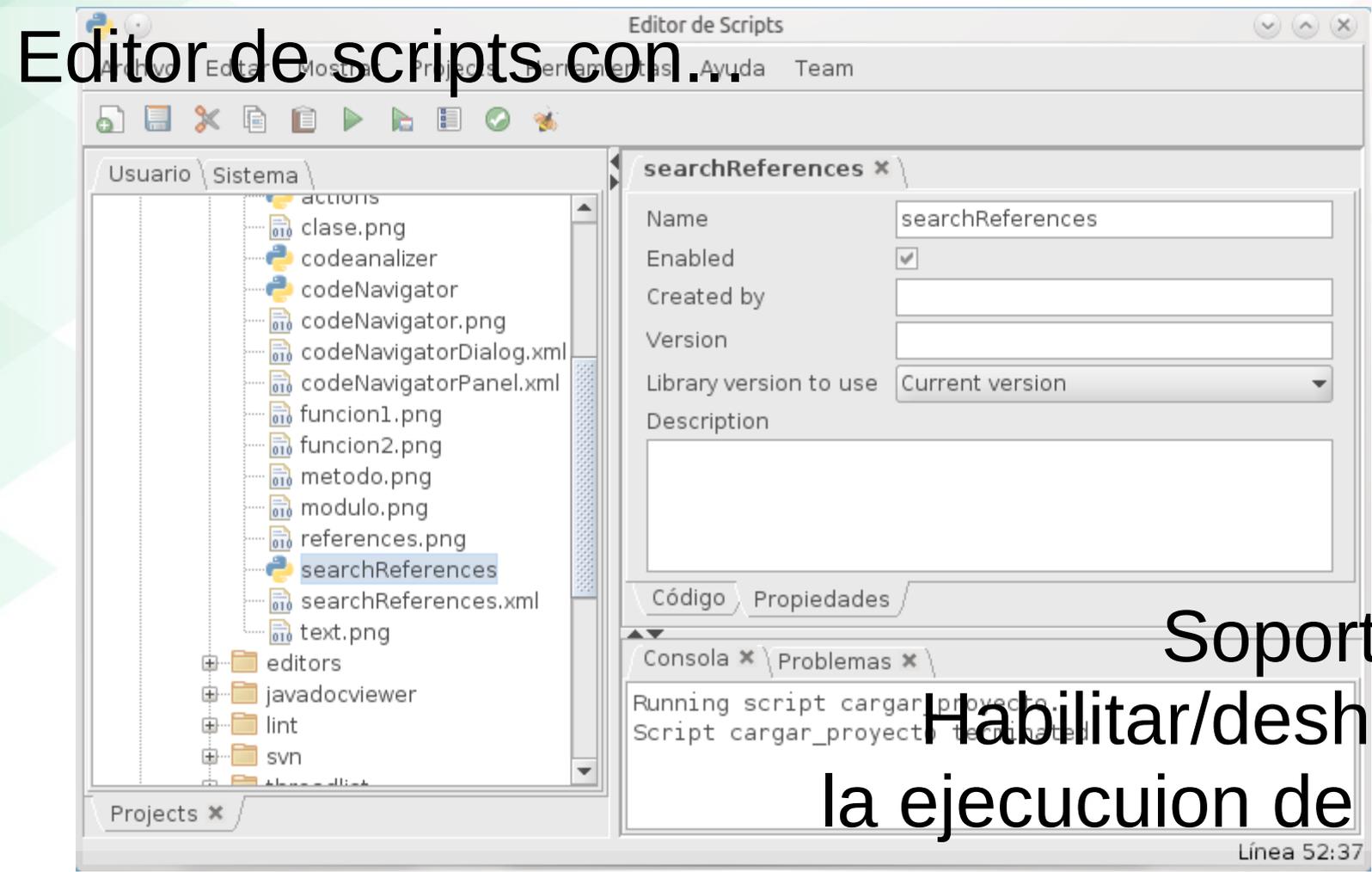
This screenshot shows the same script editor with a search for 'mapcontrol'. The search results are highlighted in yellow in the code editor. The search dialog is also visible, showing 'mapcontrol' as the search term. The console at the bottom shows the message 'Running script coordenadas. Script coordenadas terminated.'

Búsquedas y reemplazos



# El entorno de scripting

## Funcionalidades



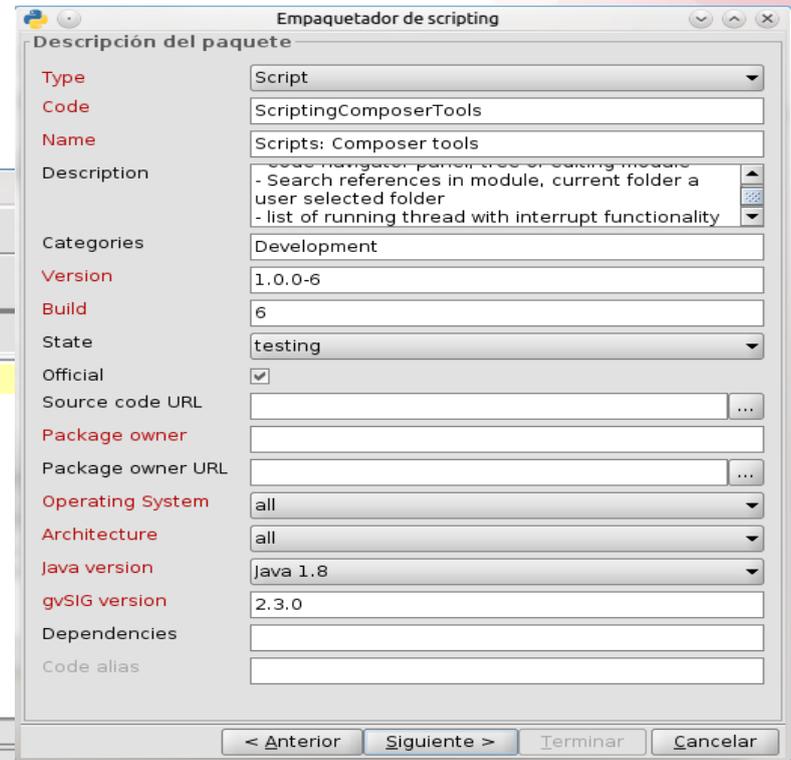
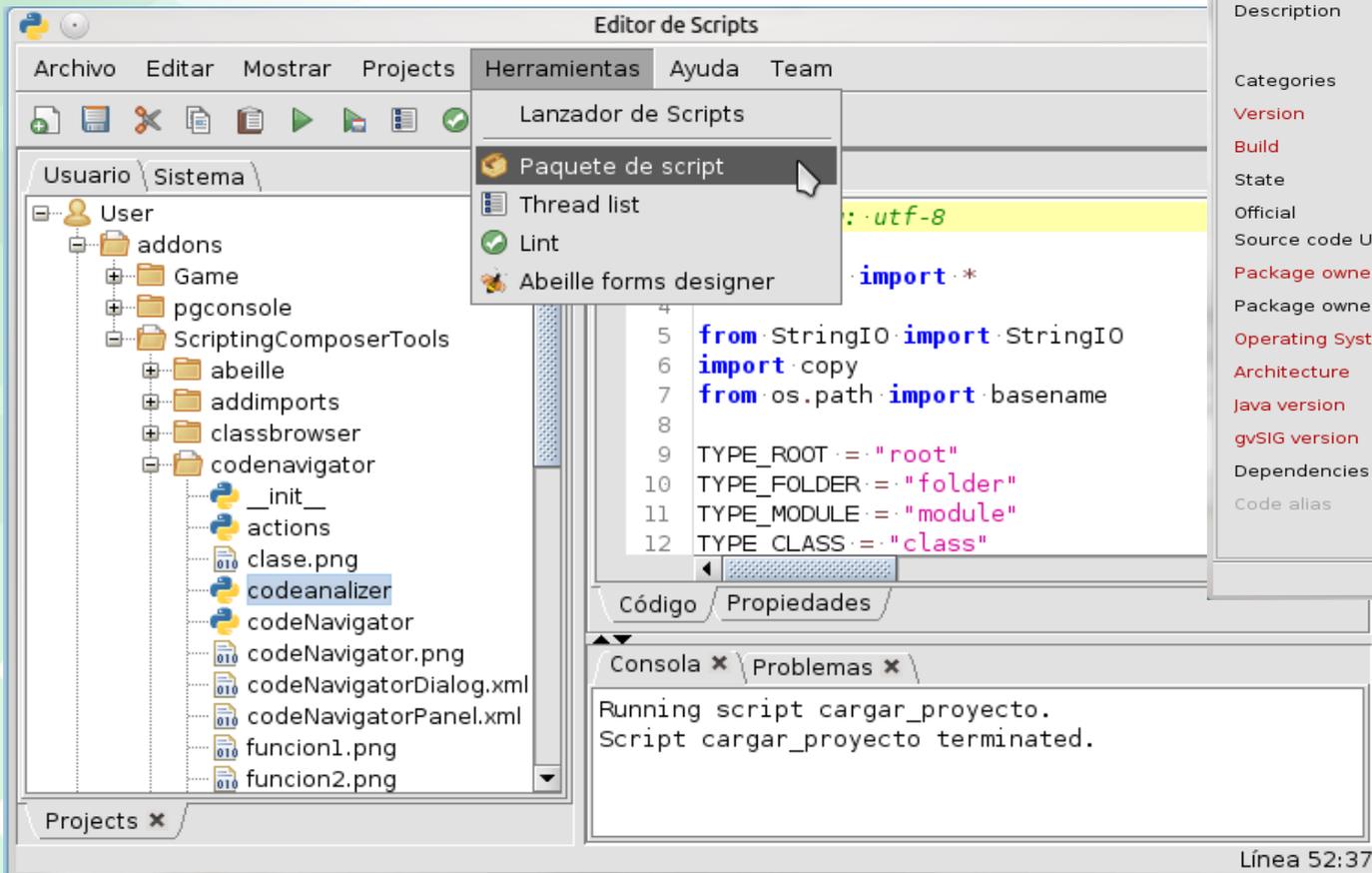
Soporte para  
**Habilitar/deshabilitar**  
**la ejecución de scripts**



# El entorno de scripting

## Funcionalidades

Editor de scripts con...



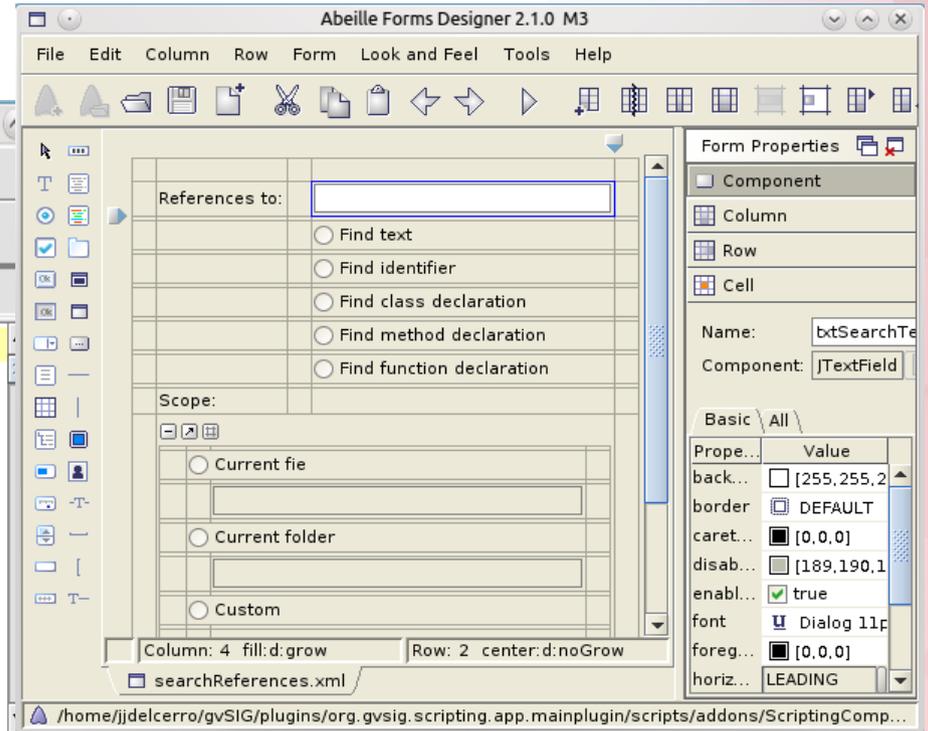
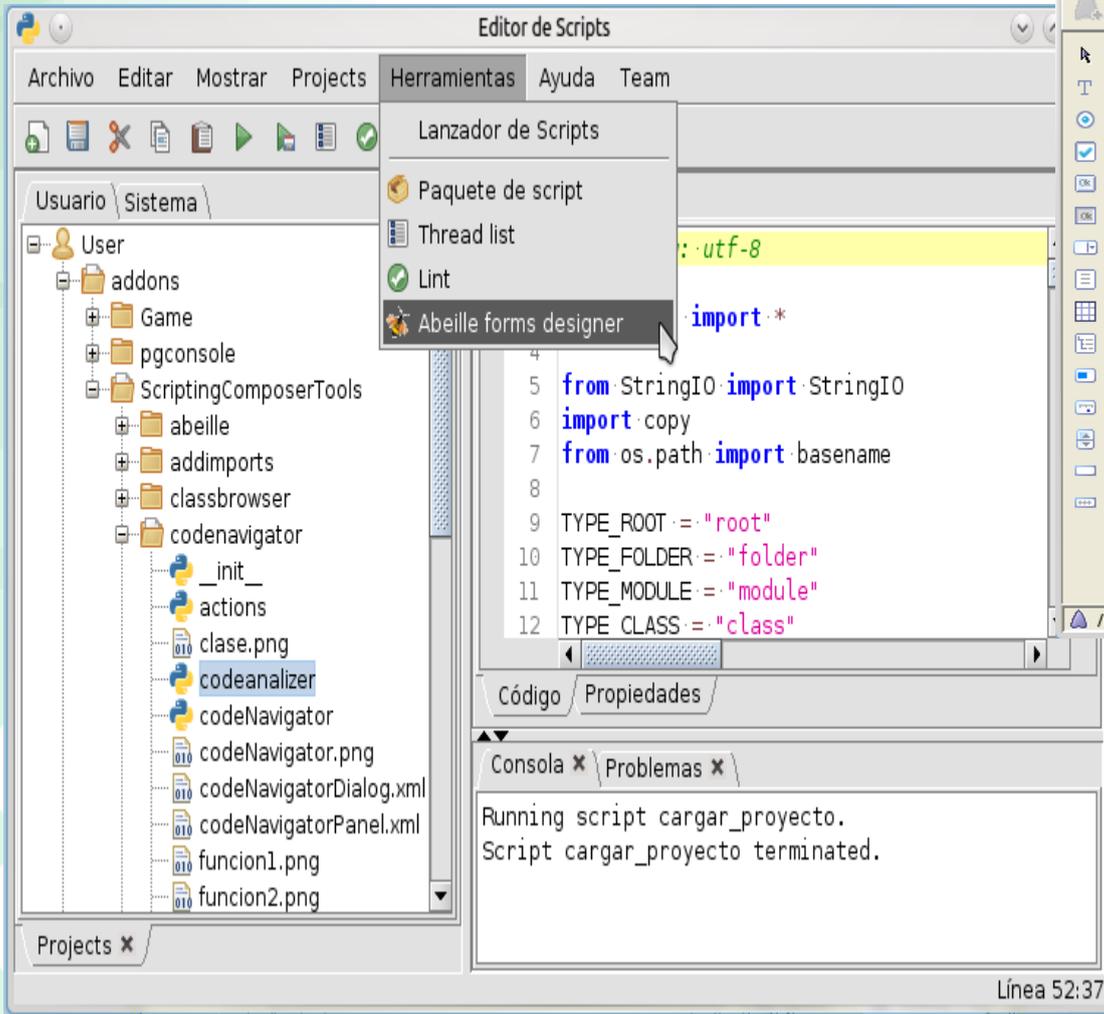
Integración con el administrador de complementos



# El entorno de scripting

## Funcionalidades

Editor de scripts con...



Integración limitada  
 con el editor visual  
 de formularios  
 "abeille"



# El entorno de scripting

## Funcionalidades

Editor de scripts...

Extensible a través de nuestros propios scripts

(ScriptingComposerTools)



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools

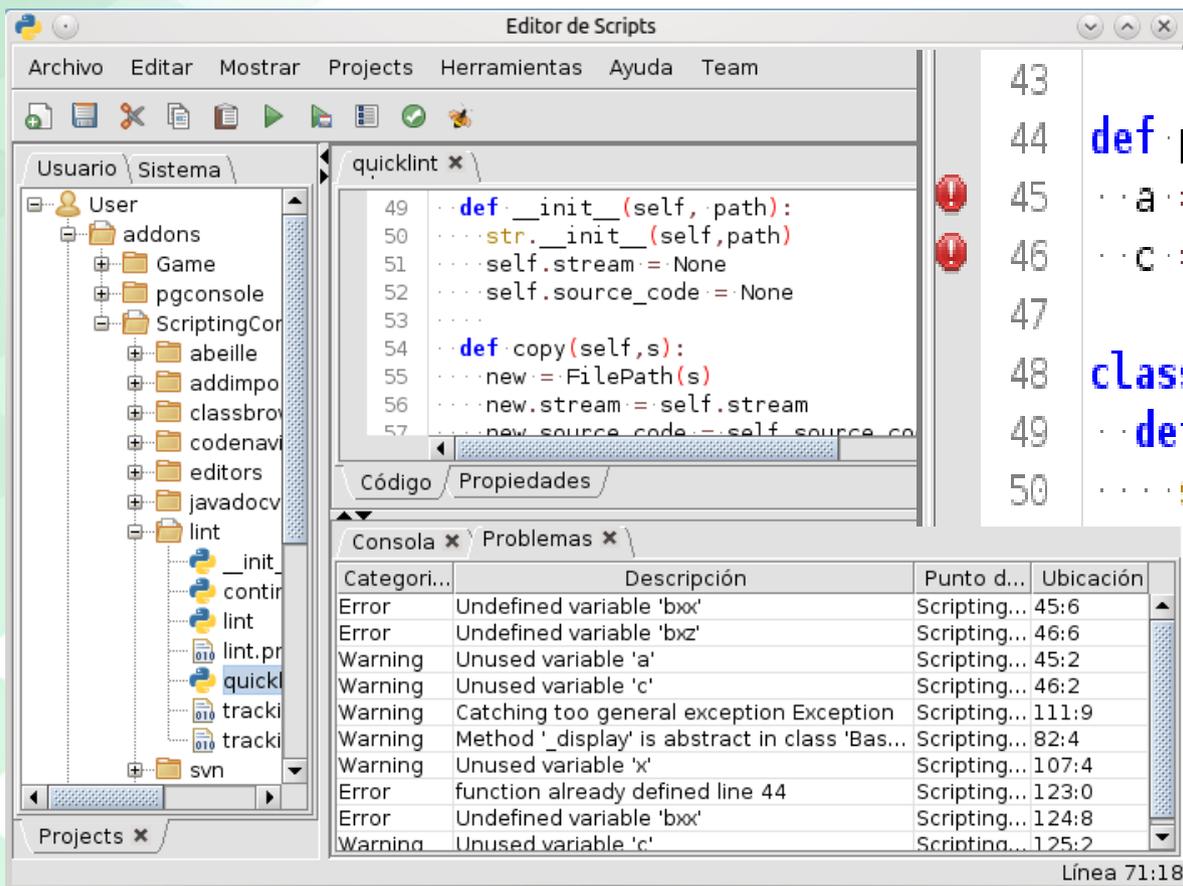
- Actualmente en desarrollo.
- Desarrollado y orientado al desarrollo en python



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools



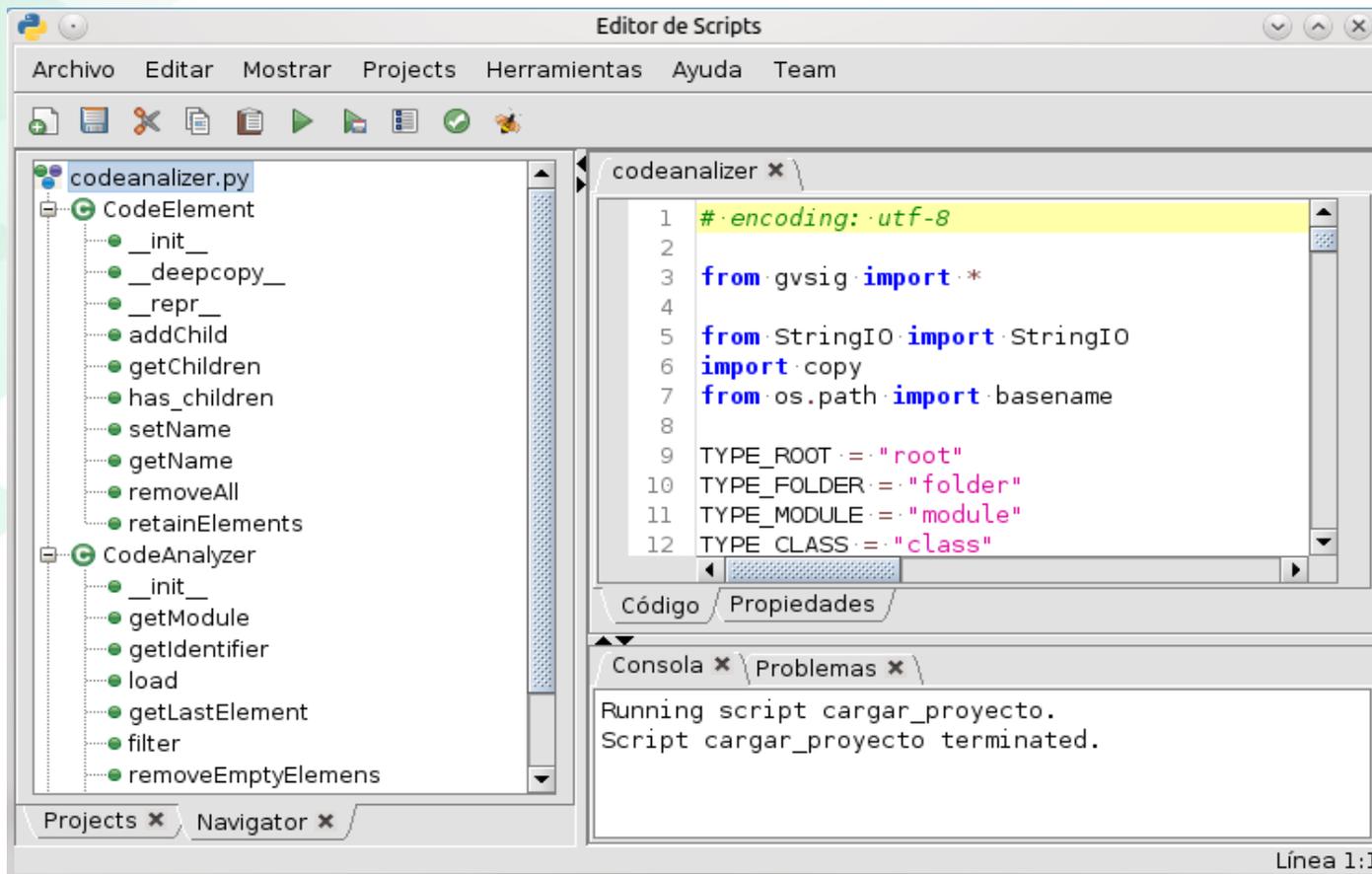
Comprobación de  
sintaxis bajo petición  
y mientras escribimos



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools



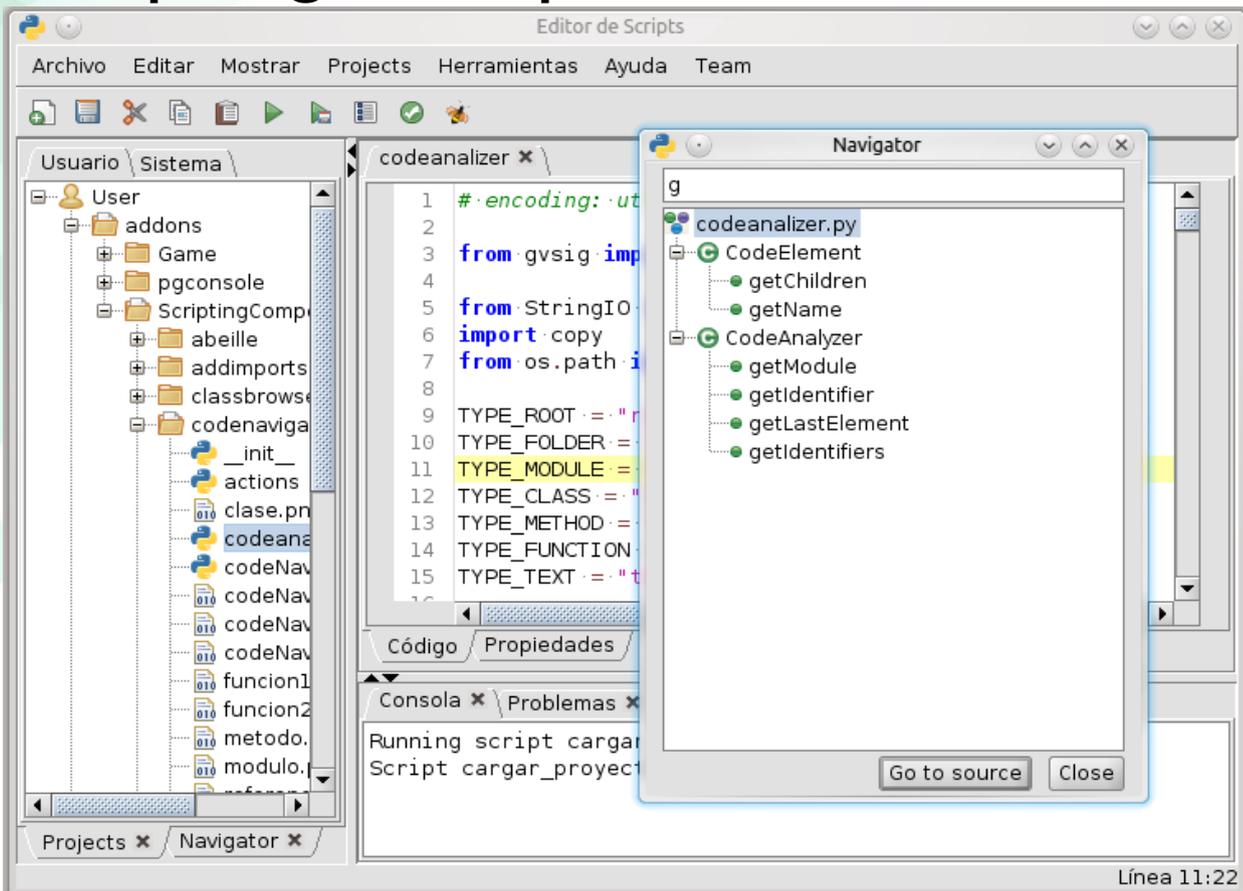
Navegador  
de código



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools



Búsqueda rápida  
de funciones  
clases y métodos  
(en el script corriente)



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools

The screenshot shows the 'Editor de Scripts' interface. The main editor displays a Python class definition for 'CodeAnalyzer' at line 71. A 'Find references' dialog is open, showing the search scope as 'Current folder' and the path 'codenavigator (addons/ScriptingComposerTools/codenavigator)'. Below the editor, a 'References' window shows the search results for 'CodeAnalyzer', listing files like 'codeanalyzer.py' and 'codeNavigator.py' with specific line numbers where the class is used.

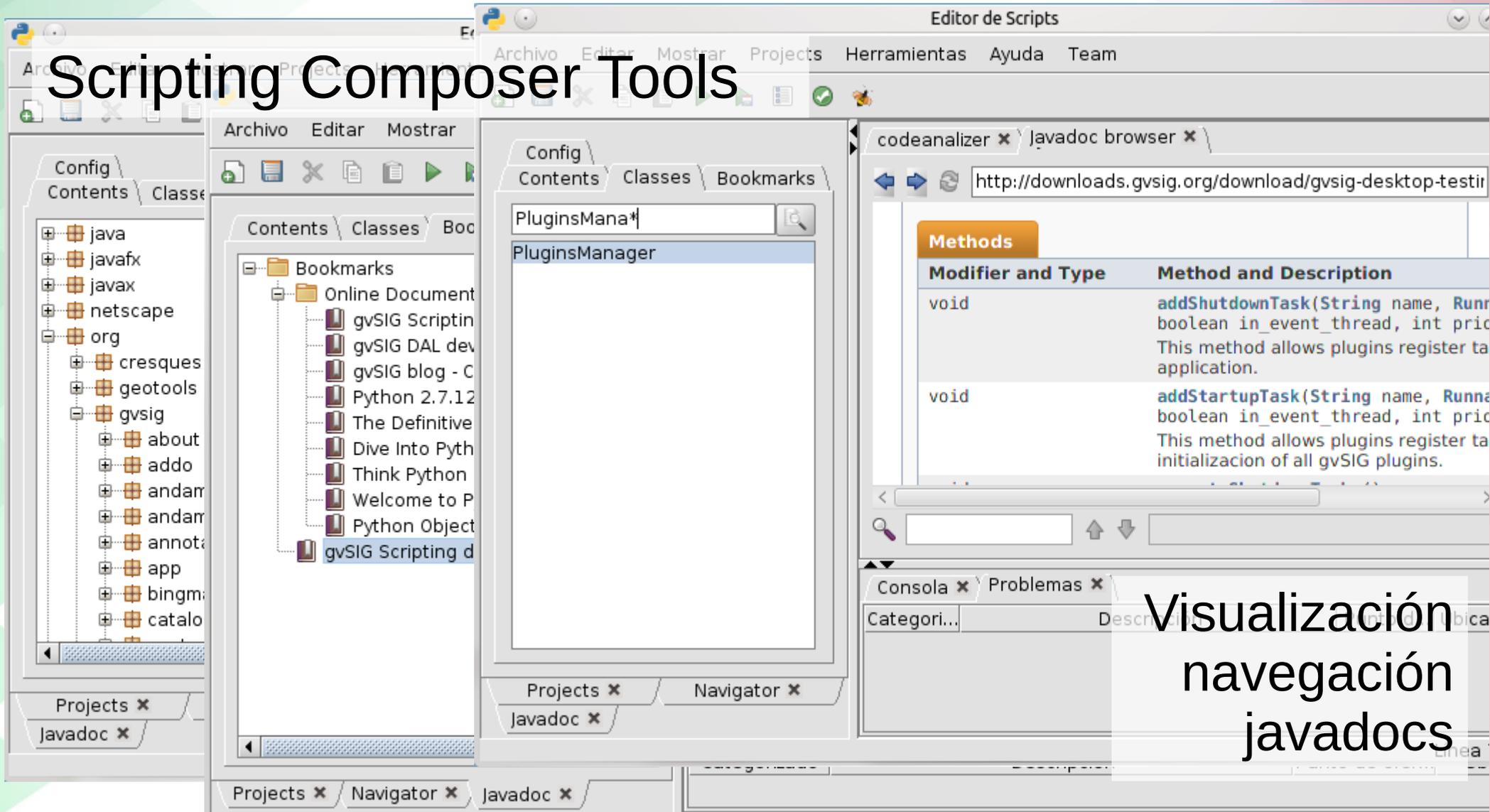
Búsqueda de funciones, clases y métodos en ficheros



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools



Visualización  
navegación  
javadocs



# El entorno de scripting

## Funcionalidades

### Scripting Composer Tools

The screenshot shows the 'Editor de Scripts' application window. The title bar reads 'Editor de Scripts'. The menu bar includes 'Archivo', 'Editar', 'Mostrar', 'Projects', 'Herramientas', 'Ayuda', and 'Team'. The toolbar contains icons for file operations and execution. The main area is split into two panes. The left pane, titled 'List of threads running', displays a list of threads with their IDs: AWT-EventQueue-0 - 17, Timer-0 - 8, WeakCollectionCleaner - 21, Timer-1 - 23, Timer-2 - 27, DestroyJavaVM - 29, MapControl Drawer Worker - 50, Thread-18 - 51, MapControl Drawer Worker - 53, weakref reaper - 85, Jython-Netty-Client-0 - 33, QuantumRenderer-0 - 138, Thread-46 - 140, JavaFX Application Thread - 141, Disposer - 143, Watchdog-Timer-1 - 149, ContinuosLint - 44, and Thread-2 - 117. Below the list are 'Interrupt' and 'Refresh' buttons. The right pane, titled 'codeanalyzer x', shows a code editor with Python code. The code includes a class definition for 'CodeAnalyzer' and a method 'def \_\_init\_\_(self):'. The status bar at the bottom right indicates 'Línea 71:18'.

```
62     ... search := search.lower()
63     ... for child in element.elements:
64     ...     self.retainElements(search, child)
65     ...     if not search in child.name:
66     ...         if len(child.elements) == 0:
67     ...             toremove.append(child)
68     ...     for x in toremove:
69     ...         element.elements.remove(x)
70     ...
71 class CodeAnalyzer(object):
72     ...
73     def __init__(self):
```

Visualización  
de hilos en  
ejecución



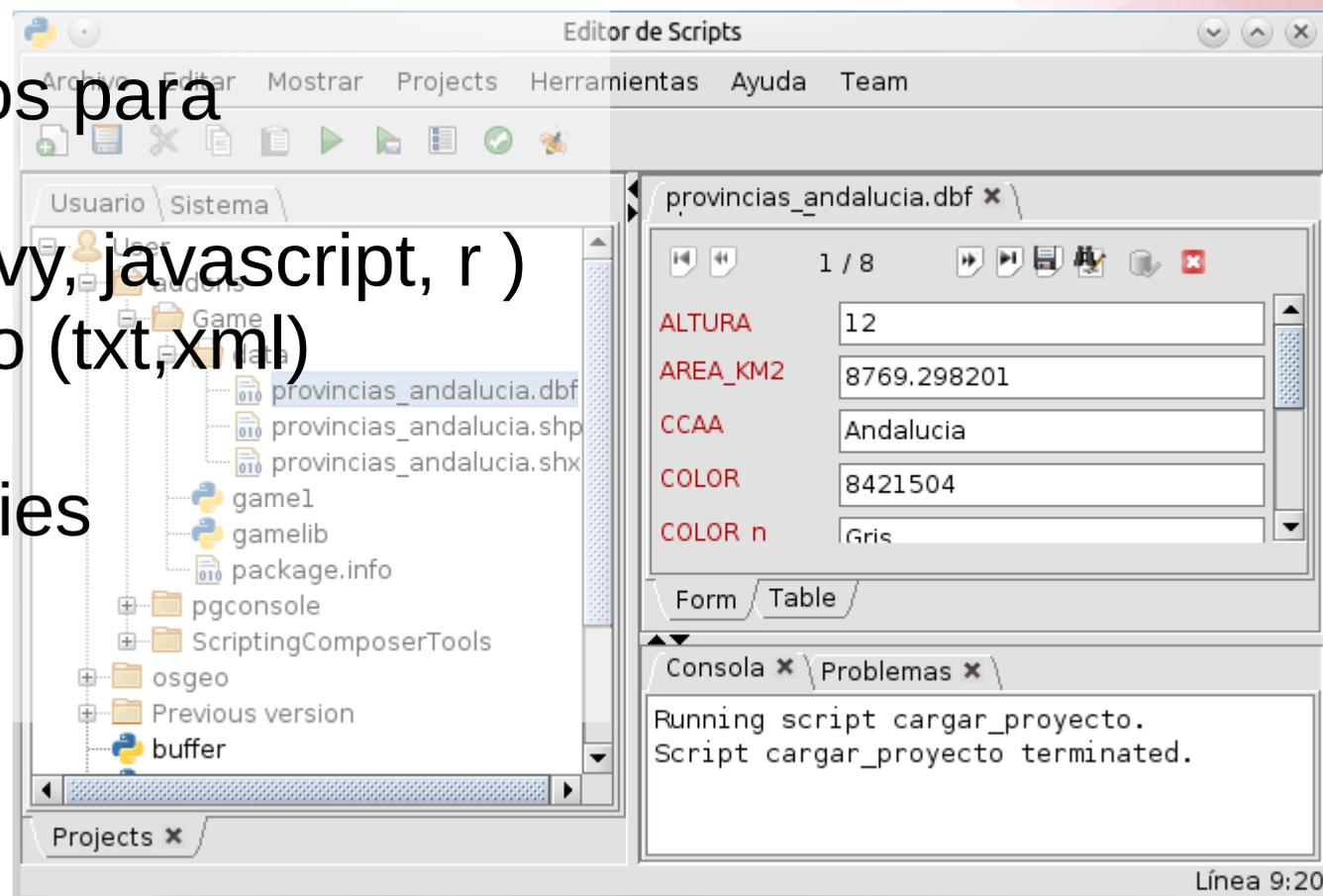
# El entorno de scripting

## Funcionalidades

Editor de scripts con...

Editores integrados para

- Scripts (py, groovy, javascript, r )
- Archivos de texto (txt,xml)
- Archivos DBF
- Archivos properties





# El entorno de scripting

## Funcionalidades

Incluidas librerías listas para ser usadas...

- geopy (geolocalización web)
- gvpy (acceso a la caja de herramientas)
- JOpenDocument (acceso a ODT/ODS)
- CSSBox (navegador web)
- cartodb
- xmltodict
- ogr2ogr

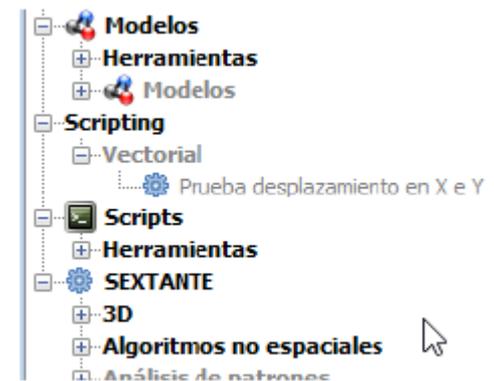


# El entorno de scripting

## Funcionalidades

Soporte para integrar scripts en:

- Menús de gvSIG
- Barra de botones de gvSIG
- Caja de herramientas (Geoprocesos)

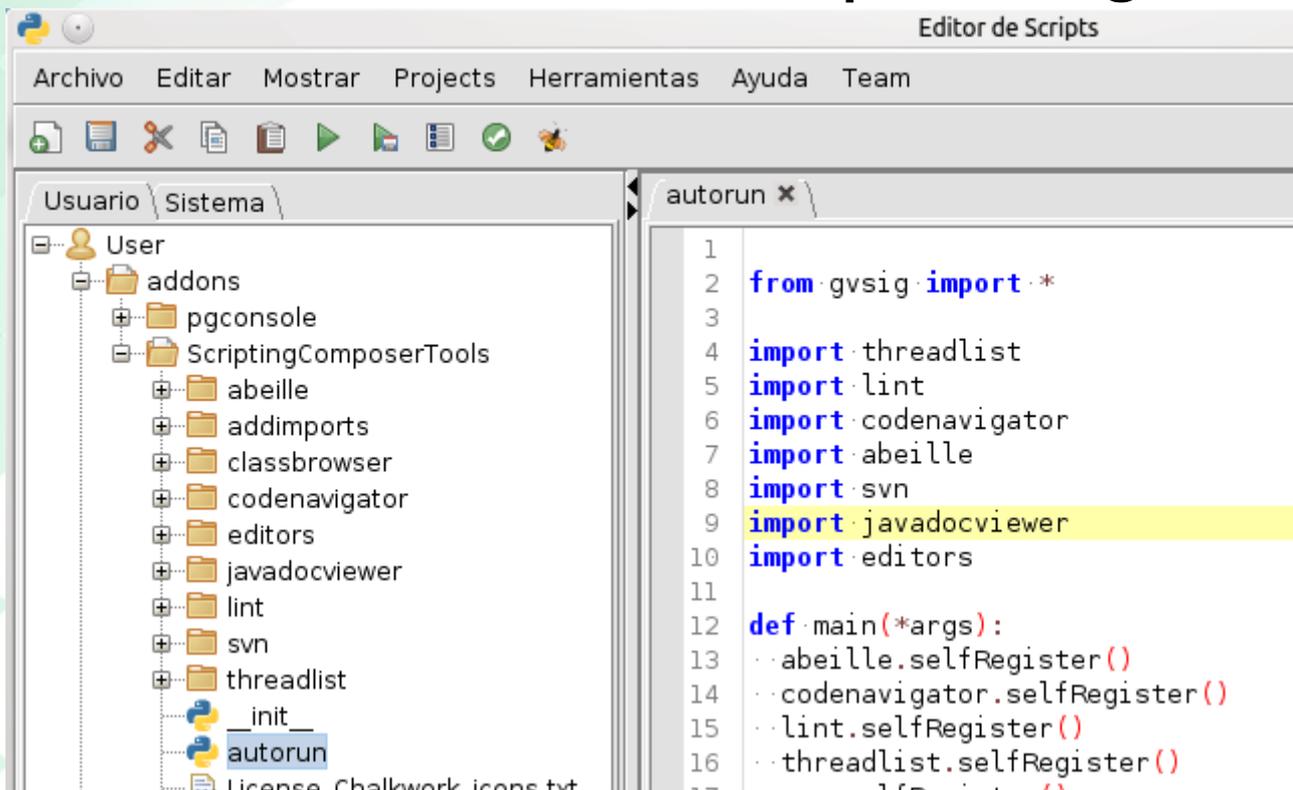




# El entorno de scripting

## Funcionalidades

Soporte para la ejecución de scripts en el arranque de gvSIG.





**ras. Jornadas gvSIG, México**  
**Geomática y software libre en México: Retos y posibilidades**





# Contacto

Contactar y dudas en

Lista de usuarios  
o

Lista de desarrolladores

(<http://www.gvsig.com/es/comunidad/listas-de-correo>)



**ras. Jornadas gvSIG, México**  
**Geomática y software libre en México: Retos y posibilidades**



**gvSIG**  
asociación

[info@gvsig.com](mailto:info@gvsig.com)

[www.gvsig.com](http://www.gvsig.com)

**Joaquin Jose del Cerro**

[jjdelcerro@gvsig.com](mailto:jjdelcerro@gvsig.com)