



Scripting en gvSIG

Joaquín José del Cerro
jjdelcerro@gvsig.com

Óscar Martínez
omartinez@gvsig.com
Twitter: [@masquesig](https://twitter.com/masquesig)

Qué podemos hacer con scripting

Scripting no es una herramienta GIS.

¿ Para qué sirve ?

¿Qué podemos hacer con ella ?

Qué podemos hacer con scripting

Scripting no es una herramienta GIS.

¿ Para qué sirve ?

¿Qué podemos hacer con ella ?

Podemos crear nuestras propias herramientas.

Qué 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. The status bar at the bottom indicates a scale of 1:1.857.045, units in Metros, and coordinates Lon = -0° 51' 47" and Lat = 41° 54' 12" in EPSG:4326.

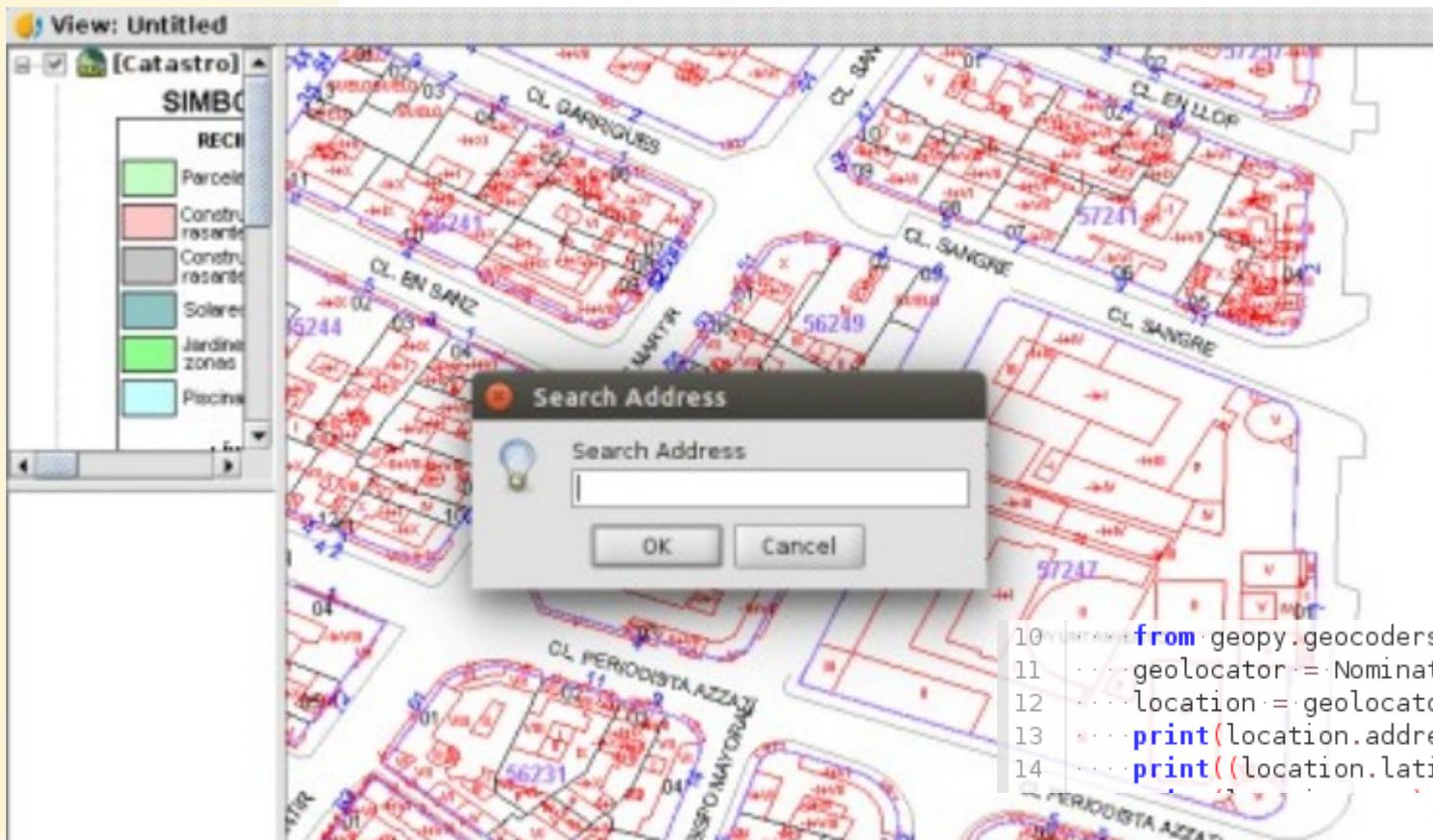
```

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

```

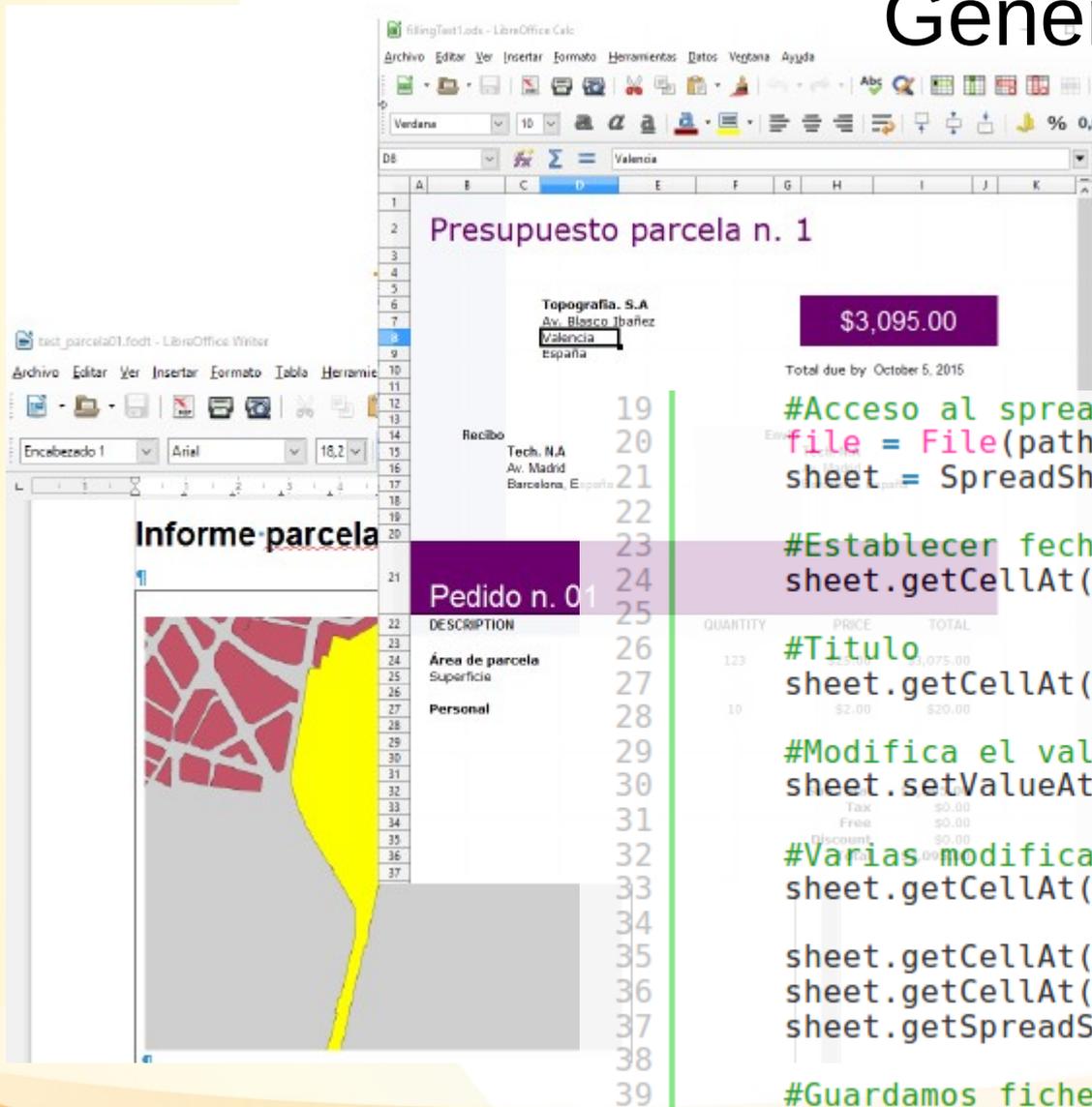
Qué podemos hacer con scripting

Usar servicios web
especializados
geopy



Qué 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)

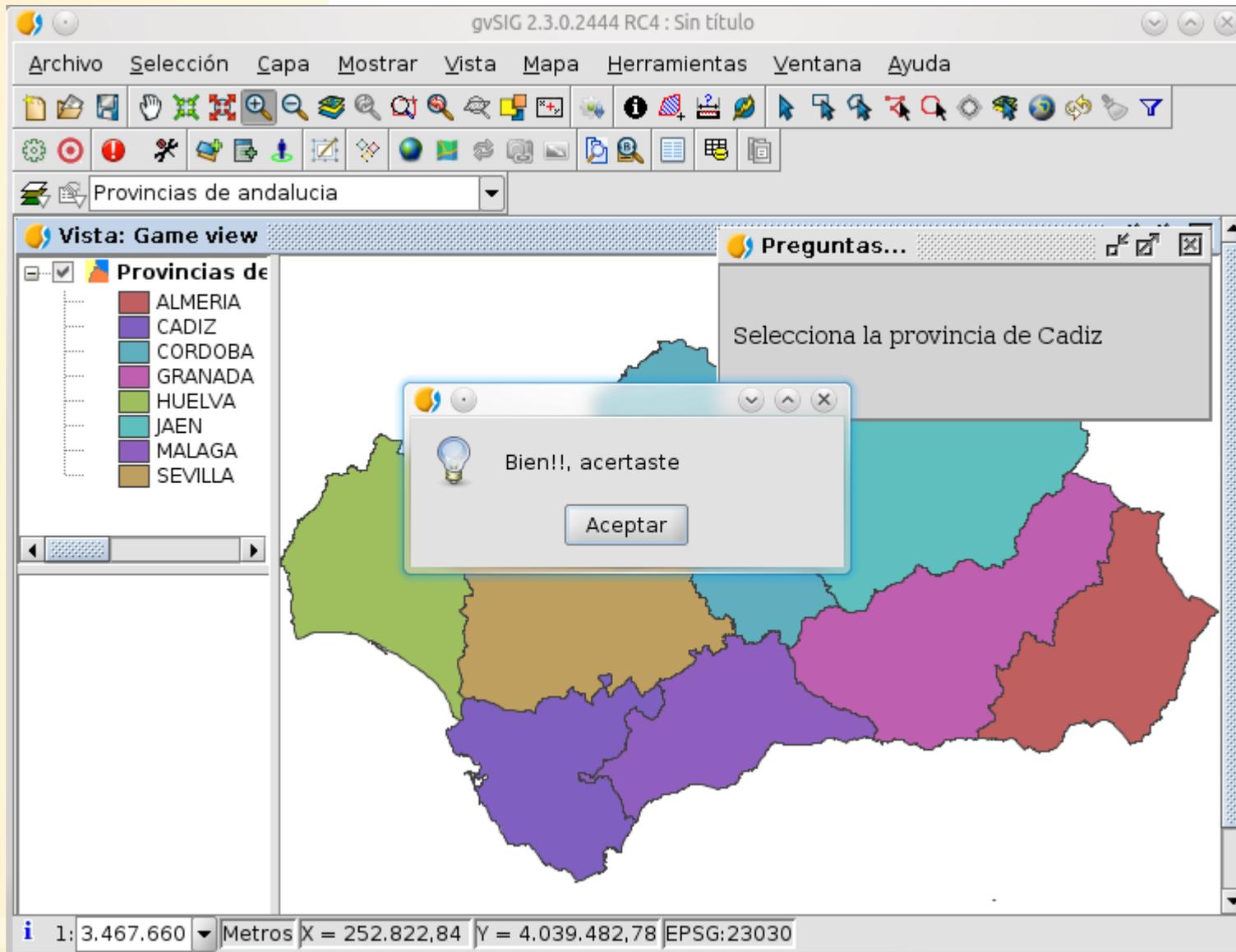
sheet.getCellAt("B27").setValue("Personal")
sheet.getCellAt("H27").setValue(2)
sheet.getSpreadSheet().getTableModel("Products").setVa

#Guardamos fichero

```

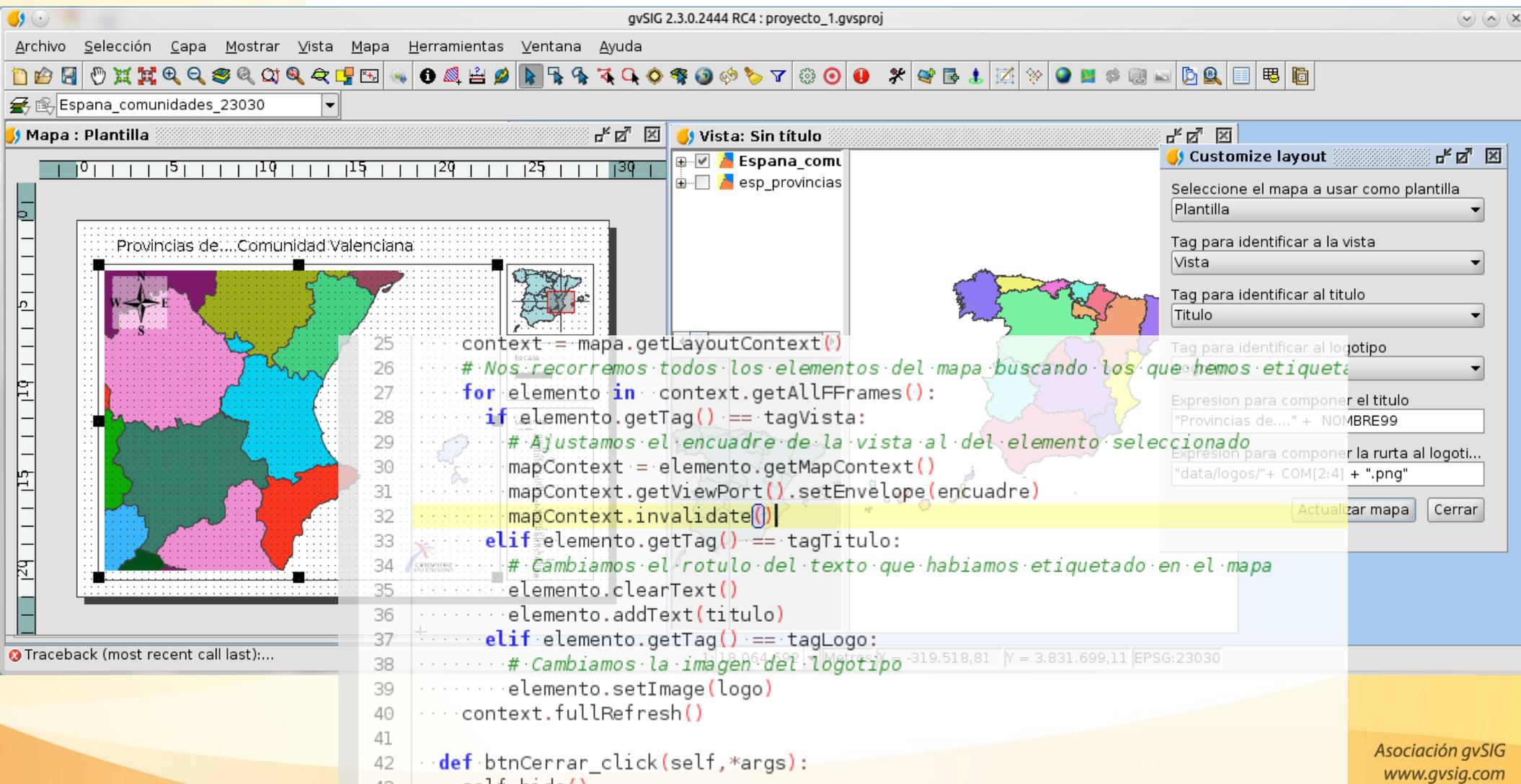
Qué podemos hacer con scripting

Juegos
didácticos



Qué 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:

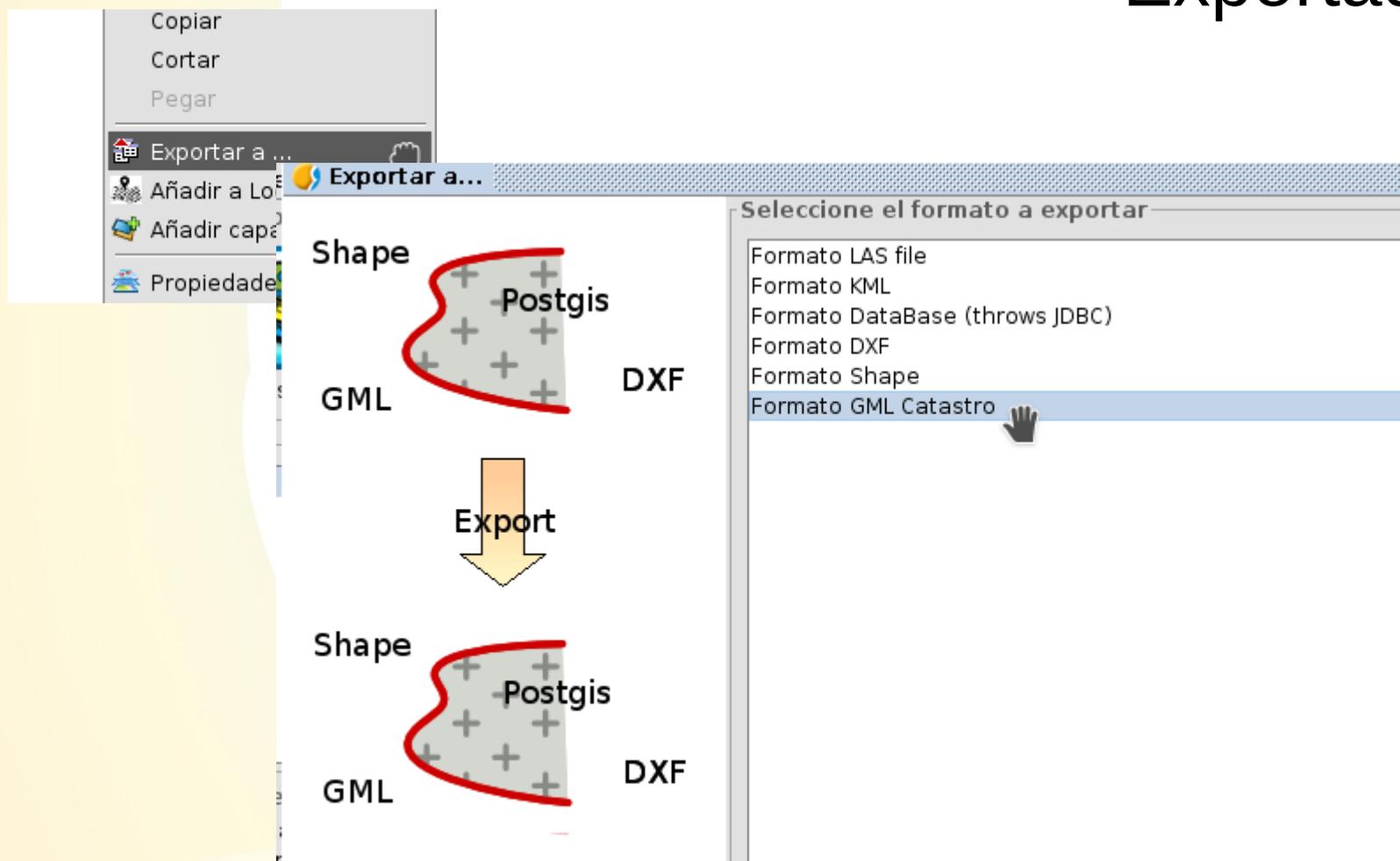
- Seleccione el mapa a usar como plantilla:
- Tag para identificar a la vista:
- Tag para identificar al titulo:
- Tag para identificar al logotipo:
- Expresion para componer el titulo:
- Expresion para componer la ruta al logotipo:

Buttons:

At the bottom left, a traceback message is visible: `Traceback (most recent call last):...`

Qué podemos hacer con scripting

Exportador GML



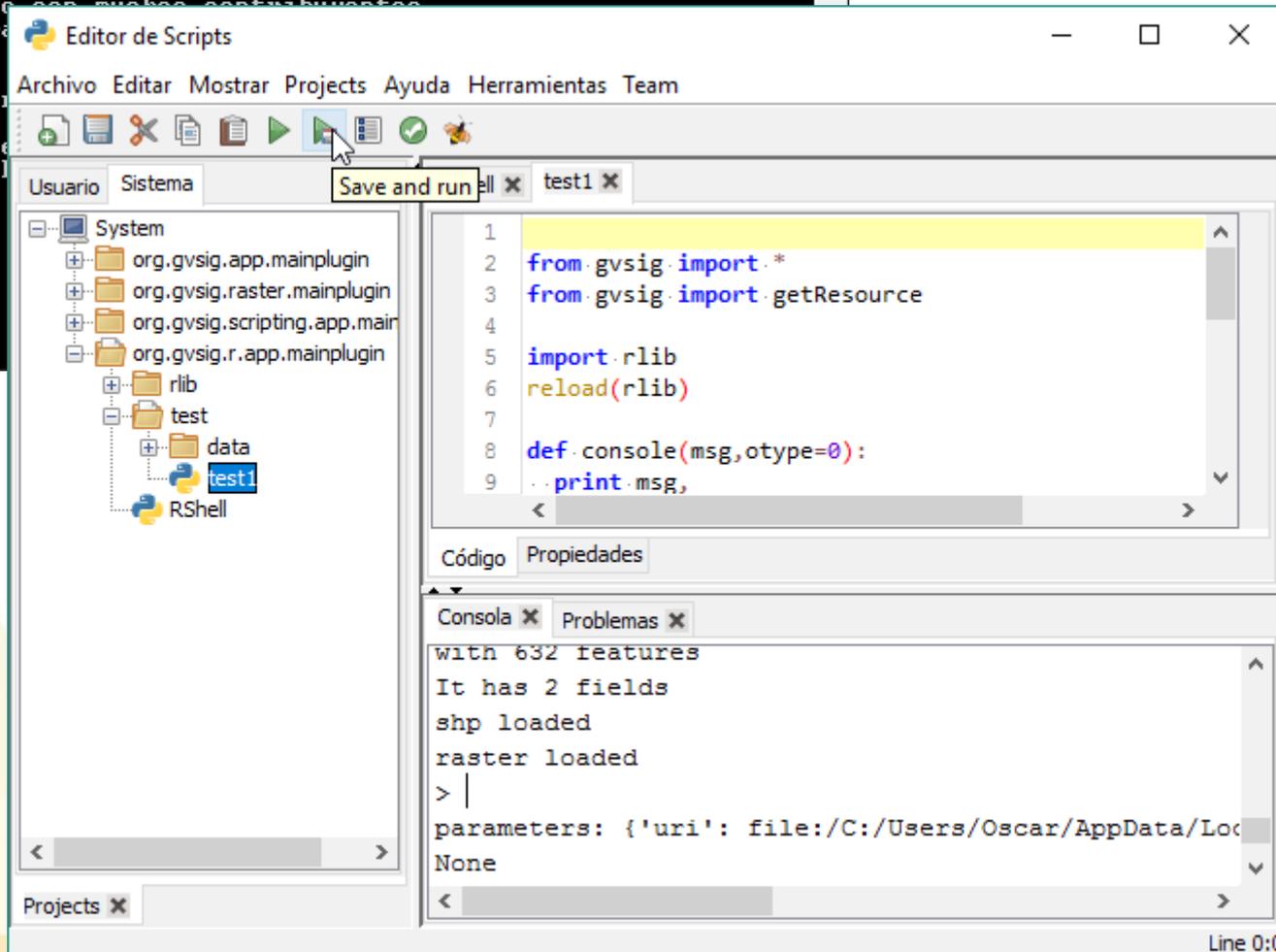
Qué podemos hacer con scripting

Integración con R

```
Rterm (64-bit)
R version 3.3.1 (2016-06-21) -- "Bug in Your Hair"
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R es un software libre y viene sin GARANTIA ALGUNA.
Usted puede redistribuirlo bajo ciertas circunstancias.
Escriba 'license()' o 'licence()' para detalles de distribución.

R es un proyecto colaborativo con muchos contribuyentes.
Escriba 'contributors()' para ver los nombres de los contribuyentes.
Escriba 'citation()' para saber cómo citar este software.
Escriba 'demo()' para demostrar algunas funciones.
Escriba 'help.start()' para abrir el sitio web de ayuda.
Escriba 'q()' para salir de R.
>
```



The screenshot shows the 'Editor de Scripts' window with a file named 'test1'. The code in the editor is as follows:

```
1
2 from gvsig import *
3 from gvsig import getResource
4
5 import rlib
6 reload(rlib)
7
8 def console(msg, otype=0):
9     print msg,
```

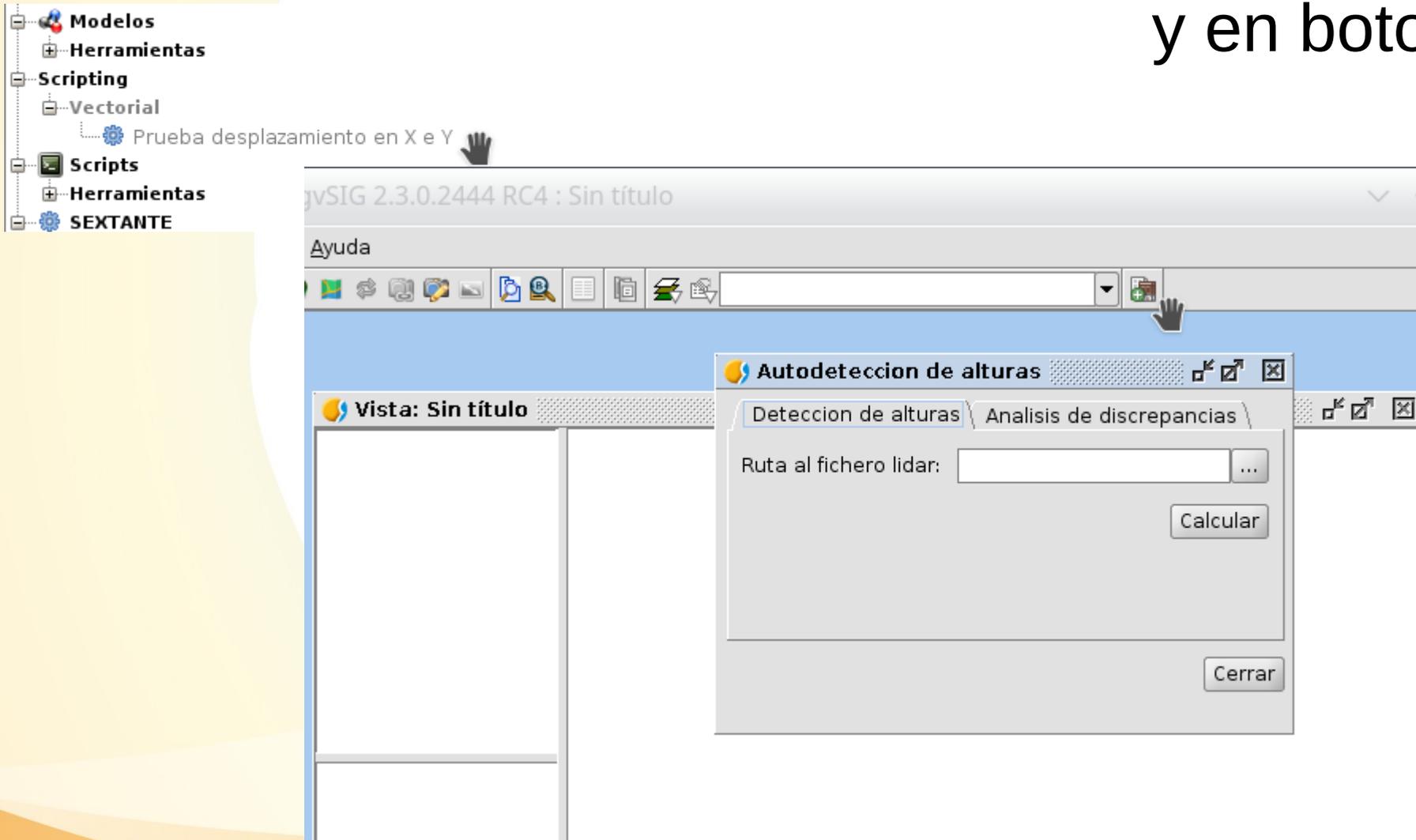
The console output shows the execution of the script:

```
with 632 features
It has 2 fields
shp loaded
raster loaded
> |
parameters: {'uri': file:/C:/Users/Oscar/AppData/Local/Temp/...}
None
```

The interface also shows a file explorer on the left with a tree view of the system structure, including folders like 'org.gvsig.app.mainplugin', 'org.gvsig.raster.mainplugin', 'org.gvsig.scripting.app.mair', and 'org.gvsig.r.app.mainplugin', and a file named 'test1' under a 'test' folder.

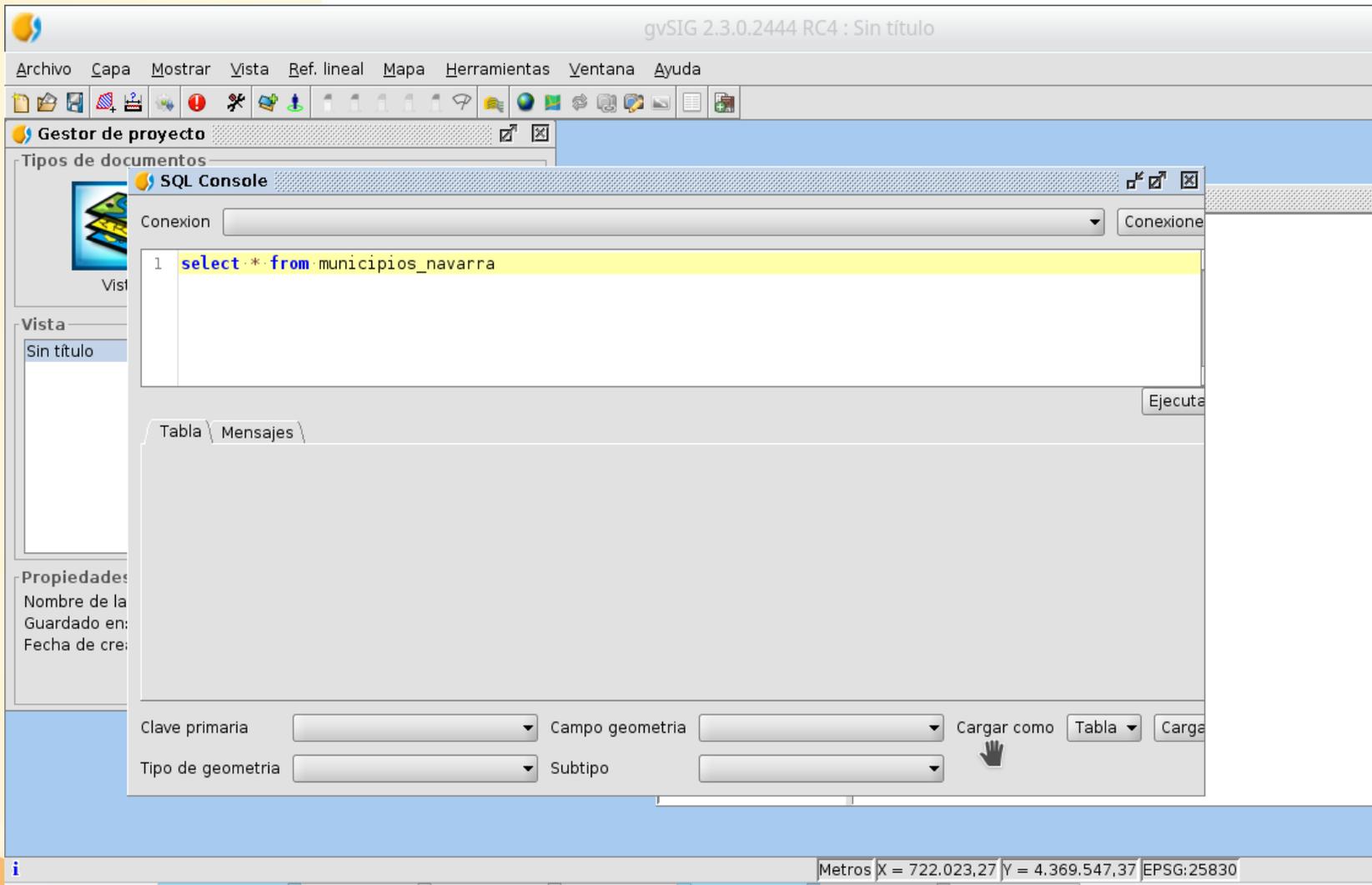
Qué podemos hacer con scripting

Integración en caja de Herramientas
y en botones



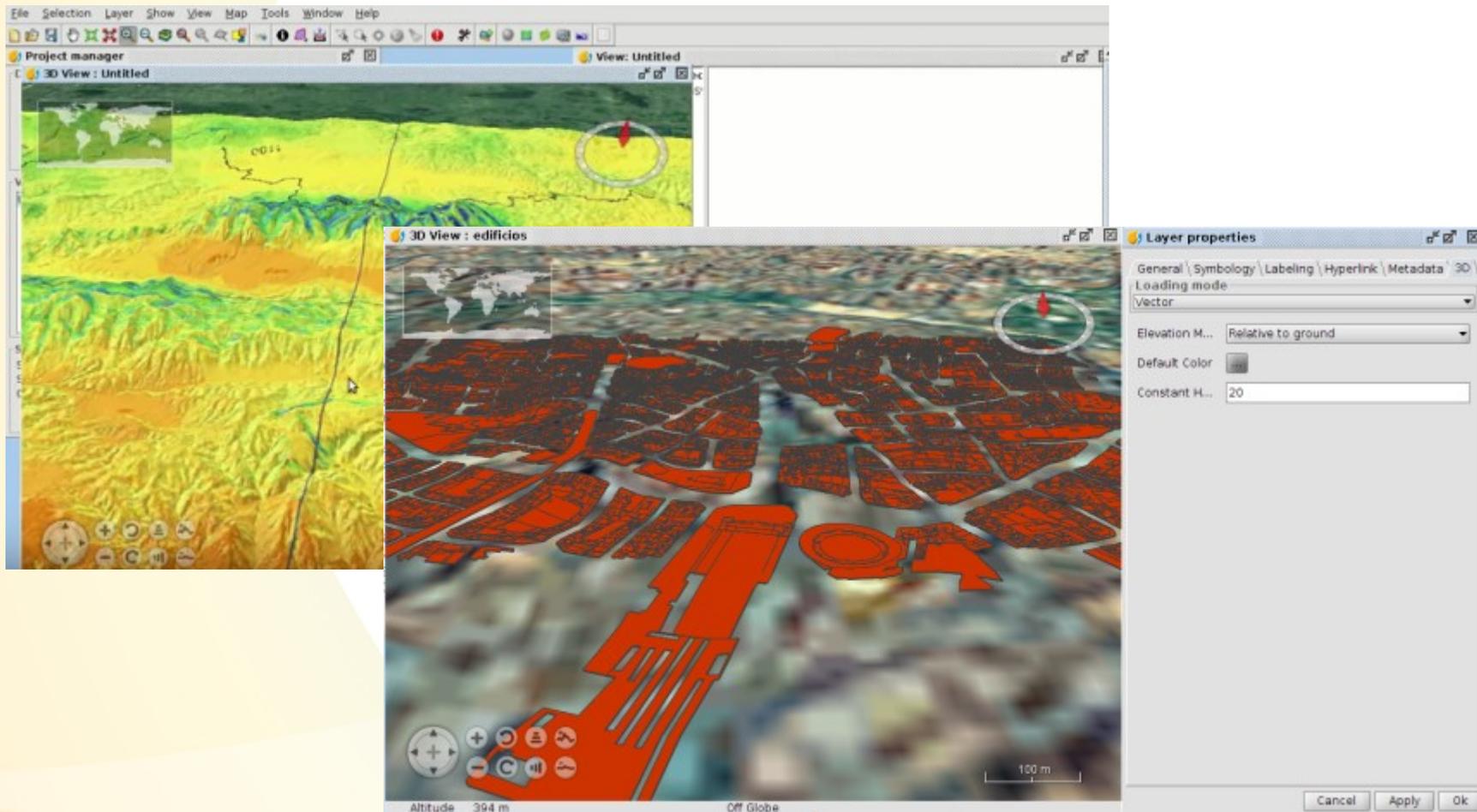
Qué podemos hacer con scripting

Herramientas de bases de datos



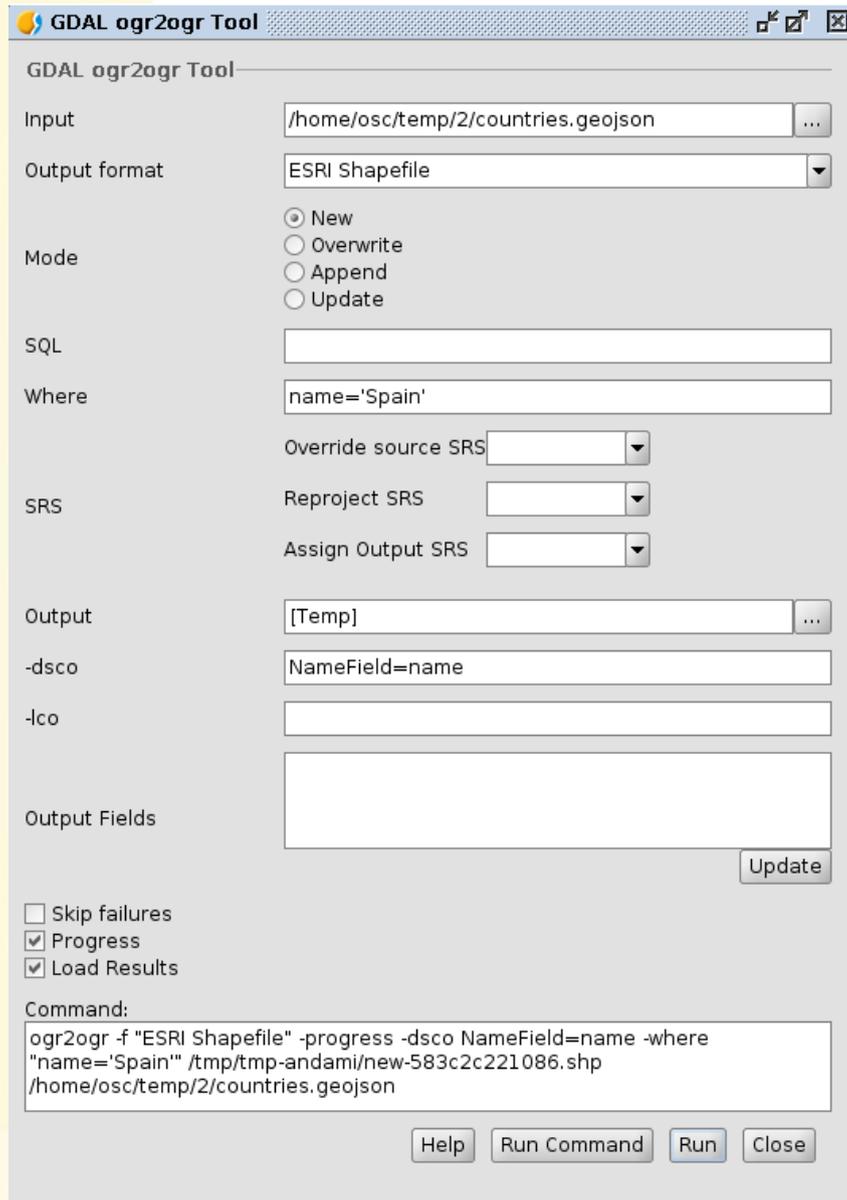
Qué podemos hacer con scripting

Interacción con otras extensiones



Qué podemos hacer con scripting

Herramientas basadas en GDAL



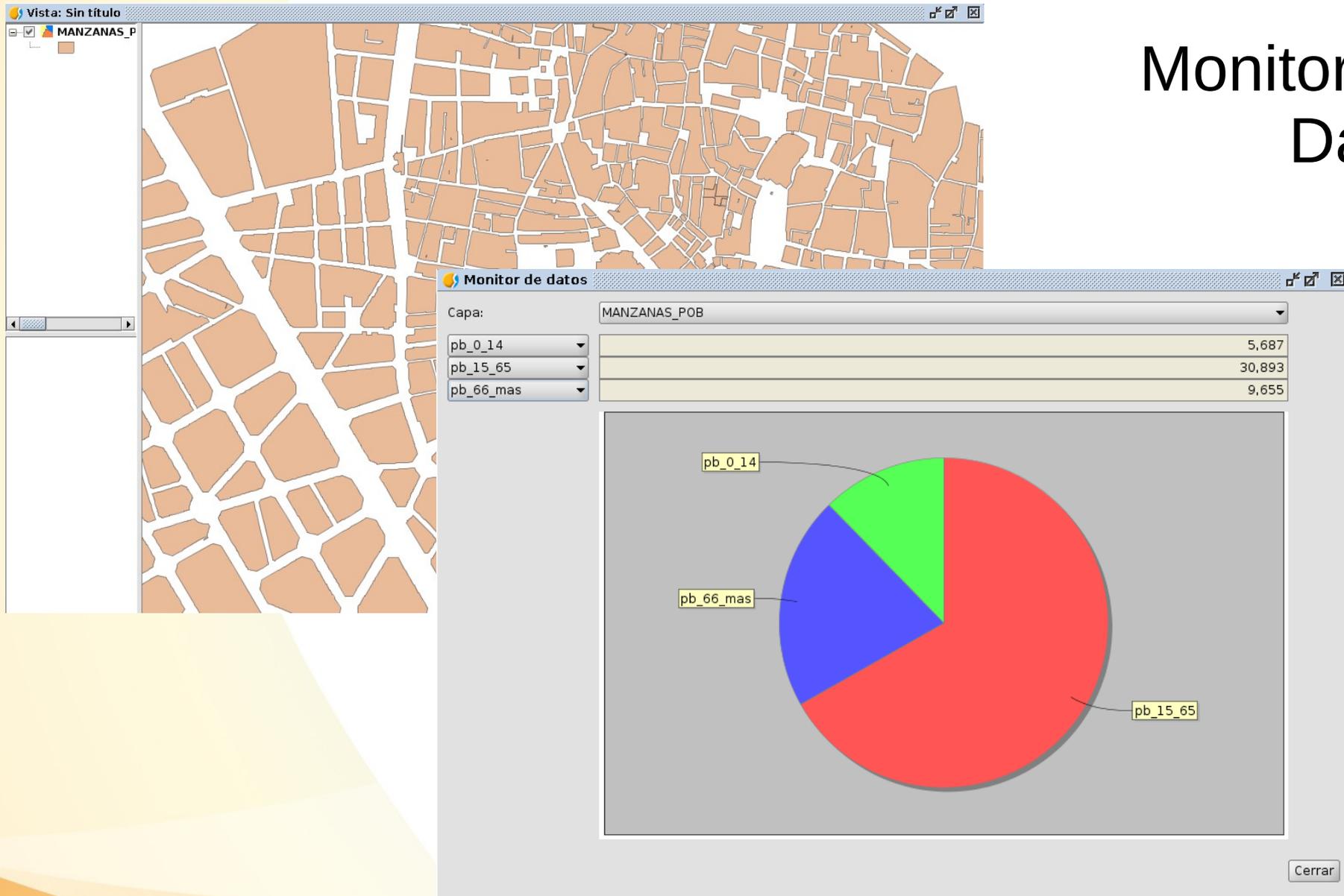
The screenshot shows the GDAL ogr2ogr Tool dialog box. The title bar reads "GDAL ogr2ogr Tool". The dialog contains the following fields and options:

- Input:** /home/osc/temp/2/countries.geojson
- Output format:** ESRI Shapefile
- Mode:** New, Overwrite, Append, Update
- SQL:** (empty)
- Where:** name='Spain'
- SRS:** Override source SRS, Reproject SRS, Assign Output SRS (all dropdown menus)
- Output:** [Temp]
- dsco:** NameField=name
- lco:** (empty)
- Output Fields:** (empty)
- Options:** Skip failures, Progress, Load Results
- Command:**

```
ogr2ogr -f "ESRI Shapefile" -progress -dsco NameField=name -where "name='Spain'" /tmp/tmp-andami/new-583c2c221086.shp /home/osc/temp/2/countries.geojson
```
- Buttons:** Help, Run Command, Run, Close

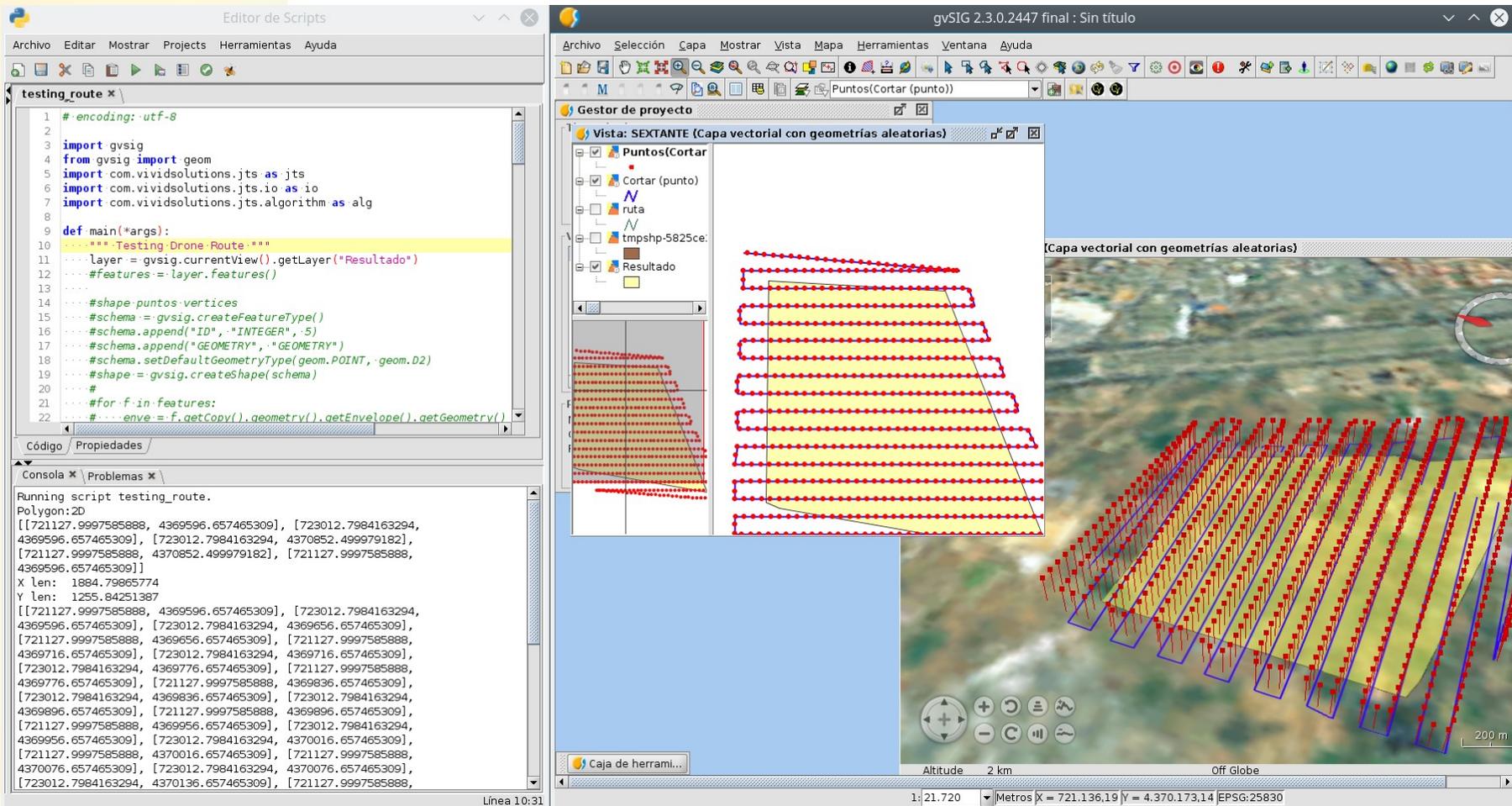
Qué podemos hacer con scripting

Monitorizar
Datos



Qué podemos hacer con scripting

Herramientas de cálculo de rutas para drones



The image displays the gvSIG 2.3.0.2447 interface with a Python script in the 'Editor de Scripts' window and a 3D map view in the 'Vista: SEXTANTE' window.

Scripting Code (testing_route.py):

```

1 #.encoding: utf-8
2
3 import gvsig
4 from gvsig import geom
5 import com.vividsolutions.jts as jts
6 import com.vividsolutions.jts.io as io
7 import com.vividsolutions.jts.algorithm as alg
8
9 def main(*args):
10     """Testing Drone Route"""
11     layer = gvsig.currentView().getLayer("Resultado")
12     #features = layer.features()
13
14     #shape: puntos vertices
15     schema = gvsig.createFeatureType()
16     schema.append("ID", "INTEGER", 5)
17     schema.append("GEOMETRY", "GEOMETRY")
18     schema.setDefaultGeometryType(geom.POINT, geom.D2)
19     shape = gvsig.createShape(schema)
20
21     #for f in features:
22     env = f.getCopy().getEnvelope().getGeometry()
  
```

Console Output:

```

Running script testing_route.
Polygon:2D
[[[721127.9997585888, 4369596.657465309], [723012.7984163294,
4369596.657465309], [723012.7984163294, 4370852.499979182],
[721127.9997585888, 4370852.499979182], [721127.9997585888,
4369596.657465309]]]
X len: 1884.79865774
Y len: 1255.84251387
[[[721127.9997585888, 4369596.657465309], [723012.7984163294,
4369596.657465309], [723012.7984163294, 4369656.657465309],
[721127.9997585888, 4369656.657465309], [721127.9997585888,
4369716.657465309], [723012.7984163294, 4369716.657465309],
[723012.7984163294, 4369776.657465309], [721127.9997585888,
4369776.657465309], [721127.9997585888, 4369836.657465309],
[723012.7984163294, 4369836.657465309], [723012.7984163294,
4369896.657465309], [721127.9997585888, 4369896.657465309],
[721127.9997585888, 4369956.657465309], [723012.7984163294,
4369956.657465309], [723012.7984163294, 4370016.657465309],
[721127.9997585888, 4370016.657465309], [721127.9997585888,
4370076.657465309], [723012.7984163294, 4370076.657465309],
[723012.7984163294, 4370136.657465309], [721127.9997585888,
  
```

The 3D map view shows a terrain with a yellow polygon representing the flight area. A red path is overlaid on the terrain, consisting of a series of parallel lines with red dots at the vertices, representing the drone's flight route. The map includes a scale bar (200 m) and altitude information (2 km).

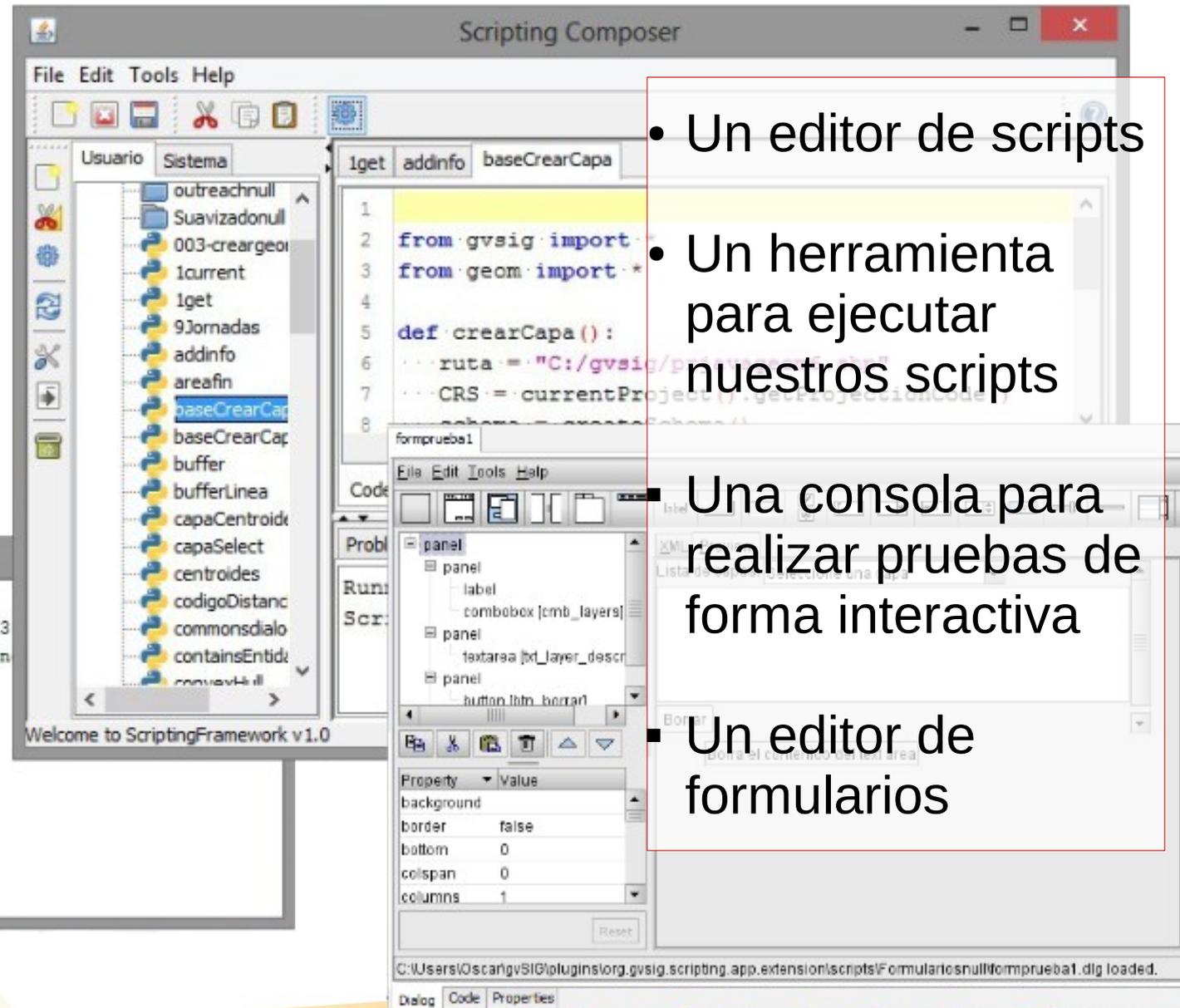
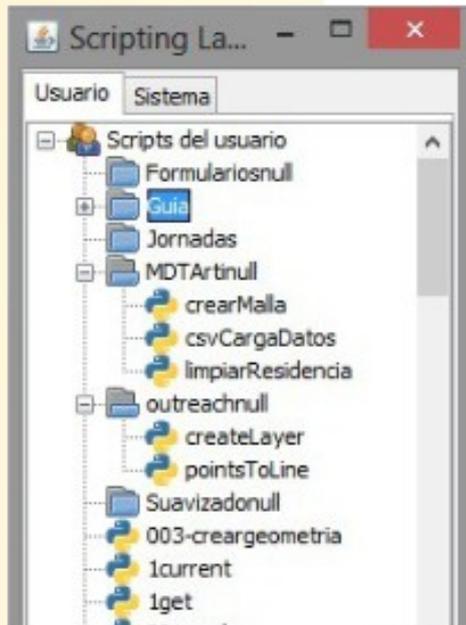
El entorno de scripting

Y...

Para hacer estas cosas...

¿ de qué entorno disponemos ?

El entorno de scripting

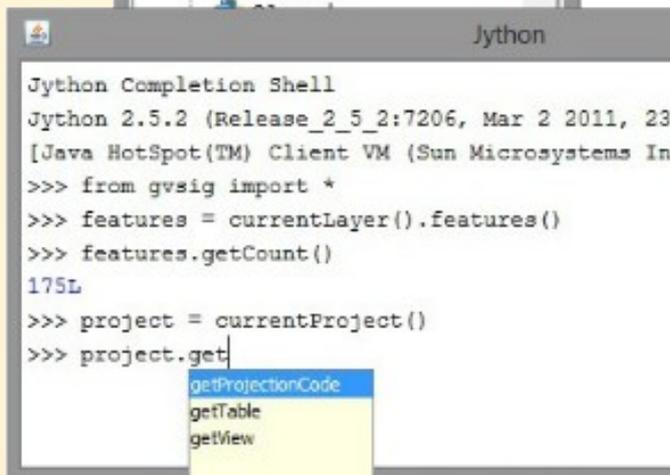


- Un editor de scripts

- 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...

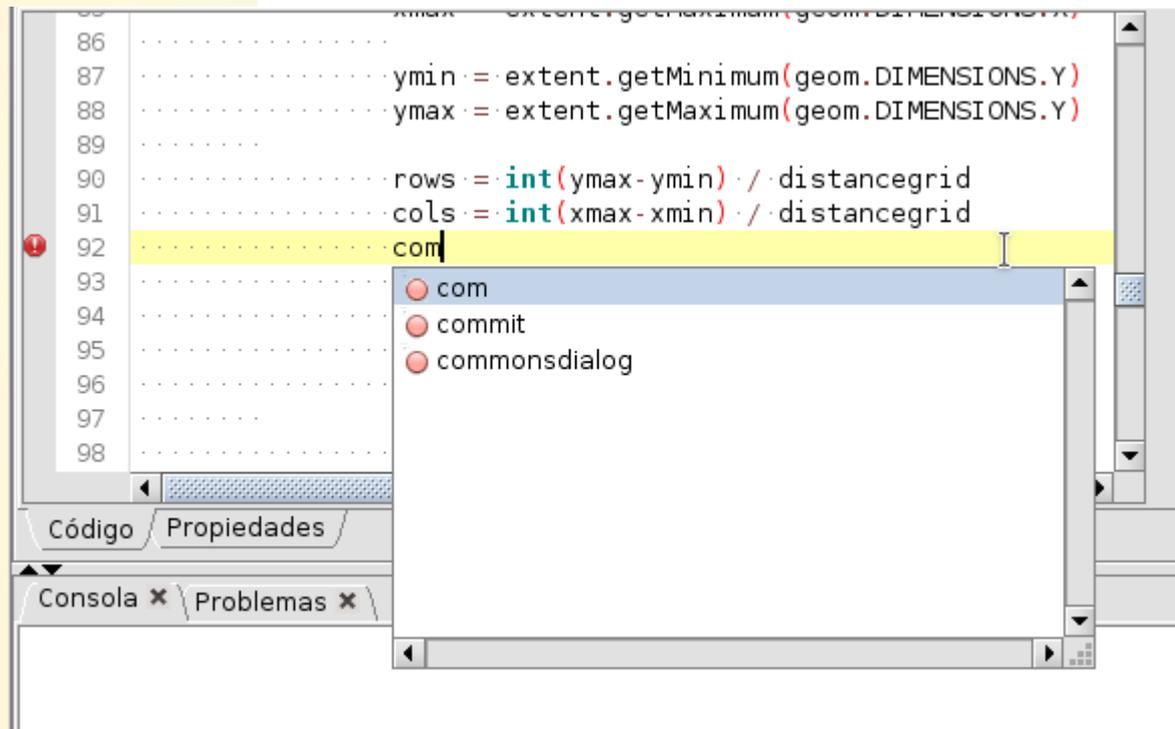
```
15 class MonitorSuma(ViewPortListener,FormPanel):
16
17     def __init__(self, mapContext, layer, label, fieldname):
18         FormPanel.__init__(self,script.getResource("monitor_suma3.xml"))
19         ....
20         self.mapContext = mapContext
21         self.layer = layer
22         self.fieldname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

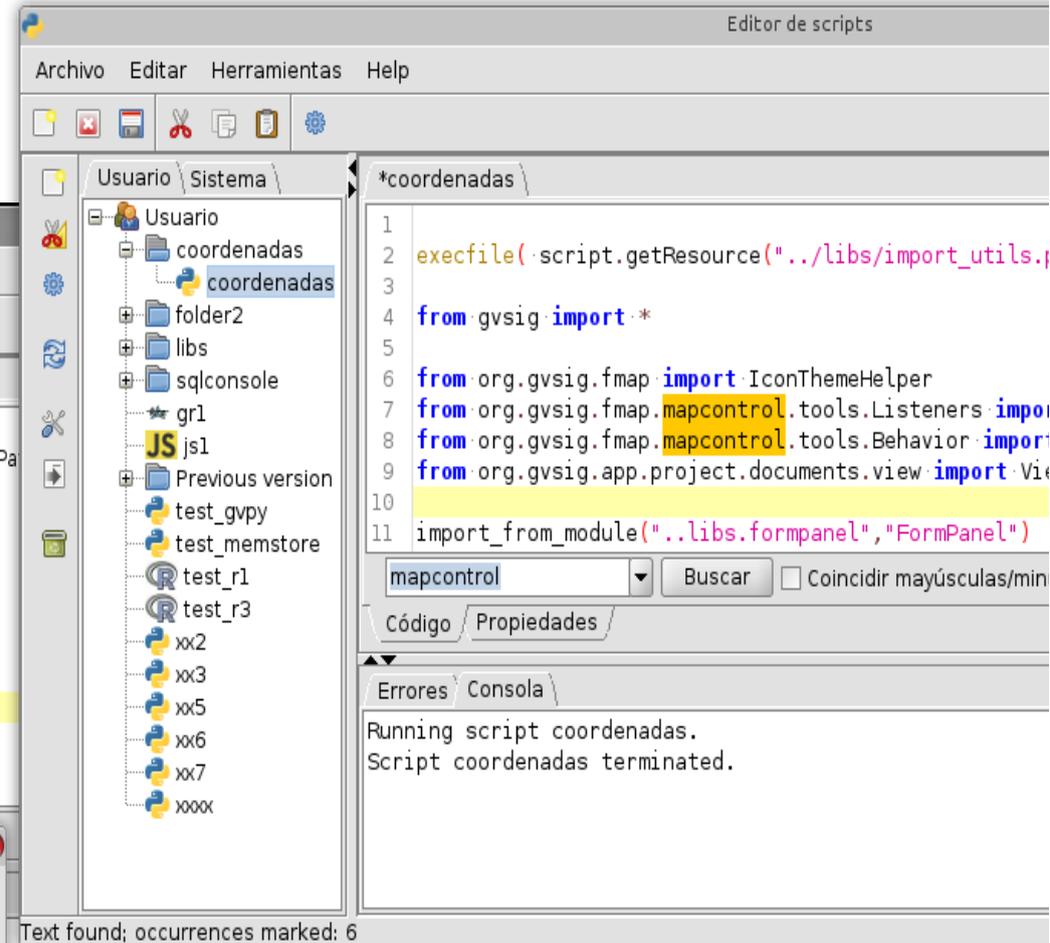
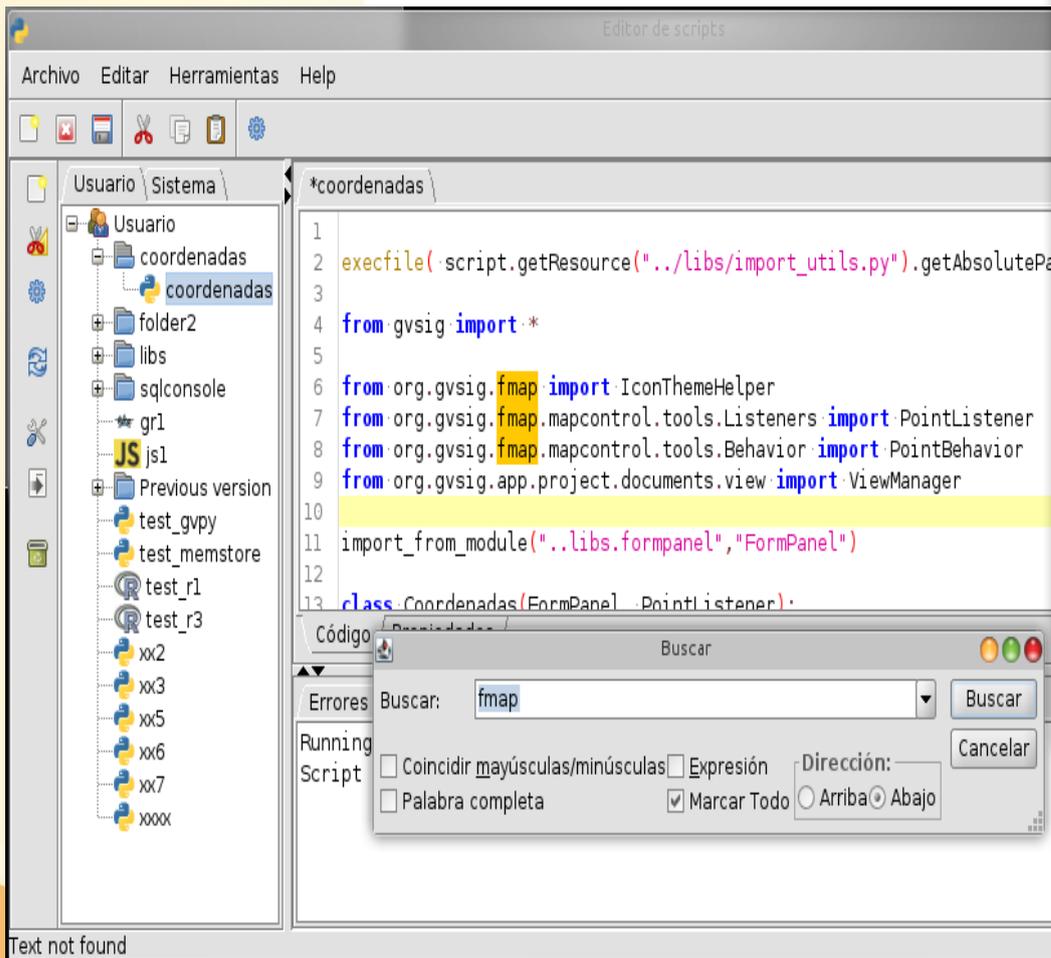


Autocompletado

El entorno de scripting

Funcionalidades

Editor de scripts con...

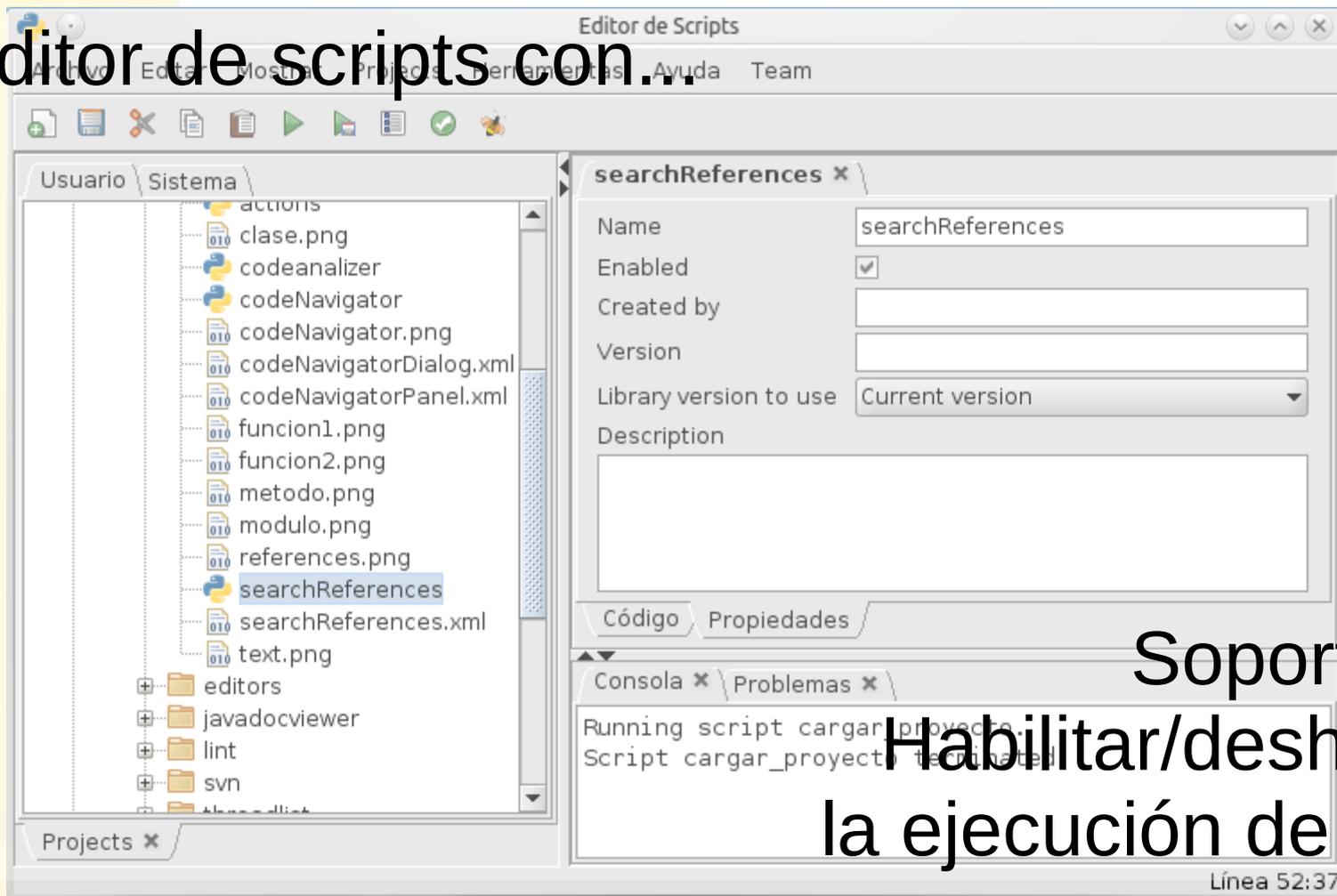


Búsquedas y reemplazos

El entorno de scripting

Funcionalidades

Editor de scripts con...

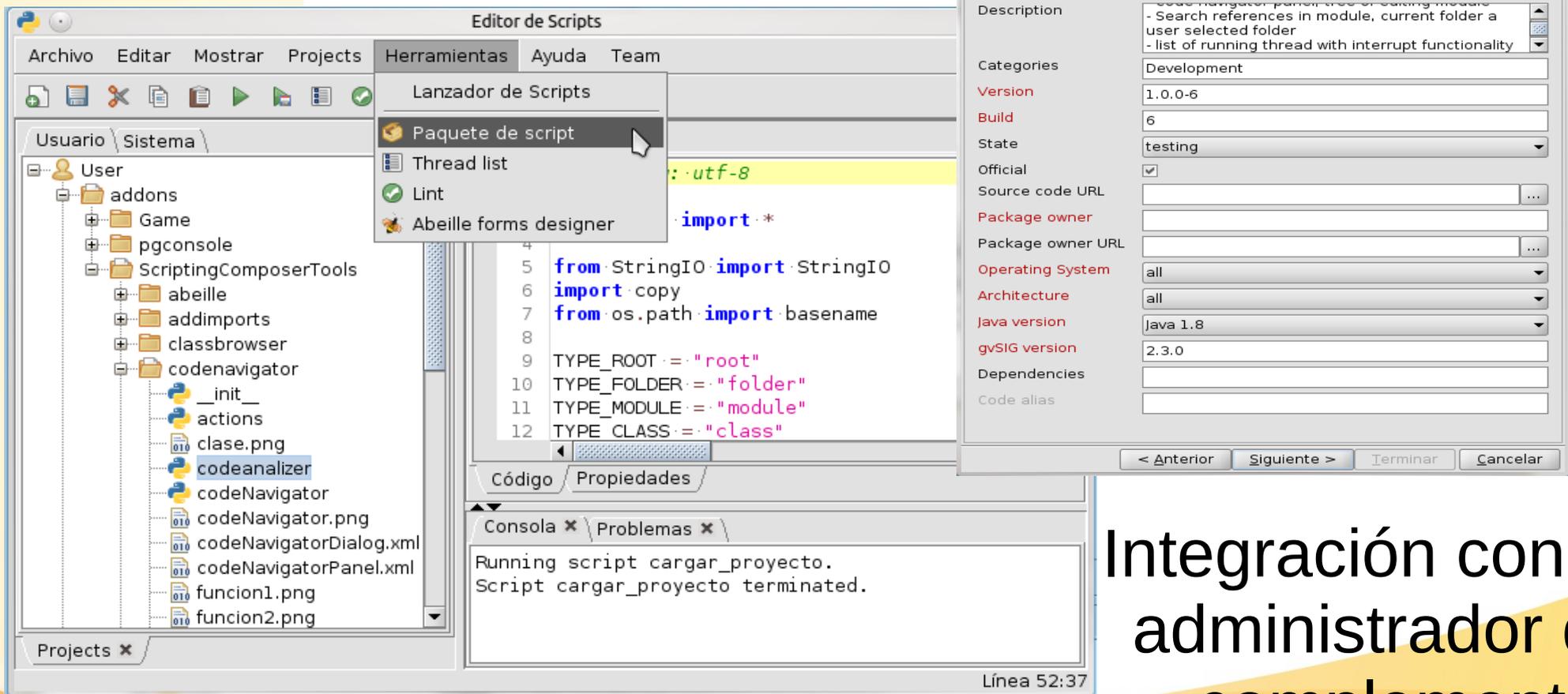


Soporte para
Habilitar/deshabilitar
la ejecución de scripts

El entorno de scripting

Funcionalidades

Editor de scripts con...



The screenshot displays the 'Editor de Scripts' interface. On the left, a project tree shows the 'ScriptingComposerTools' package. The main editor window shows a Python script with the following code:

```
1  # coding: utf-8
2
3  import sys
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"
```

The 'Herramientas' menu is open, showing options like 'Lanzador de Scripts', 'Paquete de script', 'Thread list', 'Lint', and 'Abeille forms designer'. The 'Paquete de script' option is highlighted. To the right, the 'Empaquetador de scripting' dialog is open, showing the 'Descripción del paquete' tab with the following details:

Property	Value
Type	Script
Code	ScriptingComposerTools
Name	Scripts: Composer tools
Description	Code Navigator, package tree or saving module - 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	<input checked="" type="checkbox"/>
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	

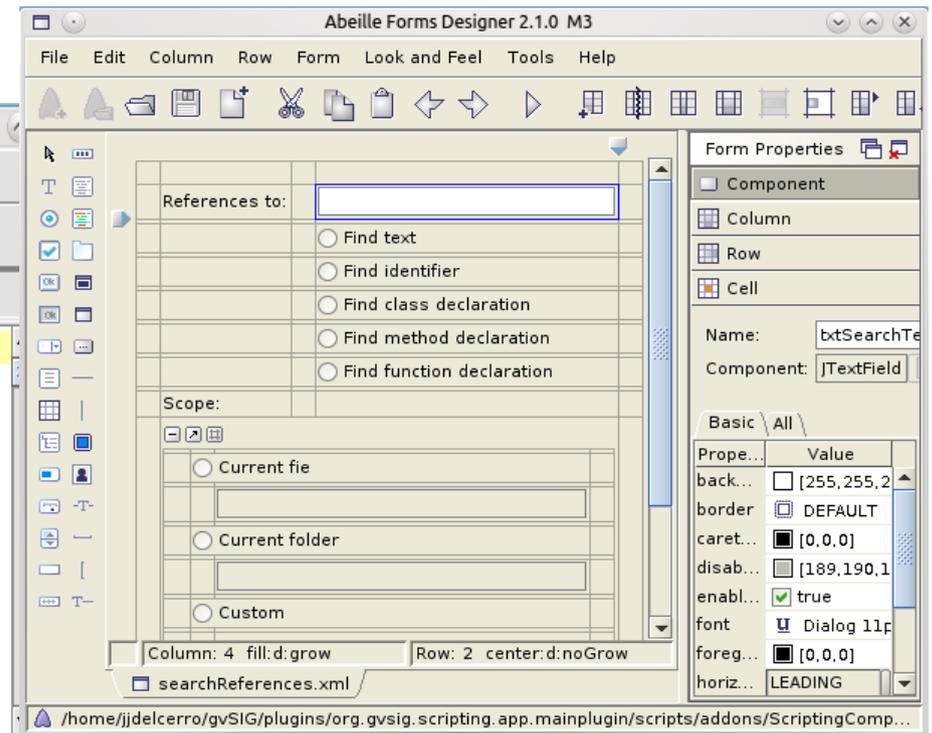
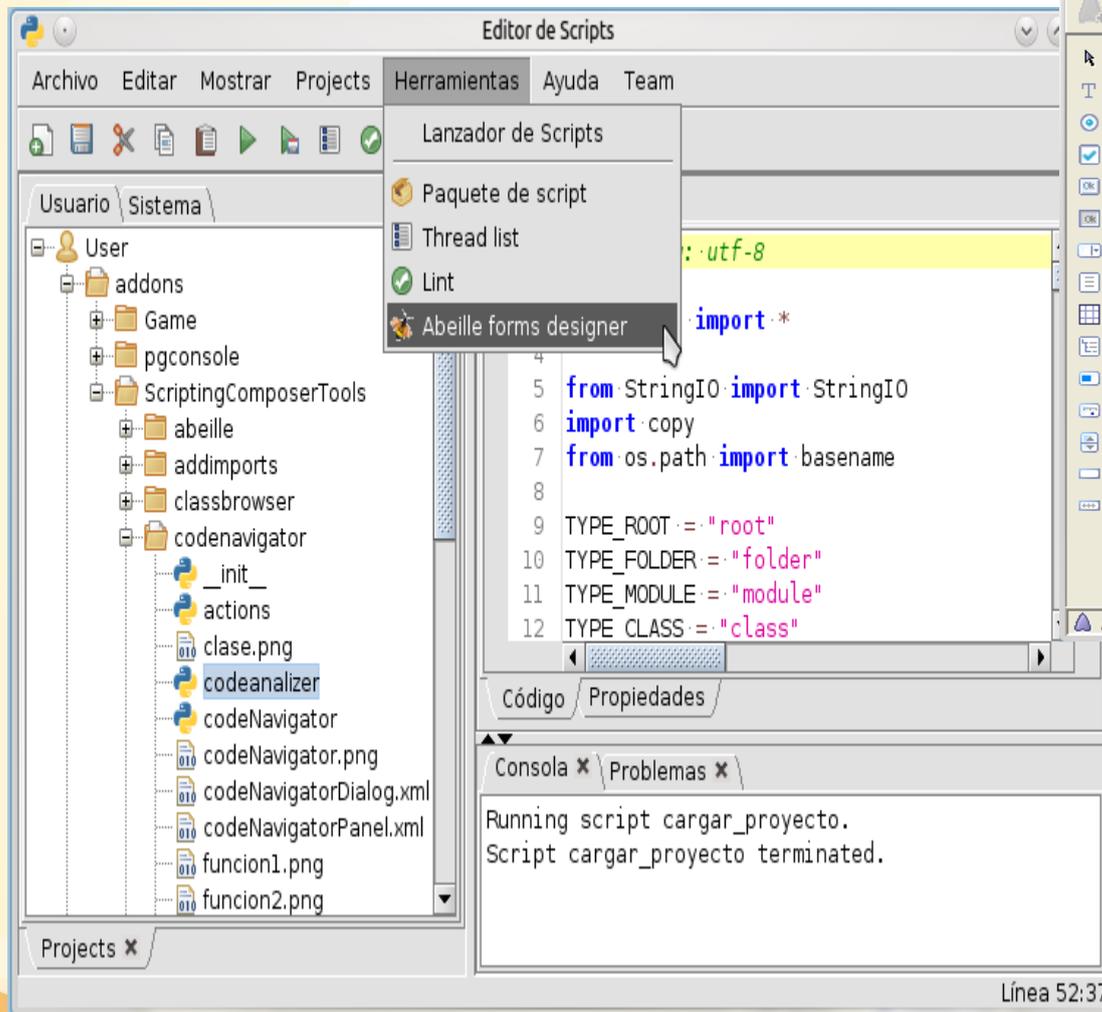
The console at the bottom shows the output of running the script: 'Running script cargar_proyecto. Script cargar_proyecto terminated.'

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

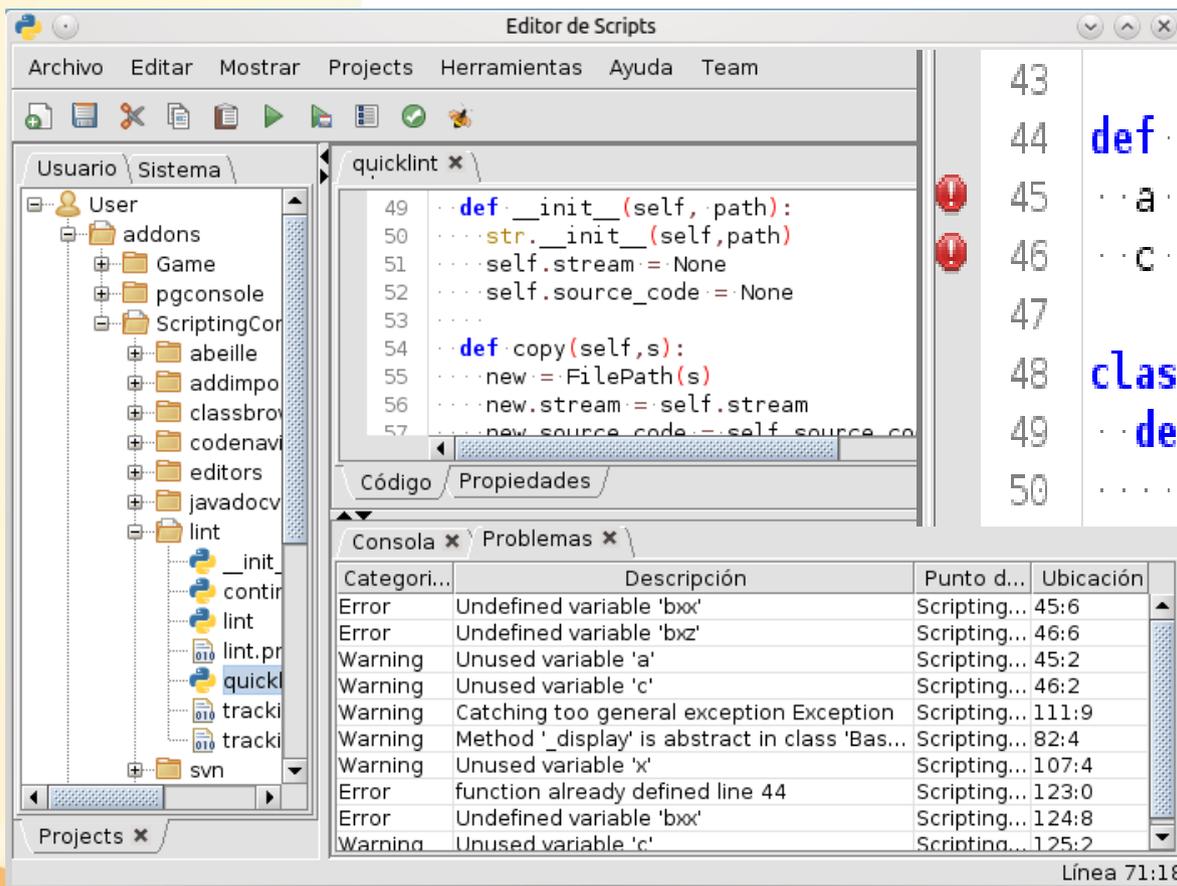
Scripting Composer Tools

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

El entorno de scripting

Funcionalidades

Scripting Composer Tools

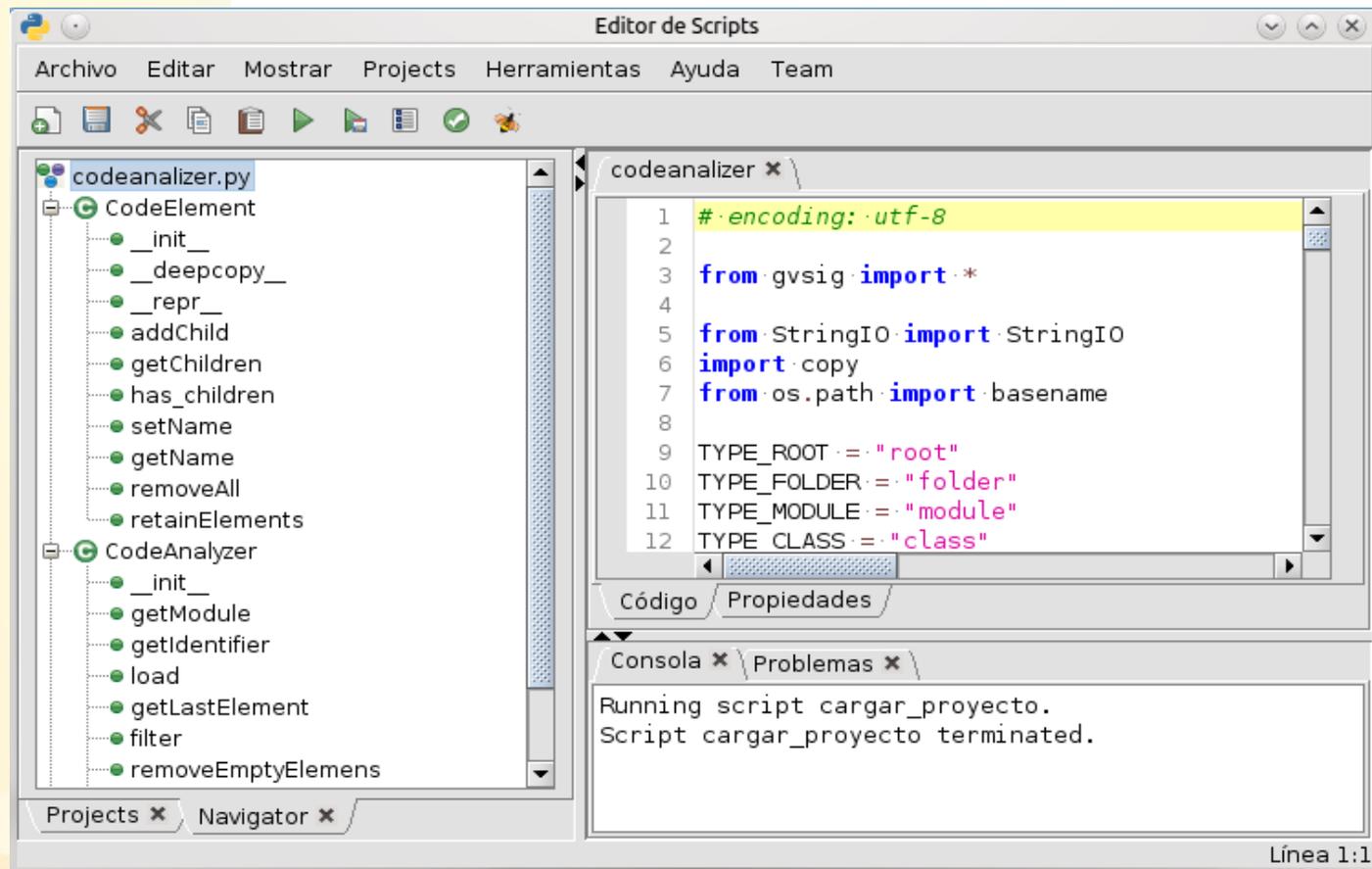


Comprobación de sintaxis bajo petición y mientras escribimos

El entorno de scripting

Funcionalidades

Scripting Composer Tools

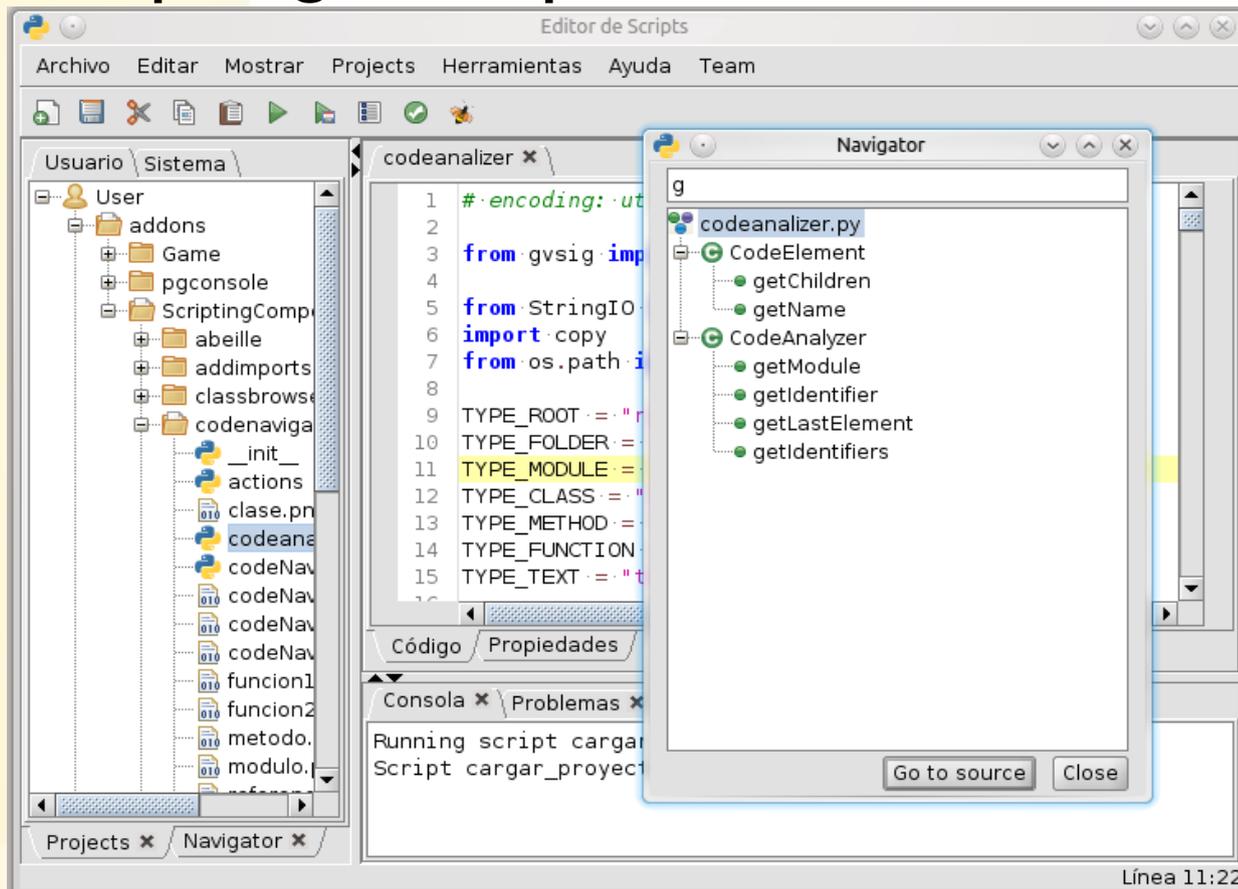


Navegador
de código

El entorno de scripting

Funcionalidades

Scripting Composer Tools

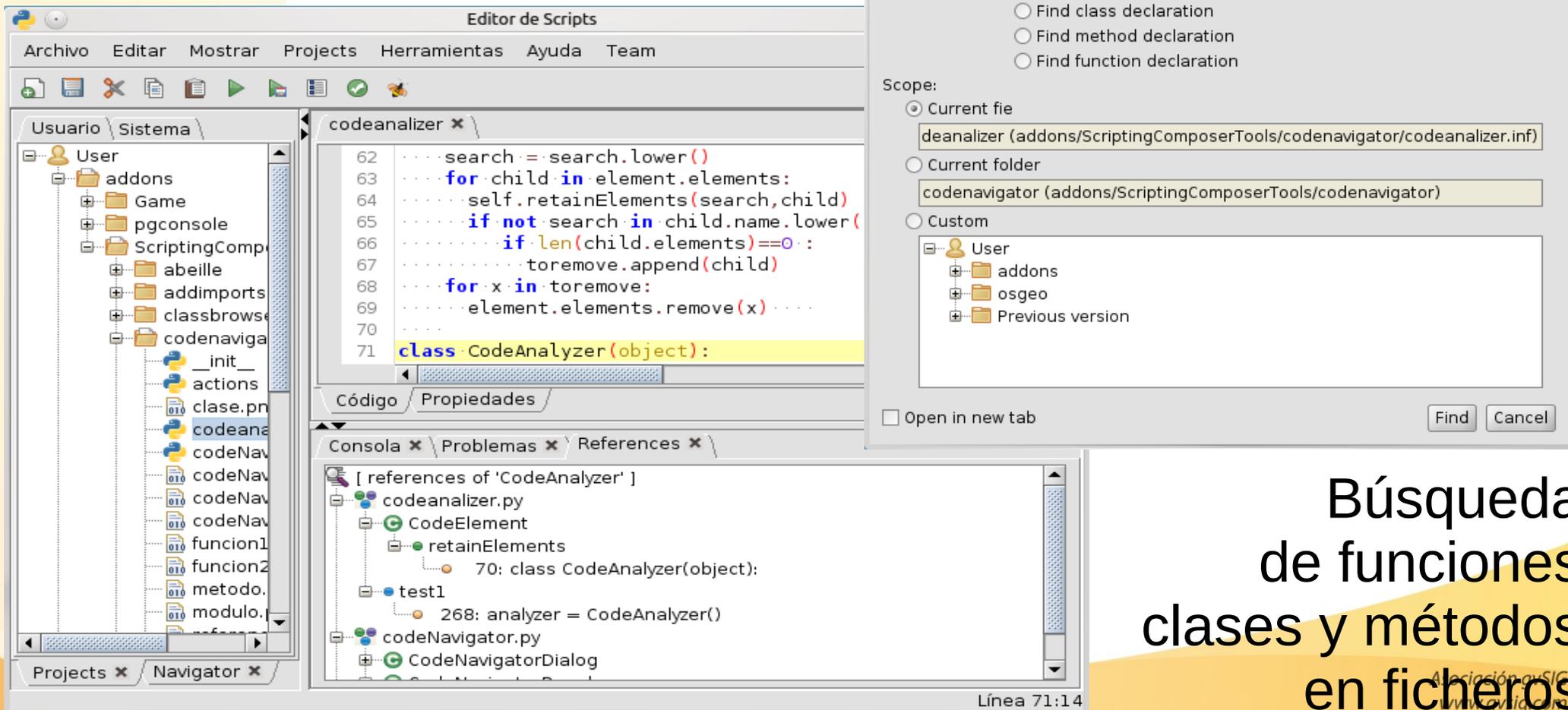


Búsqueda rápida
de funciones
clases y métodos
(en el script corriente)

El entorno de scripting

Funcionalidades

Scripting Composer Tools



The screenshot displays the Scripting Composer Tools interface. The main window is titled 'Editor de Scripts' and contains a code editor with the following Python code:

```
62 search = search.lower()
63 for child in element.elements:
64     self.retainElements(search, child)
65     if not search in child.name.lower():
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):
```

The 'Find references' dialog box is open, showing the search criteria:

- References to: CodeAnalyzer|
- Find text (selected)
- 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).

The 'References' panel shows the following results:

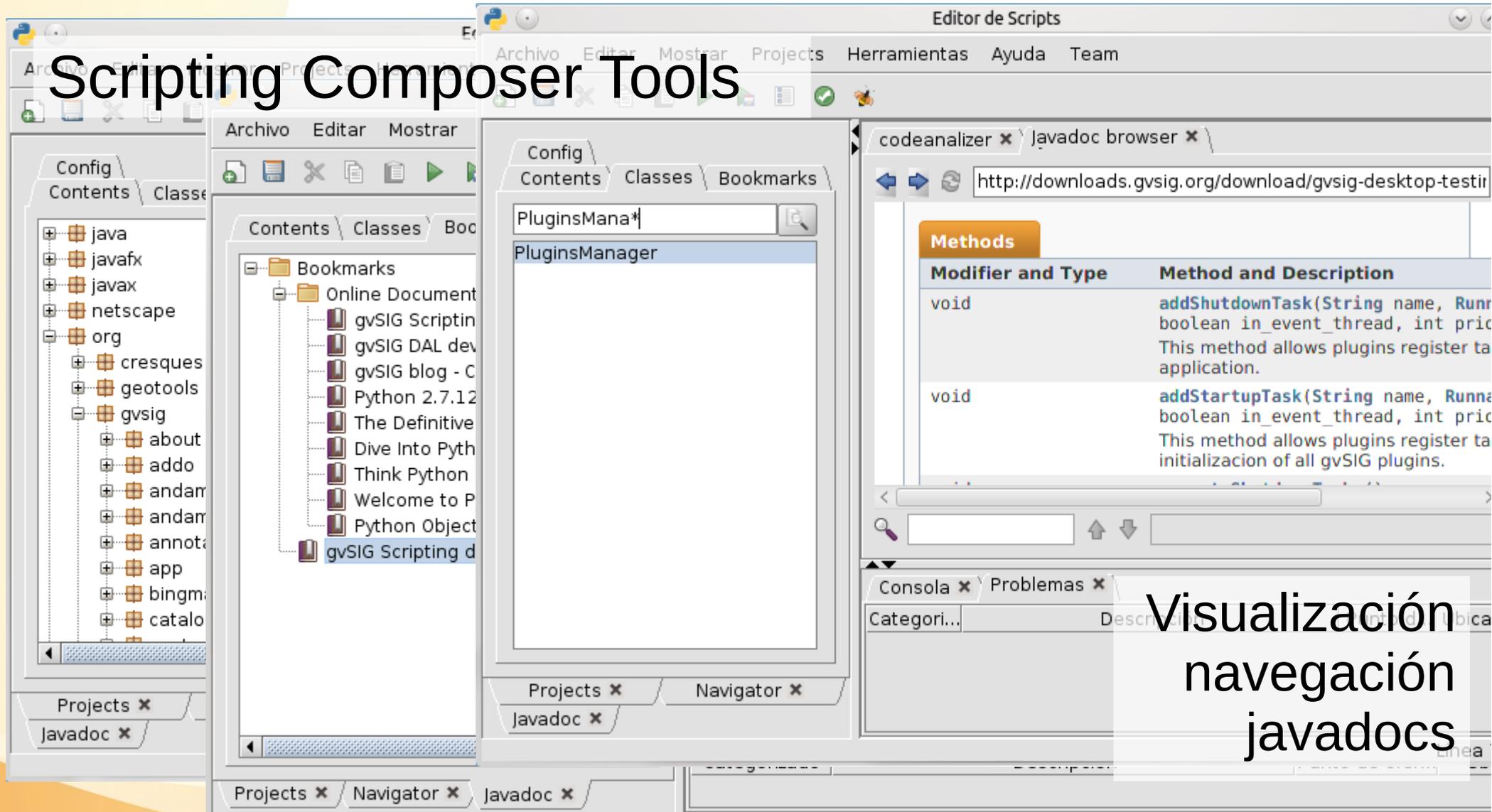
- [references of 'CodeAnalyzer']
- codeanalyzer.py
 - CodeElement
 - retainElements
 - 70: class CodeAnalyzer(object):
 - test1
 - 268: analyzer = CodeAnalyzer()
- codeNavigator.py
 - CodeNavigatorDialog

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

gvSIG Documentation » Documentación de ... »

[previous](#) | [next](#) | [index](#)



Table Of Contents

Documentación de Scripting en gvSIG 2.3

- [Tabla de contenidos](#)
- [Enlaces](#)
- [Índices](#)

Previous topic

[13. Geoprosesos](#)

Next topic

[1. Introducción](#)

This Page

[Show Source](#)

Quick search

Documentación de Scripting en gvSIG 2.3

Tabla de contenidos

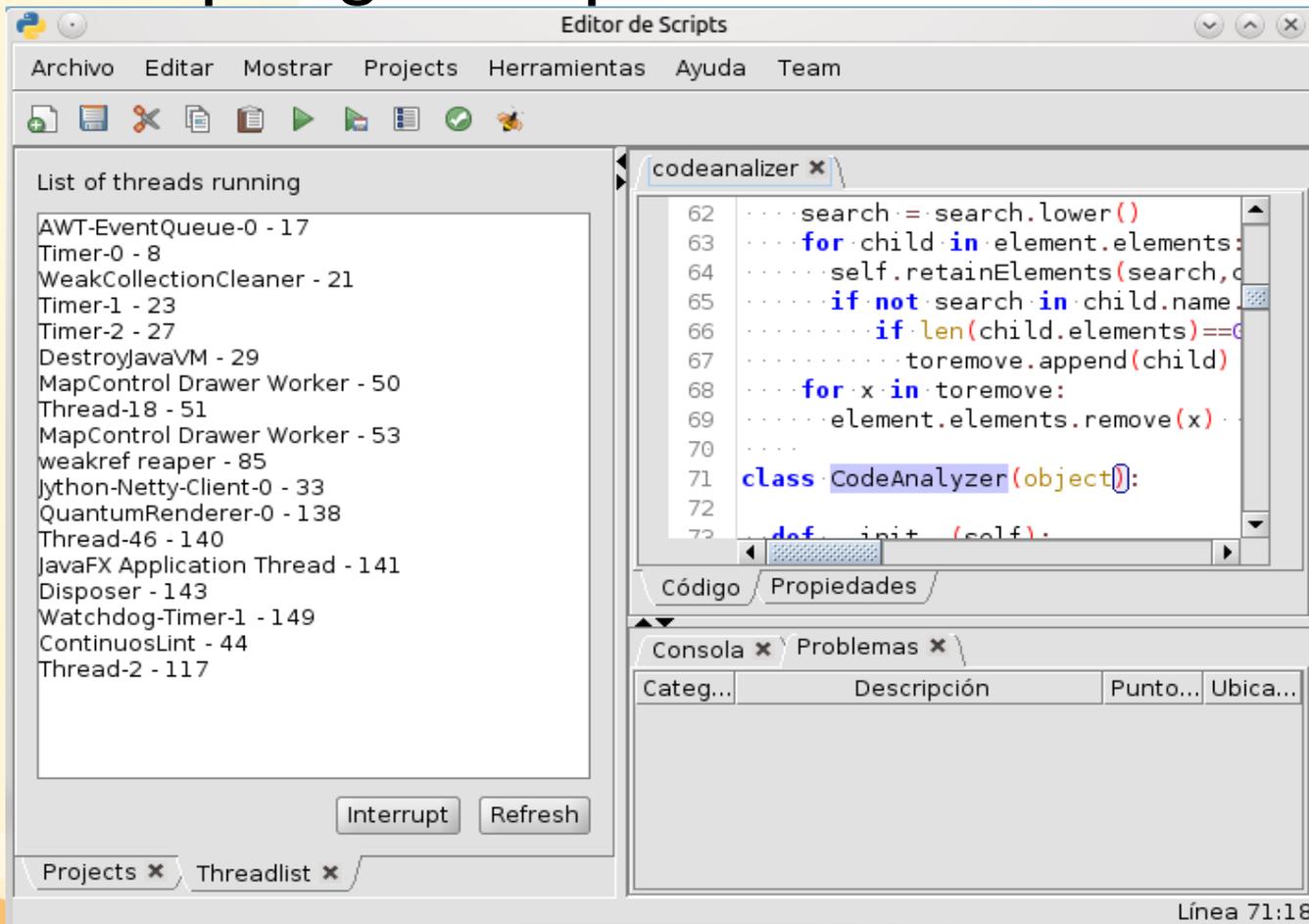
- [1. Introducción](#)
 - [1.1. Qué es el Módulo de Scripting](#)
 - [1.2. Qué necesitas](#)
 - [1.3. Qué nos permite](#)
 - [1.4. Qué lenguaje utilizamos](#)
 - [1.5. Cómo acceder al Módulo de Scripting](#)
 - [1.6. Librerías de gvSIG para Scripting](#)
 - [1.7. Organización de la librería gvsig](#)
 - [1.8. Creación de un script](#)
 - [1.9. Generando paquetes de scripts](#)
- [2. Modelo de objetos simplificado](#)
 - [2.1. Cómo interpretar el diagrama](#)
- [3. Modelo de clases simplificado](#)
 - [3.1. Plugins y acciones](#)
 - [3.2. Proyecto y documentos](#)
 - [3.3. Mapcontrol](#)
 - [3.4. MapContext](#)
 - [3.5. Geometrías](#)

Visualización
navegación
javadocs

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 main area is split into two panes. The left pane, titled 'List of threads running', contains 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 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 shows 'Línea 71:18'.

Visualización
de hilos en
ejecución

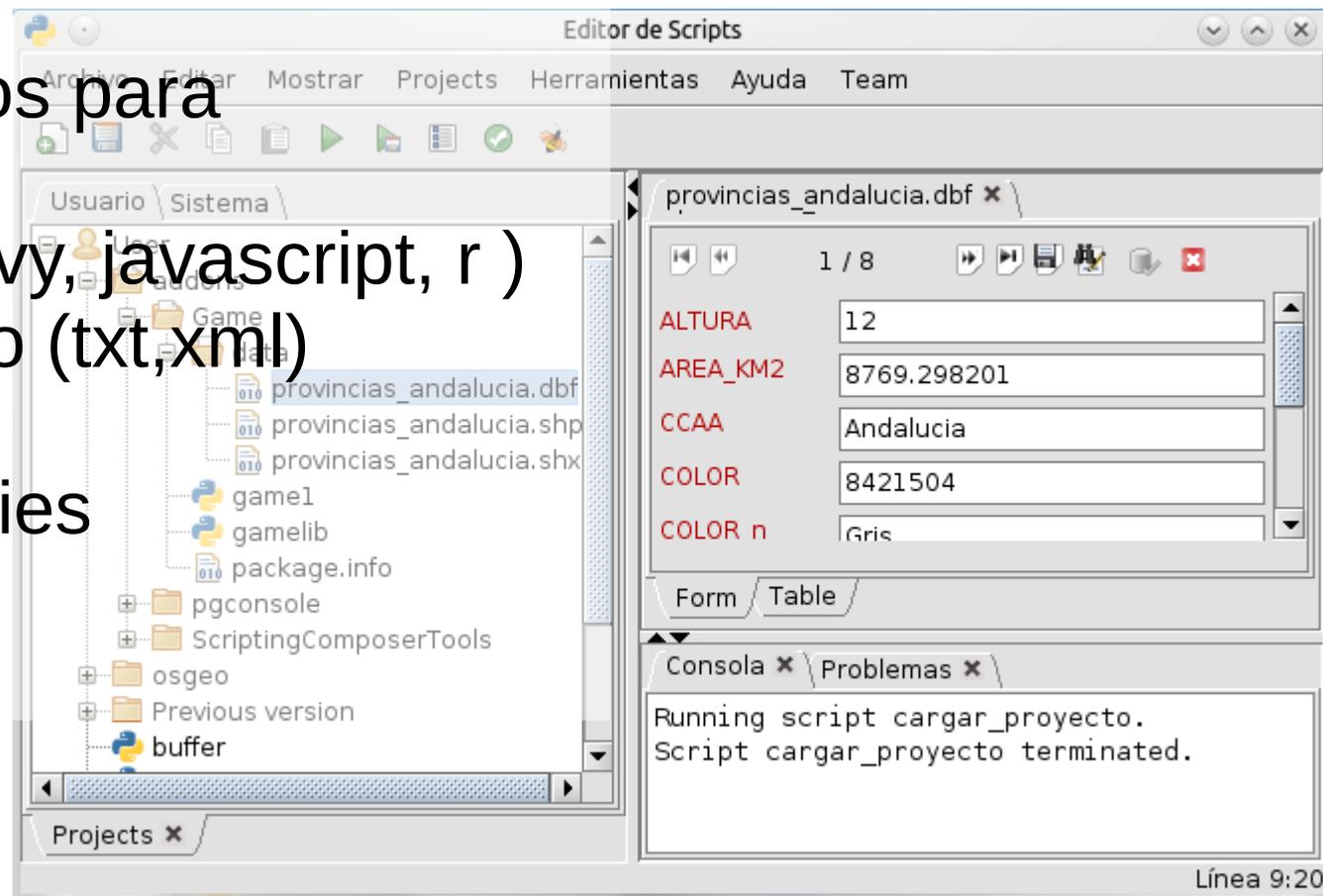
El entorno de scripting

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...

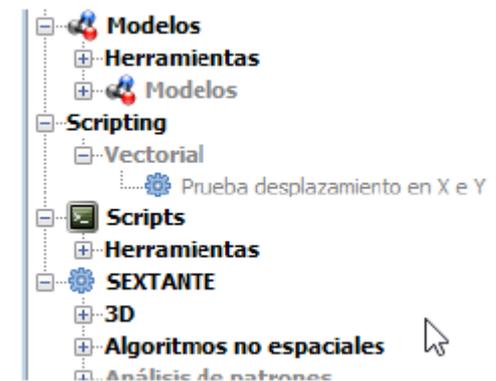
- geopy (geolocalización web)
- gvpy (acceso a la caja de herramientas)
- JOpenDocument (acceso a ODT/ODS)
- CSSBox (navegador web)
- cartodb
- xmltodict
- ogr2ogr

El entorno de scripting

Funcionalidades

Soporte para integrar scripts en:

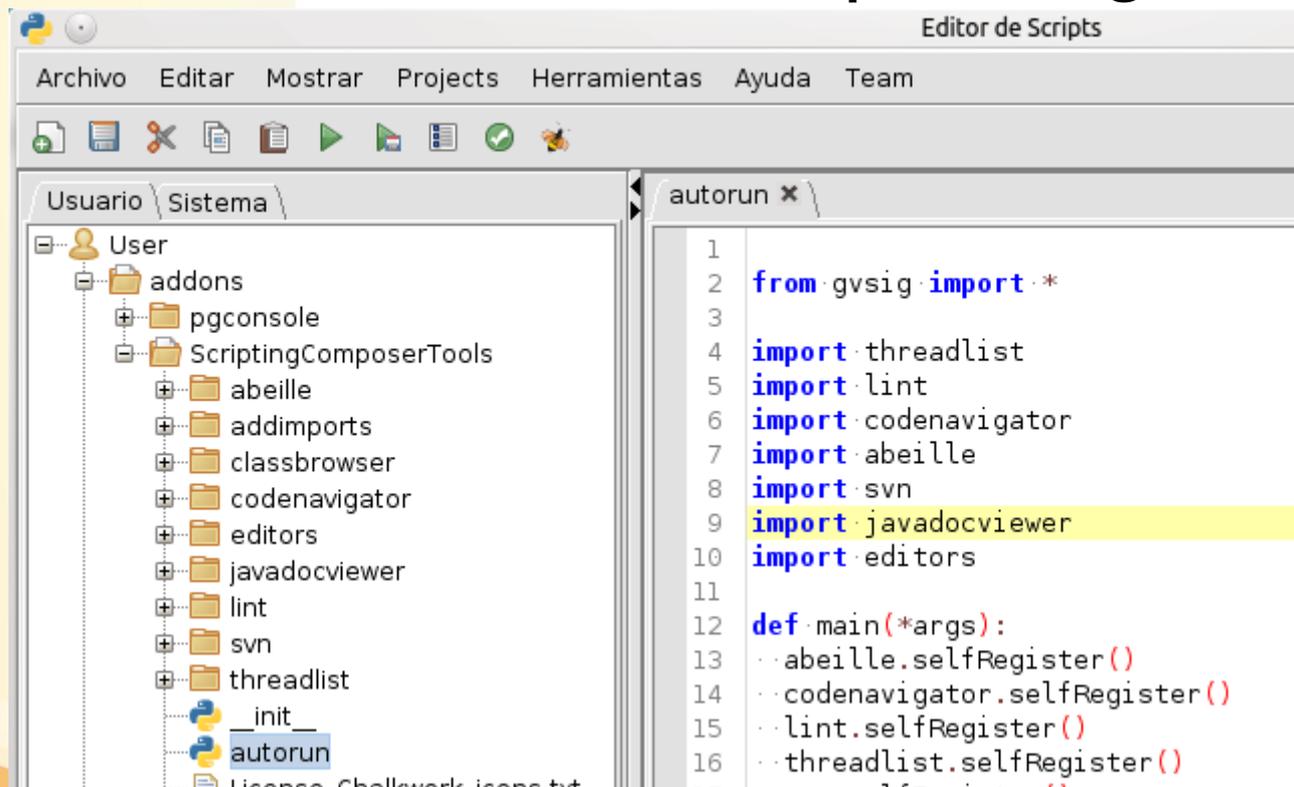
- Menús de gvSIG
- Barra de botones de gvSIG
- Caja de herramientas (Geoprocesos)



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
blog.gvsig.com

Joaquin Jose del Cerro
jjdelcerro@gvsig.com

Óscar Martínez
omartinez@gvsig.com
Twitter: @masquesig