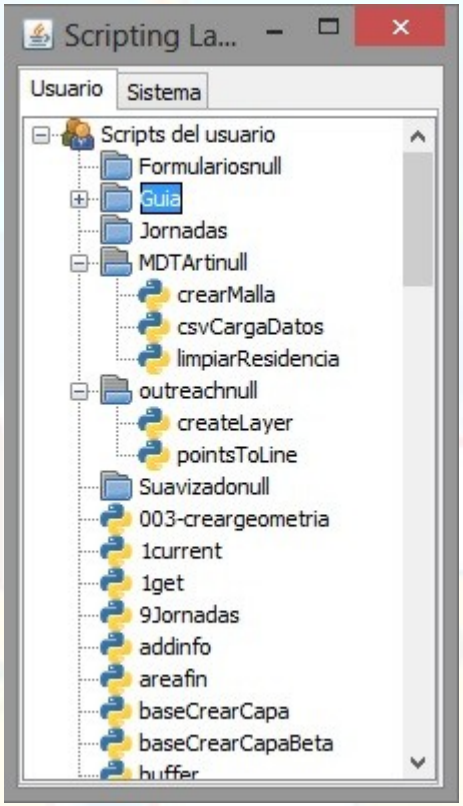
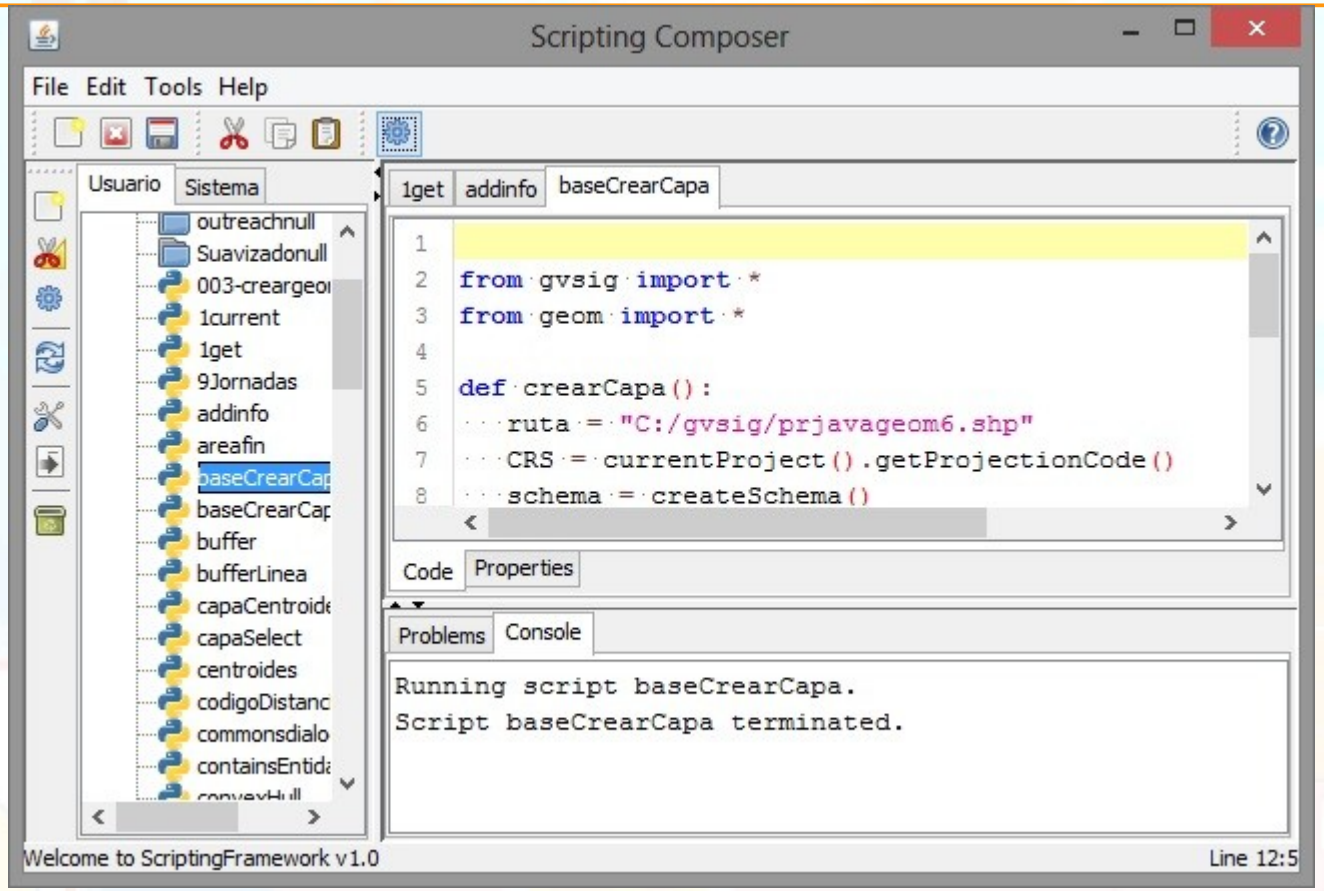
CC BY SA

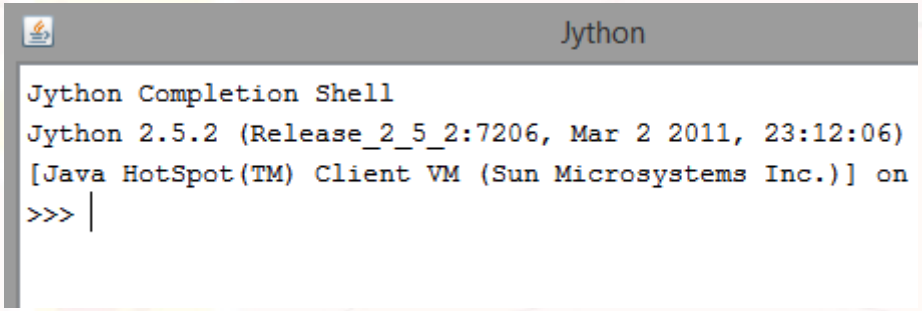
Scripting in gvSIG 2.3: New features, improvements and enhancements

Óscar Martínez
Twitter: @masquesig
Email: omartinez@gvsig.com

Scripting Module







Scripting Module

- Integrated programming module (IDE)
- Create & Execute & Organize scripts
- Different allowed languages:
  - Jython (Java + Python)
  - Groovy
  - R
  - .. and more

Scripts

- Scripts for:
  … create new geoprocess
  … automate tasks
  … develop new tools
  … add visual interface (Abeille)

- Access to all gvSIG functions

Scripting in gvSIG

Useful for:

Users

&

Developers

New features

Jython

- Updated version Jython 2.7.1 beta3

more info: http://fwierzbicki.blogspot.com.es/

- Full compatibility with Java libraries … and some of Python (without C)

- add Python libs to:
gvsig\gvSIG\extensiones\org.gvsig.scripting.app.mainplugin\scripting\lib

- or jars to:
gvsig\gvSIG\extensiones\org.gvsig.scripting.app.mainplugin\lib

- Future: JyNI (compatibility with Numpy and with almost all Python libraries)

more info: http://jyni.org/

R

- R through Renjin
  - more info: http://www.renjin.org/

  Renjin is a JVM-based interpreter for the R language
  - *Full integrated with Java/gvSIG*
  - *Not full integrated with R packages*

```
import (org.gvsig.app.project.documents.view.ViewDocument)
import (org.gvsig.app.ApplicationLocator)
import (org.gvsig.fmap.mapcontext.MapContext)
main <- function() {

    cat( "View access\n" );
    application <- ApplicationLocator$getManager();

    view = application$getActiveDocument(ViewDocument);
    print(view$name);
    prj <- view$projection;
    print(prj);
    crs <- prj$fullCode;
    print(crs);
    mapcontext <- view$mapContext;
    layers <- mapcontext$layers;
    layer <- layers[[1]];
```

- R native
  - Full integrated with R
  - Not full integrated with Java/gvSIG
  - One installation of R will come with gvSIG

GDAL

- GDAL integrated in gvSIG

  ..so we can use gdal in our scripts

  http://gdal.org/java/

- Also, we could use:
  - Ogr2ogr
  - Gdaltranslate
  - ..

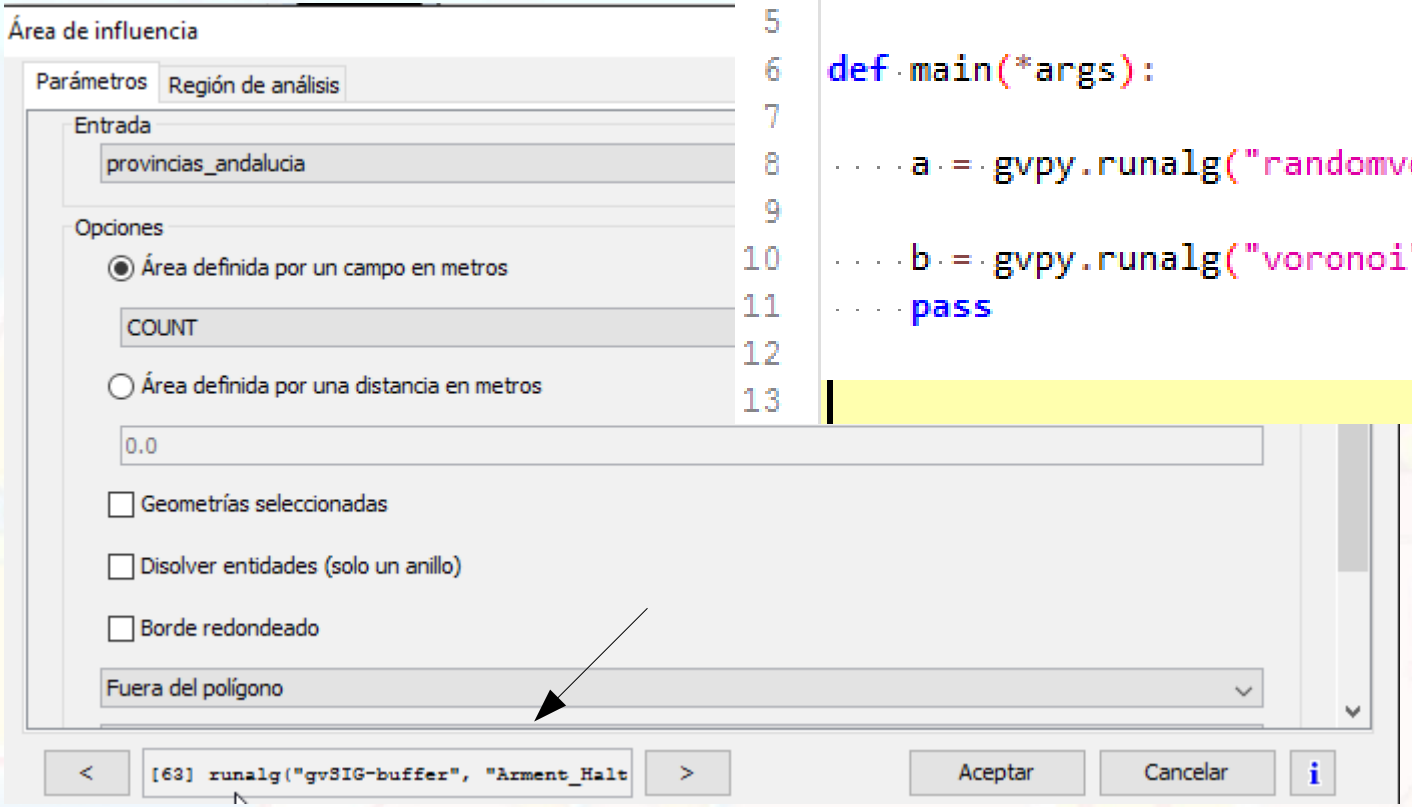
```
def main(*args):

    """ogr2ogr from gdal"""

    ogr2ogr.main(["-t_srs", "CRS:84",
                  "-f",
                  "ESRI Shapefile",
                  "C:/temp/j1",
                  "C:/temp/countries.geojson",
                  "-overwrite",
                  "-skipfailures"
                  ])


    gdalapp("ogr2ogr -t_srs 'CRS:84' -f 'ESRI S
```

```
1  from gvsig import *
2  import uselib
3  uselib.use_plugin("org.gvsig.gdal.app.mainplugin")
4
5  from org.gdal.gdal import gdal
6  from org.gdal.ogr import ogr
7  from org.gdal import osr
8
9  def main(*args):
10     #Book: Python Geospatial Analysis Cookbook
11     gdal.AllRegister()
12
13     # get raster data source
14     rasterfile = r"C:\gvdata\data_example\mdt_jaen.tif"
15     open_image = gdal.Open(rasterfile)
16     input_band = open_image.GetRasterBand(1)
17
18     # create output data source
19     output_shp = getTempFile("newshp","")
20     shp_driver = ogr.GetDriverByName("ESRI Shapefile")
21
22     # create output file name
23     output_shapefile = shp_driver.CreateDataSource(output_shp + '
24
25     sp = osr.SpatialReference('PROJCS["ETRS89 / UTM zone 30N",GEO
26
27     new_shapefile = output_shapefile.CreateLayer(output_shp, sp)
28     #mask_band = input_band.GetMaskBand()
```

Geoprocess

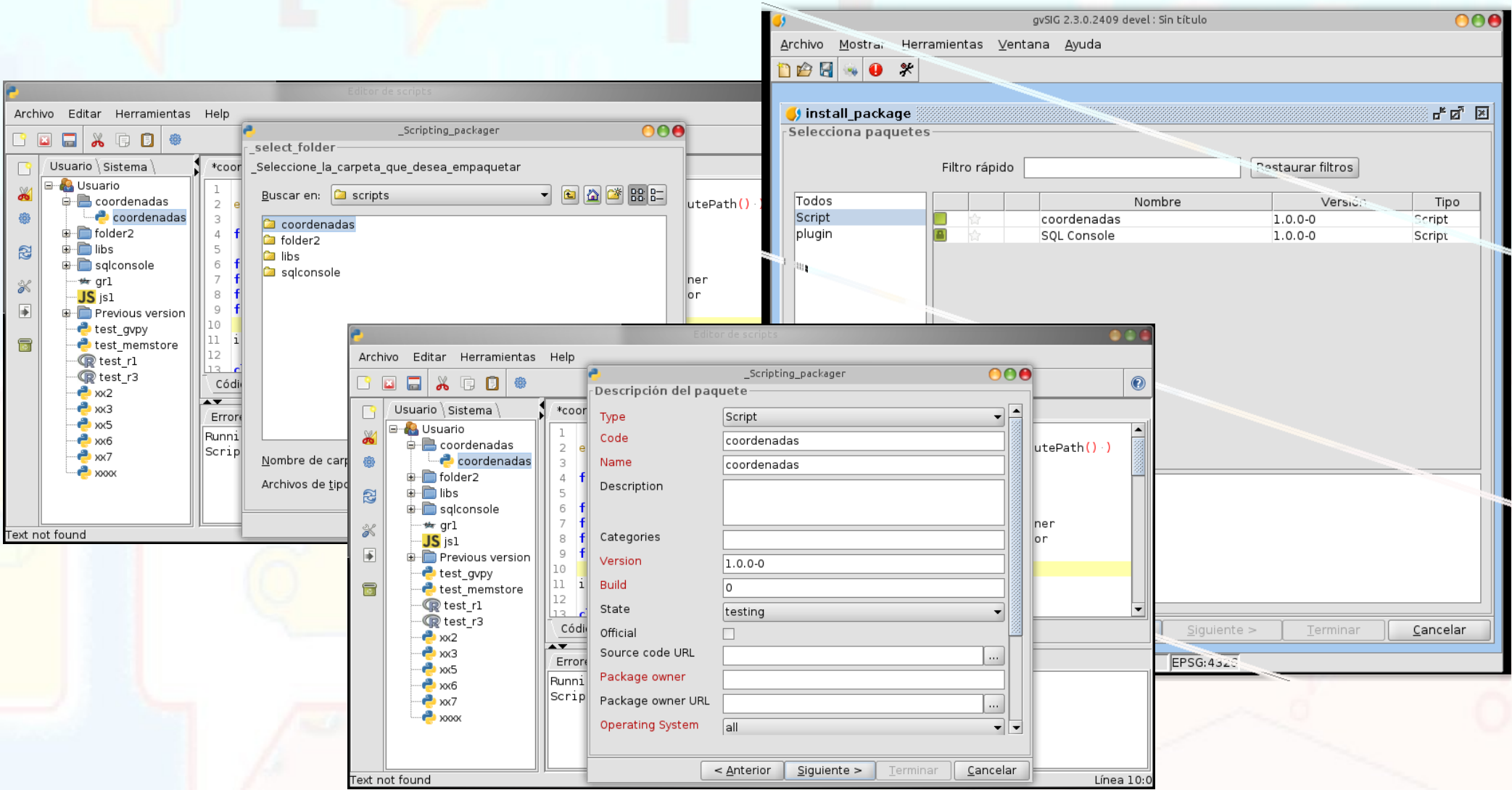
Call geoprocess that already exist inside gvSIG

```
1
2  from gvsig import *
3  import gvpy
4
5
6  def main(*args):
7
8      a = gvpy.runalg("randomvector", COUNT=10 , TYPE=2 )
9
10     b = gvpy.runalg("voronoi", a)
11     pass
12
13
```



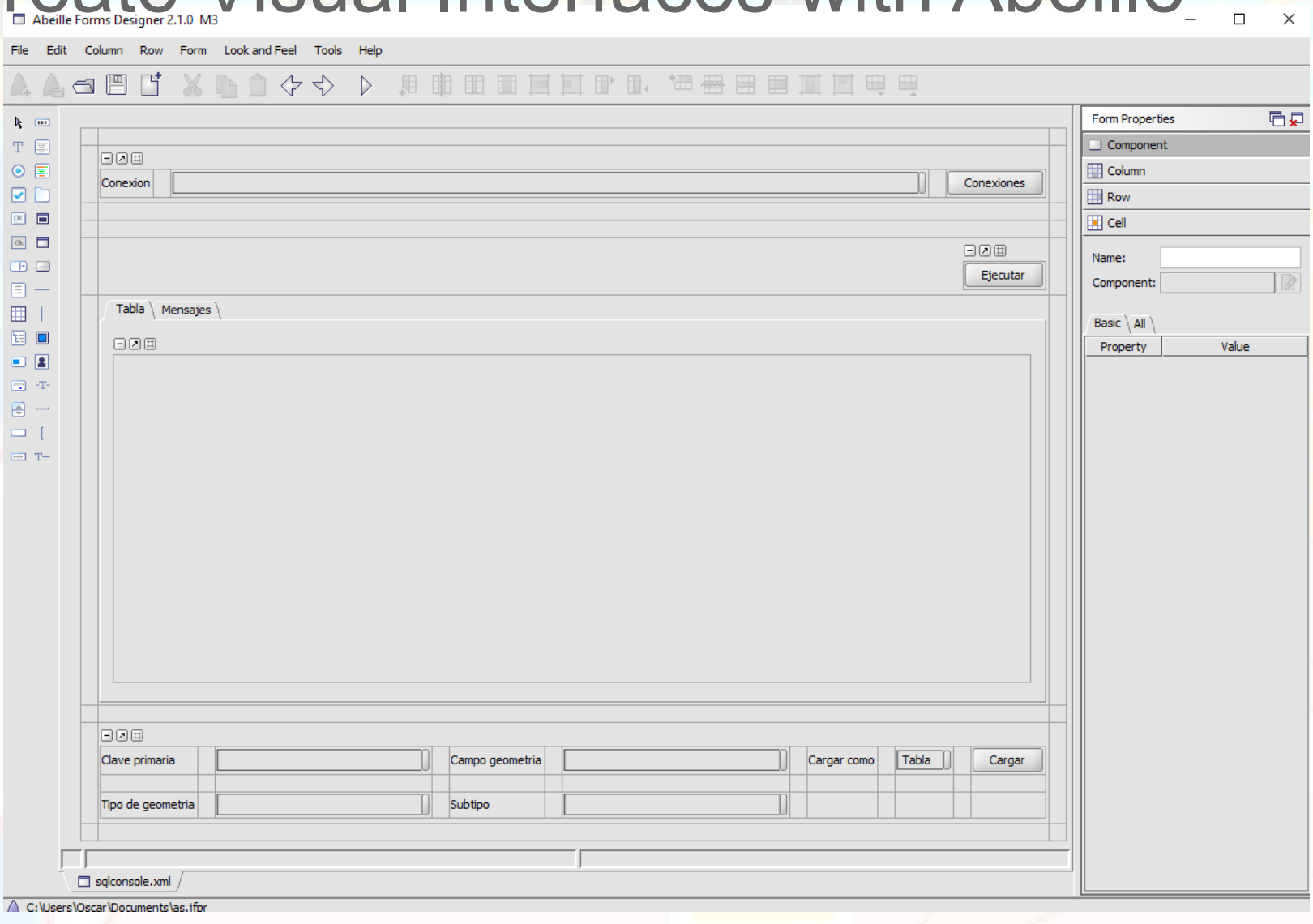
Addons

Create Package (Addons) from Scripts



Visual Interfaces

Create Visual Interfaces with Abeille



Visual Interfaces

New lib development “FormPanel”

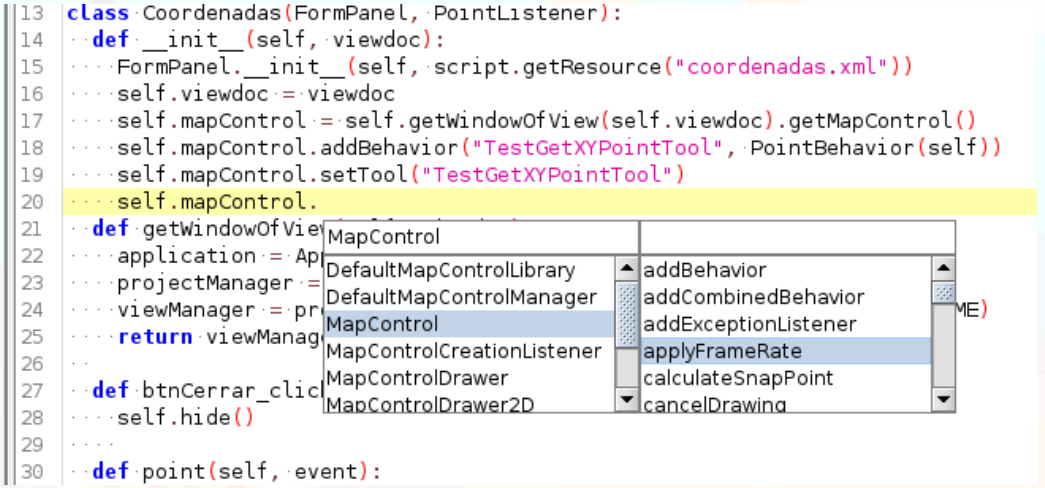
- Create easily visual interfaces
- Assign actions to buttons (action click)
- Manage listeners (action change)

```
1
2  from gvsig import *
3  from libs.formpanel import FormPanel
4
5  class Panel(FormPanel):
6      def __init__(self):
7          FormPanel.__init__(self, script.getResource("ui_basic.xml"))
8
9      def btnCalcular_click(self, *args):
10         self.txtField.setText("Clicked!")
11         print "Clicked!"
12         
13     def btnClose_click(self,*args):
14         self.hide()
15
16     def spinerValue(self, *args):
17         print "Value has change!"
18         
19 def main(*args):
20     l = Panel()
21     l.showTool("Visual")
22     pass
```

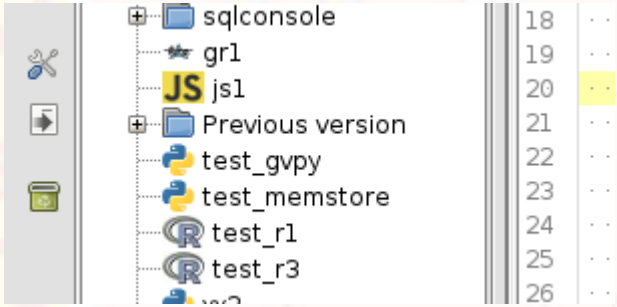
Composer

IDE Improvements

Autocomplete
(New: also with variables names)



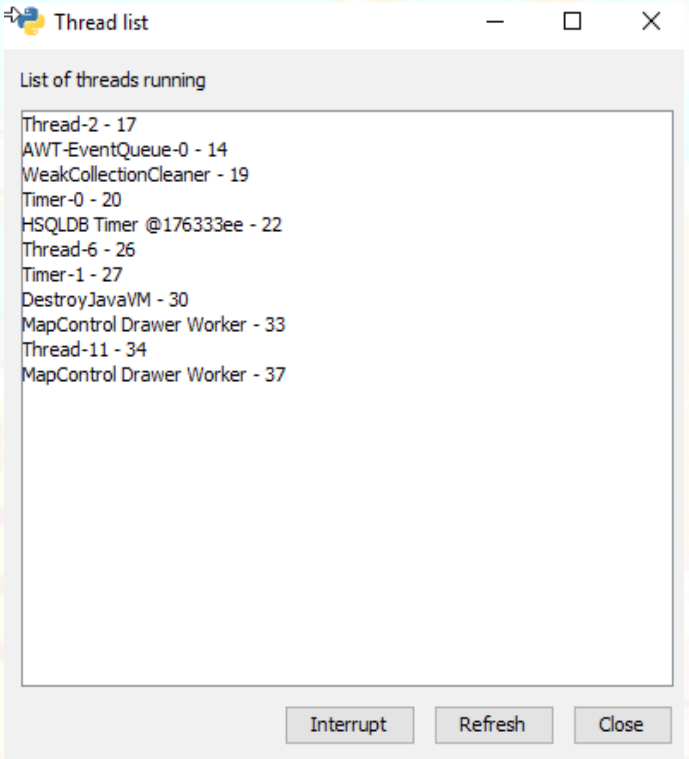
Tree scripts



Composer

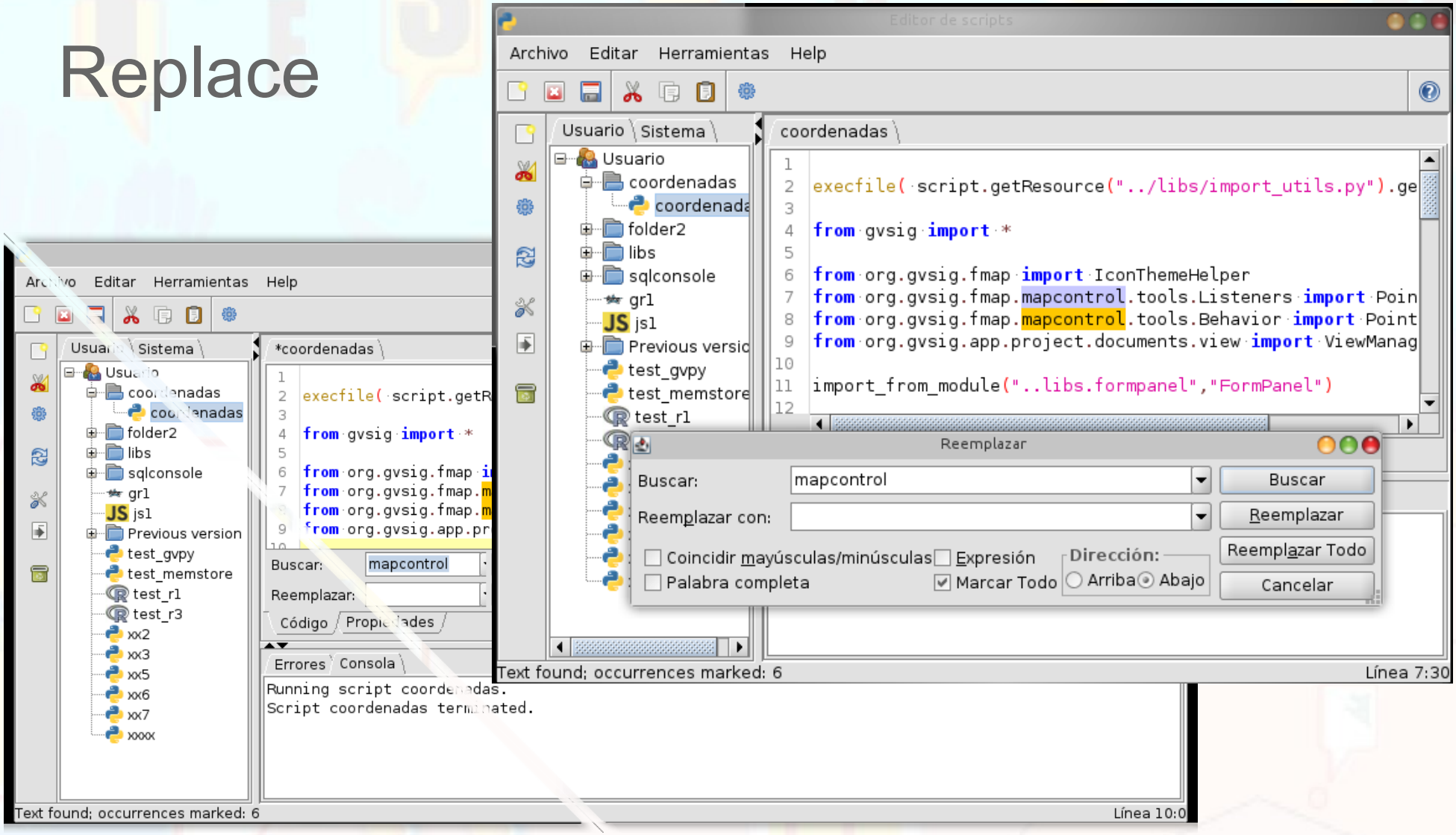
IDE Improvements

- Threads Killer
- Code navigator



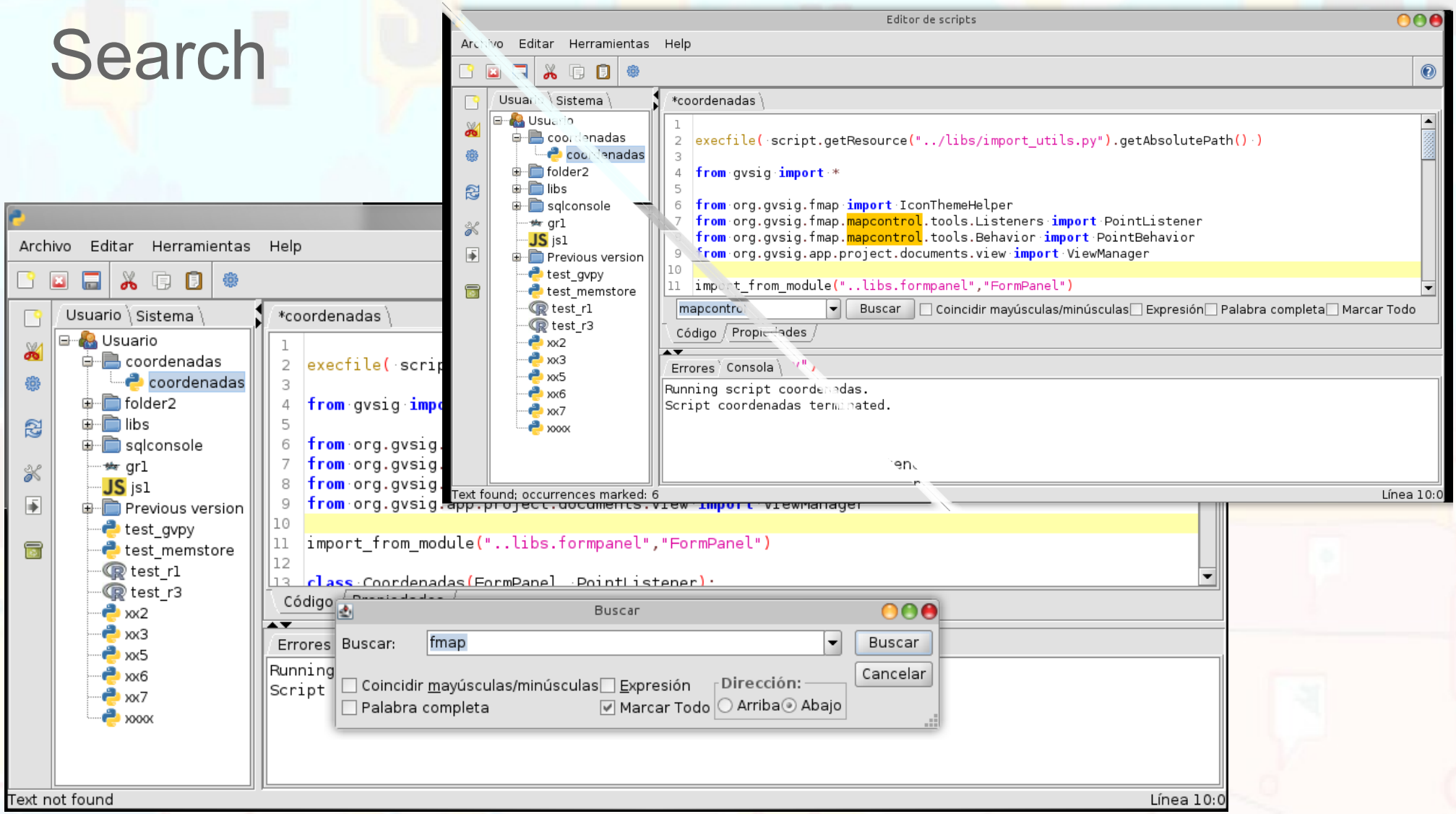
Composer

Replace



Composer

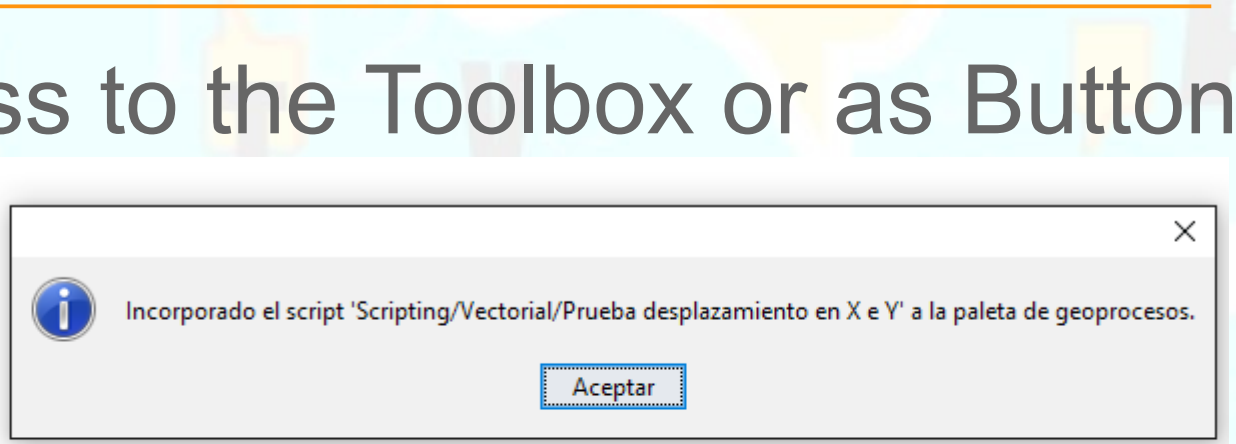
Search

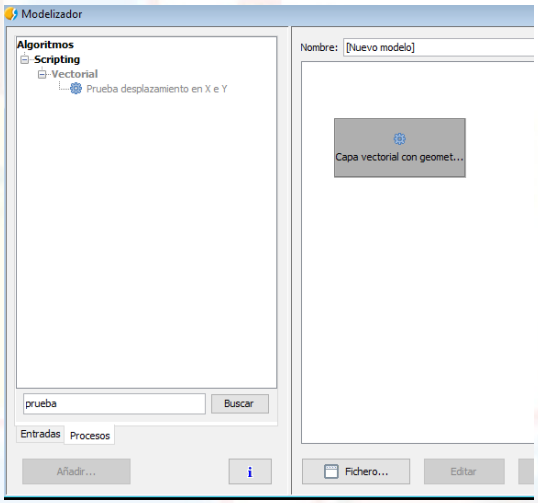


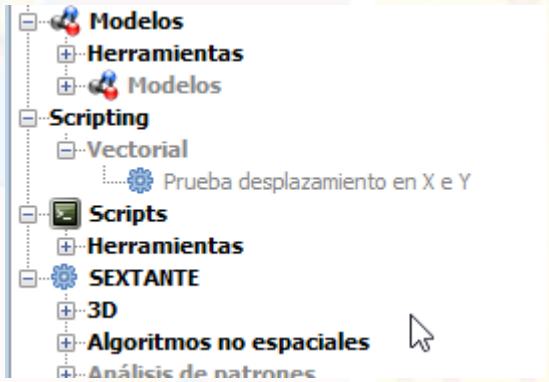
Toolbox and Buttons

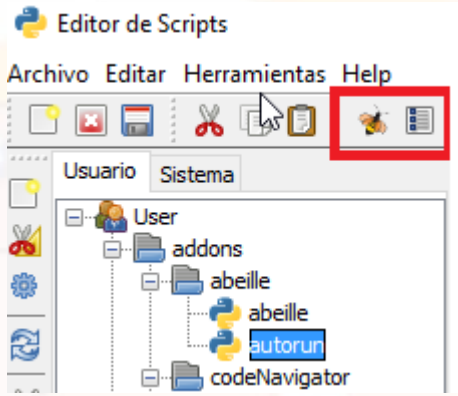
Add geoprocess to the Toolbox or as Buttons

```
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 netrada y salida que va a precisar nuetro proceso.
        """
        #Fijamos el nombre con el que se va a mostrar nuestro proceso
        self.setName("Prueba desplazamiento en X e Y")
```

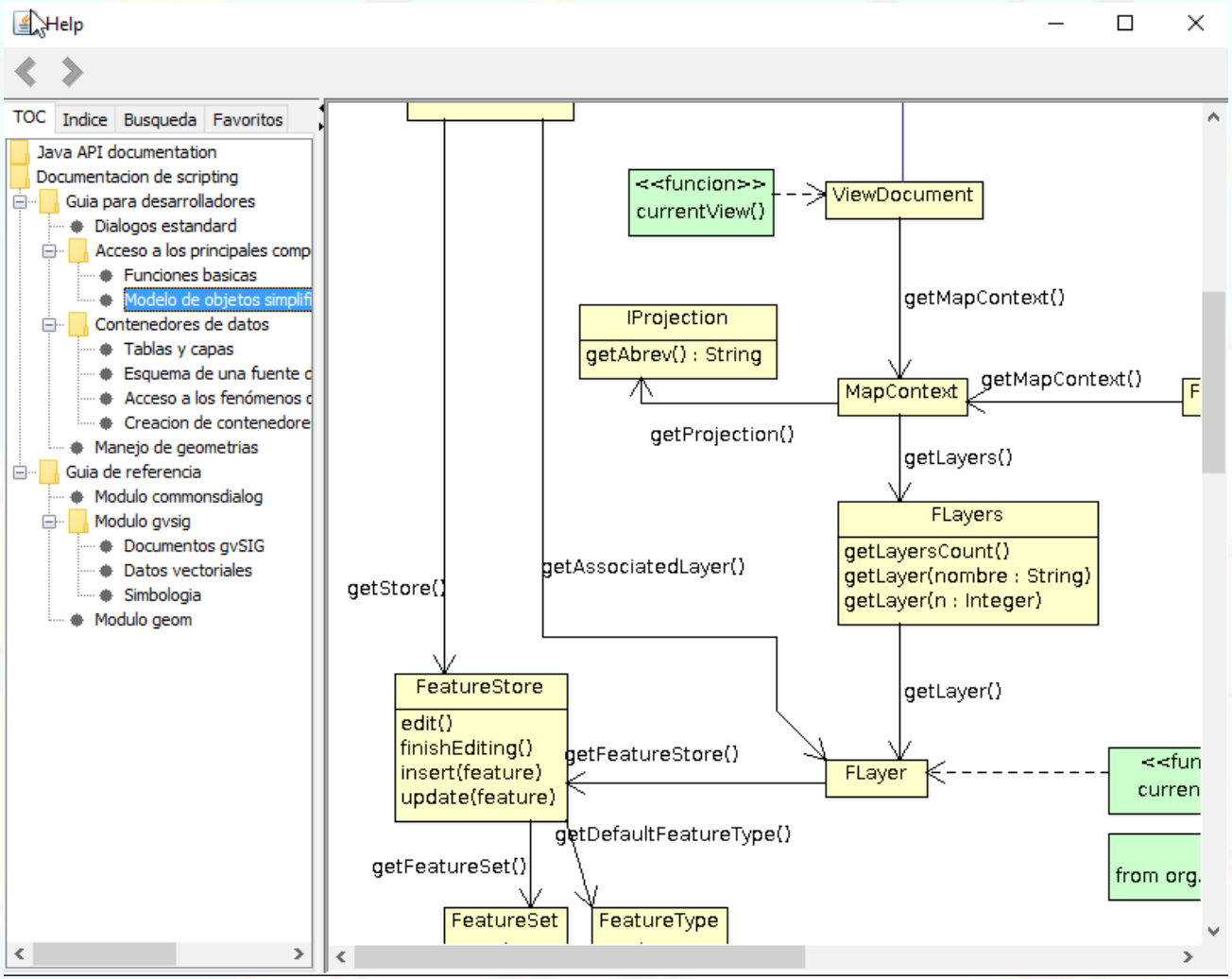








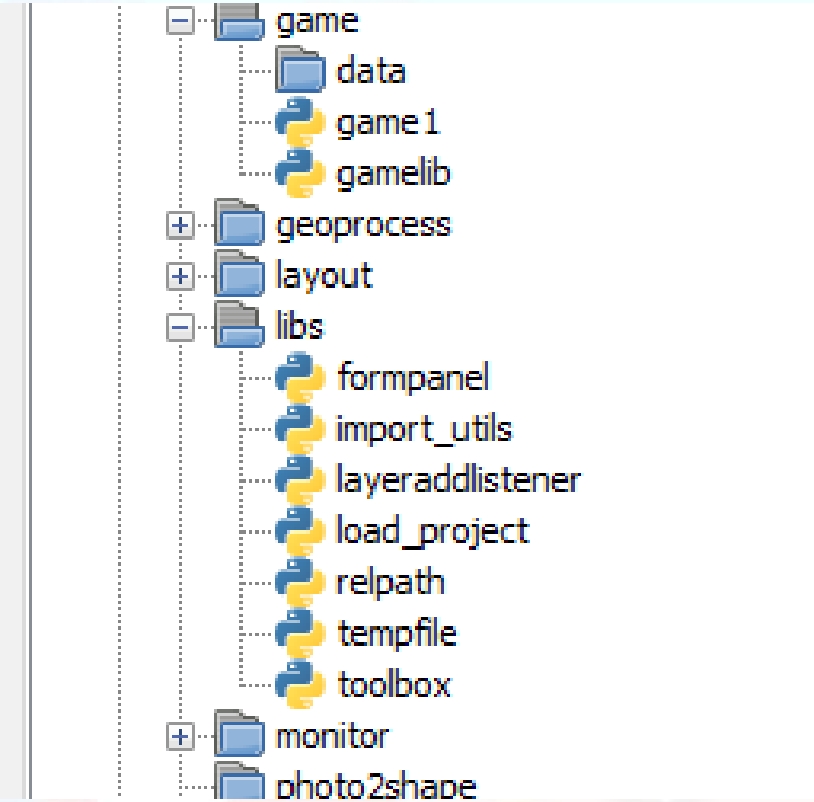
Help documentation

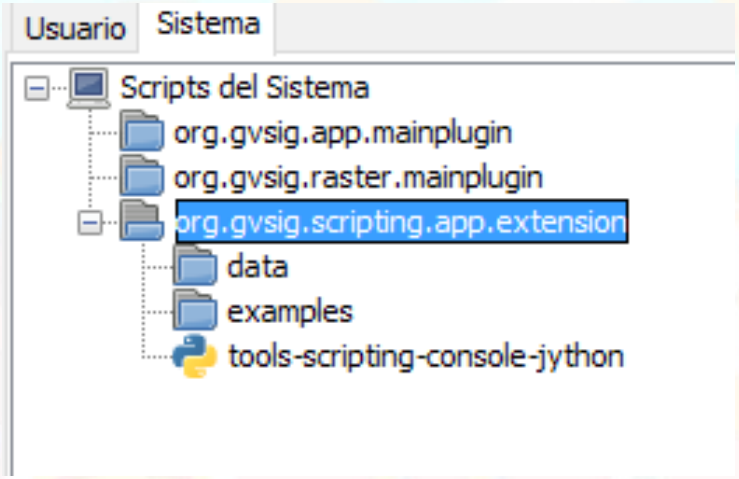


Scripting and Javadocs documentation

Imports

Import functions from others scripts or libs directly with *import libname*





Use plugin & jar

Use plugins already installed inside gvSIG
(**use_plugin**)

Use new jars from outside gvSIG Java (jars)
(**use_jar**)

Libraries

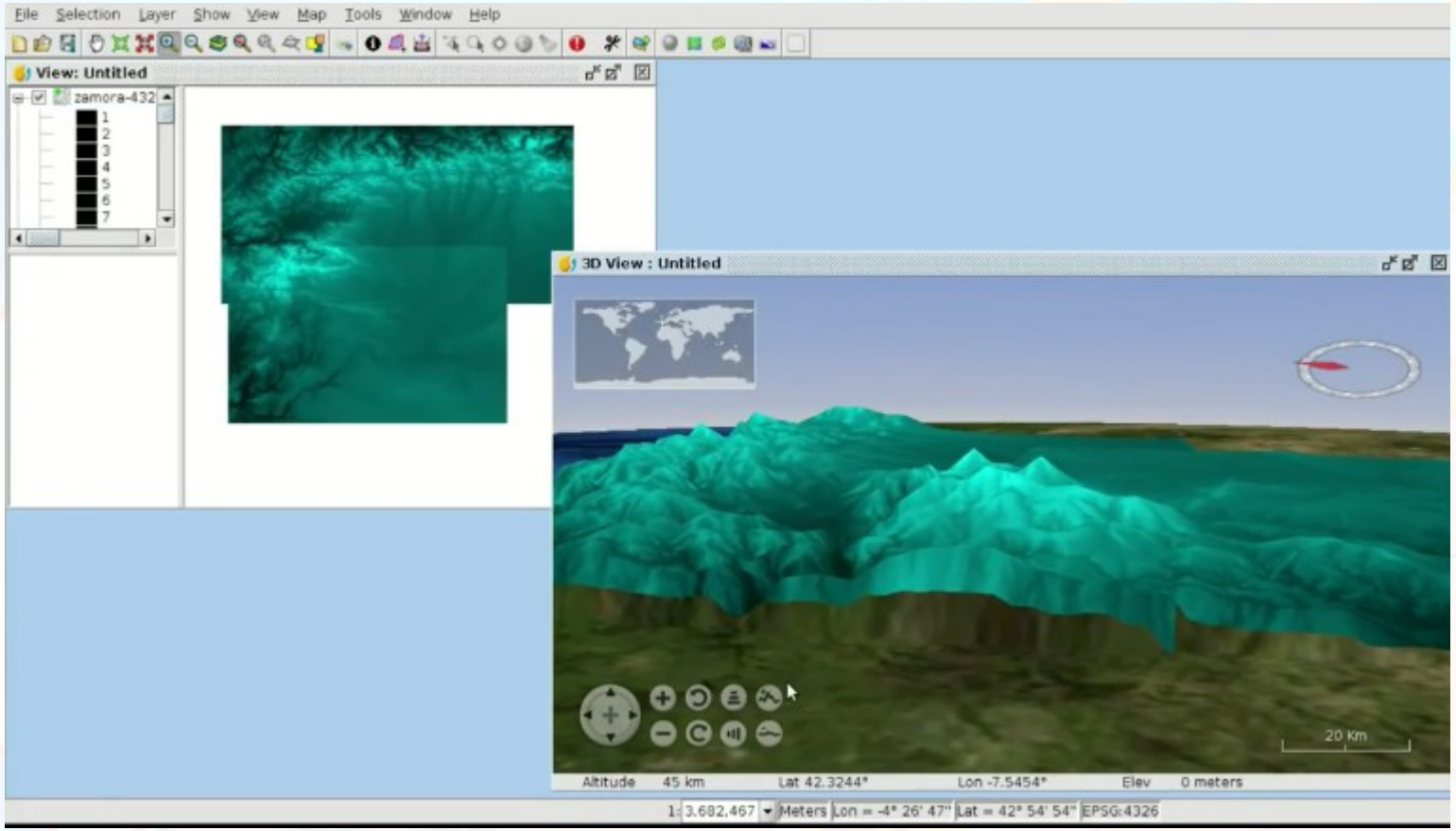
Included some libraries:

- **geopy**: address geocodification
- **jOpenDocument**: reports
- **cartodb**: upload files and more

WorldWind View Plugin

Still not fully developed: Accessing to the 3D View from scripts

It will be a big feature



Big Improvements

Jython

- Updated libraries (gvsig and geom)
- All the module has been updated
- We tried don't lose compatibility (same syntax)

- **New approached to gvsig lib**:
  Improve Java API with Python functions

Jython

Before (gvSIG 2.2)

- Less powerful
- Requires a lot of development
- Rewrite all classes (mix of Python and Java Classes)
- Hard if one method doesn't exist

```
class View(WrapperToJava):
  """
  Represents gvSIG view document
  (org.gvsig.app.project.documents.view.DefaultViewDocument).
  """
  def __init__(self,view):
    WrapperToJava.__init__(self,view)

  def getLayer(self, name=None):
    """
    Returns one view layer. If name is None returns active
    layer if the view has one, if name is not None returns laye
    else returns None.
    :param name: view name in Table Of Contents
    :type name: string
    :return: View
    :return: None
    """
    map = self.getMapContext();
    if name == None:
      activeLayers = map.getLayers().getActives()
      if len(activeLayers) != 1 :
```

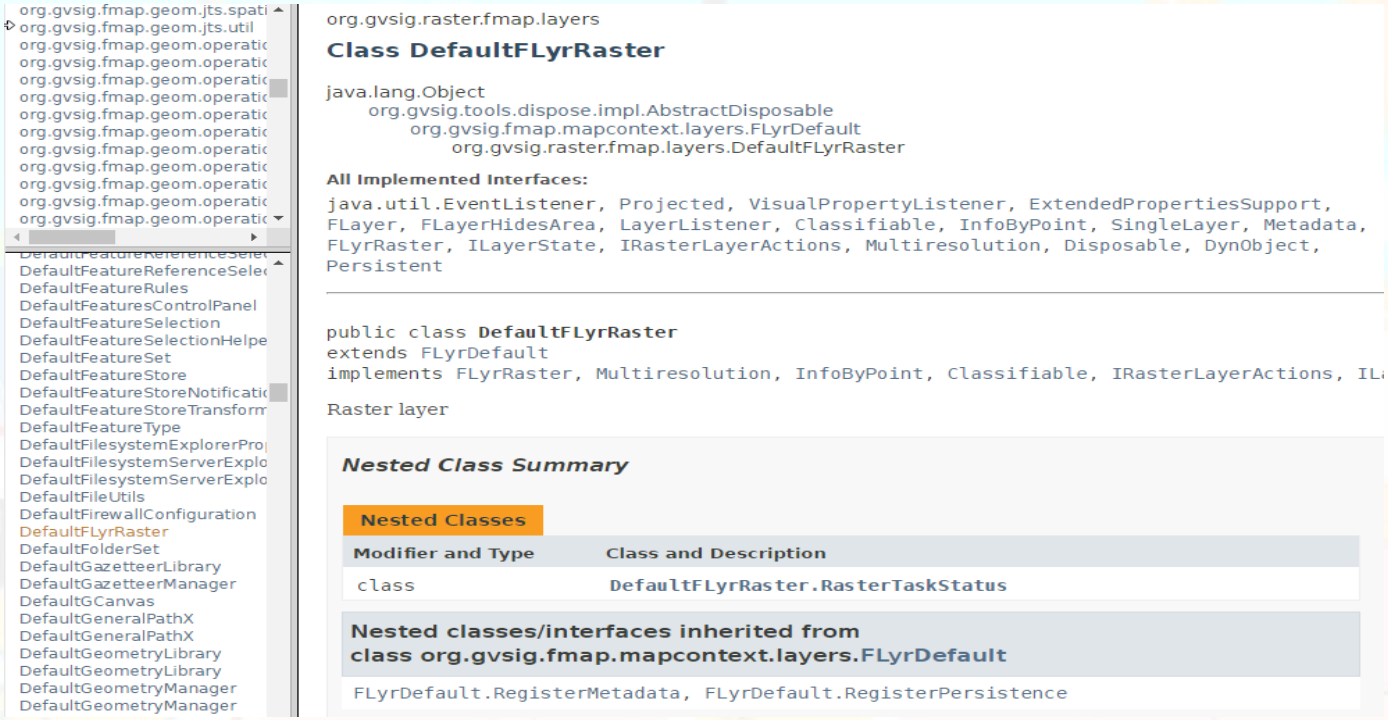
After (gvSIG 2.3 and more)

- All of the gvSIG and Java power
- Doesn't require big developments
- All gvSIG Objects are Java
- Added Python methods to Java classes

```
def __getWindowOfView(self):
    application = ApplicationLocator.getManager()
    projectManager = application.getProjectManager()
    viewManager = projectManager.getDocumentManager(V
    return viewManager.getMainWindow(self,None)

def currentLayer():
  try:
    return currentView().getLayer()
  except:
    return None


#
# Inject new methods in the class JViewDocument
#
JViewDocument.centerView = __centerView
JViewDocument.showWindow = __showWindow
JViewDocument.getLayer = __getLayer
JViewDocument.getLayers = __getLayers
```

Jython

So.. you could use directly all Java classes and methods from the Javadocs



+ Pythons functions and methods that will make your scripts so easy to develop and maintain

```
13    layer = createShape(schema, prefixname="asd", CRS='EPSG:25830')
```

Development examples

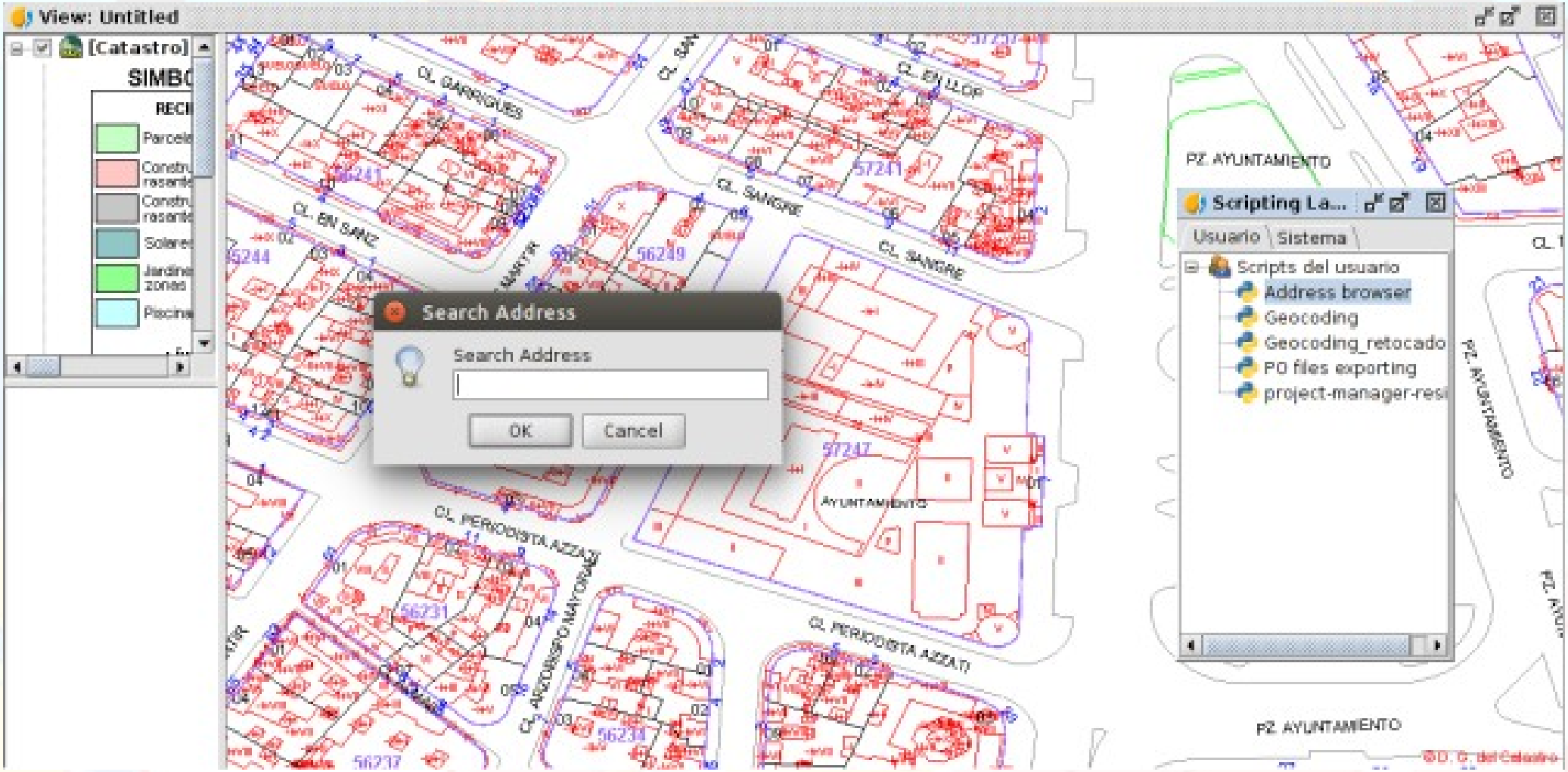
Intersection

"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?"

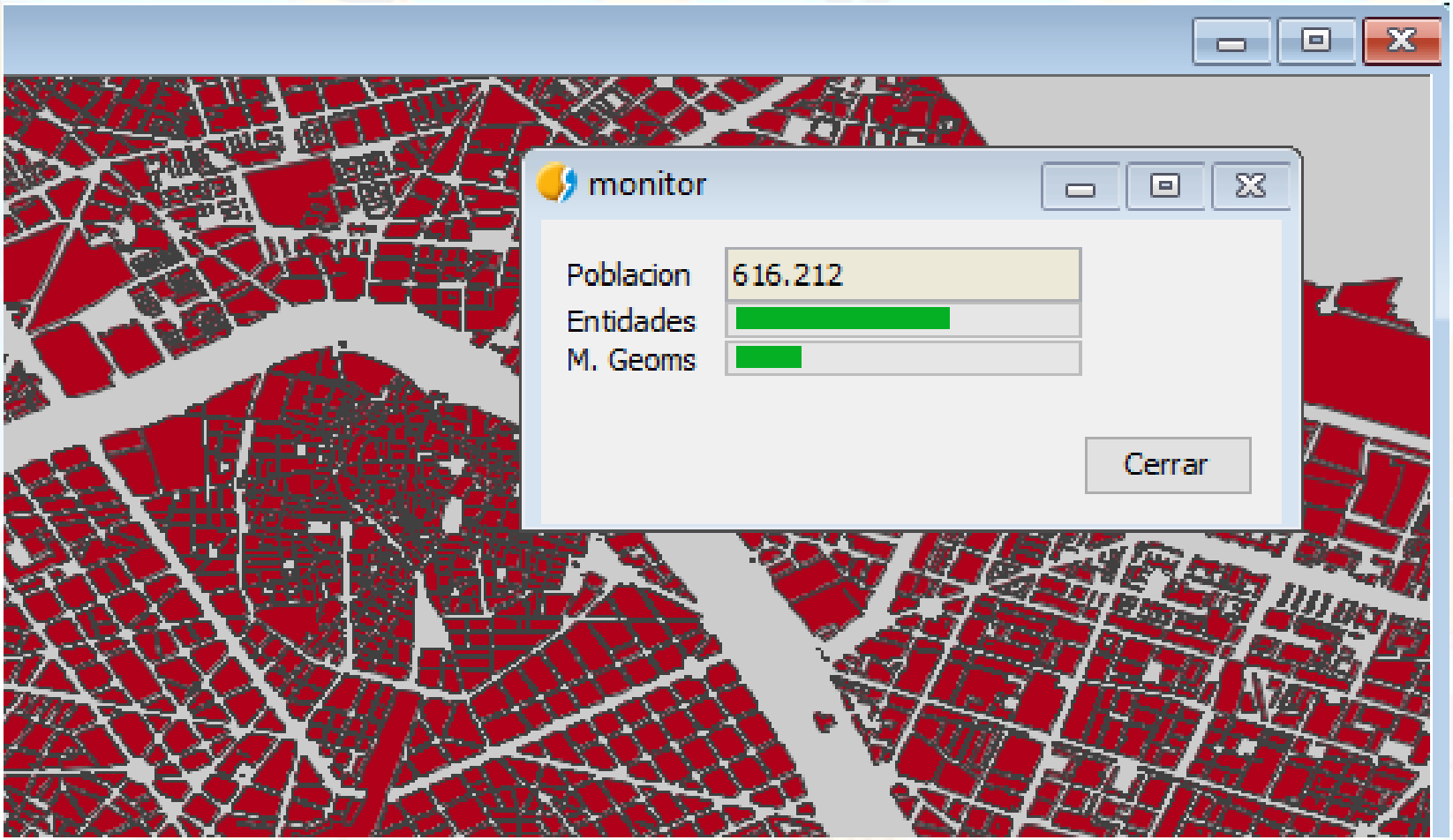
```
Point info:  {u'campo2': u'Venecia', u'campo1': 101L}
- Layer:  valencia_index
--  {u'fecha': u'1980-01-01', u'location': u'/cdrom/dat
--  {u'fecha': u'1992-01-01', u'location': u'/cdrom/dat
--  {u'fecha': u'2002-01-01', u'location': u'/cdrom/dat
- Layer:  parcelas_Valencia
--  {u'PARCELA': u'03', u'AREA': 327L, u'FECHAALTA': 2(
- Layer:  manzanas_valencia
--  {u'AREA': 1061L, u'FECHAALTA': 20011119, u'MUNICIPI
```

```python
from gvsig import *

def infoInter(featurePoint, lstLayers):
    pointValues = featurePoint.getValues()
    pointValues.pop('GEOMETRY')
    gfeat = featurePoint.geometry()
    print "\nPoint info: ", pointValues
    for layer in lstLayers:
        print "- Layer: ", layer.name
        for pol in layer.features():
            polValues = pol.getValues()
            gpol = pol.geometry()
            polValues.pop('GEOMETRY')
            if gpol.intersects(gfeat):
                print "-- ", polValues

def main(*args):
    #Intersection info
    lyrPoints = currentLayer()

    #Capas diferentes a la de puntos
    lstLayers = []
    for capa in currentView().getLayers():
        if not capa.name == lyrPoints.name:
            lstLayers.append(capa)

    #intersecar puntos
    for point in lyrPoints.features():
        infoInter(point, lstLayers)
```

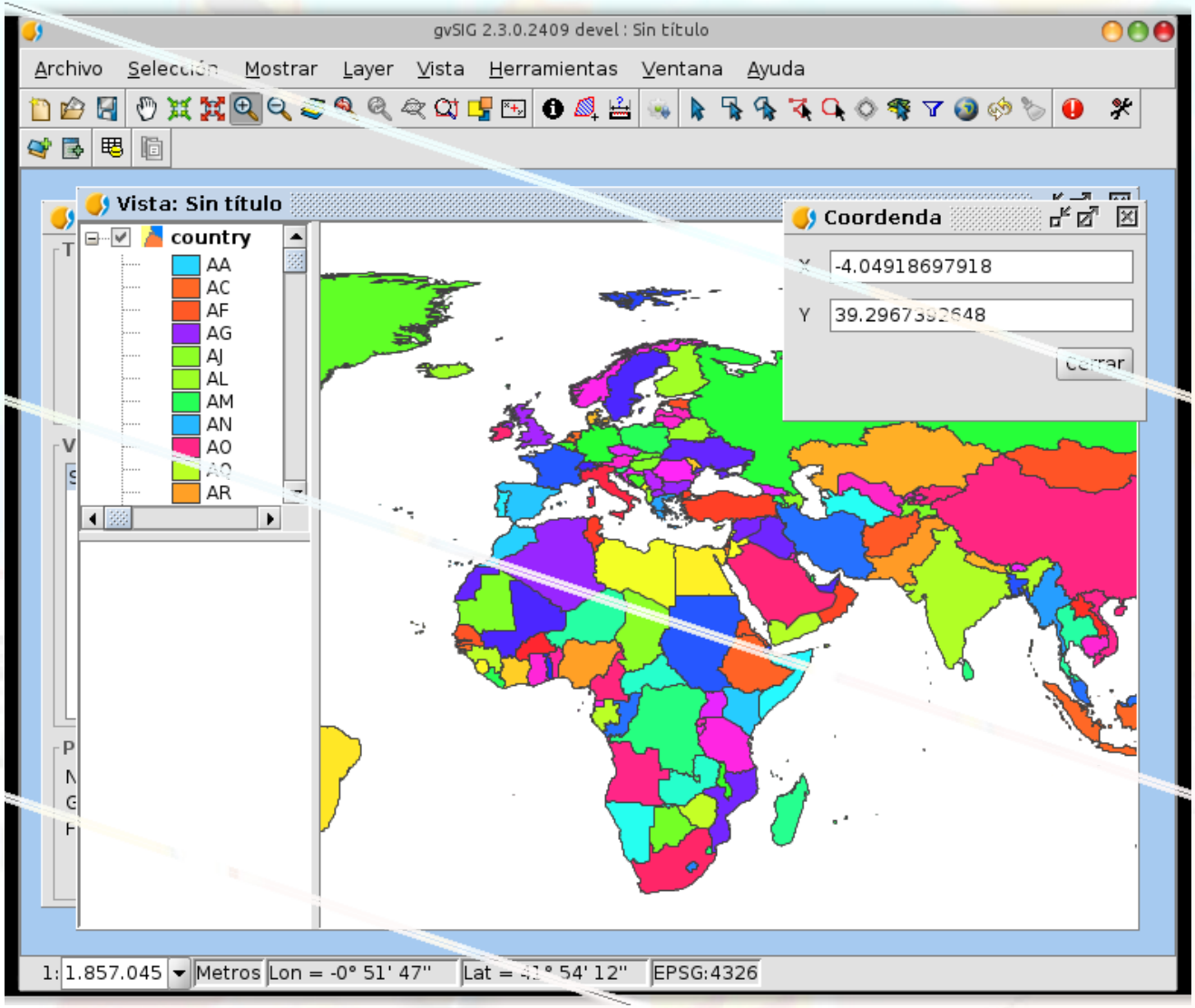
geopy


View: Untitled
[Catastro]
Search Address
Search Address
OK
Cancel
Scripting La...
Usuario
Sistema
Scripts del usuario
Address browser
Geocoding
Geocoding_retocado
PO files exporting
project-manager-resi
PZ. AYUNTAMIENTO
CL. SANGRE
CL. PERIODISTA AZZATI

Monitor



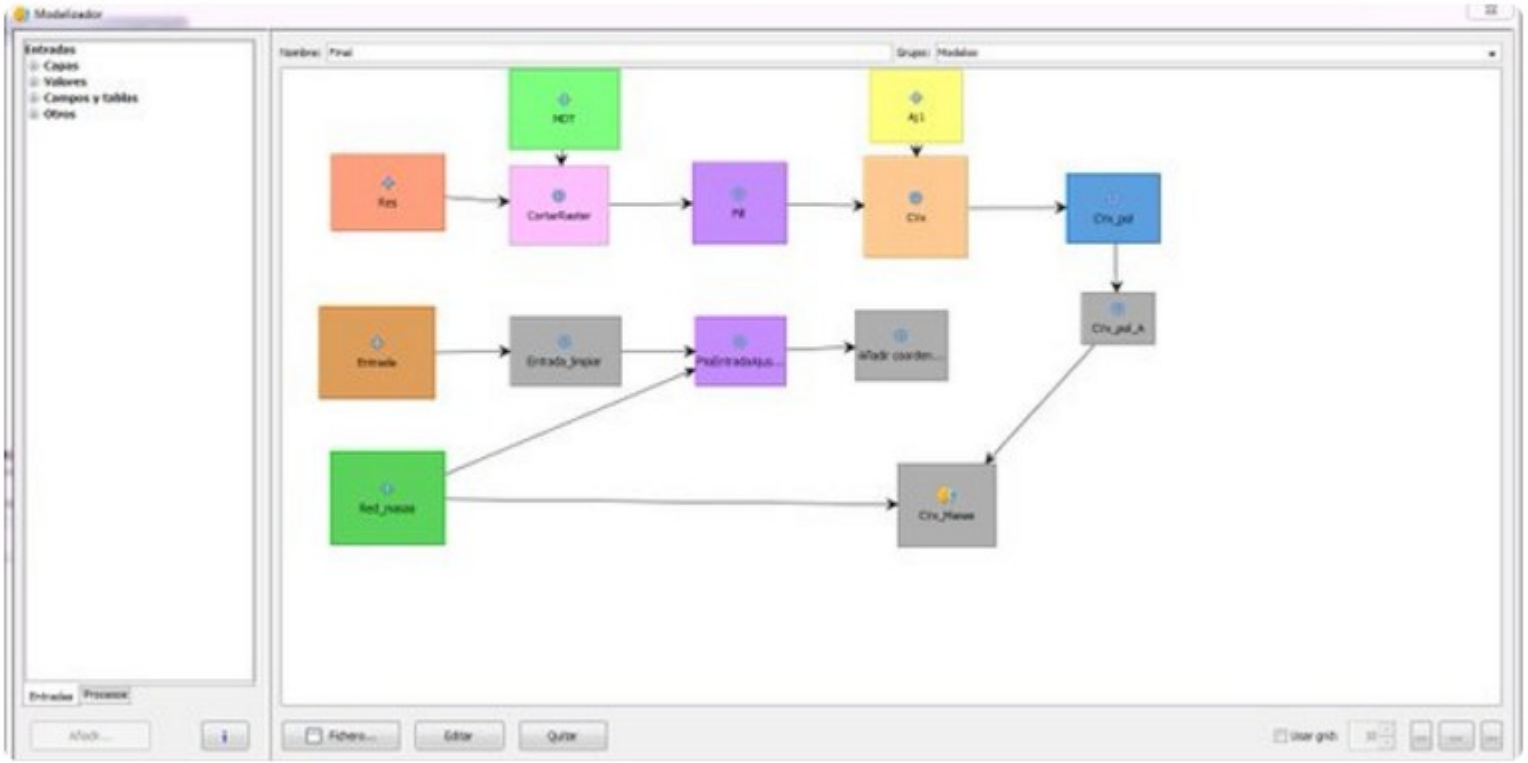
Coordinates



Ideya

ideyared @ideyared · 7 oct.

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



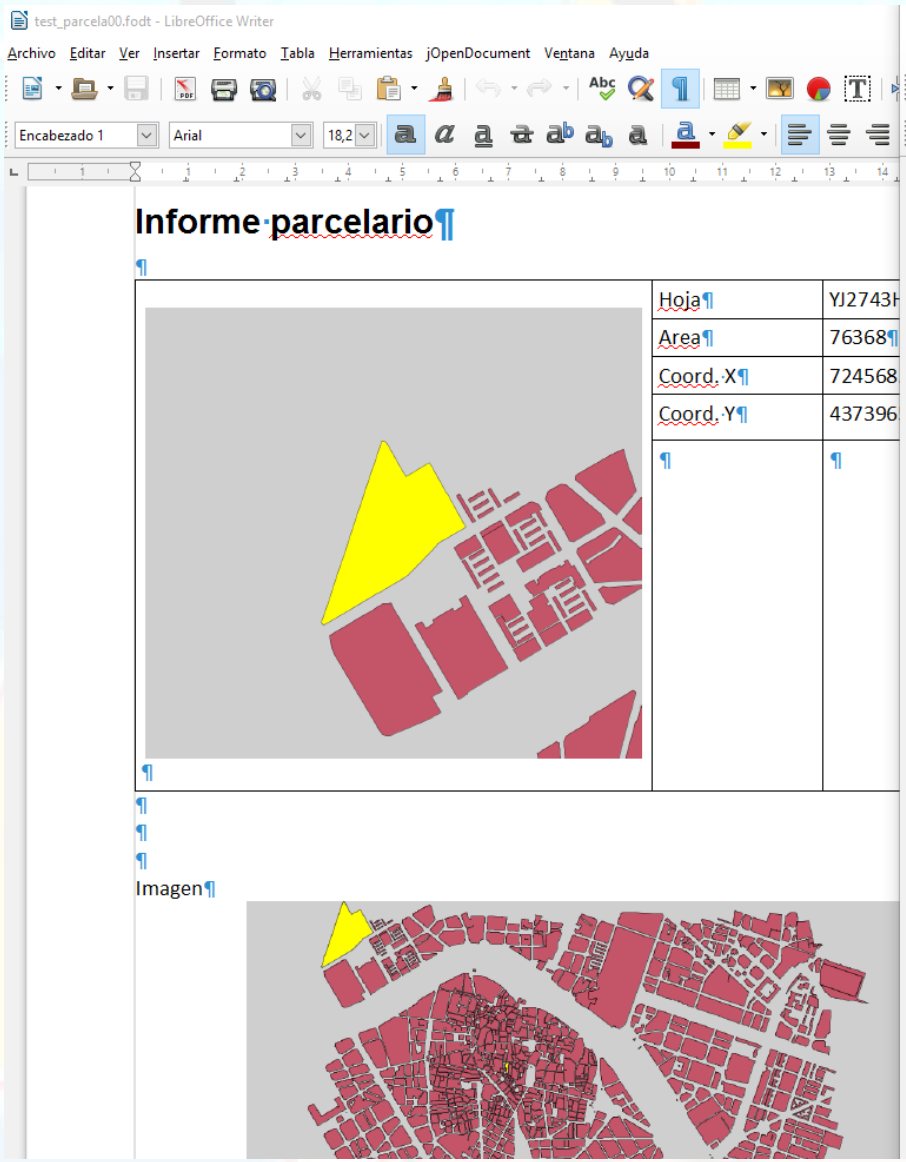
CartoDB

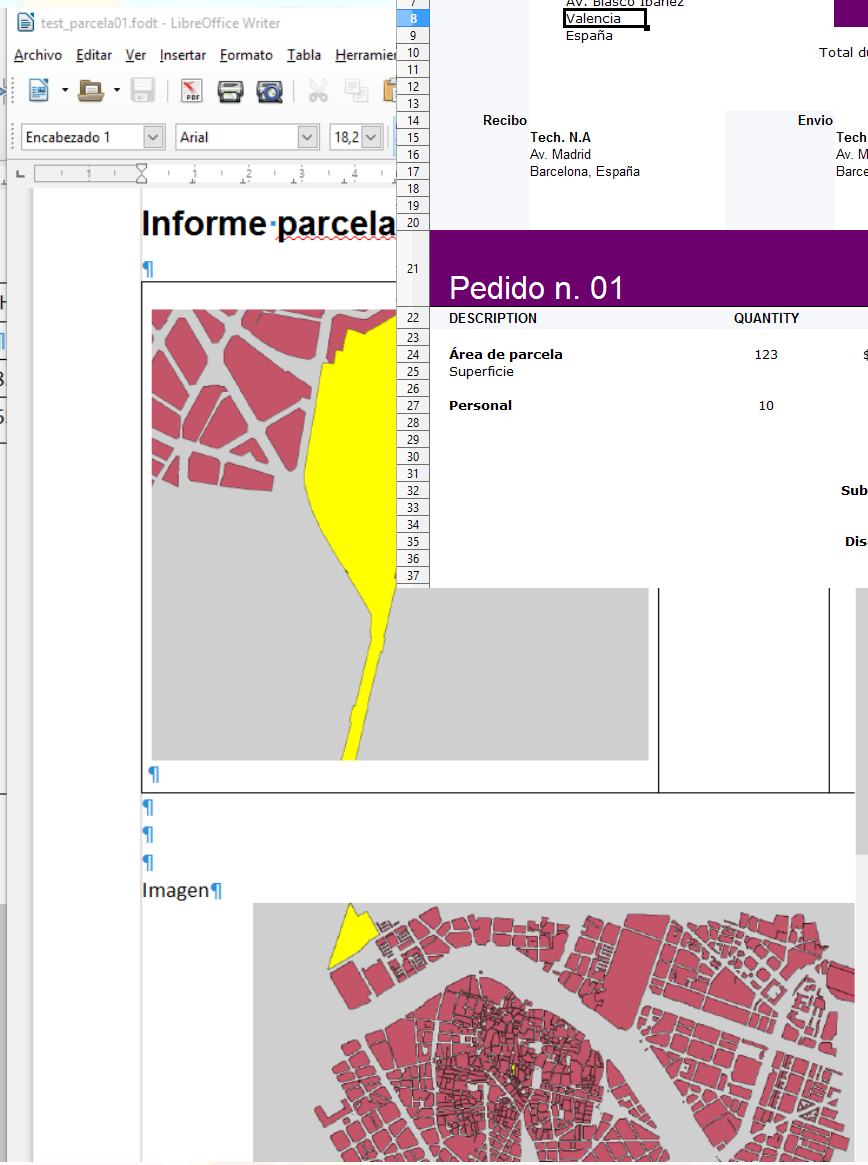
```
1
2  from gvsig import *
3  import cartodb1
4
5  def main(*args):
6      API_KEY ='****'
7      cartodb_domain = 'user'
8      cl = CartoDBAPIKey(API_KEY, cartodb_domain)
9      print(cl.sql('select * from mytable'))
10
```

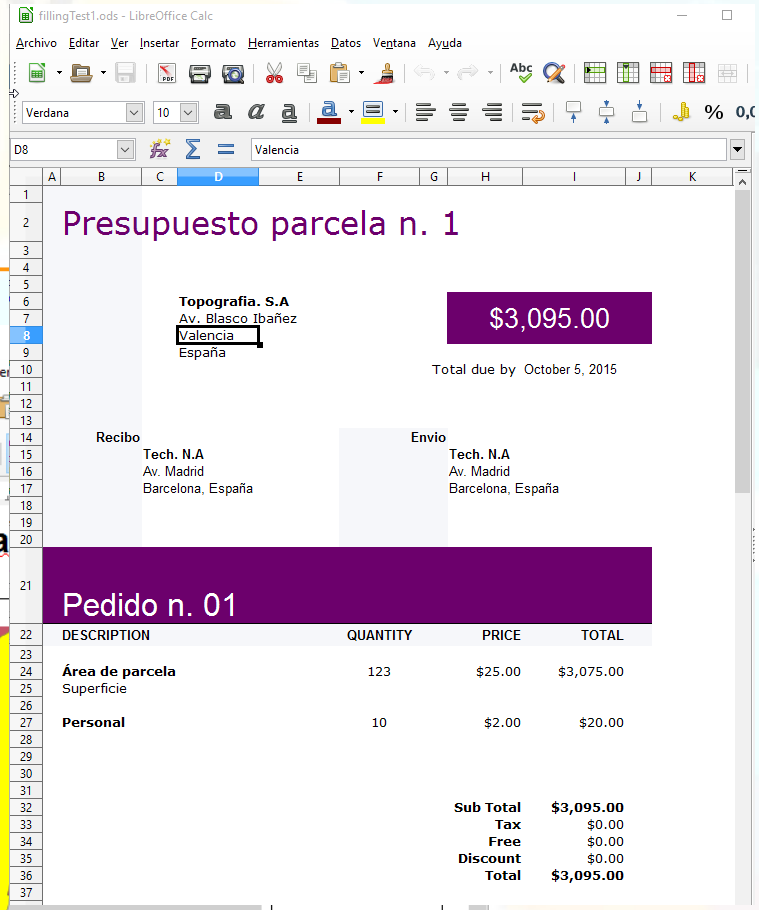
CARTODB
Geospatial on the cloud

https://github.com/CartoDB/cartodb-python

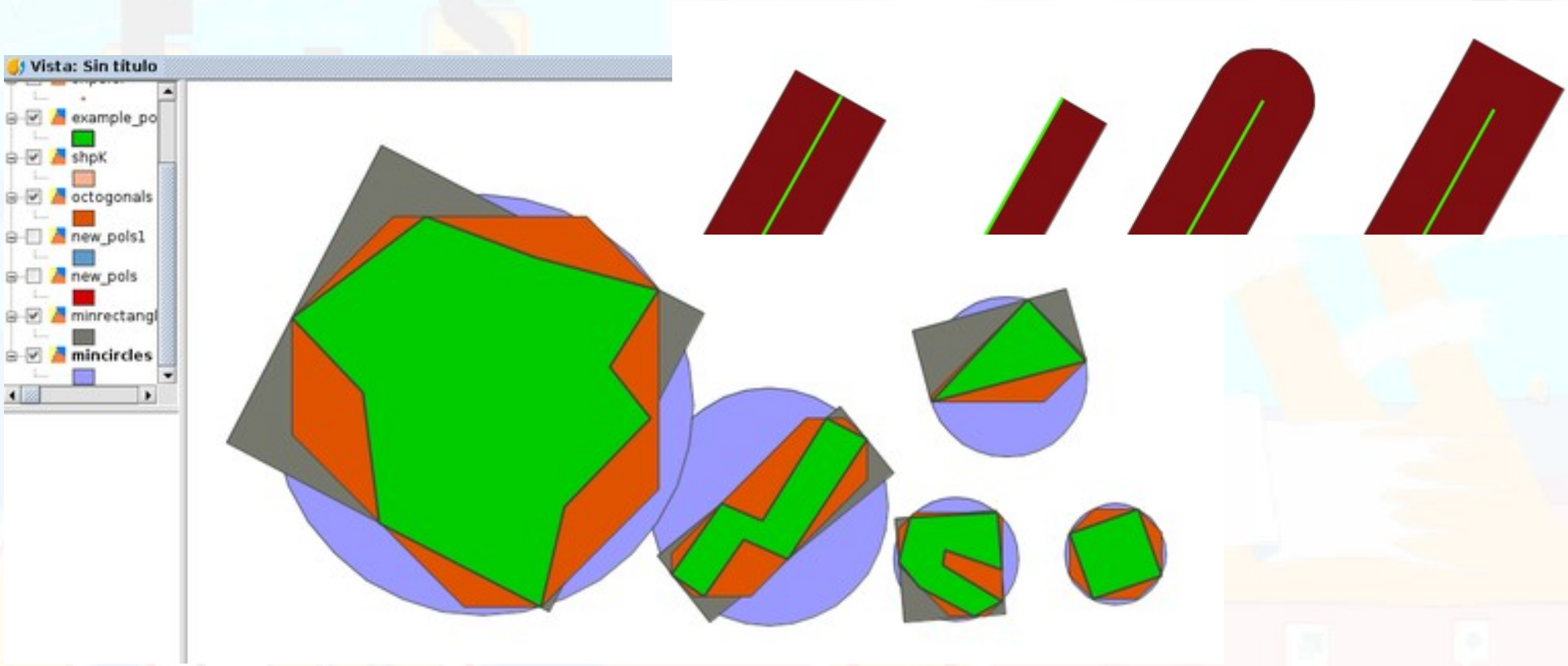
jOpenDocument







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

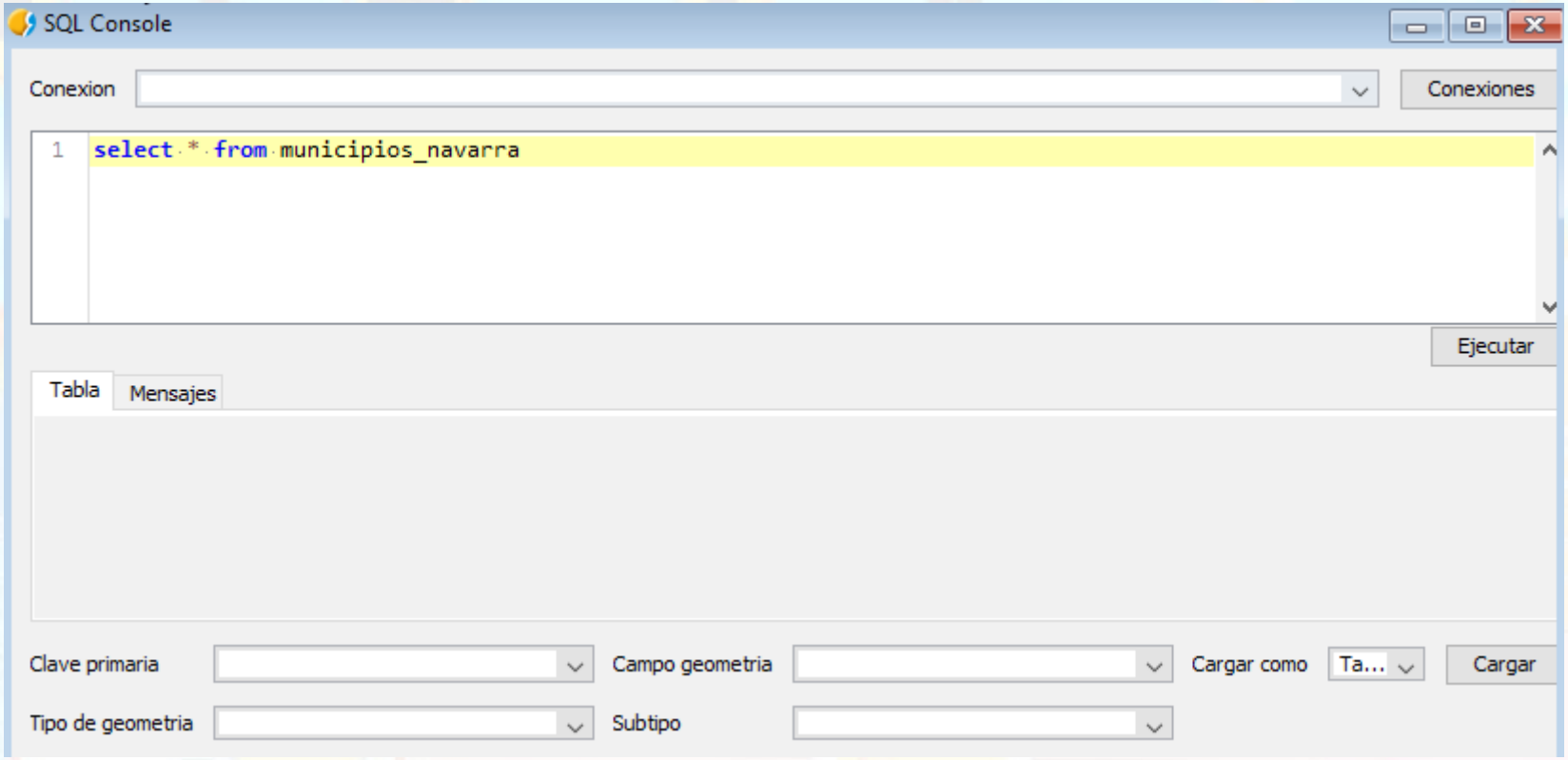
```
from gvsig import *
from com.vividsolutions.jts.io import WKTReader

from java.util import List

from org.geotools.data.simple import SimpleFeatureCollection
from org.geotools.feature import DefaultFeatureCollection
from org.geotools.feature.simple import SimpleFeatureBuilder
from org.geotools.feature.simple import SimpleFeatureTypeBuilder
from org.geotools.referencing.crs import DefaultGeographicCRS
from org.jgrasstools.gears.modules.v.smoothing import OmsLineSmoother Jaitools
from org.jgrasstools.gears.modules.v.smoothing import OmsLineSmoother McMaster
from org.jgrasstools.gears.utils.features import FeatureUtilities
from org.opengis.feature.simple import SimpleFeature
from org.opengis.feature.simple import SimpleFeatureType

from com.vividsolutions.jts.geom import Geometry
from com.vividsolutions.jts.geom import LineString

from java.lang import Integer

def main(*args):
    #https://code.google.com/p/jgrasstools/source/browse/jgrassgears/src/test/java/org

    print "Test JGrassTools: TestLineSmoother Jaitools\n"
    b = SimpleFeatureTypeBuilder()
```

Code | Properties

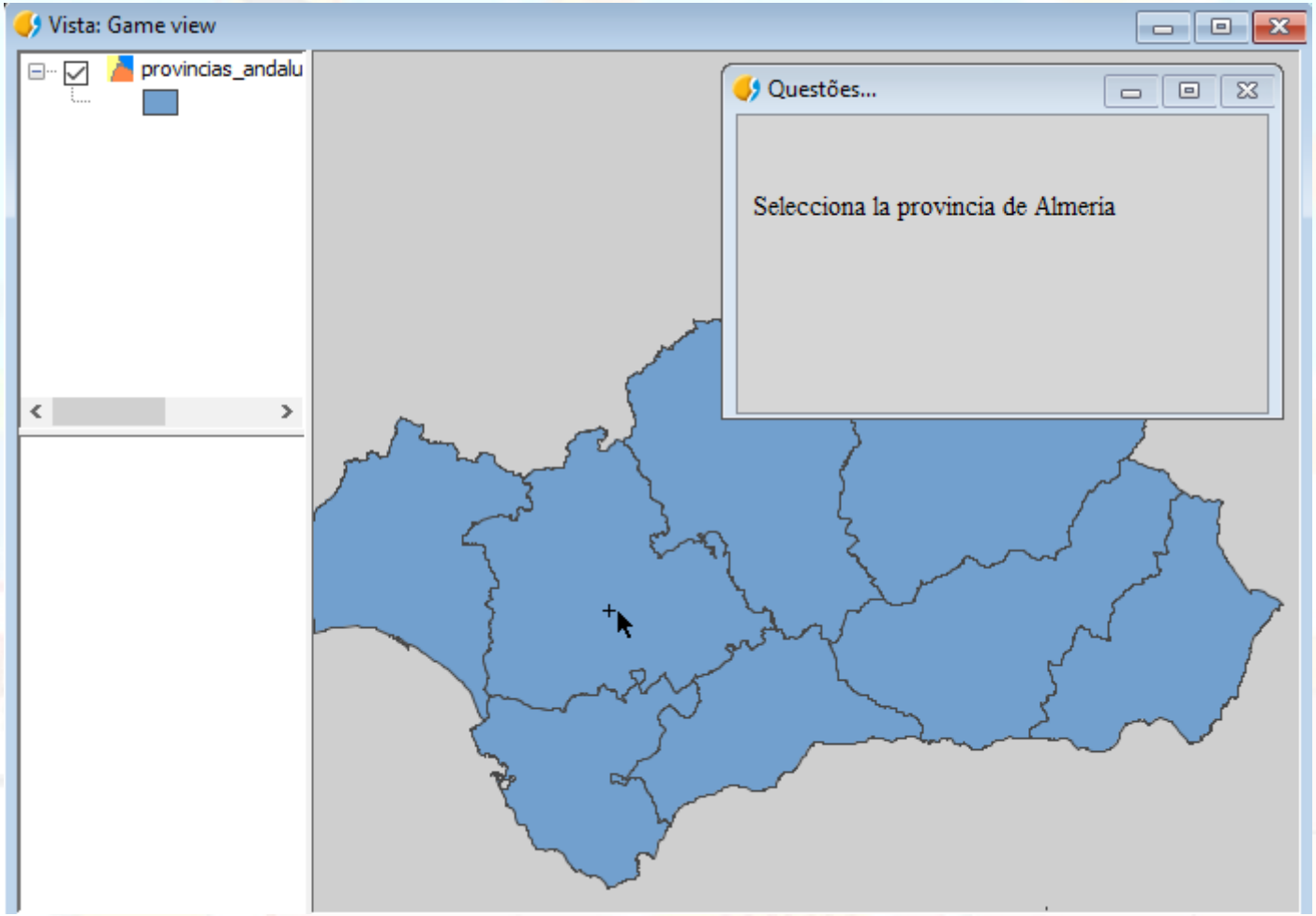
Problems | Console

```
Running script test_jgrasstools_2.
Test JGrassTools: TestLineSmoother Jaitools

Collecting geometries...
Smoothing features...
100%...
Finished.

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

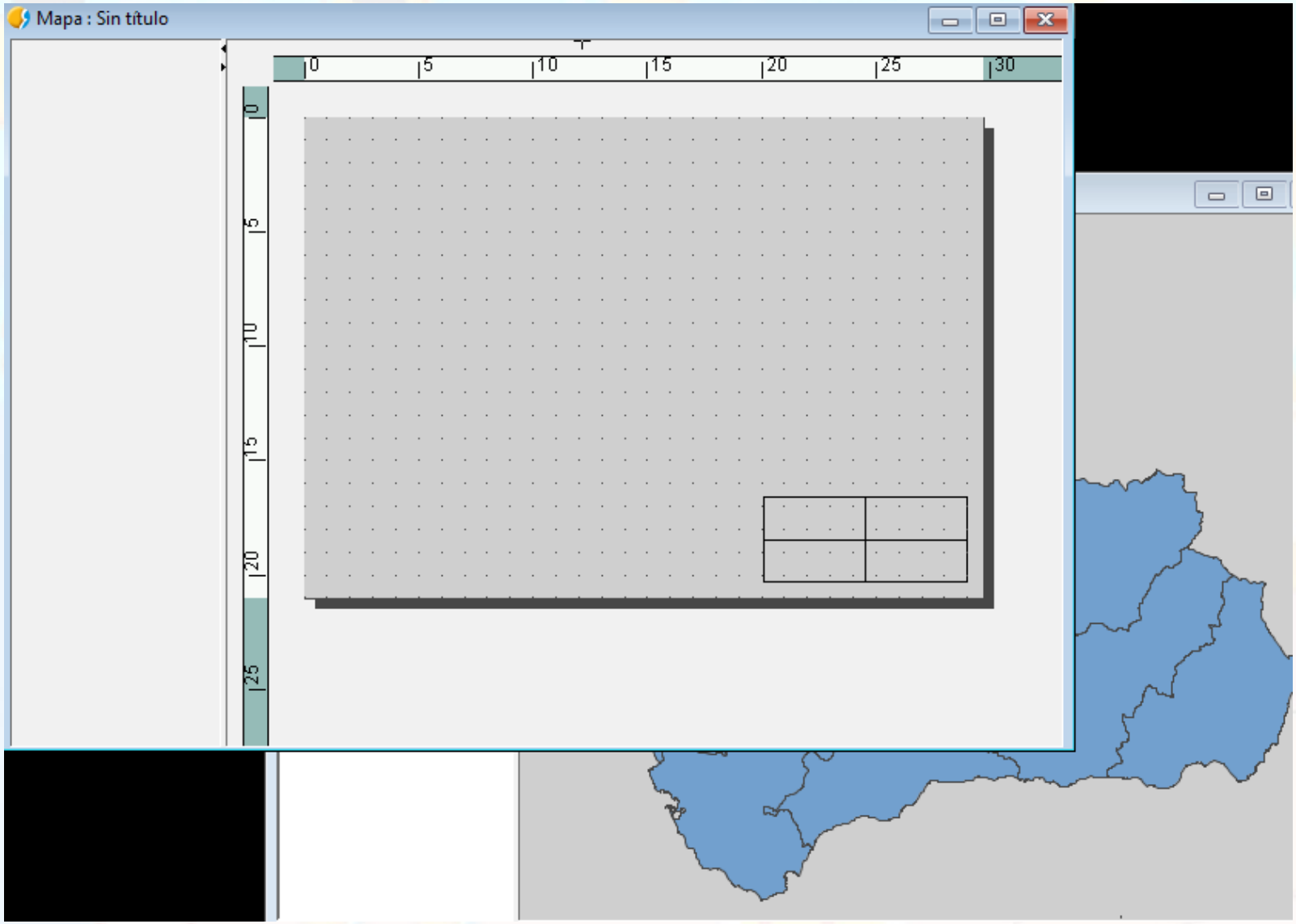
SQL Console


SQL Console
Conexion
Conexiones
1
select * from municipios_navarra
Ejecutar
Tabla
Mensajes
Clave primaria
Campo geometria
Cargar como
Ta...
Cargar
Tipo de geometria
Subtipo

Game


Vista: Game view
provincias_andalu
Questões...
Selecciona la provincia de Almeria

Automate maps creation



Geotagging Digital Photos

```
    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 from MOOC students

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)

More info:

- Mooc: Scripting in gvSIG
- Blog gvSIG
  http://blog.gvsig.org/
- Outreach: Scripts examples
  http://outreach.gvsig.org/scripts/
- Emails lists: Users and developers
  http://www.gvsig.com/es/comunidad/listas-de-correo

CC BY SA

Thanks!

Óscar Martínez
@masquesig
omartinez@gvsig.com

info@gvsig.com
www.gvsig.com