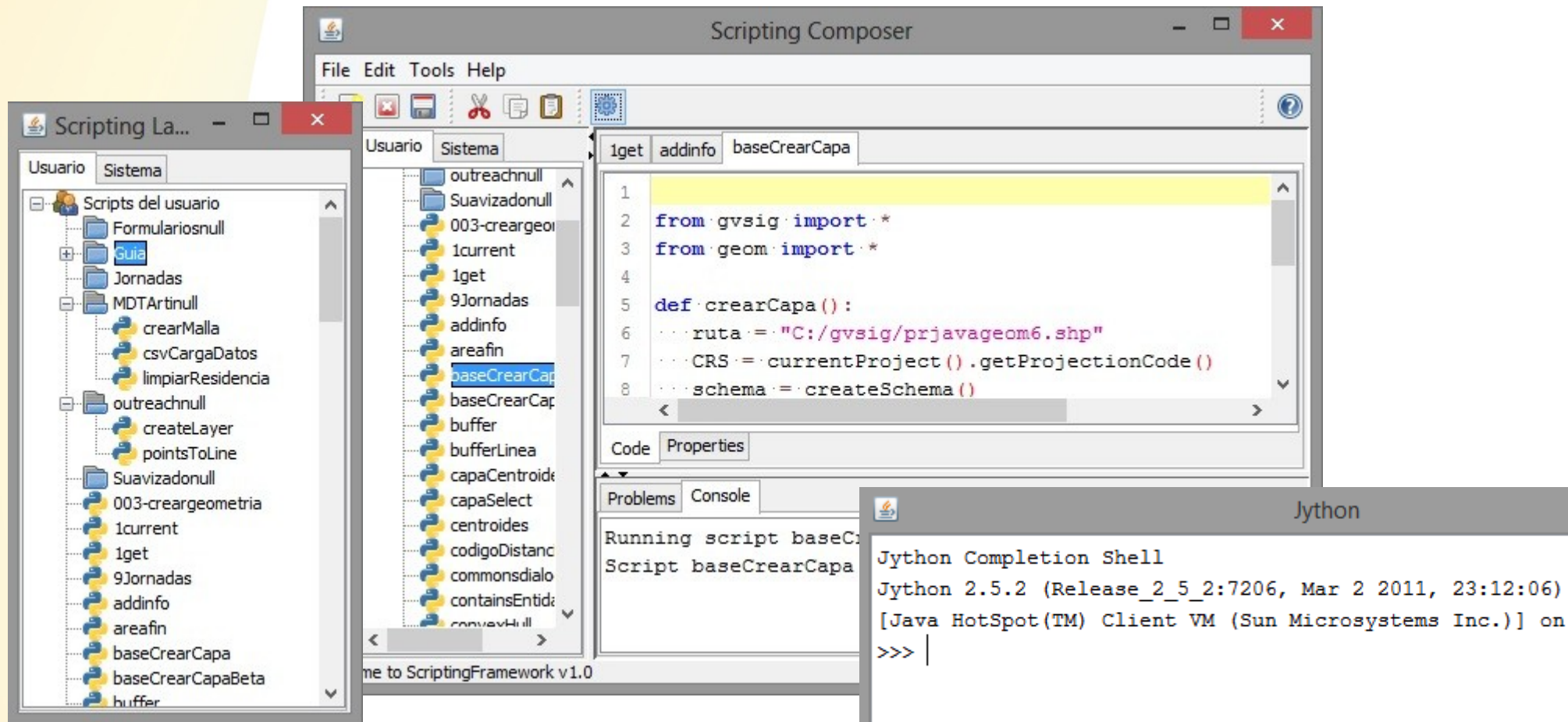




Scripting en gvSIG: Novedades y ejemplos de desarrollo

Módulo de Scripting



The screenshot displays the 'Scripting Composer' window in gvSIG. The left pane shows a tree view of scripts under 'Scripts del usuario'. The main editor shows a script named 'baseCrearCapa' with the following code:

```

1
2 from gvsig import *
3 from geom import *
4
5 def crearCapa():
6     ... ruta = "C:/gvsig/prjavageom6.shp"
7     ... CRS = currentProject().getProjectionCode()
8     ... schema = createSchema()

```

Below the code editor, there are tabs for 'Problems' and 'Console'. The 'Console' tab shows the Jython completion shell output:

```

Jython Completion Shell
Jython 2.5.2 (Release_2_5_2:7206, Mar 2 2011, 23:12:06)
[Java HotSpot(TM) Client VM (Sun Microsystems Inc.)] on
>>>

```

Python + Java = **Jython**

Módulo de Scripting

- Compatibilidad con librerías Java
- ... y Python
- Entorno totalmente integrado
- Base de desarrollo para:
 - ... geoprocesos
 - ... automatización de tareas
 - ... y herramientas
- Script + Interfaz visuales

Módulo de Scripting

Orientado para
usuarios
y
para
desarrolladores

Ejemplos de desarrollo

Intersección

“Es decir elijo la capa [...] pedir que se interseque con otras capas y que me indique cuáles de esos puntos se encuentran dentro de una cuenca hidrográfica, un área protegida, una provincia, entre otros
 ¿existe alguna forma de hacer esto automatizado?”

```
Point info: {u'campo2': u'Venecia', u'campo1': 101L}
- Layer: valencia_index
-- {u'fecha': u'1980-01-01', u'location': u'/cdrom/dat
-- {u'fecha': u'1992-01-01', u'location': u'/cdrom/dat
-- {u'fecha': u'2002-01-01', u'location': u'/cdrom/dat
- Layer: parcelas_Valencia
-- {u'PARCELA': u'03', u'AREA': 327L, u'FECHAALTA': 20
- Layer: manzanas_valencia
-- {u'AREA': 1061L, u'FECHAALTA': 20011119, u'MUNICIP
```

```
1
2 from gvsig import *
3
4 def infoInter(featurePoint, lstLayers):
5     pointValues = featurePoint.getValues()
6     pointValues.pop('GEOMETRY')
7     gfeat = featurePoint.geometry()
8     print "\nPoint info:", pointValues
9     for layer in lstLayers:
10         print "- Layer:", layer.name
11         for pol in layer.features():
12             polValues = pol.getValues()
13             gpol = pol.geometry()
14             polValues.pop('GEOMETRY')
15             if gpol.intersects(gfeat):
16                 print "--", polValues
17
18 def main(*args):
19     #Intersection info
20     lyrPoints = currentLayer()
21
22     #Capas diferentes a la de puntos
23     lstLayers = []
24     for capa in currentView().getLayers():
25         if not capa.name == lyrPoints.name:
26             lstLayers.append(capa)
27
28     #intersecar puntos
29     for point in lyrPoints.features():
30         infoInter(point, lstLayers)
31
```

Librería Python: pygeoj

PyGeoj

PyGeoj is a simple Python GeoJSON file reader and writer intended for high-level objects with convenience methods, so the user does not have to

Platforms

So far only tested on Python version 2.x.

<https://github.com/karimbahgat/PyGeoj>

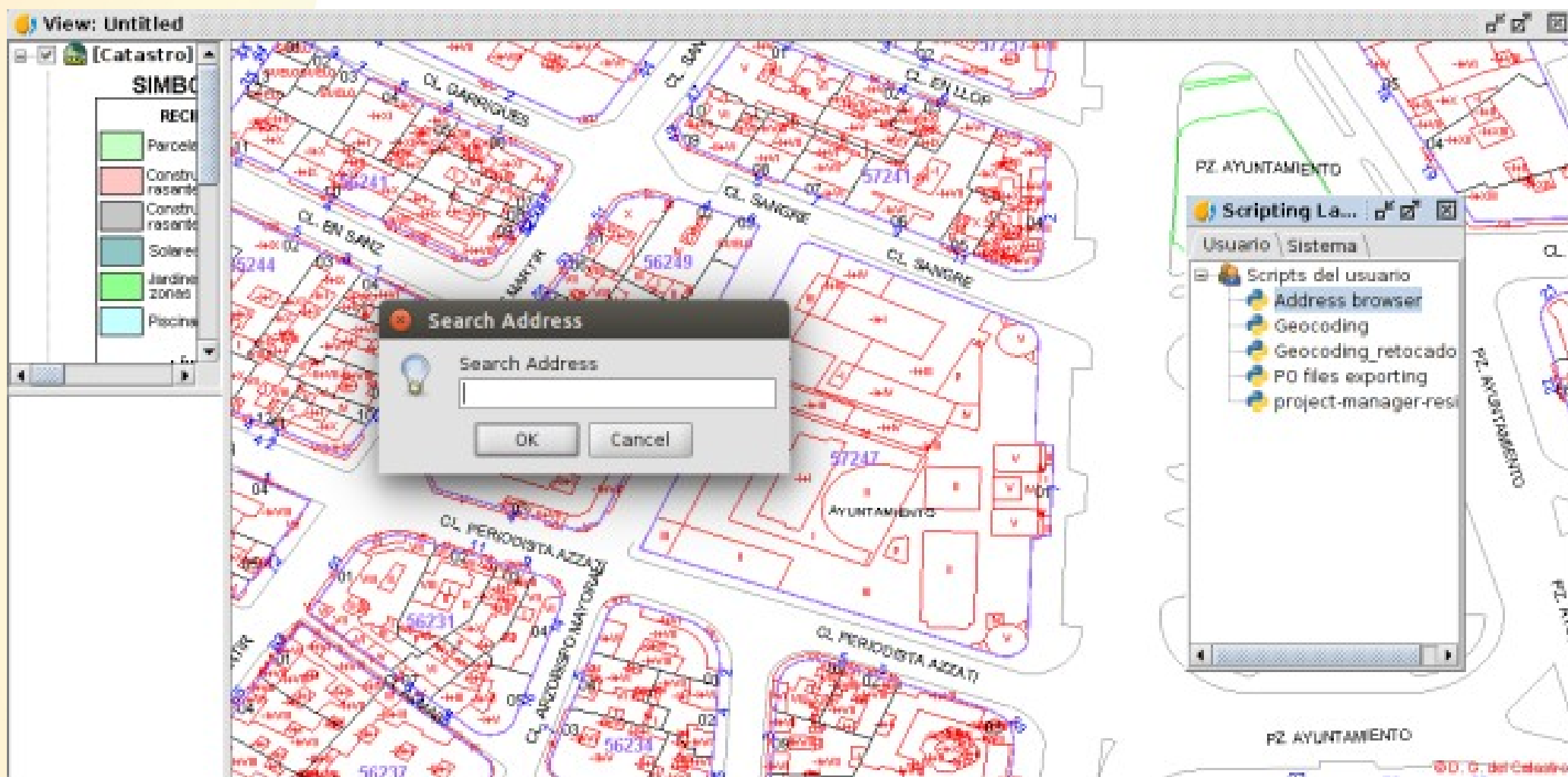
Dependencies

Pure Python, no dependencies.

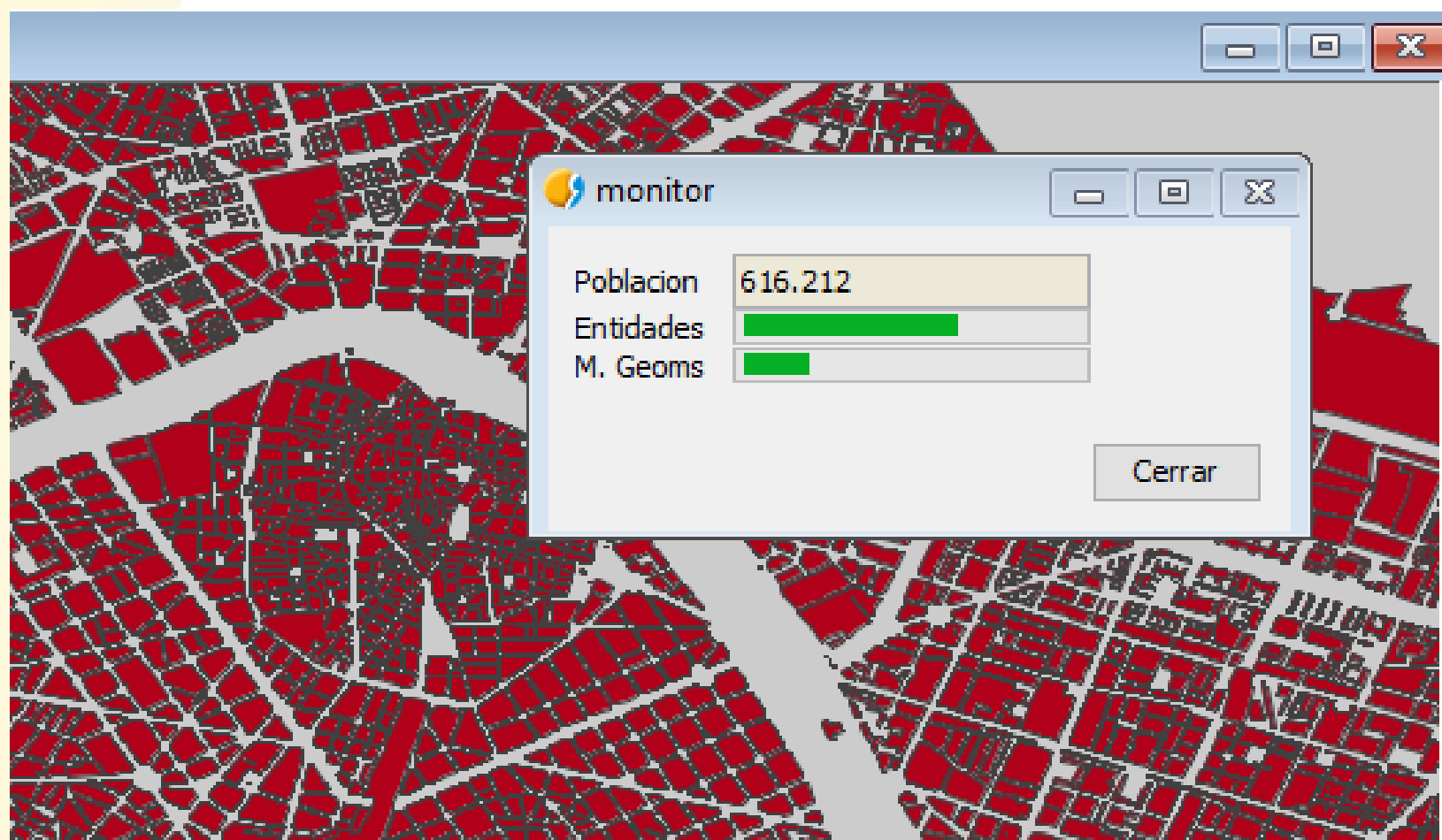
Installing it

PyGeoj is installed with pip from the commandline:

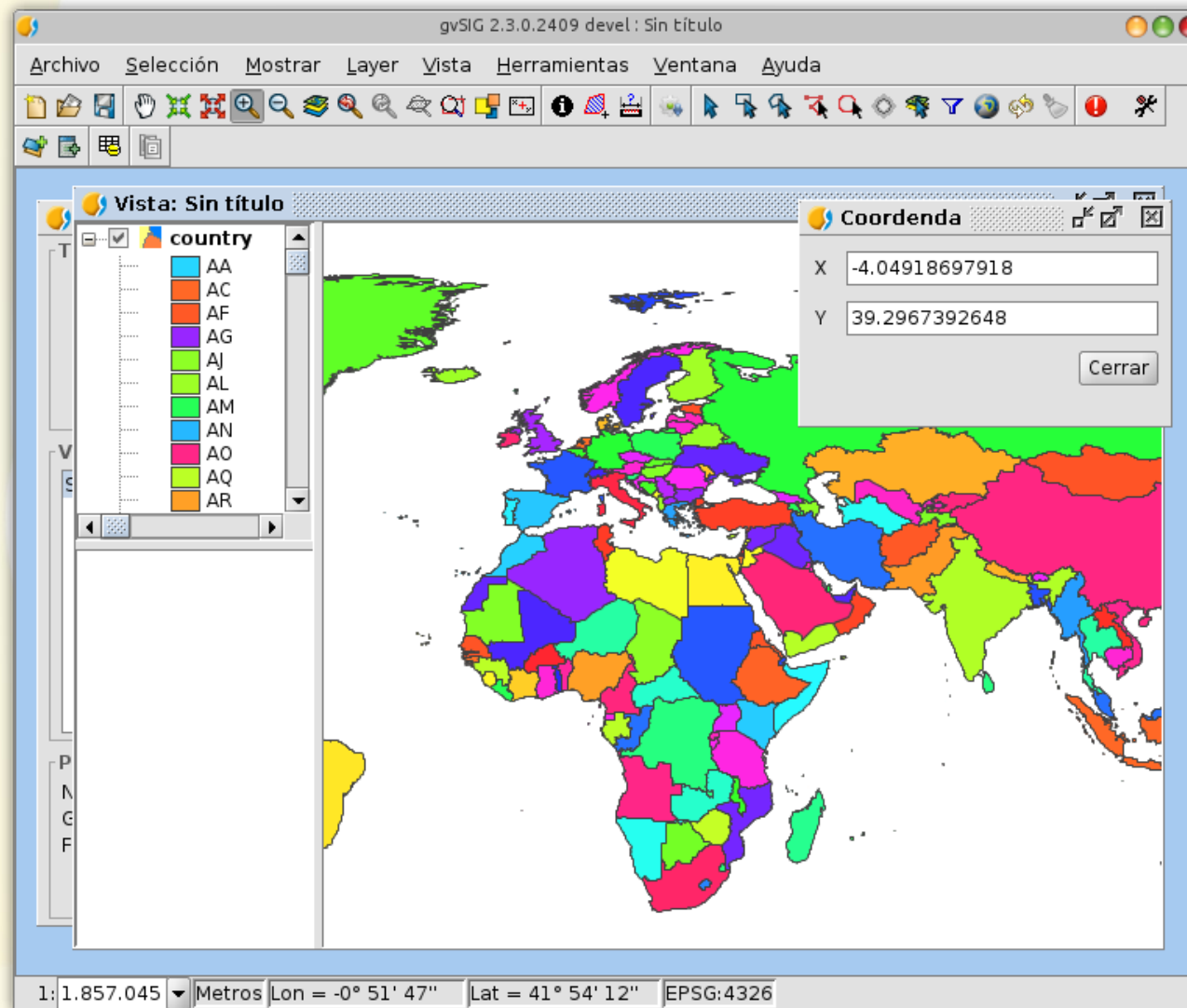
Librería Python: geopy



Monitor



Coordenadas

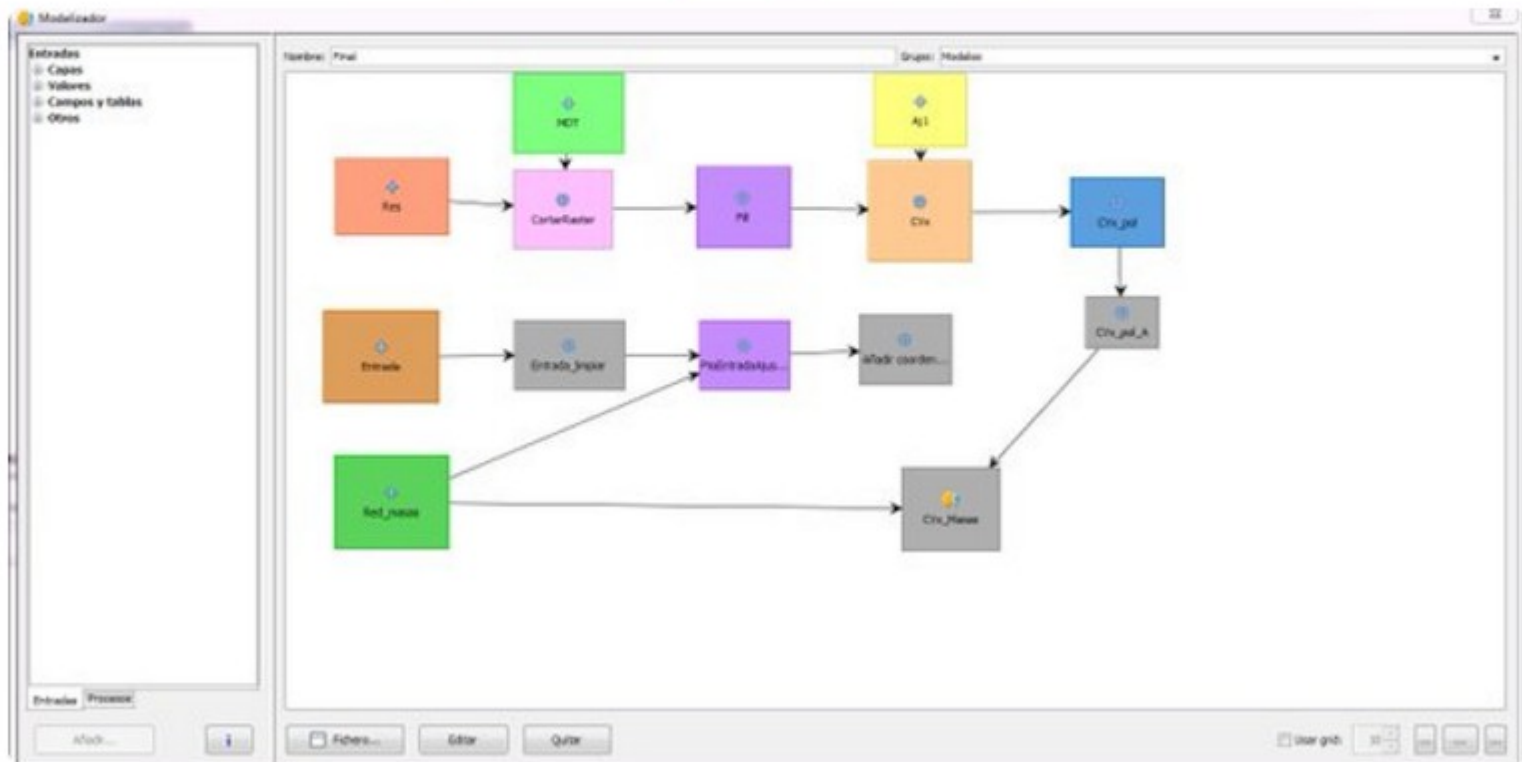


Ideya



ideyared @ideyared · 7 oct.

Probando la potencia del Model Builder de @gvsig en la aplicación de proyectos hídricos. @AlvaroAnguix @masquesig



CartoDB

```
1
2 from gvsig import *
3 import cartodb1
4
5 def main(*args):
6     ....API_KEY = '****'
7     ....cartodb_domain = 'user'
8     ....cl = CartoDBAPIKey(API_KEY, cartodb_domain)
9     ....print(cl.sql('select *.from mytable'))
10
```



CARTO^{DB}
Geospatial on the cloud

jOpenDocument

fillingTest1.ods - LibreOffice Calc

Archivo Editar Ver Insertar Formato Herramientas Datos Ventana Ayuda

Verdana 10

D8 Valencia

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3											
4											
5											
6											
7											
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											
21											
22											
23											
24											
25											
26											
27											
28											
29											
30											
31											
32											
33											
34											
35											
36											
37											

Presupuesto parcela n. 1

Topografía. S.A
Av. Blasco Ibañez
Valencia
España

\$3,095.00

Total due by October 5, 2015

Recibo
Tech. N.A
Av. Madrid
Barcelona, España

Envío
Tech. N.A
Av. Madrid
Barcelona, España

Pedido n. 01

DESCRIPTION	QUANTITY	PRICE	TOTAL
Área de parcela Superficie	123	\$25.00	\$3,075.00
Personal	10	\$2.00	\$20.00
Sub Total		\$3,095.00	
Tax		\$0.00	
Free		\$0.00	
Discount		\$0.00	
Total		\$3,095.00	

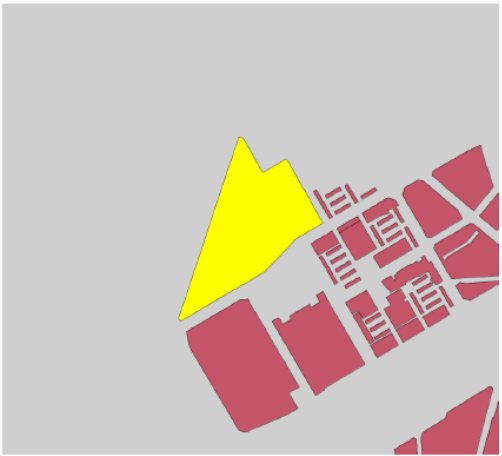
test_parcela00.fodt - LibreOffice Writer

Archivo Editar Ver Insertar Formato Tabla Herramientas jOpenDocument Ventana Ayuda

Encabezado 1 Arial 18,2

Informe parcelario

Hoja	YJ2743H
Area	76368
Coord.X	724568
Coord.Y	437396



Imagen

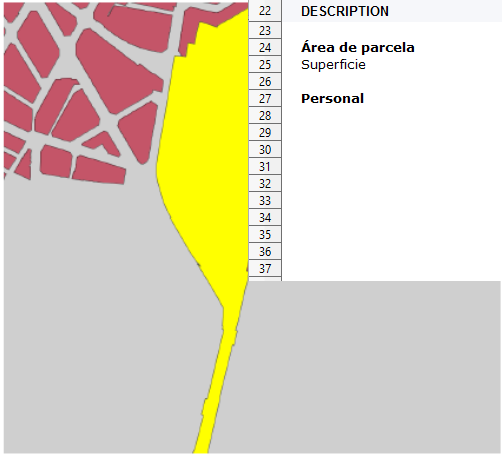


test_parcela01.fodt - LibreOffice Writer

Archivo Editar Ver Insertar Formato Tabla Herramientas jOpenDocument Ventana Ayuda

Encabezado 1 Arial 18,2

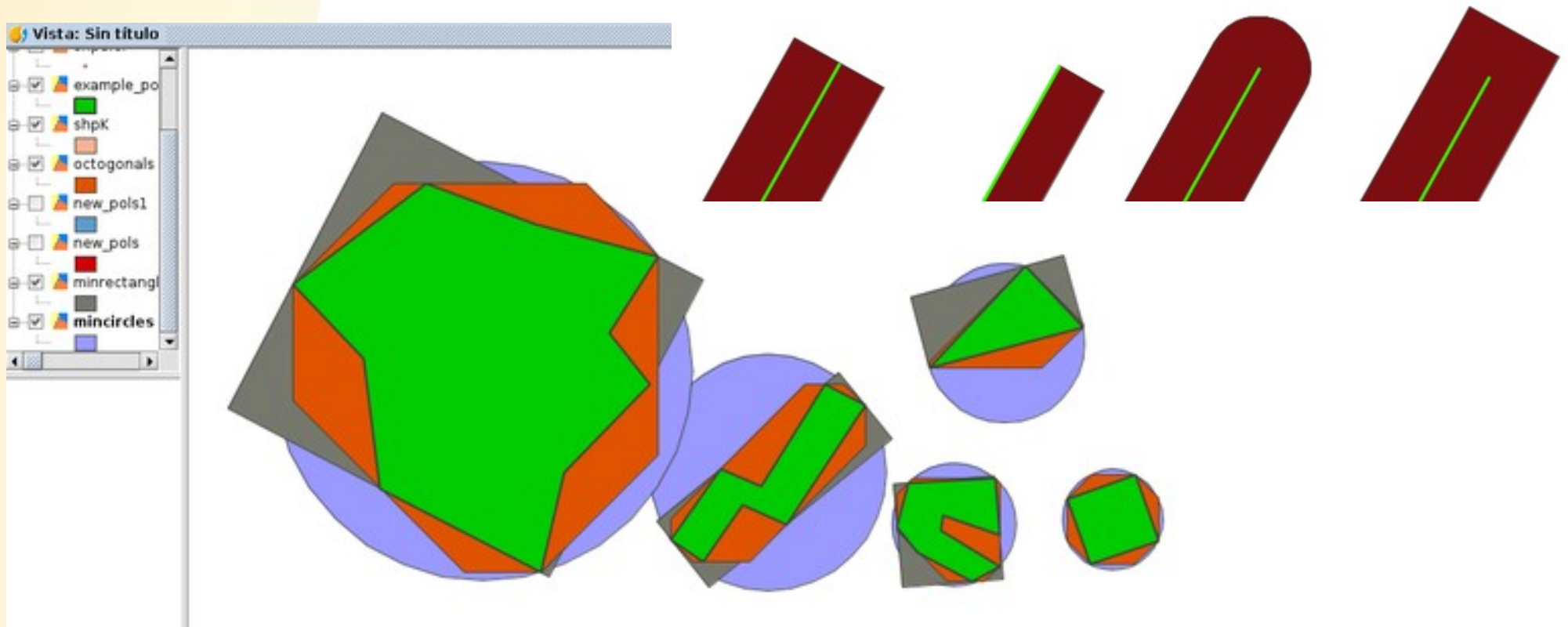
Informe parcela



Imagen



Java Topology Suite (JTS)



```
... import com.vividsolutions.jts.operation.distance.DistanceOp as DistanceOp
... d = DistanceOp(l1, l2)
... nearestPoints = d.nearestPoints()
... print nearestPoints
... print "Punto de geom. 1: ", nearestPoints[0]
... print "Punto de geom. 2: ", nearestPoints[1]
```


jgrasstools

```
1
2 from gvsig import *
3 from com.vividsolutions.jts.io import WKTRReader
4
5 from java.util import List
6
7 from org.geotools.data.simple import SimpleFeatureCollection
8 from org.geotools.feature import DefaultFeatureCollection
9 from org.geotools.feature.simple import SimpleFeatureBuilder
10 from org.geotools.feature.simple import SimpleFeatureTypeBuilder
11 from org.geotools.referencing.crs import DefaultGeographicCRS
12 from org.jgrasstools.gears.modules.v.smoothing import OmsLineSmootherJaitools
13 from org.jgrasstools.gears.modules.v.smoothing import OmsLineSmootherMcMaster
14 from org.jgrasstools.gears.utils.features import FeatureUtilities
15 from org.opengis.feature.simple import SimpleFeature
16 from org.opengis.feature.simple import SimpleFeatureType
17
18 from com.vividsolutions.jts.geom import Geometry
19 from com.vividsolutions.jts.geom import LineString
20
21 from java.lang import Integer
22
23 def main(*args):
24     ... #https://code.google.com/p/jgrasstools/source/browse/jgrassgears/src/test/java/org
25
26     ... print "Test JGrassTools: TestLineSmootherJaitools\n"
27     ... b = SimpleFeatureTypeBuilder()
```

Code Properties

Problems Console

Running script test_jgrasstools_2.

Test JGrassTools: TestLineSmootherJaitools

Collecting geometries...

Smoothing features...

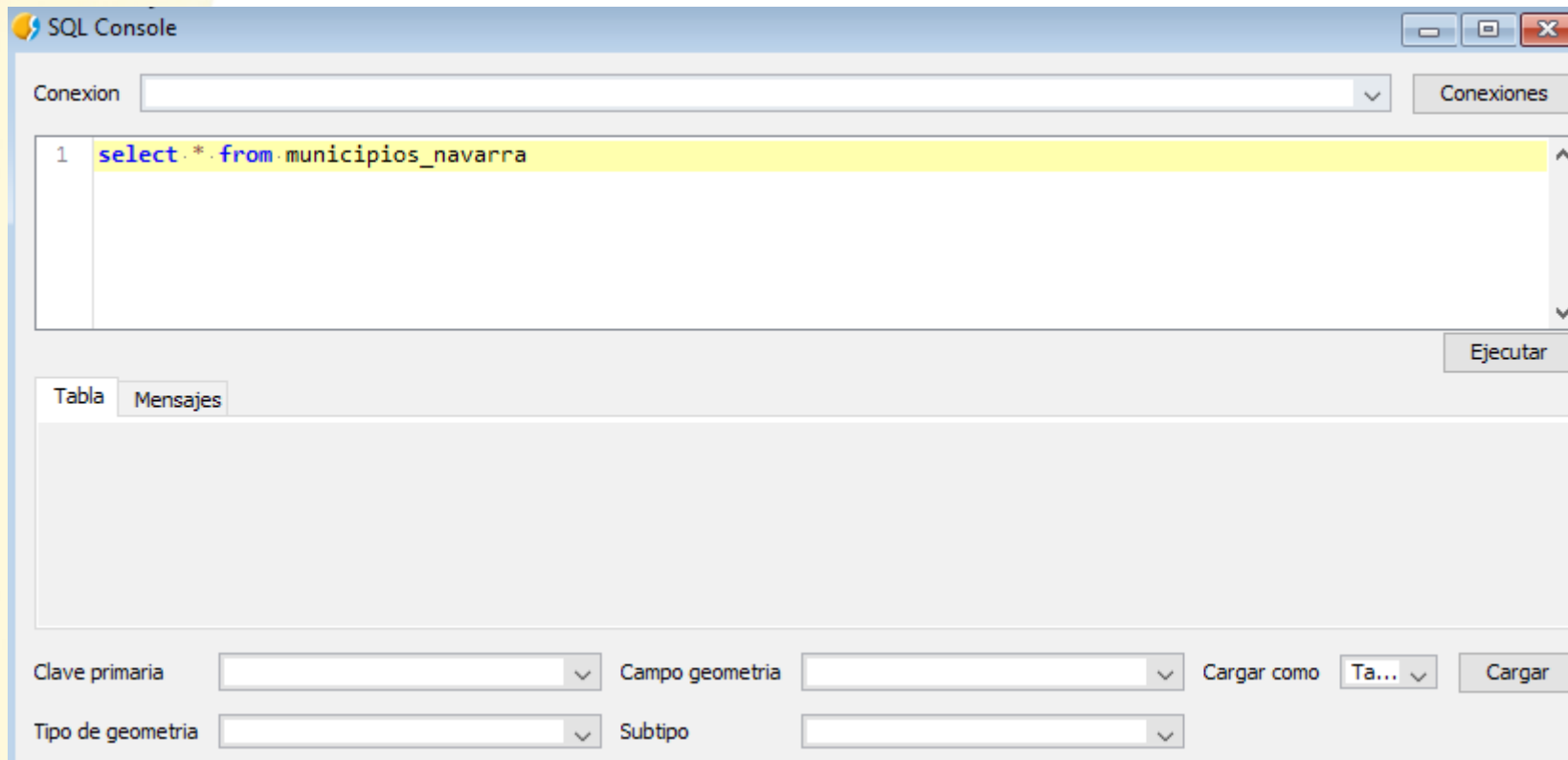
100%...

Finished.

New geom: LINESTRING (0 0, 0.0342935528120713 0.0342935528120713, 0.12620027434

10

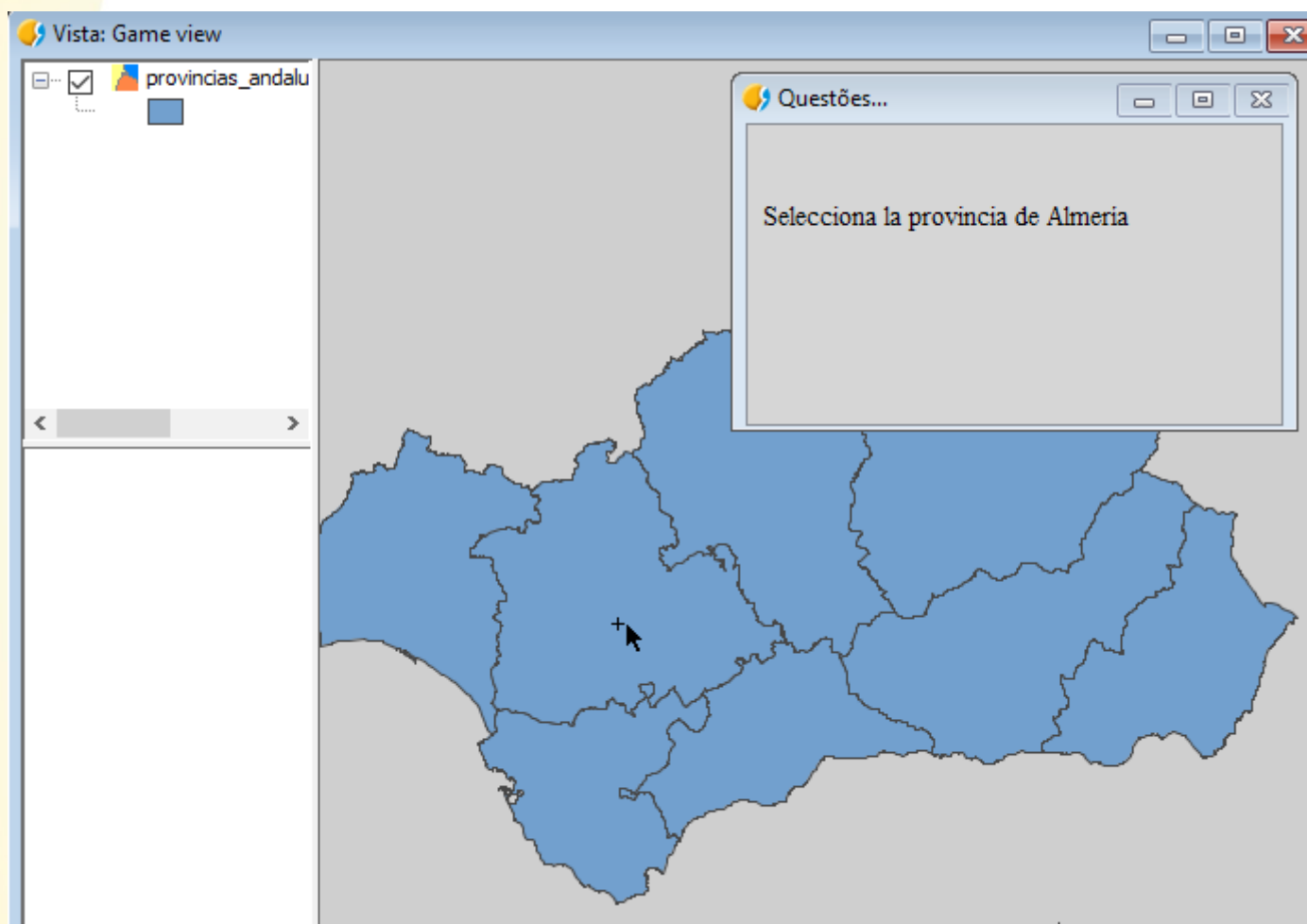
SQL Console



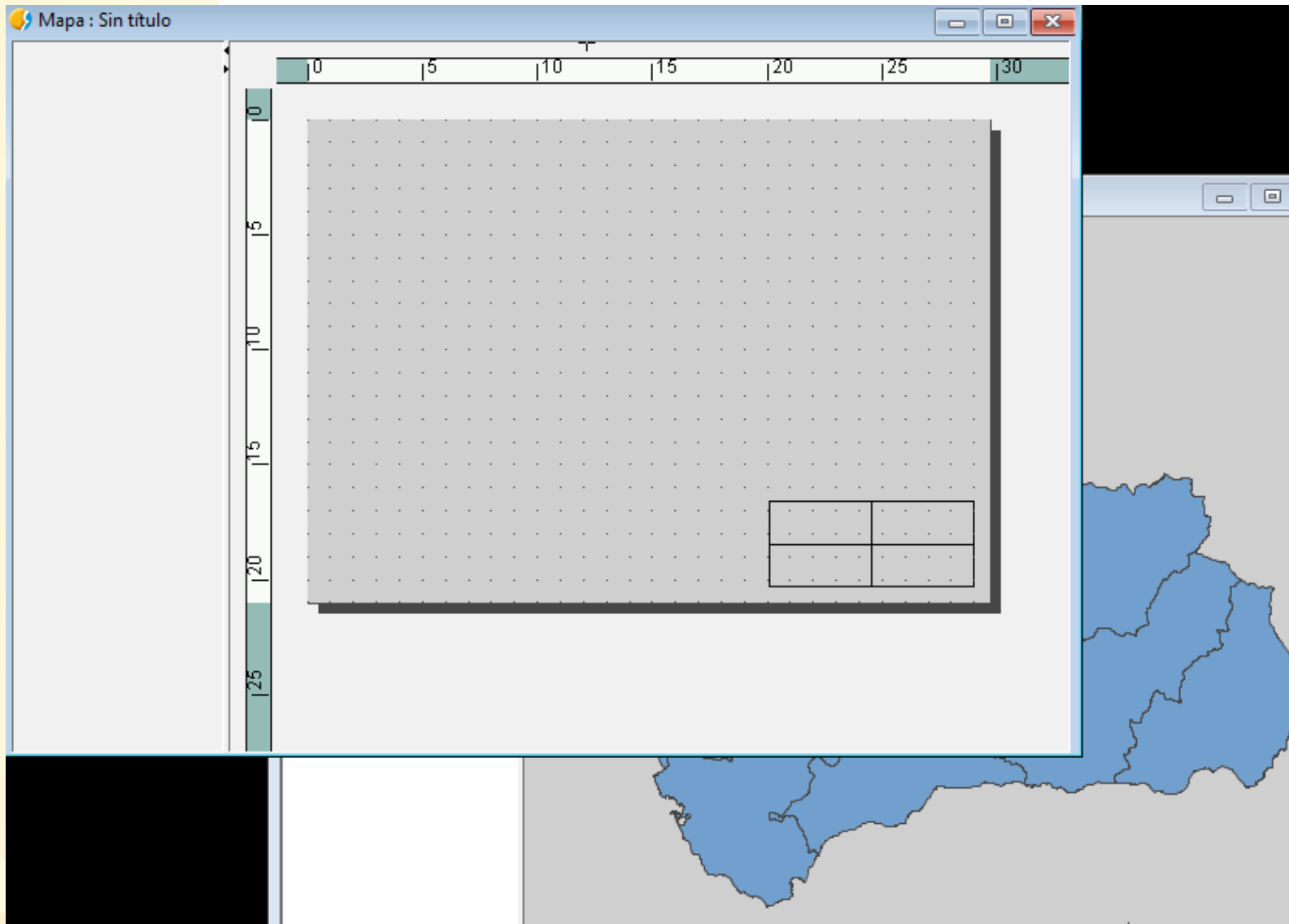
The screenshot shows the SQL Console window with the following elements:

- Conexion:** A dropdown menu for selecting a database connection.
- Conexiones:** A button to manage database connections.
- SQL Editor:** A text area containing the SQL query: `1 select.*.from:municipios_navarra`.
- Ejecutar:** A button to execute the SQL query.
- Tabla / Mensajes:** A tabbed interface for viewing query results or messages.
- Clave primaria:** A dropdown menu for selecting the primary key.
- Campo geometria:** A dropdown menu for selecting the geometry field.
- Cargar como:** A dropdown menu for selecting the load format, currently set to "Ta...".
- Cargar:** A button to load the data.
- Tipo de geometria:** A dropdown menu for selecting the geometry type.
- Subtipo:** A dropdown menu for selecting the geometry subtype.

Game



Personalizar mapa



Georeferenciar fotografías

```
..def.getLatitude(self, defaultValue=None):  
...return self.__values.get("GPS.Latitude",defaultValue)  
...  
..def.getLatitudeRef(self, defaultValue=None):  
...return self.__values.get("GPS.Latitude.Ref",defaultValue)  
...  
..def.getLongitude(self, defaultValue=None):  
...return self.__values.get("GPS.Longitude",defaultValue)  
...  
..def.getLongitudeRef(self, defaultValue=None):  
...return self.__values.get("GPS.Longitude.Ref",defaultValue)  
...  
..def.getAltitude(self, defaultValue=None):  
...return self.__values.get("GPS.Altitude",defaultValue)  
...  
..def.getAltitudeRef(self, defaultValue=None):  
...return self.__values.get("GPS.Altitude.Ref",defaultValue)  
.....  
..def.getDatum(self, defaultValue=None):  
...return self.__values.get("GPS.Map.Datum",defaultValue)  
.....  
..def.getDate(self, defaultValue=None):  
...return self.__values.get("GPS.Date.Stamp",defaultValue)  
.....  
..def.getTime(self, defaultValue=None):  
...return self.__values.get("GPS.Time-Stam",defaultValue)
```

Scripts de alumnos del MOOC

Enseñanza de la geografía política mediante ejercicios y juegos
(Manuel Maria Alvarez)

Column Statistics
(Piotr Pachól)

Importador / Exportador de geojson a shapes
(Jose Alberto Gonzalez y Francisco Puga)

Crear copia de capa con los campos deseados y en orden
(Jose Luis Lopez Janas)

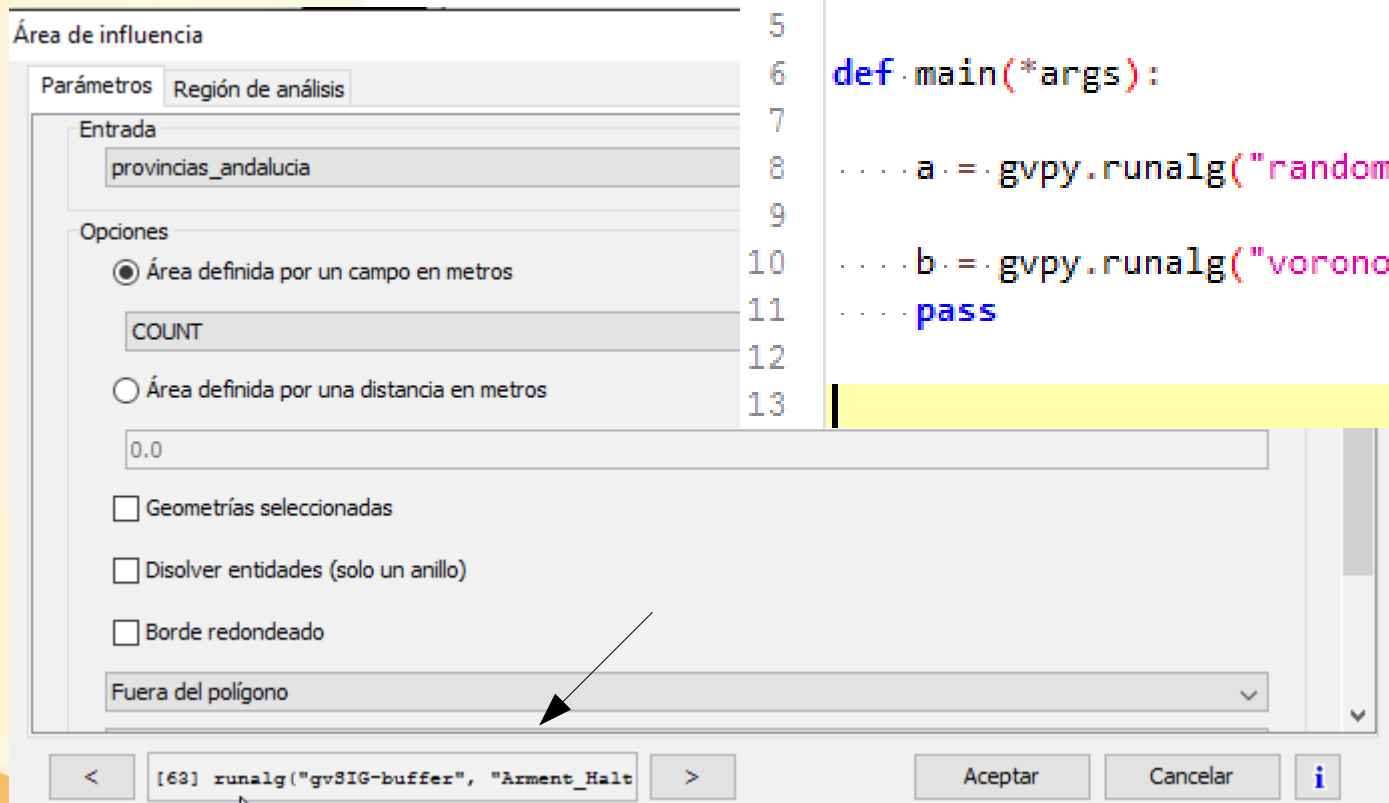
**Reporte de viaje del servicio Sub-urbano Asunción-Ypacarai
del FCPPCAL y ubicación en el tiempo**
(Sergio Spiridonoff Reyes)

Sinuosity Index Schumm classification
(Carles Clua Millan)

Novedades

Novedades

Soporte para invocar geoprocesos



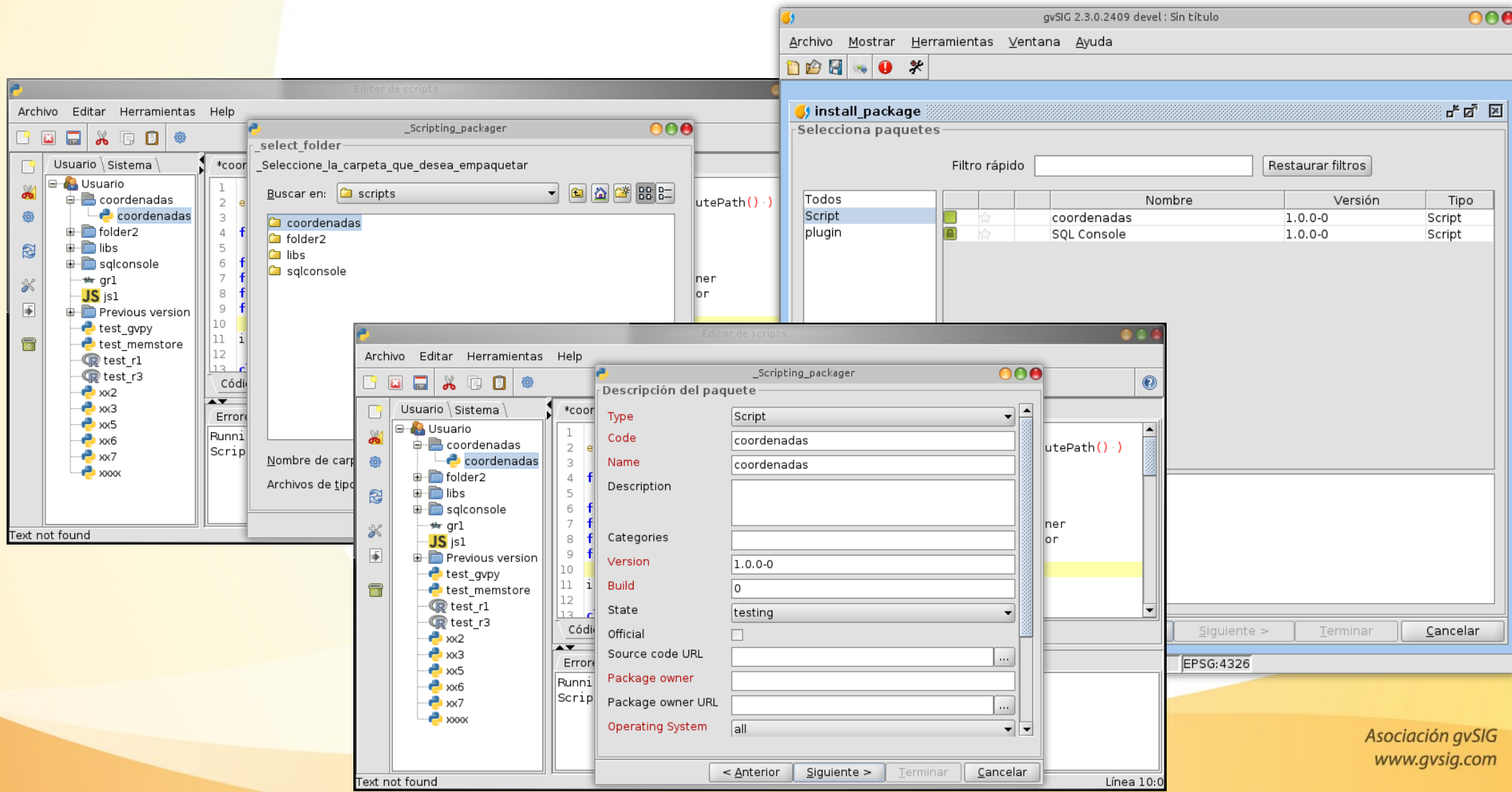
Novedades

Actualización a Jython 2.7 (antes 2.5)

- Produce mejoras de rendimiento
- Mayor compatibilidad con librerías Python

Novedades

Crear paquetes de instalación (addons)



gvSIG 2.3.0.2409 devel: Sin título

Archivo Mostrar Herramientas Ventana Ayuda

install_package
-Selecciona paquetes

Filtro rápido Restaurar filtros

	Nombre	Versión	Tipo
Todos			
Script	coordenadas	1.0.0-0	Script
plugin	SQL Console	1.0.0-0	Script

Siguiente > Terminar Cancelar

EPSG:4326

_Scripting_packager
Descripción del paquete

Type	Script
Code	coordenadas
Name	coordenadas
Description	
Categories	
Version	1.0.0-0
Build	0
State	testing
Official	☐
Source code URL	
Package owner	
Package owner URL	
Operating System	all

< Anterior Siguiente > Terminar Cancelar

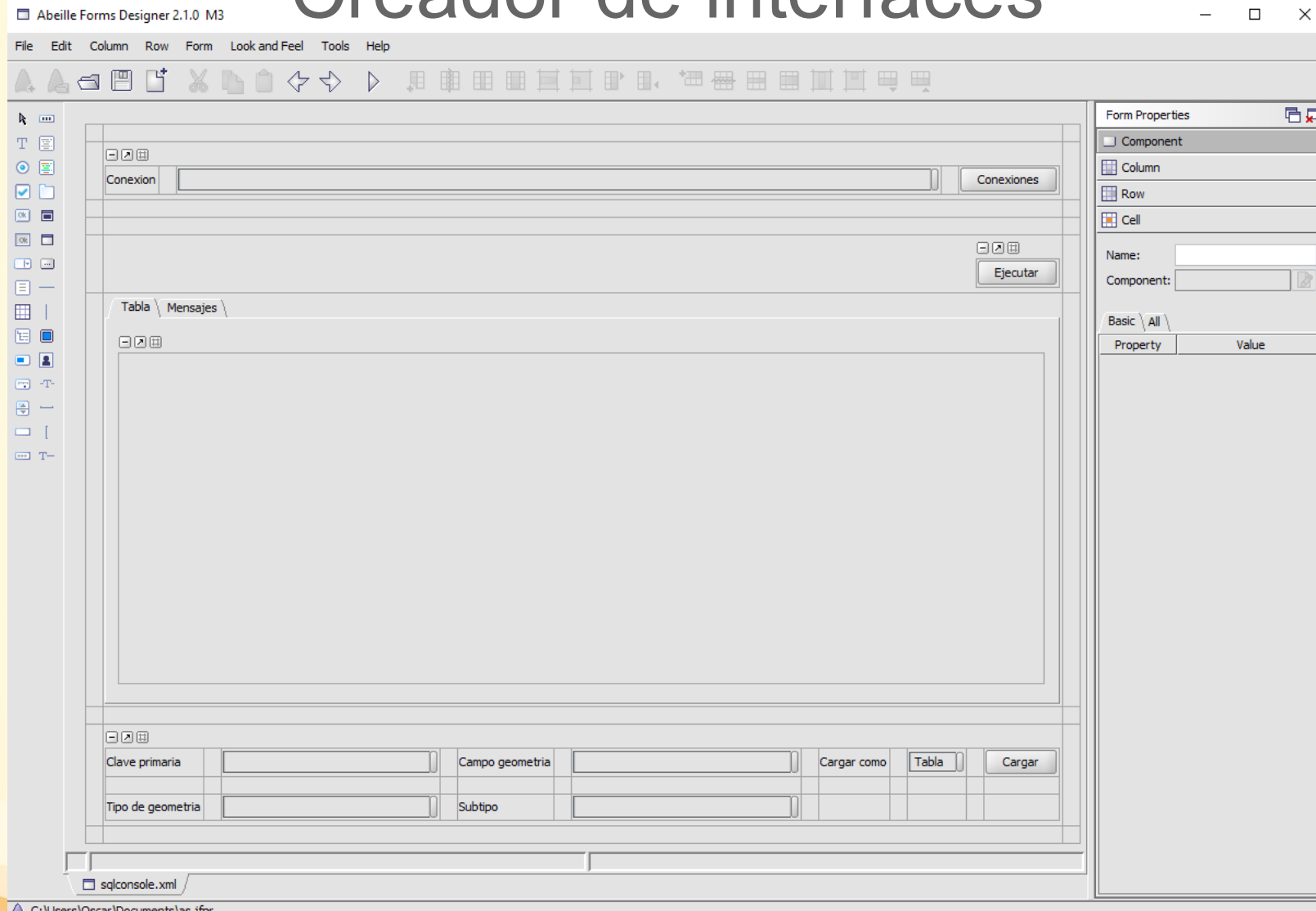
Text not found

Text not found

Línea 10:0

Novedades

Creador de interfaces



Novedades

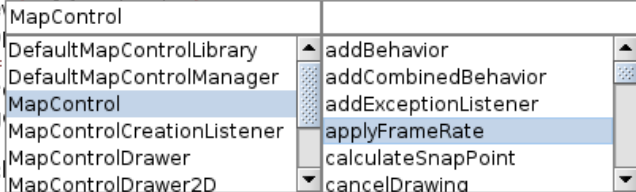
Mejoras en el editor:

Autocompletado

```

13 class Coordenadas(FormPanel, PointListener):
14     def __init__(self, viewdoc):
15         FormPanel.__init__(self, script.getResource("coordenadas.xml"))
16         self.viewdoc = viewdoc
17         self.mapControl = self.getWindowOfView(self.viewdoc).getMapControl()
18         self.mapControl.addBehavior("TestGetXYPointTool", PointBehavior(self))
19         self.mapControl.setTool("TestGetXYPointTool")
20         self.mapControl.
21     def getWindowOfView(self, viewdoc):
22         application = Application.getInstance()
23         projectManager = application.getProjectManager()
24         viewManager = projectManager.getViewManager()
25         return viewManager.getView(viewdoc.getId())
26
27     def btnCerrar_click(self, event):
28         self.hide()
29
30     def point(self, event):

```

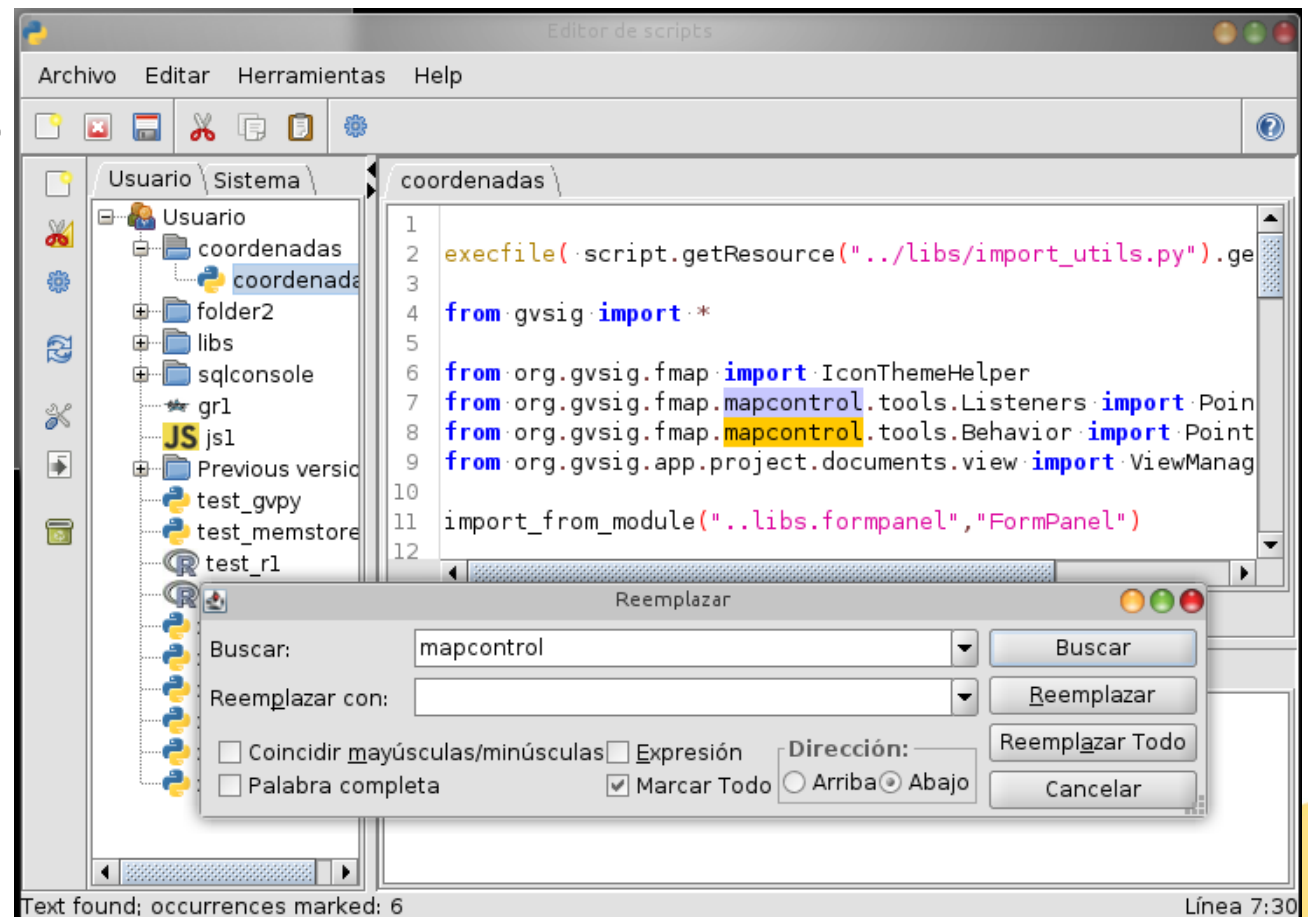


Manejo del “arbol” de scripts



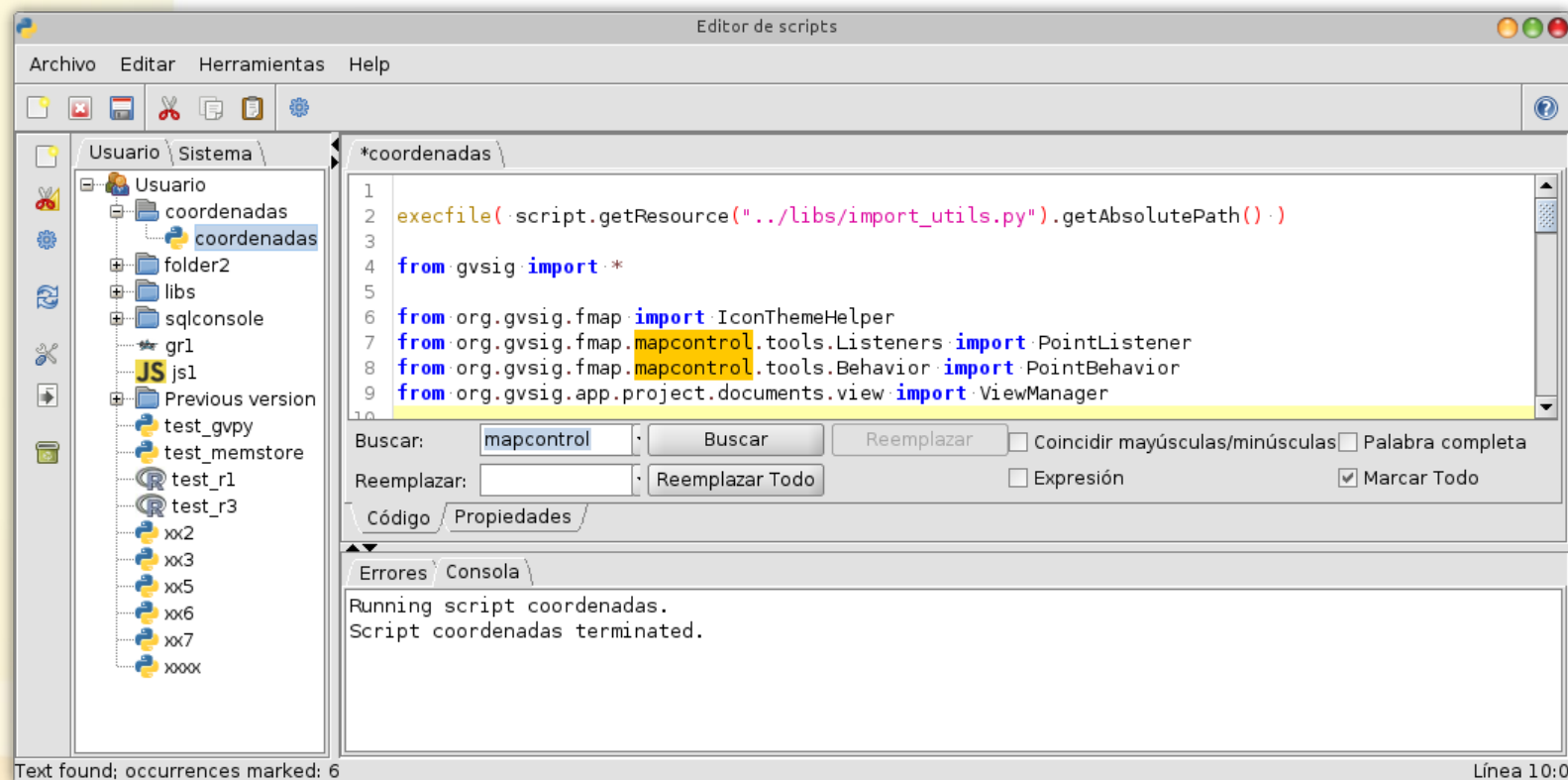
Novedades

Mejoras en el editor: Reemplazos



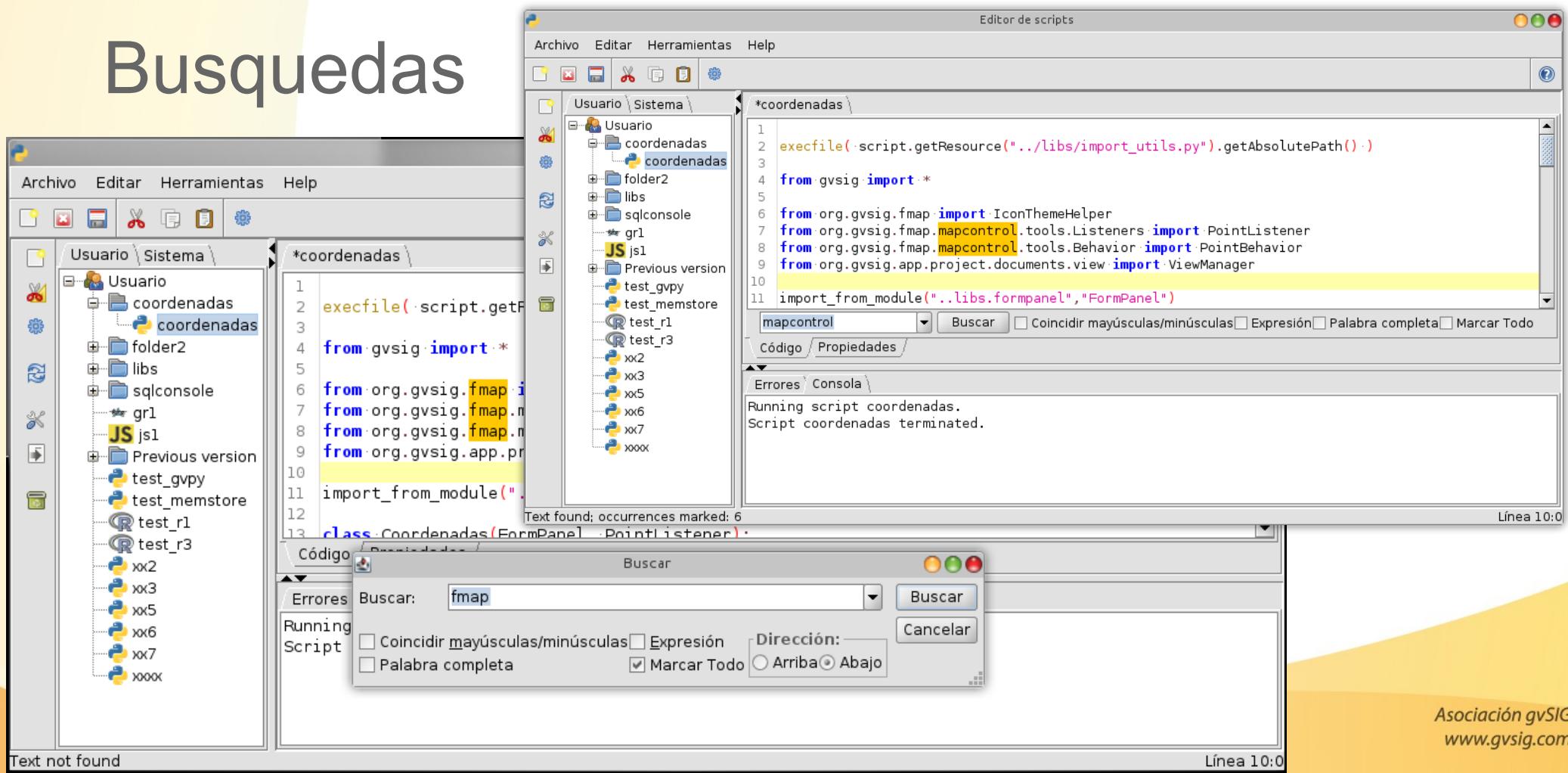
Novedades

Mejoras en el editor: Reemplazos



Novedades

Mejoras en el editor: Busquedas



Novedades

Soporte para crear geoprocesos

```
import_from_module("../libs.toolbox", "*")  
  
from gvsig import *  
from commonsdialog import *  
from es.unex.sextante.core import Sextante
```

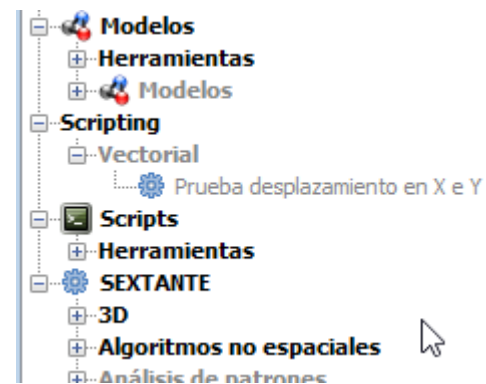
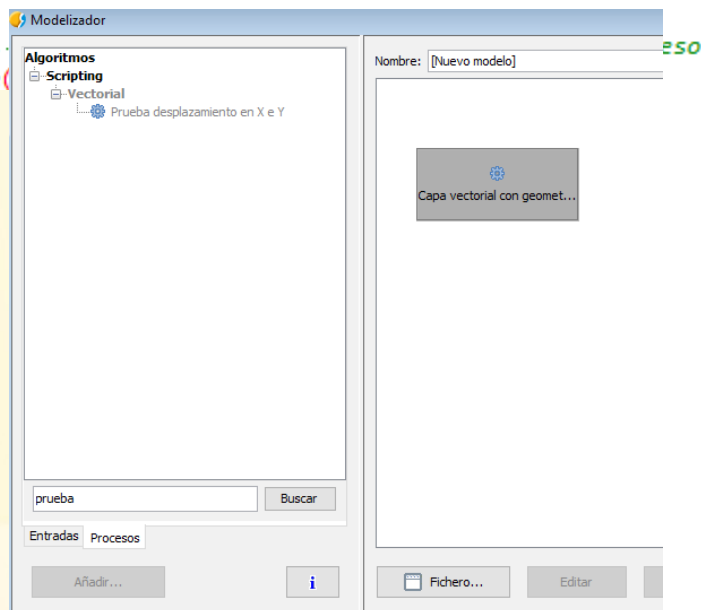
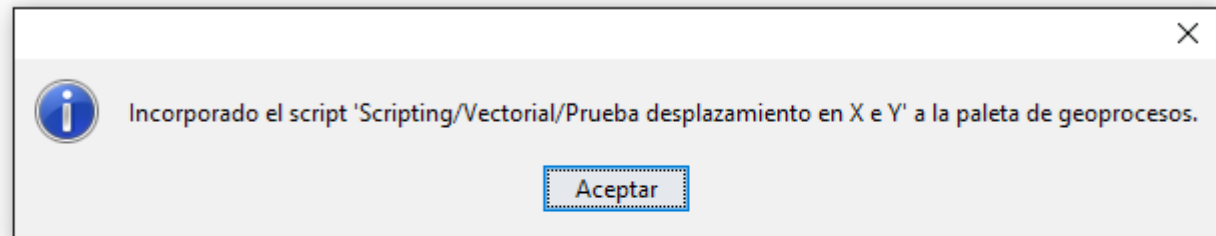
```
class XYShift(ToolboxProcess):
```

```
    def defineCharacteristics(self):
```

```
        """  
        En esta operacion debemos definir los parametros de entrada y salida que va a precisar nuestro proceso.  
        """
```

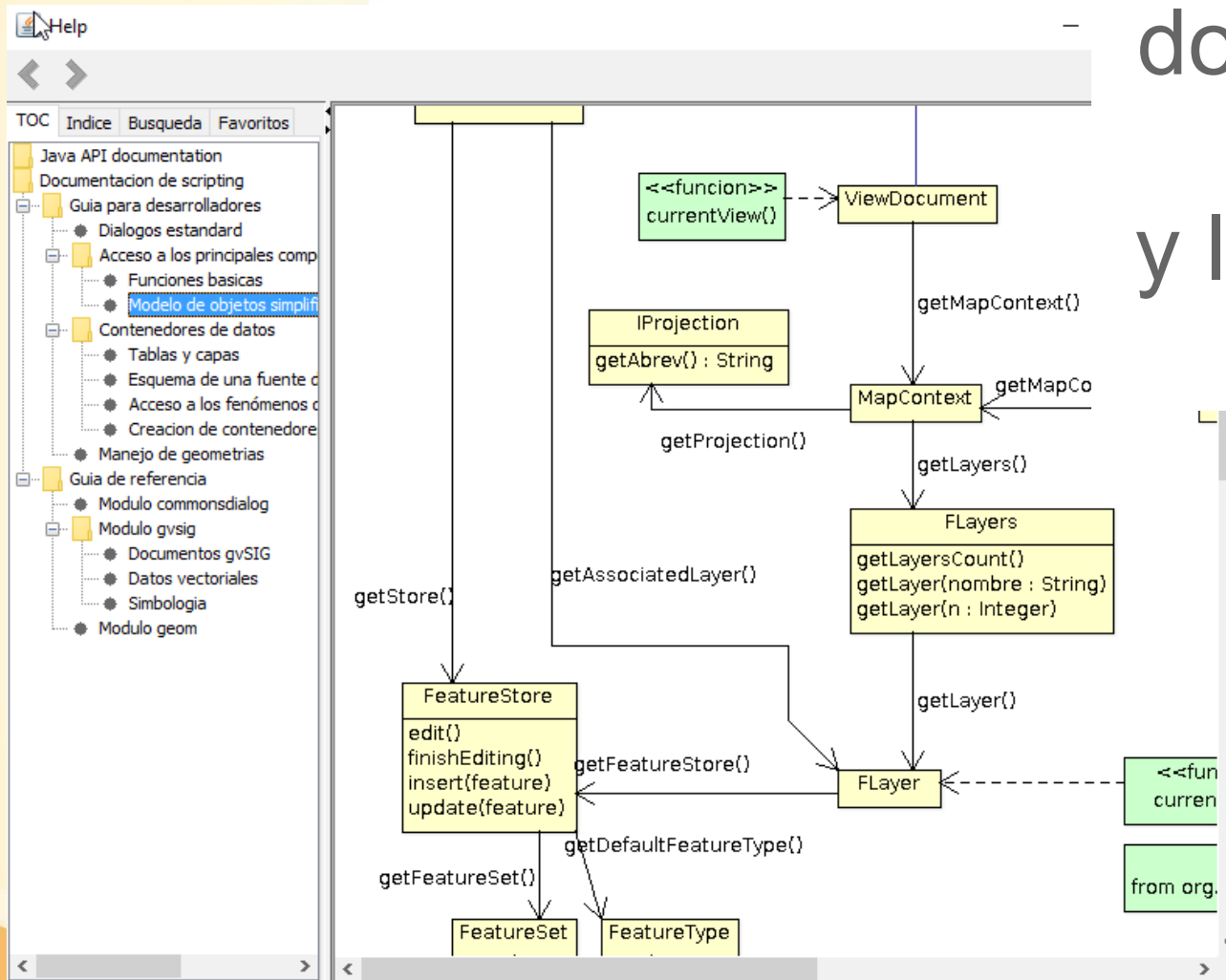
```
        # Fijamos el
```

```
        self.setName("Prueba desplazamiento en X e Y")
```



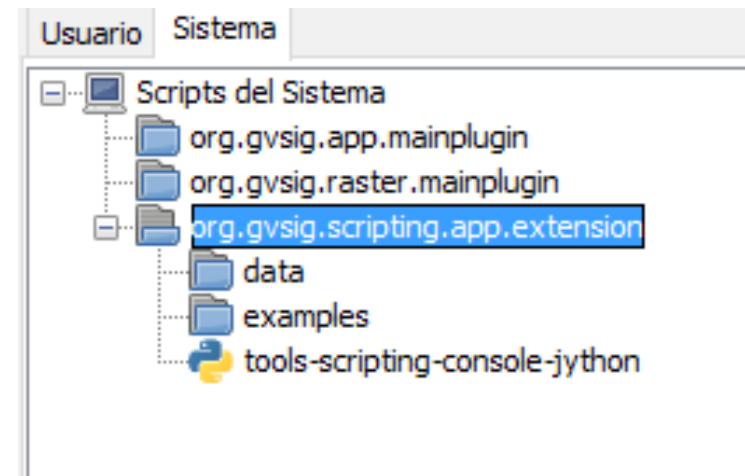
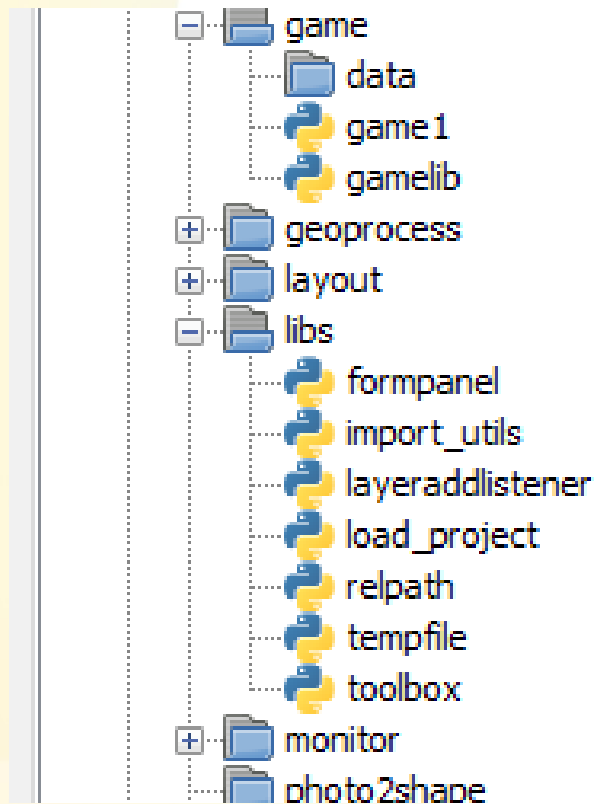
Novedades

Embebida documentación de Scripting y librerías gvSIG en Java



Novedades

Soporte para invocar otros scripts
y usarlos como módulos de librería



Novedades

Añadido soporte para usar librerías
De diferentes plugins
(**use_plugin**)

Añadido soporte para incluir librerías
Java (jars) e importar clases de ellas
(**use_jar**)

Novedades

Mejoras de rendimiento en el
módulo “gvSIG”

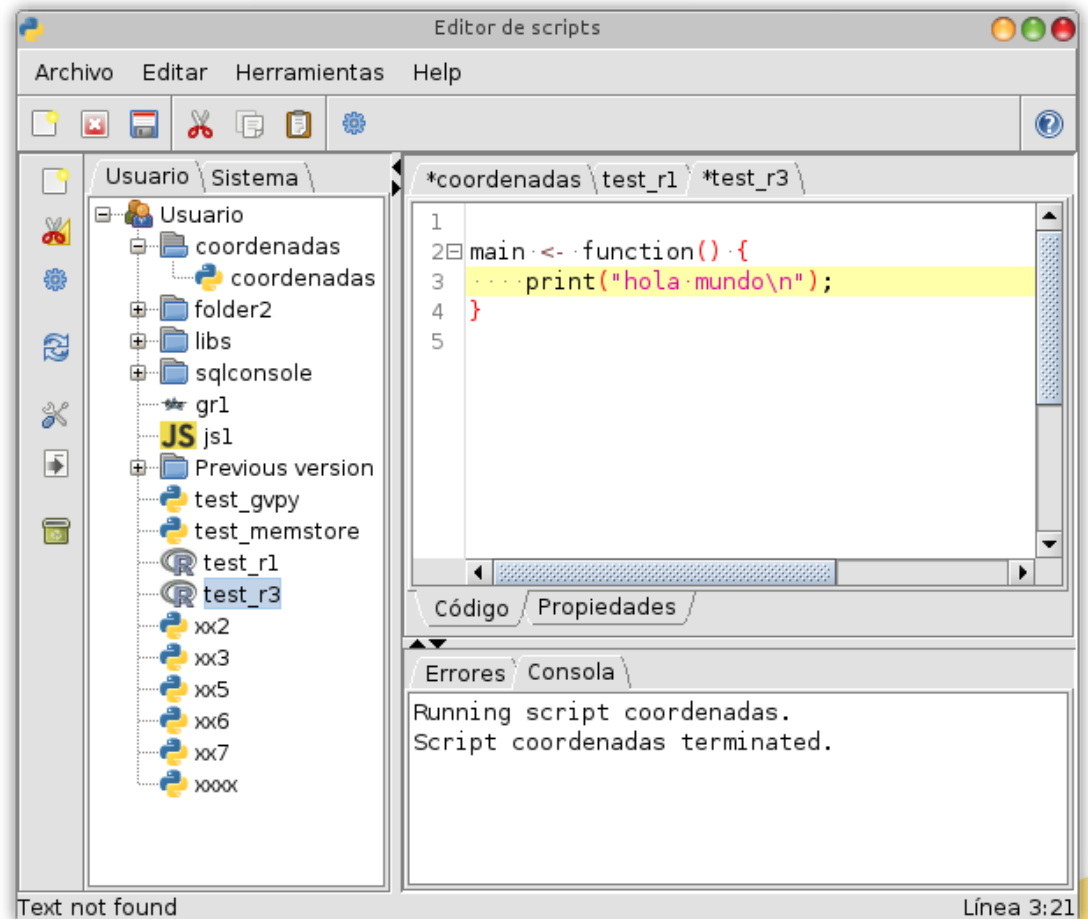
Incluidas algunas librerías de base:

- **geopy**: geocodificación de direcciones
- **jOpenDocument**: generación de informes

Novedades

Aumento de soporte
en el
Módulo de Scripting:

- **Soporte de R**
a través de la
librería **Renjin**



Más información

- Code Sprint
- Mooc de Scripting en gvSIG
- Blog de gvSIG
<http://blog.gvsig.org/>
- Outreach: Ejemplos de scripts
<http://outreach.gvsig.org/scripts/>

Contacto

Listas de correo

- Usuarios
- Desarrolladores



info@gvsig.com
www.gvsig.com

Joaquín del Cerro

Óscar Martínez
@masquesig

omartinez@gvsig.com