



Scripting en gvSIG

Joaquín José del Cerro
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

The screenshot shows the gvSIG 2.3.0.2409 interface. The main window displays a world map with a 'country' layer. A 'Coordenda' dialog box is open, showing the coordinates X: -4.04918697918 and Y: 39.2967392648. A code snippet for a Python script is overlaid on the map, showing the following code:

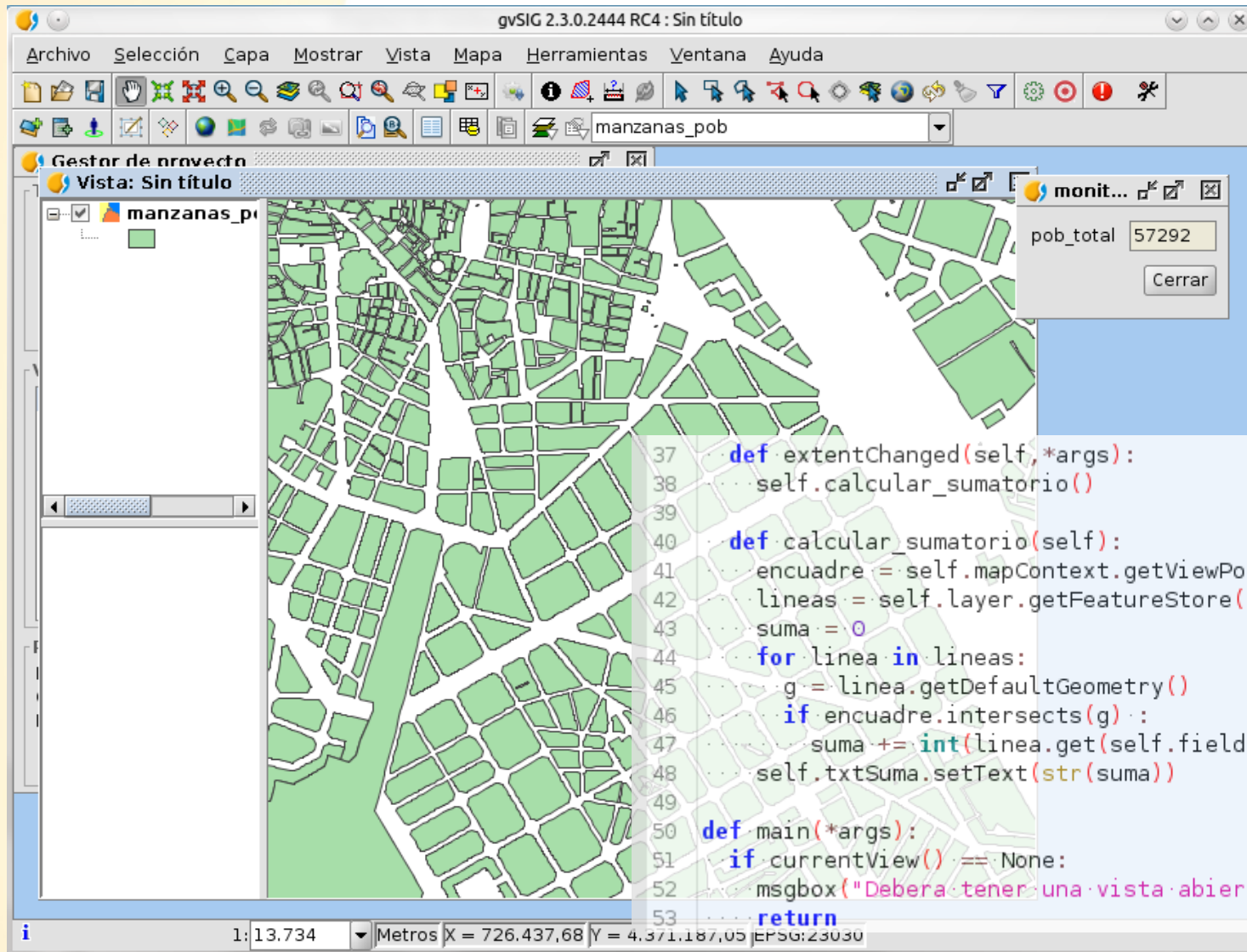
```

42
43 def pointDoubleClick(self, event):
44     """Evento de PointListener"""
45     p = event.getMapPoint()
46     self.txtX.setText(str(p.getX()))
47     self.txtY.setText(str(p.getY()))
48
49 def main(*args):
50     coordenadas = Coordenadas(currentView())
51     coordenadas.showTool("Coordenda")
52

```

The status bar at the bottom shows: 1: 1.857.045 Metros Lon = -0° 51' 47" Lat = 41° 54' 12" EPSG:4326

Monitorizar datos



The screenshot shows the gvSIG 2.3.0.2444 RC4 interface. The main window displays a map of a city grid. A script editor window is open, showing the following Python code:

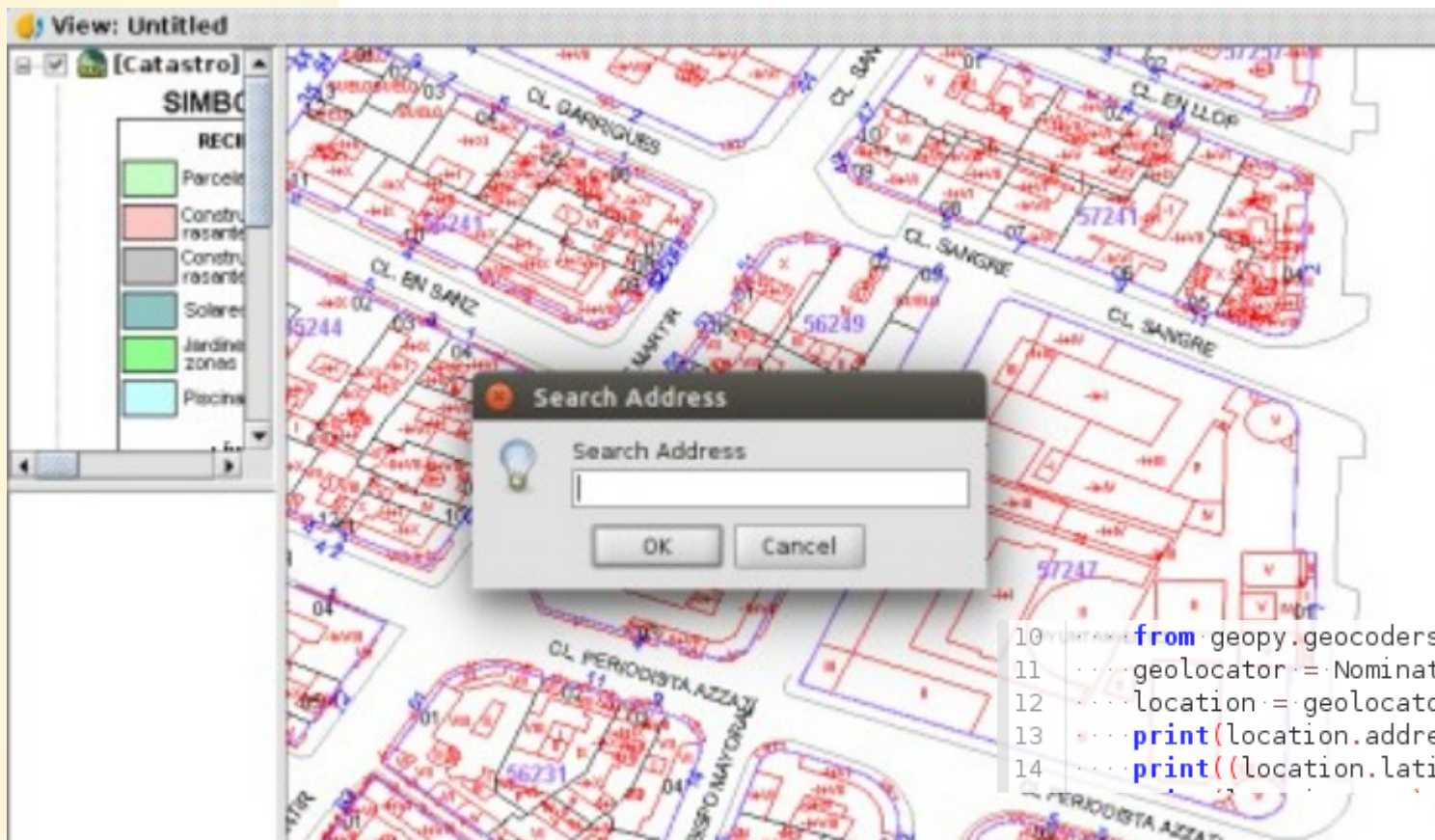
```

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
    
```

A small dialog box titled 'monit...' is open, showing 'pob_total' with the value 57292 and a 'Cerrar' button.

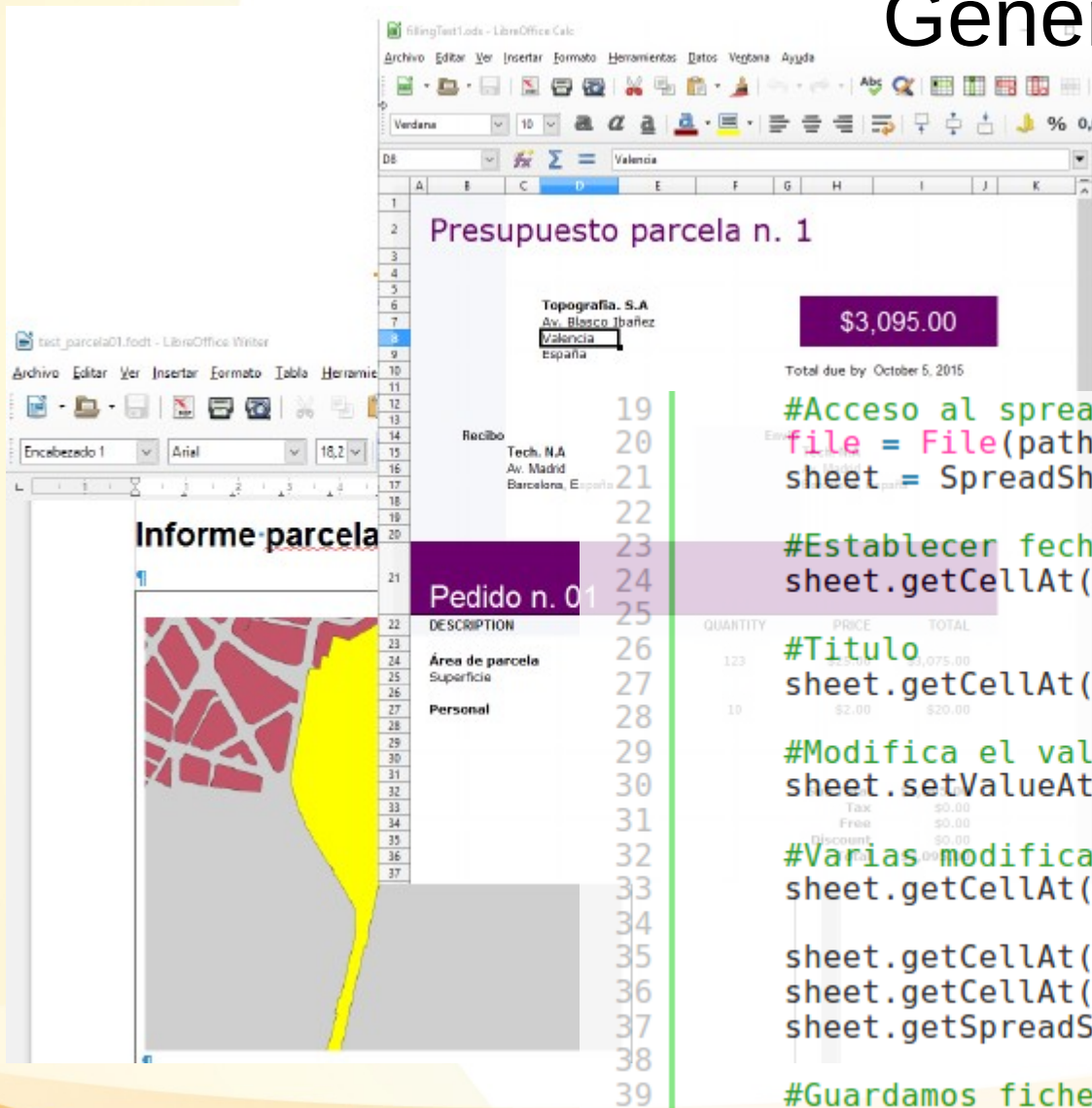
Que podemos hacer con scripting

Usar servicios web
especializados
geopy



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)

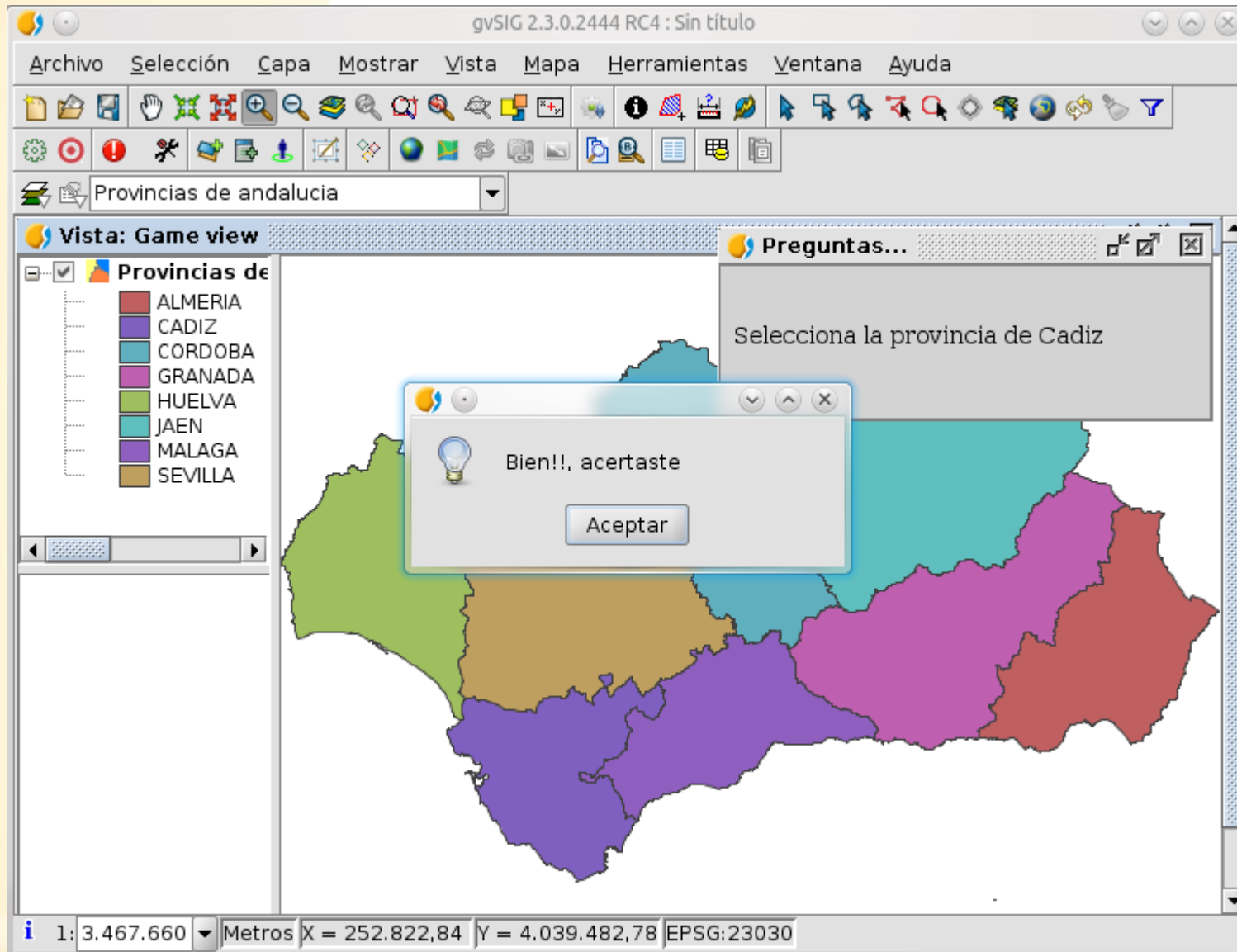
sheet.getCellAt("B27").setValue("Personal")
sheet.getCellAt("H27").setValue(2)
sheet.getSpreadSheet().getTableModel("Products").setVa

#Guardamos fichero

```

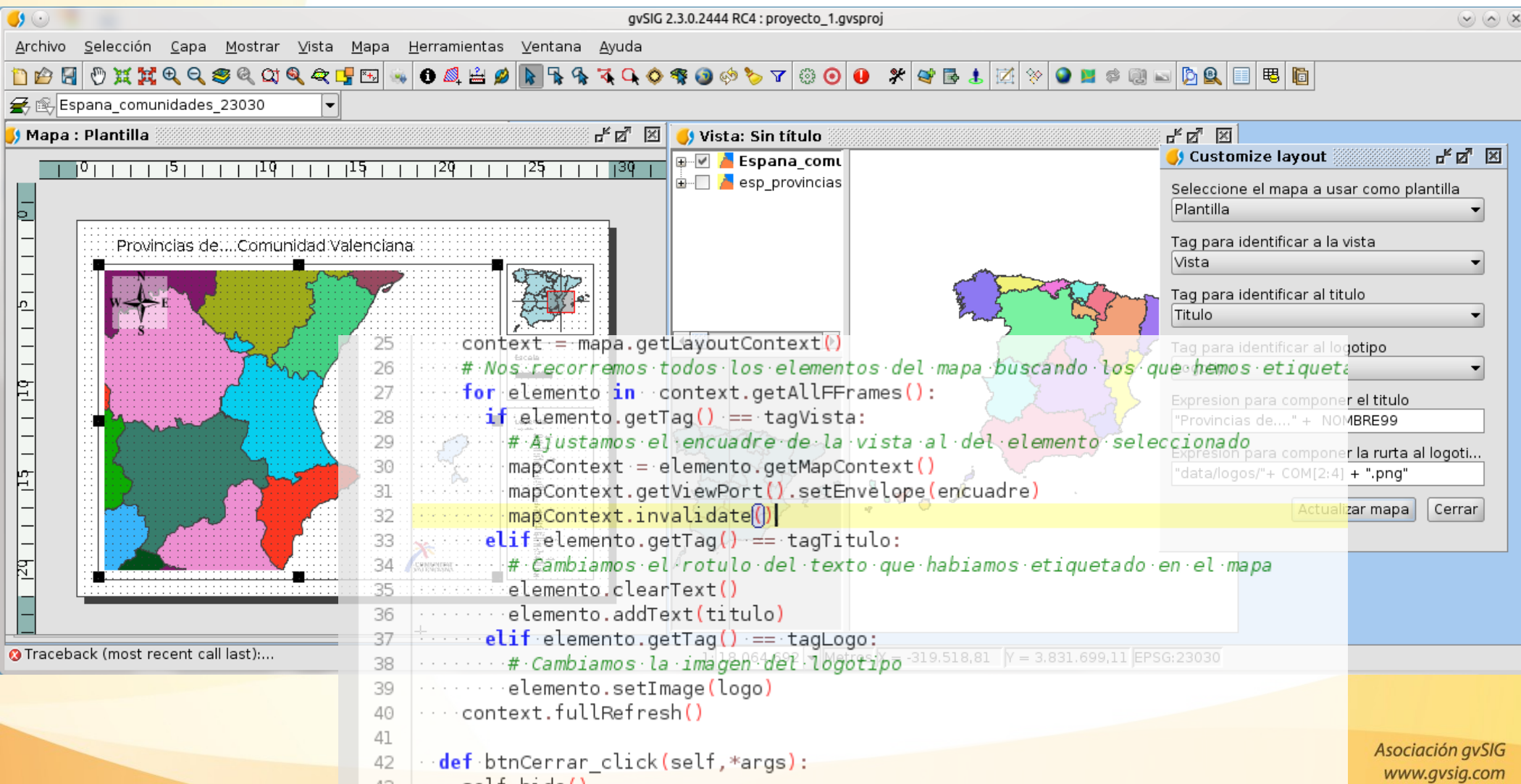
Que podemos hacer con scripting

Juegos
didácticos



Que podemos hacer con scripting

Personalizar mapas



The screenshot shows the gvSIG 2.3.0.2444 RC4 interface. The main window displays a map of Spain with a script editor overlaid. The script editor contains the following code:

```

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() == tagLogotipo:
38         # Cambiamos la imagen del logotipo
39         elemento.setImage(logotipo)
40     context.fullRefresh()
41
42 def btnCerrar_click(self, *args):
43     self.hide()

```

The 'Customize layout' dialog box is open, showing 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: (empty)
- 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

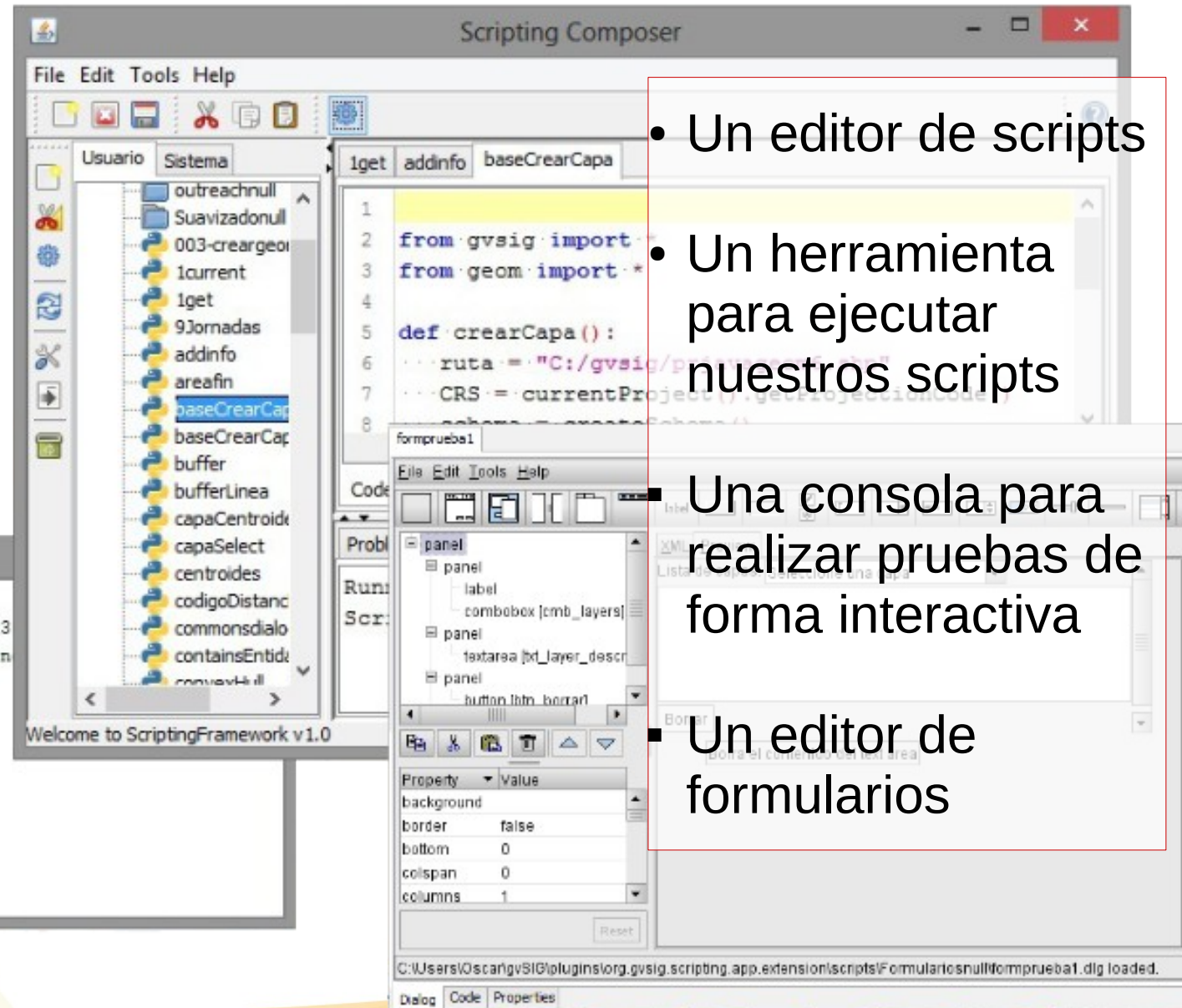
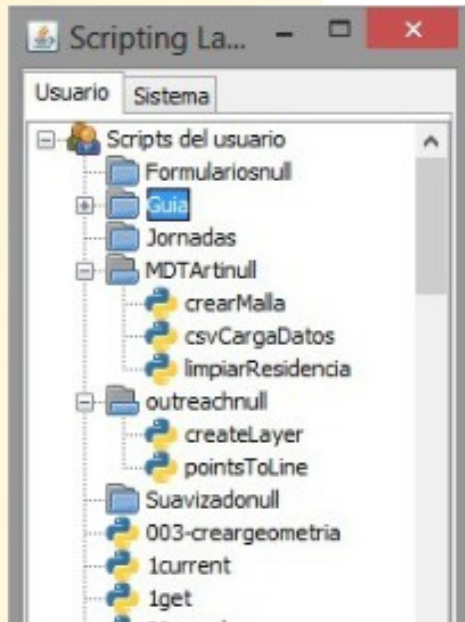
El entorno de scripting

Y...

Para hacer estas cosas...

¿ de que entorno disponemos ?

El entorno de scripting

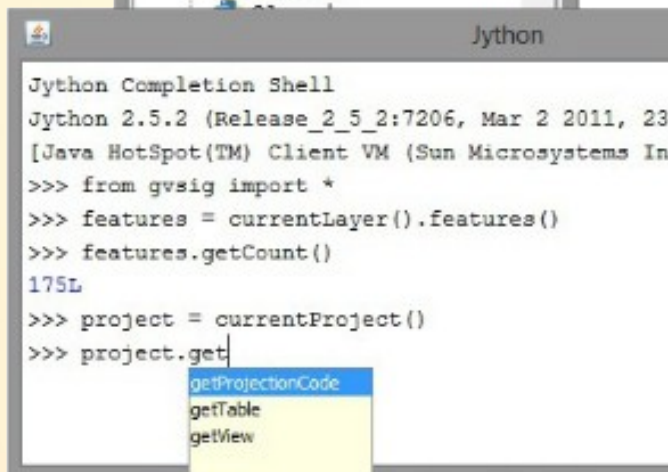


- Un editor de scripts

- 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...

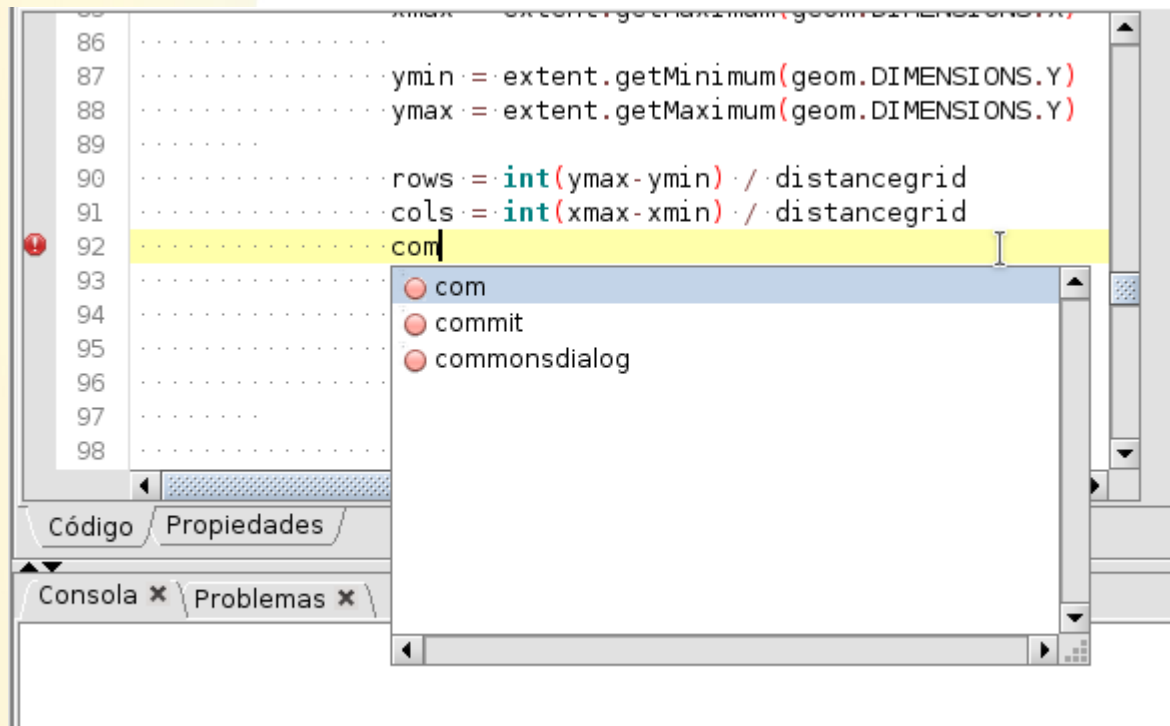
```
15 class MonitorSuma(ViewPortListener,FormPanel):
16
17     def __init__(self, mapContext, layer, label, fieldname):
18         FormPanel.__init__(self,script.getResource("monitor_suma3.xml"))
19         ....
20         self.mapContext = mapContext
21         self.layer = layer
22         self.fieldname = 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...

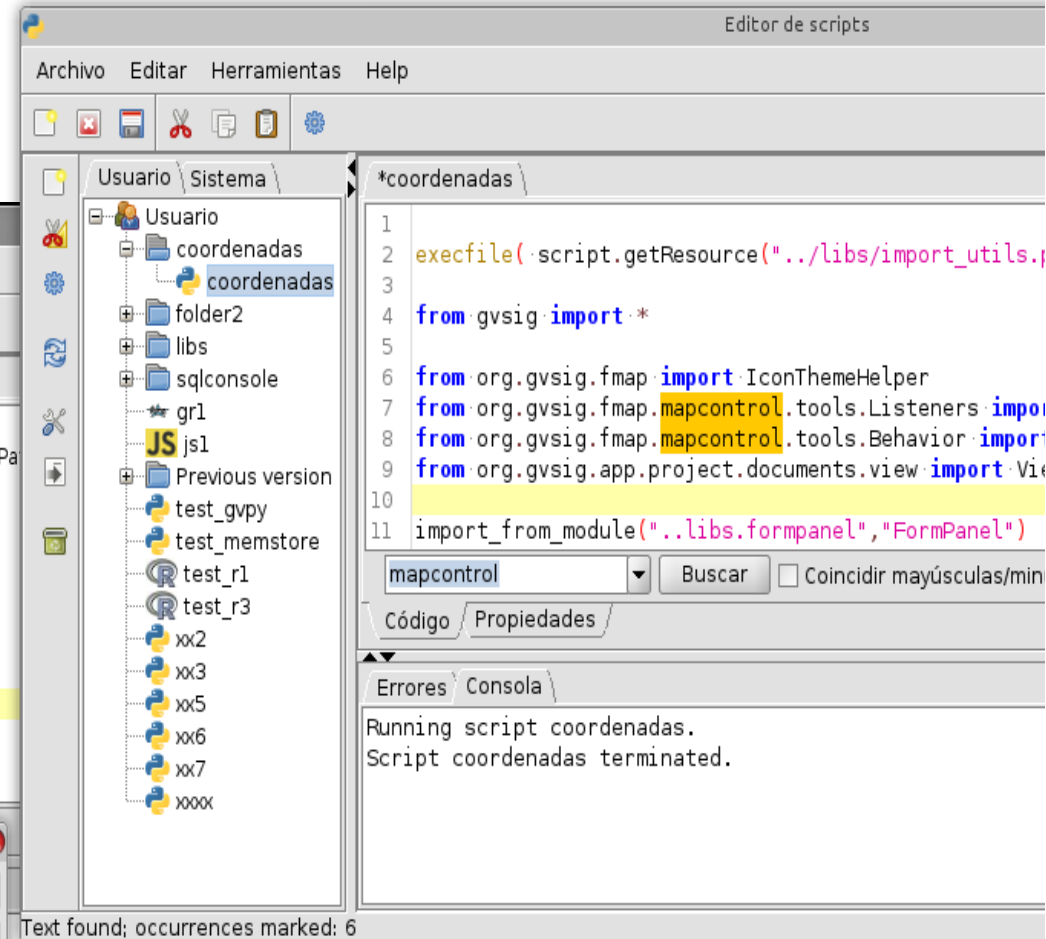
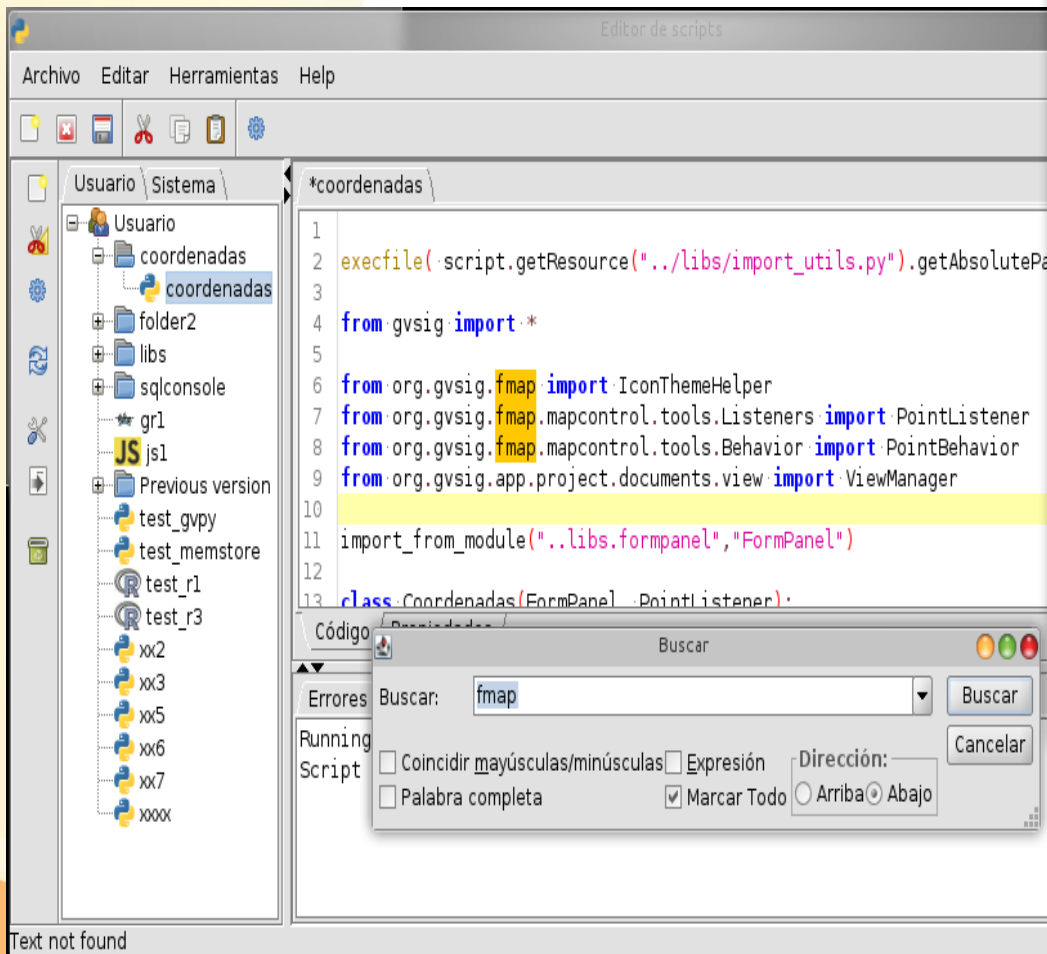


Autocompletado

El entorno de scripting

Funcionalidades

Editor de scripts con...

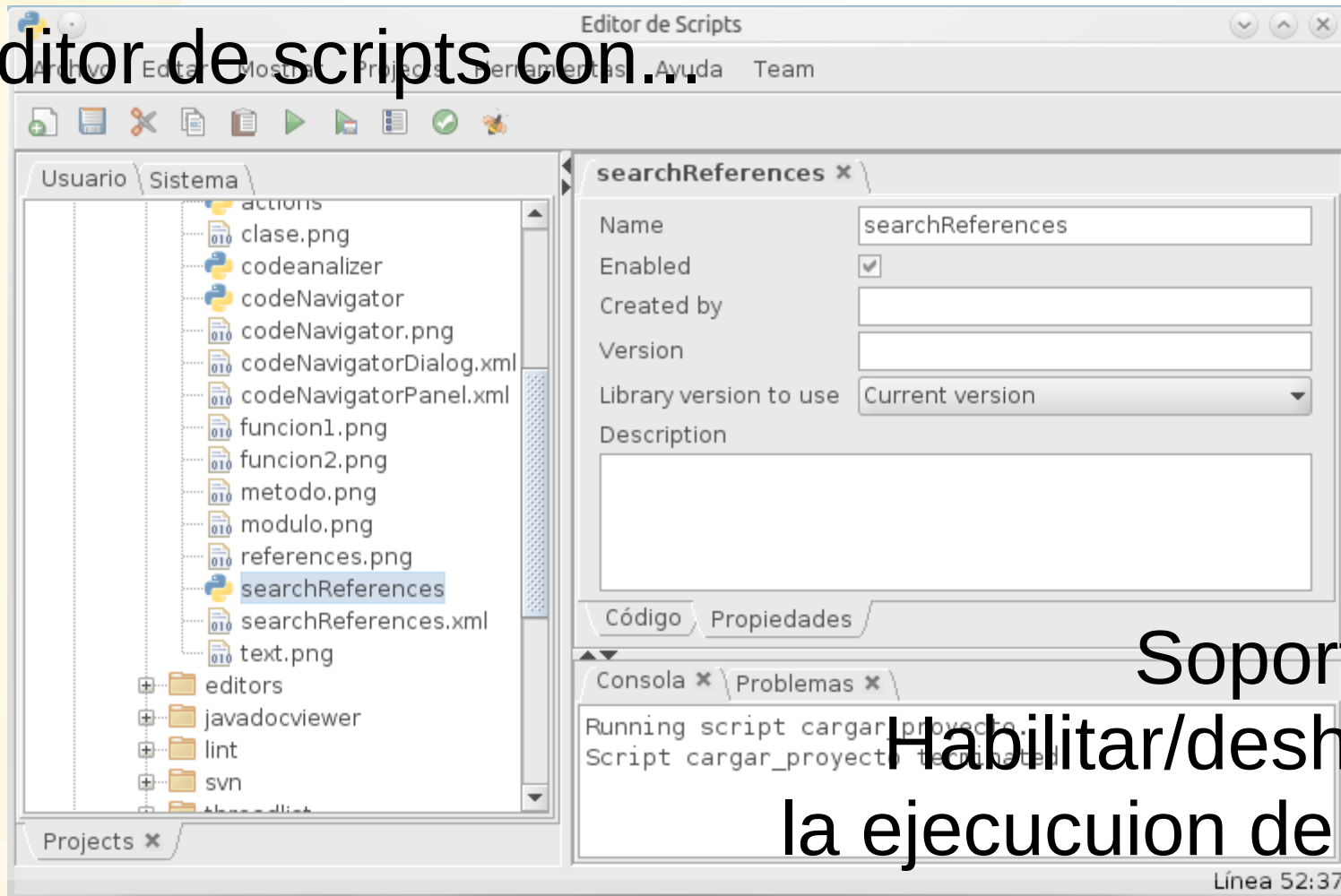


Búsquedas y reemplazos

El entorno de scripting

Funcionalidades

Editor de scripts con...

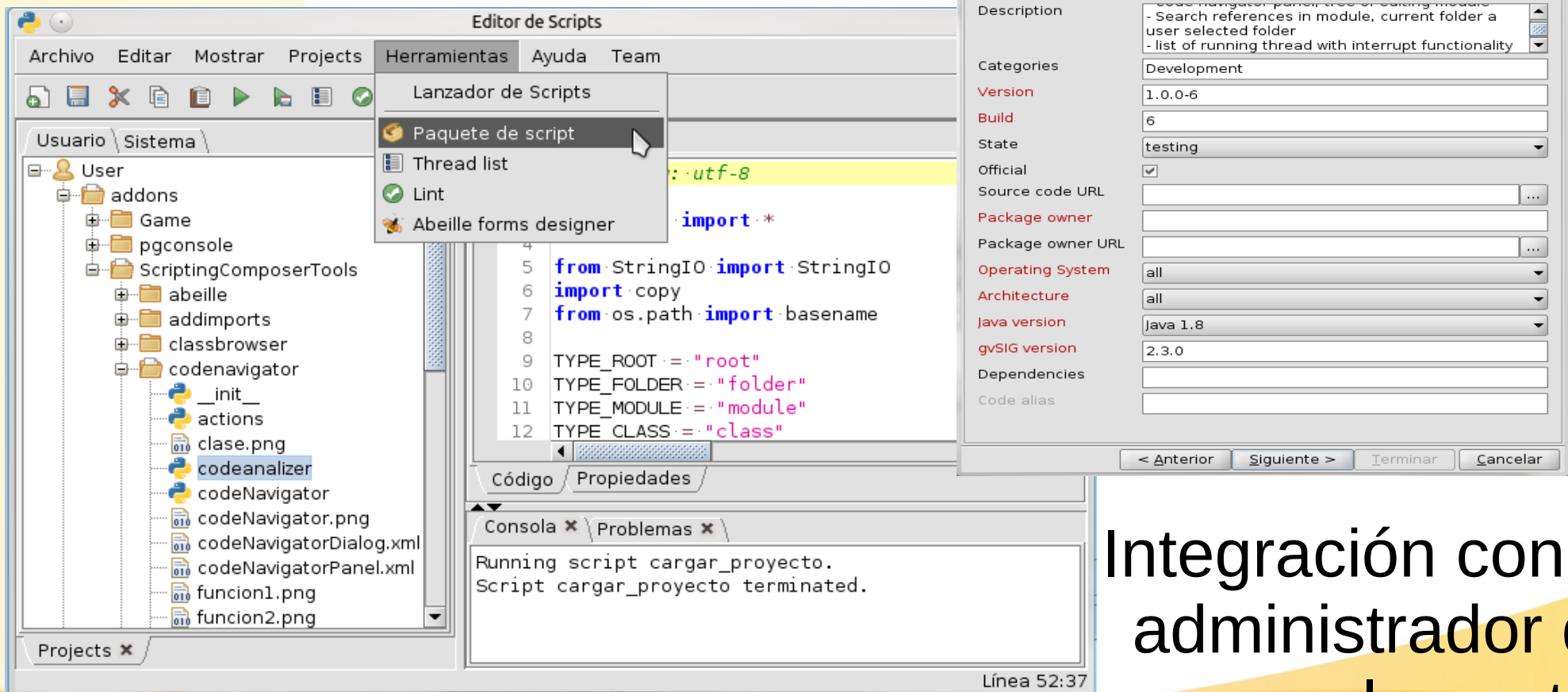


Soporte para
Habilitar/deshabilitar
la ejecución de scripts

El entorno de scripting

Funcionalidades

Editor de scripts con...



Editor de Scripts

Archivo Editar Mostrar Projects Herramientas Ayuda Team

Lanzador de Scripts
Paquete de script
Thread list
Lint
Abeille forms designer

Usuario \ Sistema \

User

- addons
 - Game
 - pgconsole
 - ScriptingComposerTools
 - abeille
 - addimports
 - classbrowser
 - codenavigator
 - _init_
 - actions
 - clase.png
 - codeanalyzer
 - codeNavigator
 - codeNavigator.png
 - codeNavigatorDialog.xml
 - codeNavigatorPanel.xml
 - funcion1.png
 - funcion2.png

```

4
5 from StringIO import StringIO
6 import copy
7 from os.path import basename
8
9 TYPE_ROOT := "root"
10 TYPE_FOLDER := "folder"
11 TYPE_MODULE := "module"
12 TYPE_CLASS := "class"
    
```

Código Propiedades

Consola x Problemas x

Running script cargar_proyecto.
Script cargar_proyecto terminated.

Línea 52:37

Empaquetador de scripting

Descripción del paquete

Type Script

Code ScriptingComposerTools

Name Scripts: Composer tools

Description

- Search references in module, current folder a user selected folder
- list of running thread with interrupt functionality

Categories Development

Version 1.0.0-6

Build 6

State testing

Official

Source code URL

Package owner

Package owner URL

Operating System all

Architecture all

Java version Java 1.8

gvSIG version 2.3.0

Dependencies

Code alias

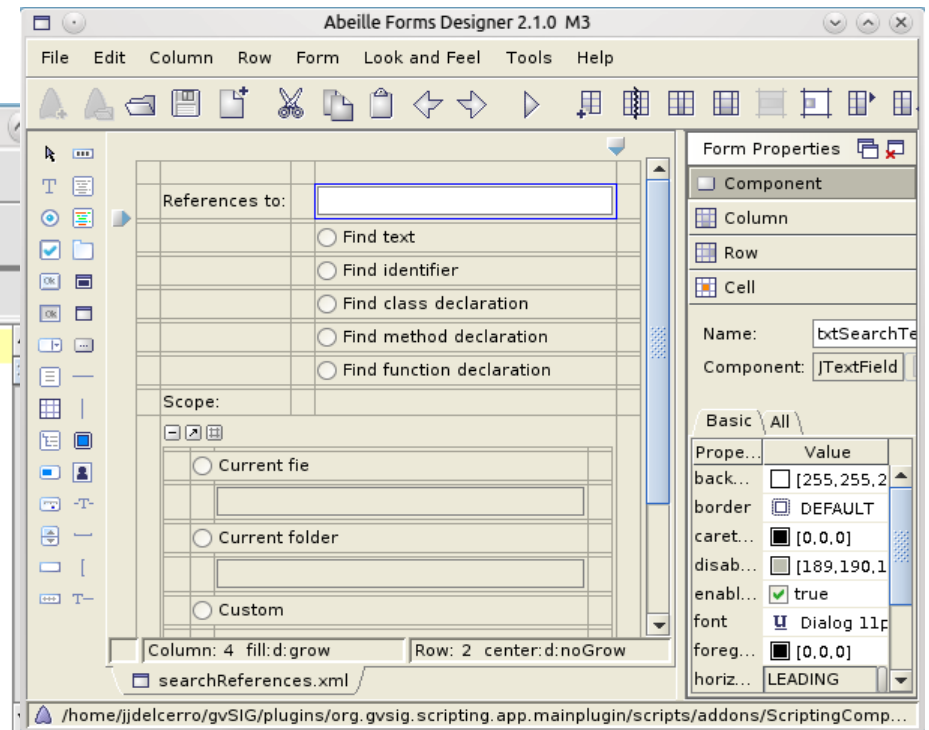
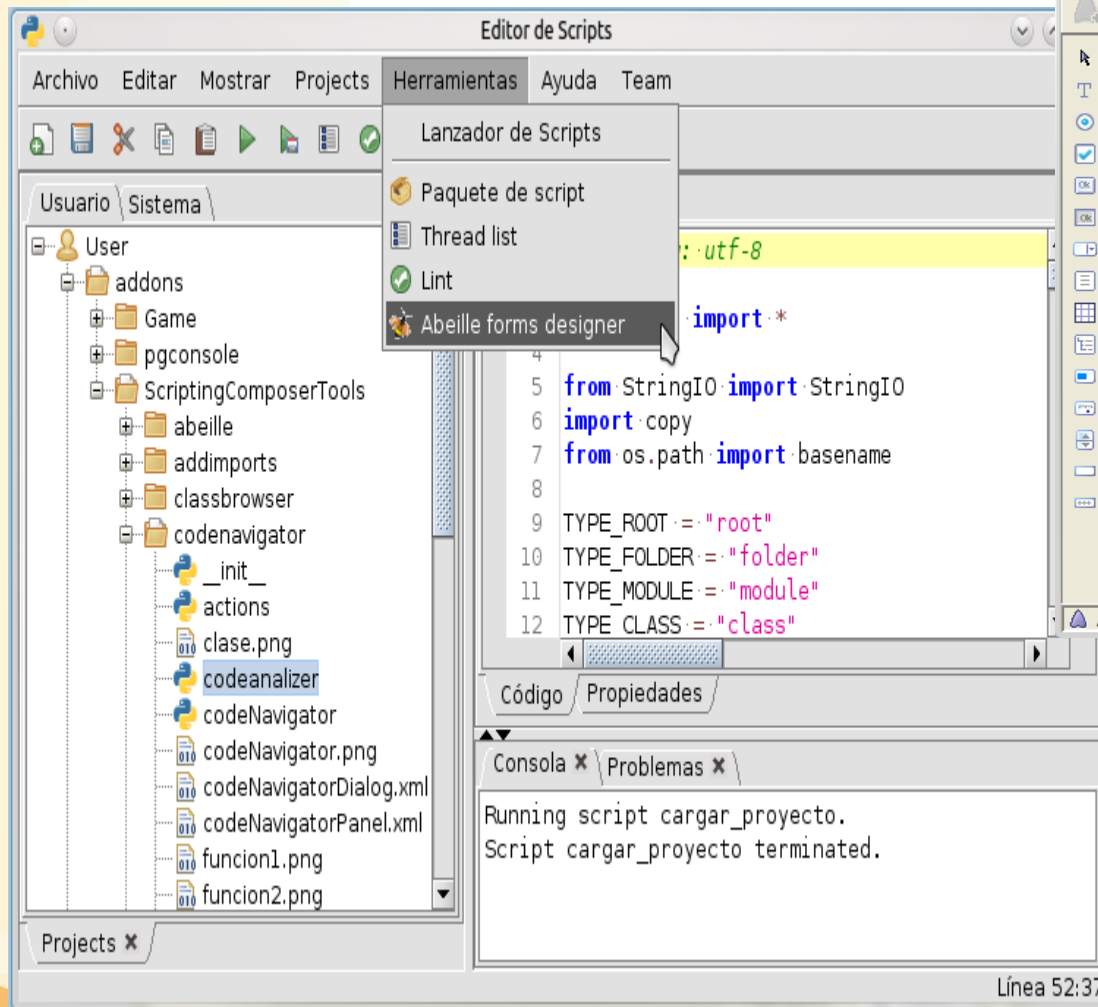
< Anterior Siguiente > Terminar Cancelar

Integración con el administrador de complementos

El entorno de scripting

Funcionalidades

Editor de scripts con...



Integración limitada
con el editor visual
de formularios

El entorno de scripting

Funcionalidades

Editor de scripts...

Extensible a través de nuestros propios scripts

(ScriptingComposerTools)

El entorno de scripting

Funcionalidades

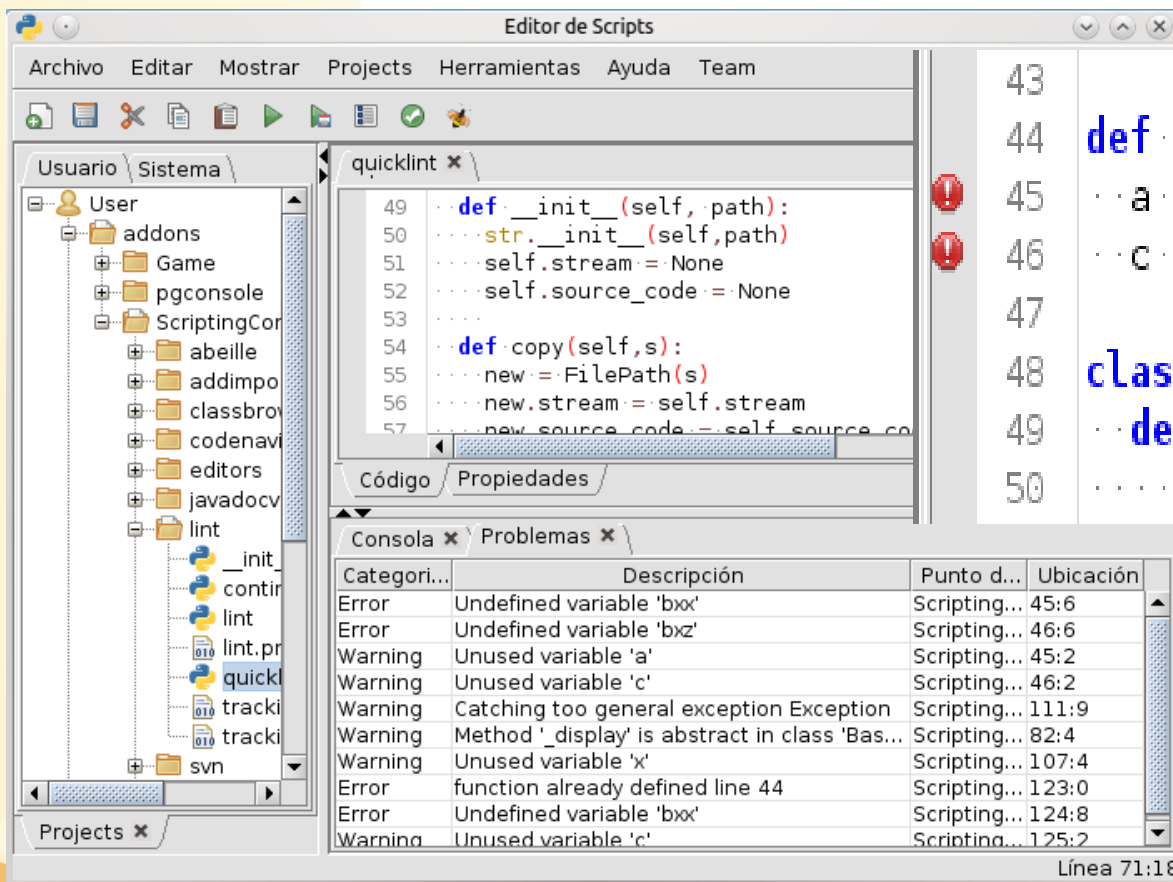
Scripting Composer Tools

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

El entorno de scripting

Funcionalidades

Scripting Composer Tools

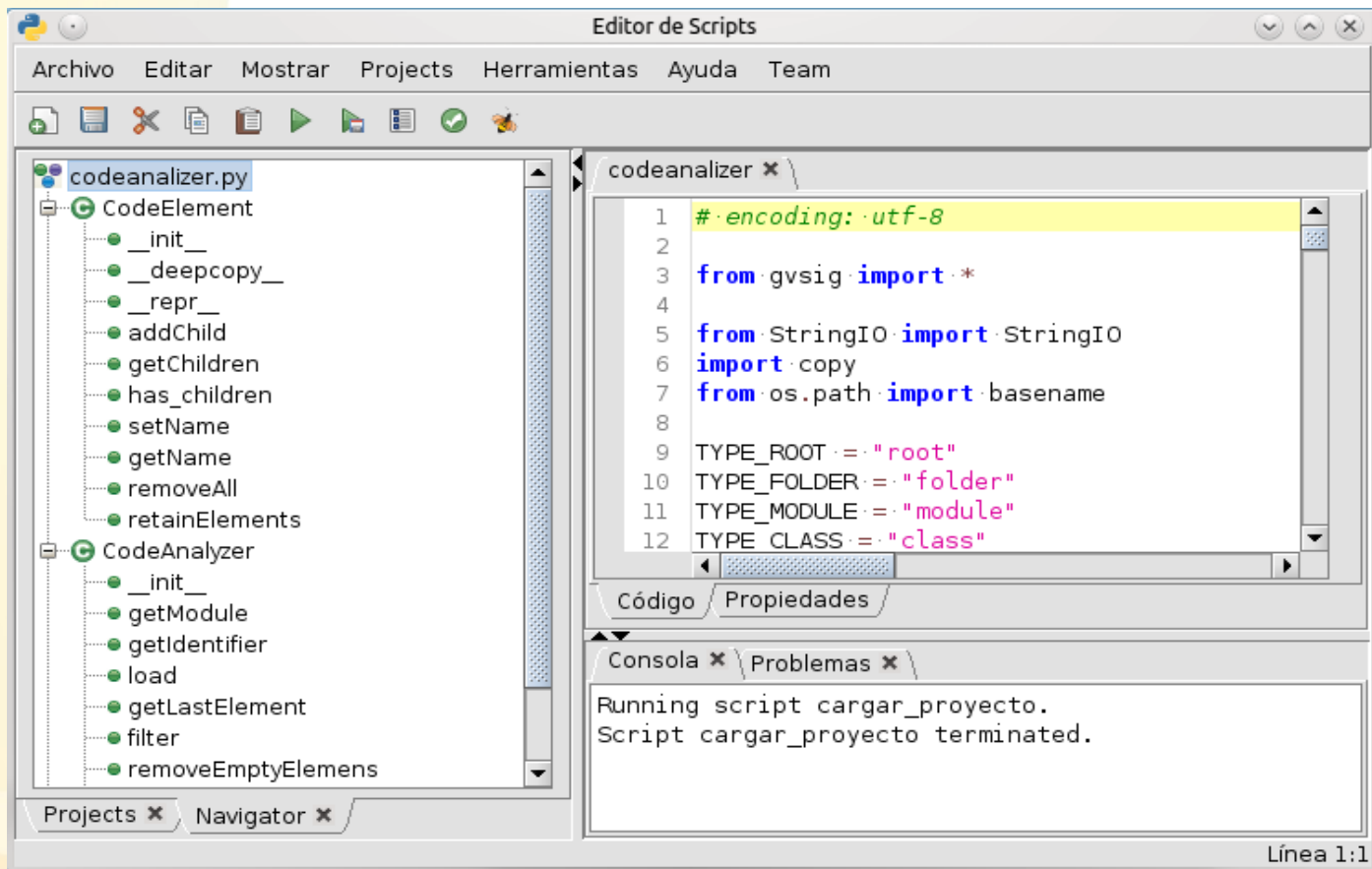


Comprobación de sintaxis bajo petición y mientras escribimos

El entorno de scripting

Funcionalidades

Scripting Composer Tools

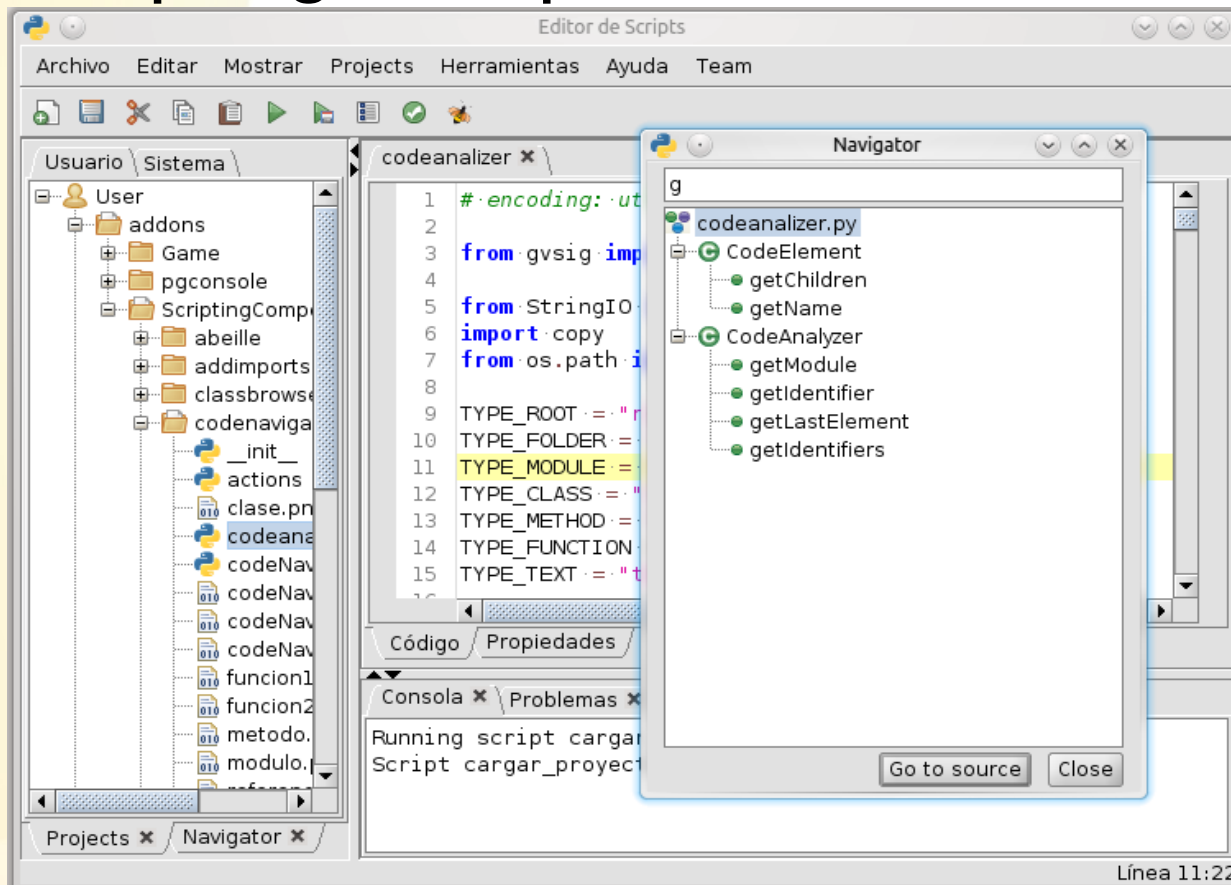


Navegador
de código

El entorno de scripting

Funcionalidades

Scripting Composer Tools

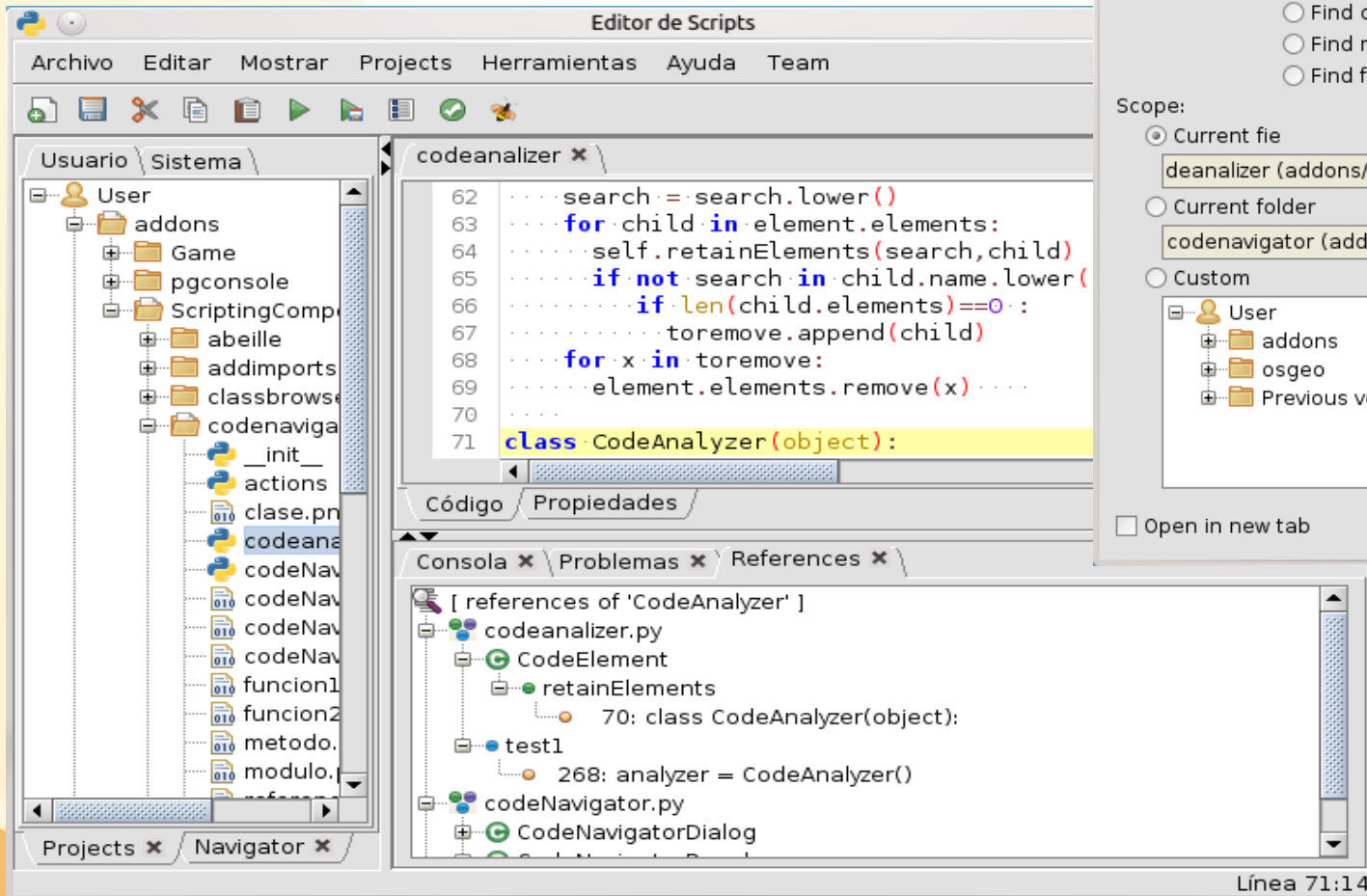


Búsqueda rápida
de funciones
clases y métodos
(en el script corriente)

El entorno de scripting

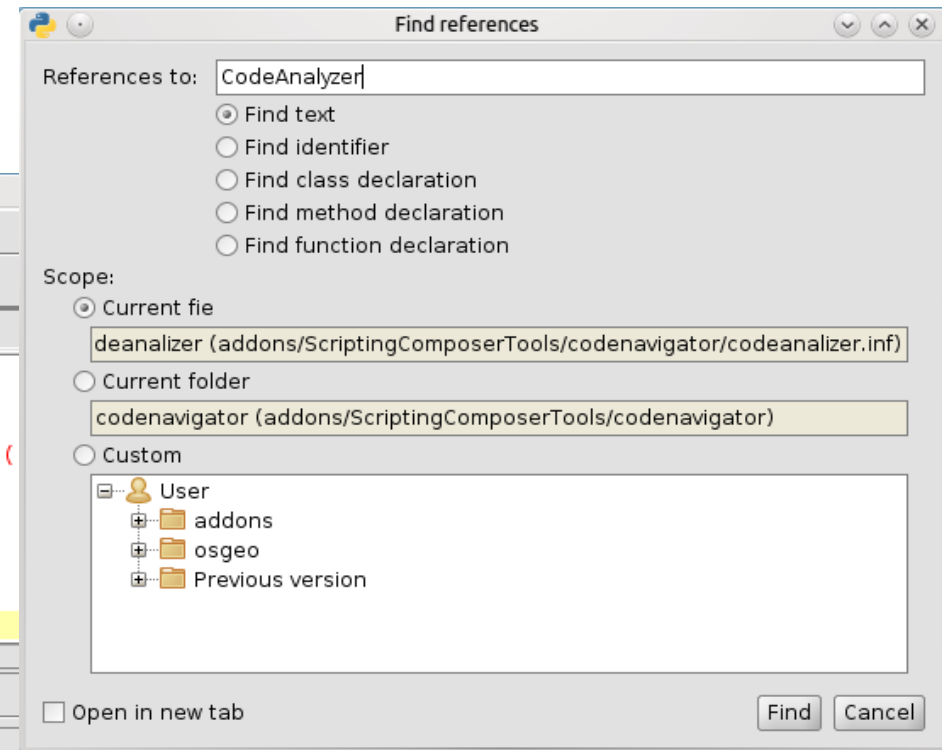
Funcionalidades

Scripting Composer Tools



The screenshot shows the 'Editor de Scripts' window with a menu bar (Archivo, Editar, Mostrar, Projects, Herramientas, Ayuda, Team) and a toolbar. The left sidebar shows a project tree with folders like 'User', 'addons', 'Game', 'pgconsole', 'ScriptingComp', 'abeille', 'addimports', 'classbrowse', 'codenaviga', and files like 'acciones', 'clase.py', 'codeana', 'codeNav', 'funcion1', 'funcion2', 'metodo.', 'modulo.'. The main editor shows Python code in 'codeanalyzer.py' with line numbers 62-71. Line 71 is highlighted: `class CodeAnalyzer(object):`. Below the editor is a 'Consola' window showing references for 'CodeAnalyzer':

```
[ references of 'CodeAnalyzer' ]
- codeanalyzer.py
  - CodeElement
    - retainElements
      - 70: class CodeAnalyzer(object):
  - test1
    - 268: analyzer = CodeAnalyzer()
- codeNavigator.py
  - CodeNavigatorDialog
```



The 'Find references' dialog box has a search field containing 'CodeAnalyzer'. The search options are:

- Find text
- Find identifier
- Find class declaration
- Find method declaration
- Find function declaration

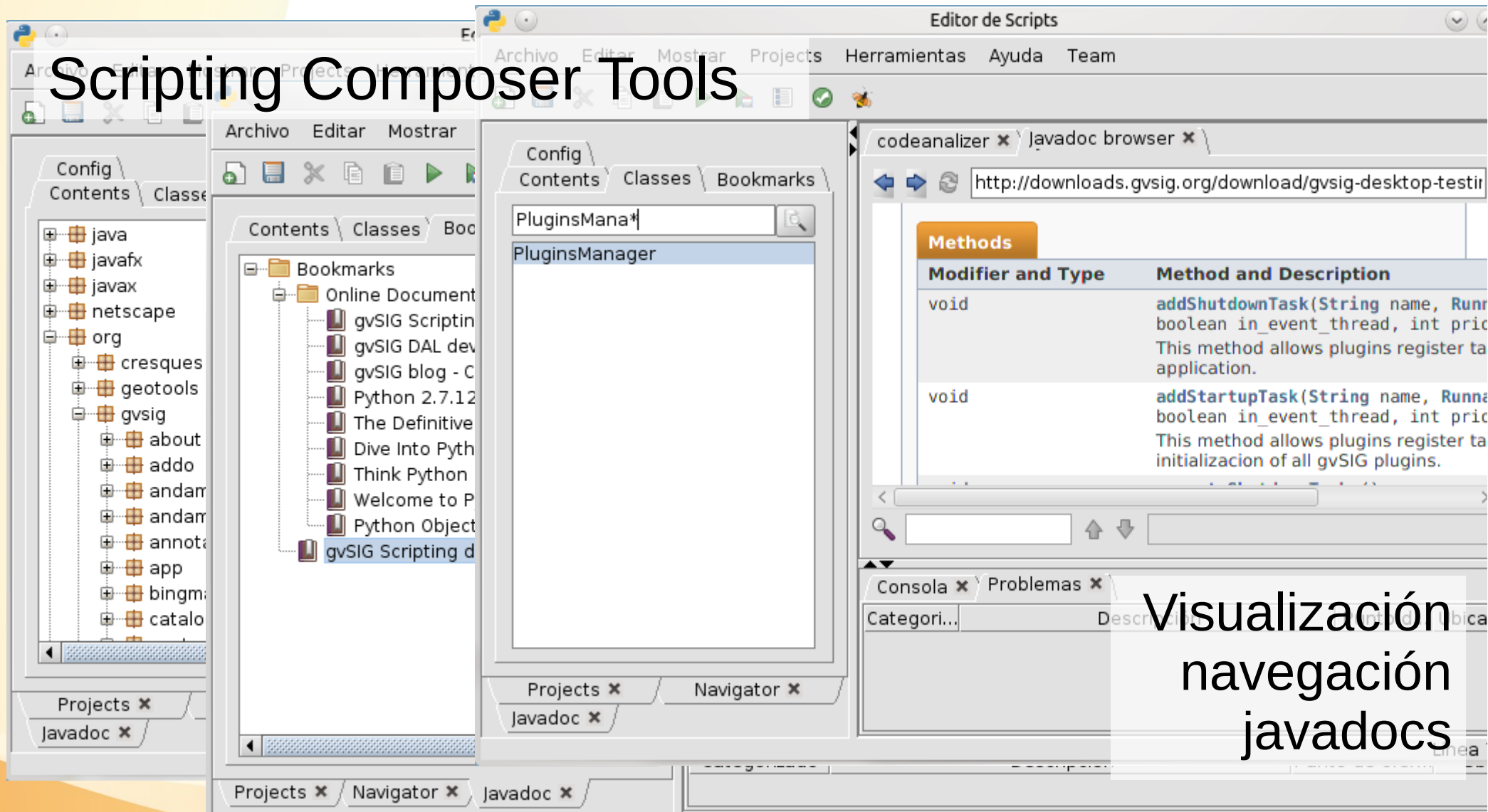
The scope is set to 'Current file' with the path: `deanalyzer (addons/ScriptingComposerTools/codenavigator/codeanalyzer.inf)`. Other scope options include 'Current folder' (path: `codenavigator (addons/ScriptingComposerTools/codenavigator)`) and 'Custom' (showing a tree with 'User', 'addons', 'osgeo', 'Previous version'). There is an 'Open in new tab' checkbox and 'Find' and 'Cancel' buttons.

Búsqueda de funciones, clases y métodos en ficheros

El entorno de scripting

Funcionalidades

Scripting Composer Tools



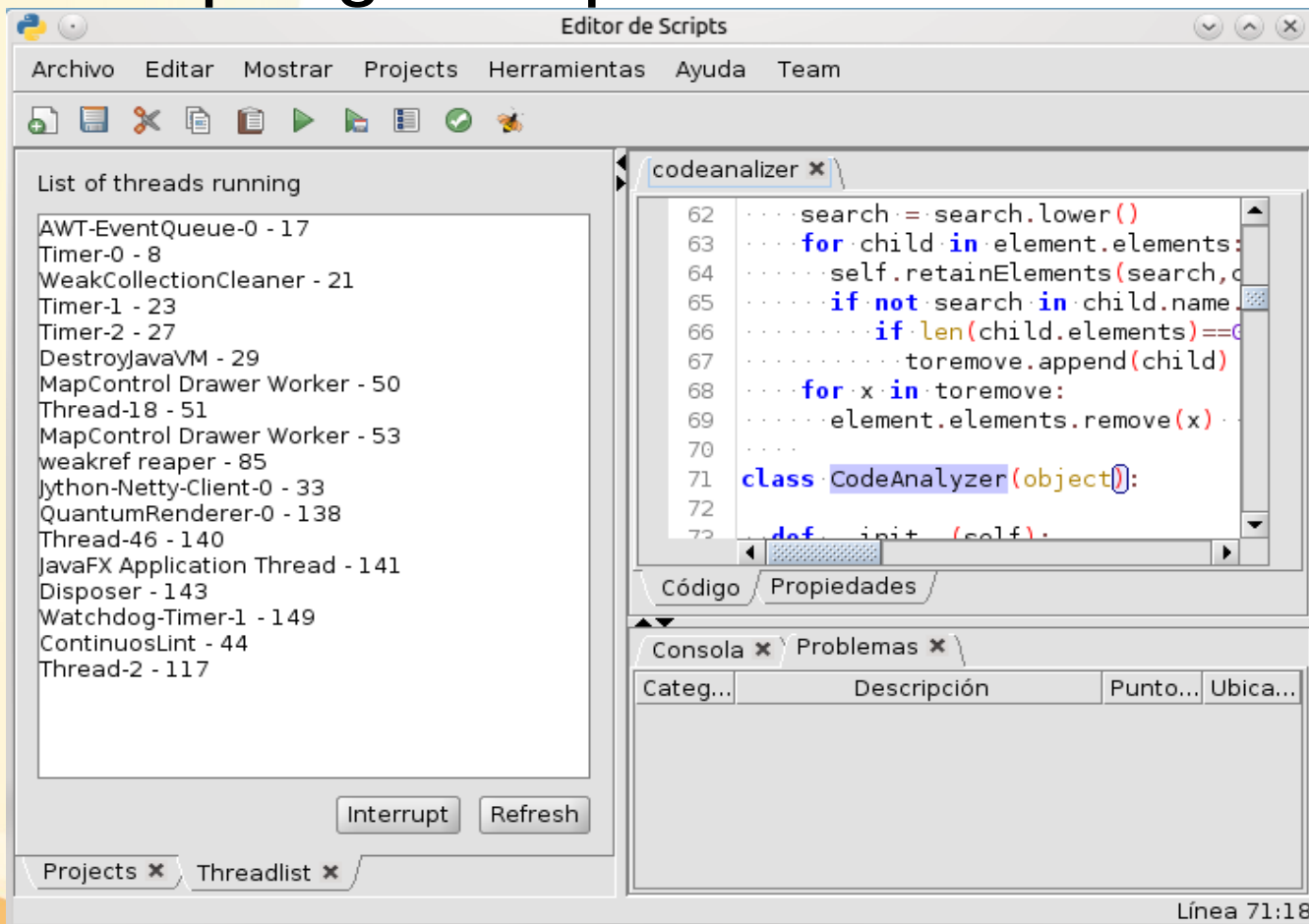
Modifier and Type	Method and Description
void	addShutdownTask(String name, Runnable in_event_thread, int priority) This method allows plugins register to application.
void	addStartupTask(String name, Runnable in_event_thread, int priority) This method allows plugins register to inicializacion of all gvSIG plugins.

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 and names, such as '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'. At the bottom of this pane are 'Interrupt' and 'Refresh' buttons. The right pane, titled 'codeanalyzer x', shows a code editor with the following code:

```
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):
```

Below the code editor are tabs for 'Código' and 'Propiedades'. At the bottom of the window, there are tabs for 'Consola x' and 'Problemas x', and a table with columns 'Categ...', 'Descripción', 'Punto...', and 'Ubica...'. The status bar at the bottom right indicates 'Línea 71:18'.

Visualización
de hilos en
ejecución

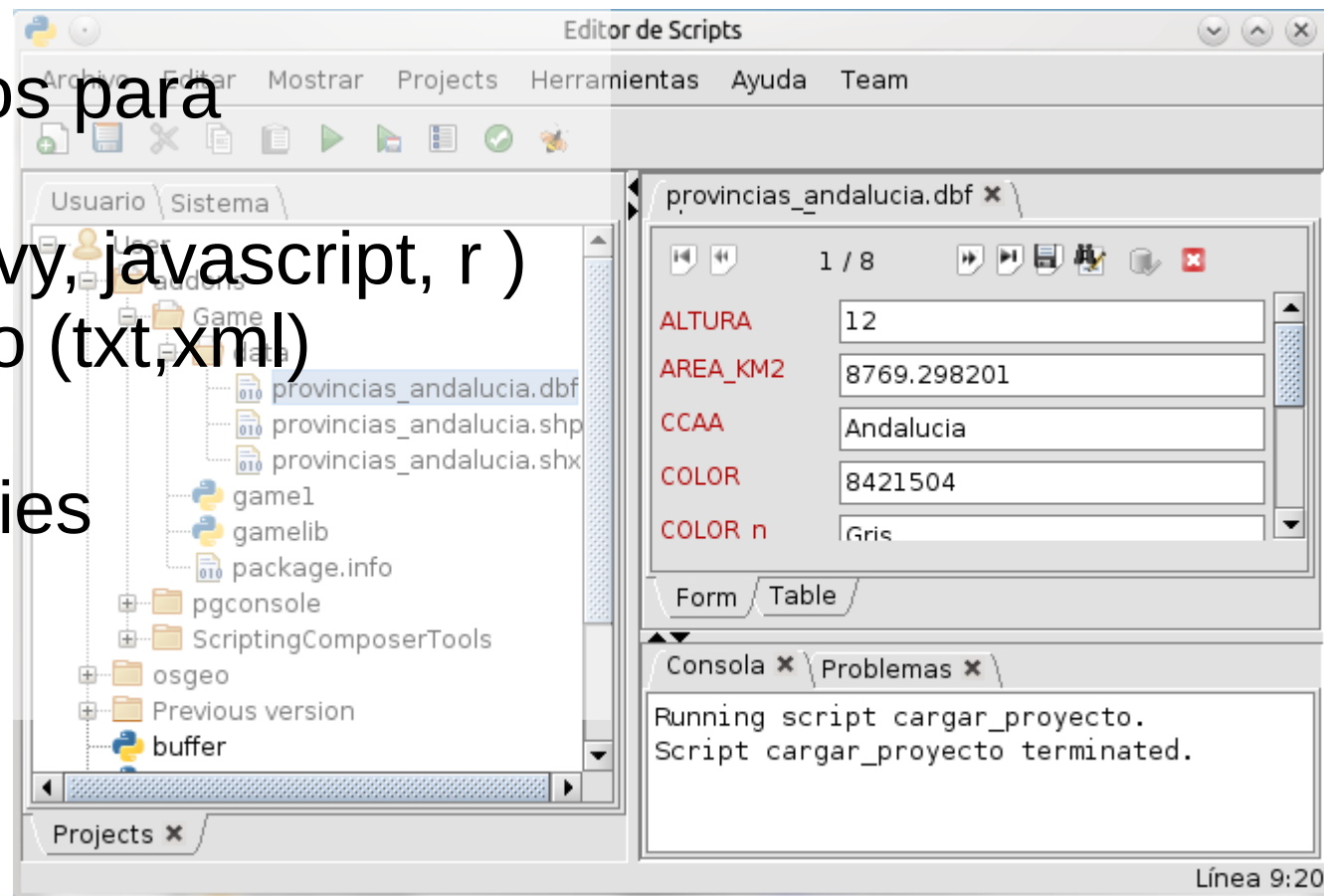
El entorno de scripting

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...

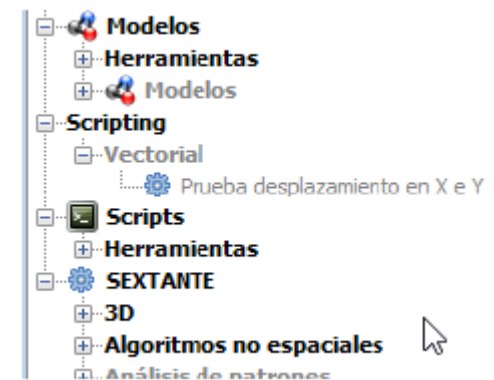
- geopy (geolocalización web)
- gvpy (acceso a la caja de herramientas)
- JOpenDocument (acceso a ODT/ODS)
- CSSBox (navegador web)
- cartodb
- xmltodict
- ogr2ogr

El entorno de scripting

Funcionalidades

Soporte para integrar scripts en:

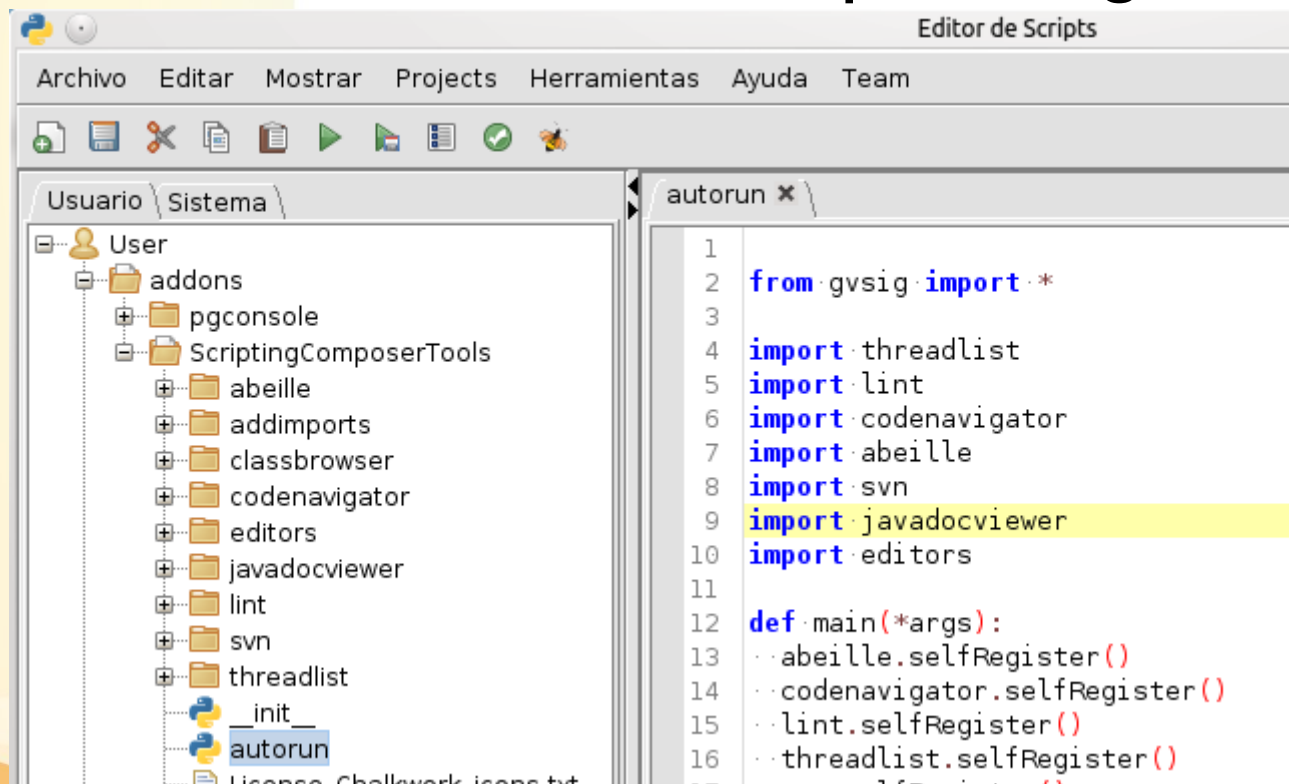
- Menús de gvSIG
- Barra de botones de gvSIG
- Caja de herramientas (Geoprocesos)



El entorno de scripting

Funcionalidades

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





That's all Folks!

Contacto

Contactar y dudas en

Lista de usuarios
o

Lista de desarrolladores

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



info@gvsig.com

www.gvsig.com

Joaquin Jose del Cerro
jjdelcerro@gvsig.com