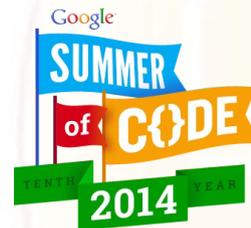
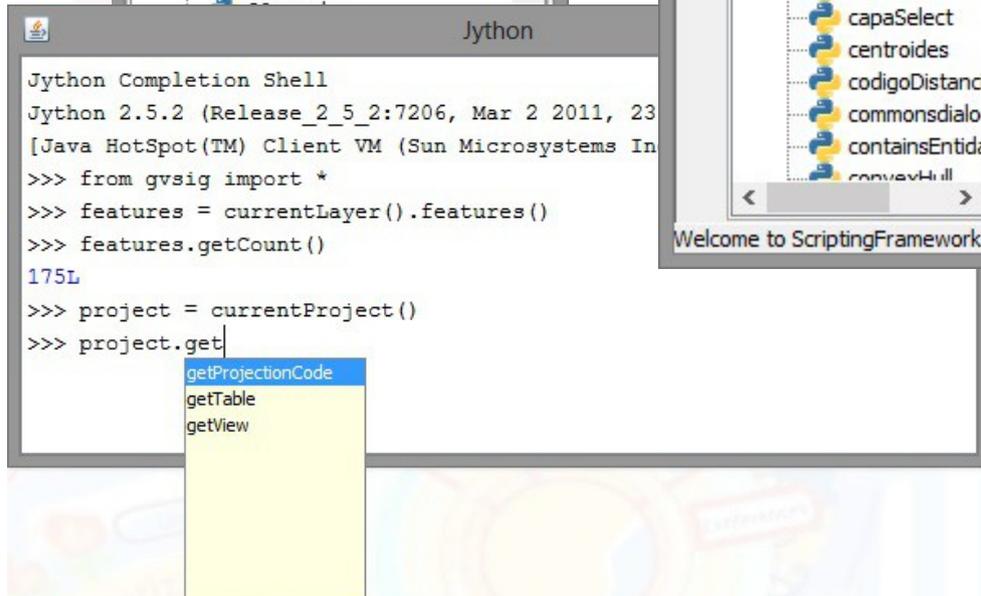
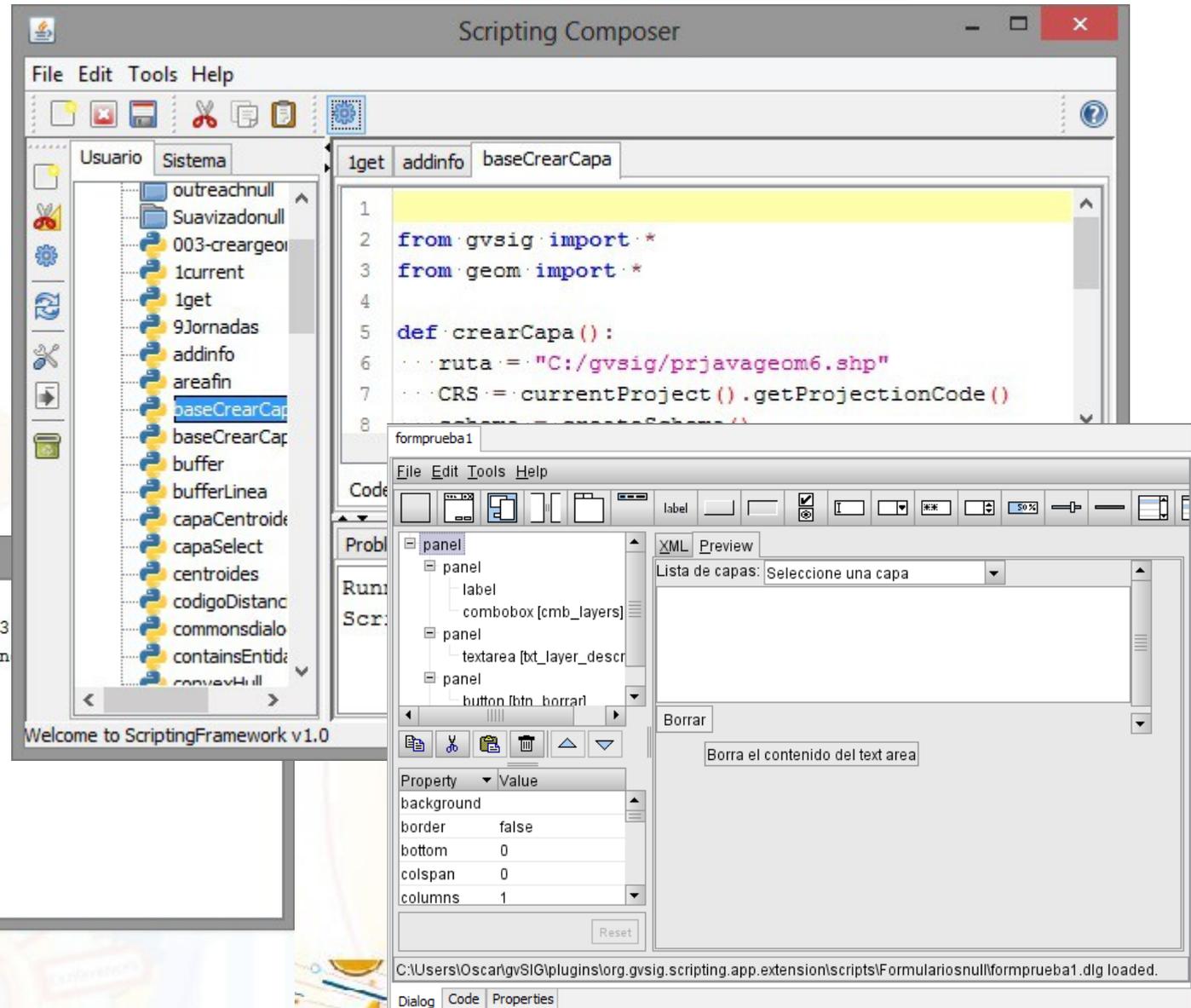
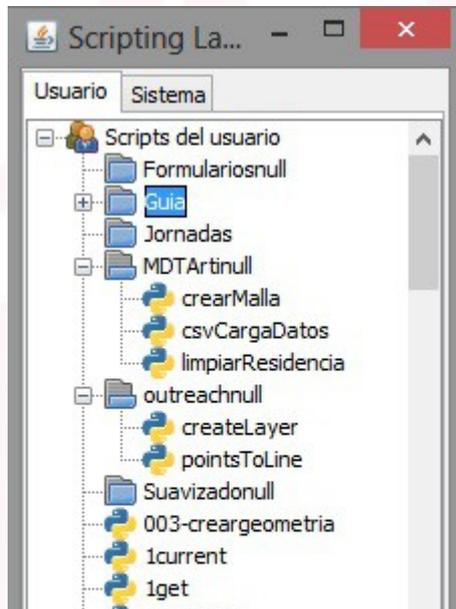


# Introducción al Módulo de Scripting en gvSIG 2.1



# Qué es el Módulo de Scripting



# Que las herramientas de gvSIG 2..

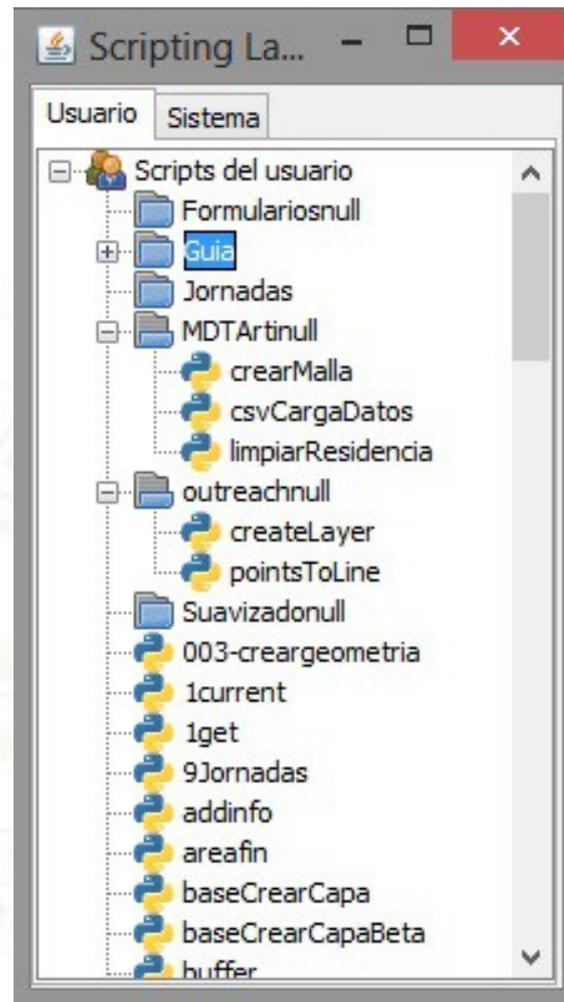
The screenshot displays the gvSIG 2.10.2246 testing interface. The main window shows a map of a region with a network of roads and red dots representing nuclei. The 'Tabla de atributos: NUCLEOS' window is open, showing a table with columns for AREA, PERIMETER, NUCLEOS\_ID, NOMBRE, CODMUN, and CODPOB. The 'Caja de herramientas - 355 H...' window is also open, showing a list of algorithms and tools.

NUC	AREA	PERIMETER	NUCLEOS_	NUCLEOS_ID	NOMBRE	CODMUN	CODPOB
1	0,000	0,000	32	400	PUEBLA D...	49166	0002
2	0,000	0,000	188	414	FABERO	24070	0002
3	0,000	0,000	189	407	TORENO	24169	0009
4	0,000	0,000	190	409	VILLAFRA...	24209	0017
5	0,000	0,000	191	408	CACABELOS	24030	0002
6	0,000	0,000	192	410	TORAL DE...	24206	0301
7	0,000	0,000	193	415	PONFERR...	24115	0021
8	0,000	0,000	194	412	BEMBIBRE	24014	0002
9	0,000	0,000	195	404	VILLAR DE...	24115	0035
...	0,000	0,000	196	403	PEÑALBA ...	24115	0019
...	0,000	0,000	258	420	PUEBLA D...	24121	0101
...	0,000	0,000	260	494	PORTILLA ...	24020	0006
...	0,000	0,000	338	558	VILLASAN...	09410	0057

0 / 459 Total registros seleccionados.

Metros Lon = 513578° 32' 56" Lat = 4482754° 50' 13" EPSG:4326

..no sean el límite



# Objetivo del Módulo

- Acceso a todas las herramientas
  - Oportunidad de crear nuevas
- Rápido acceso y automatización de tareas
- Módulo de programación orientado a Usuarios

# Scripts

```

pygeoj_01
1
2 import pygeoj
3
4 def main(*args):
5     testfile = pygeoj.load("E://bars.geojson")
6
7     for feature in testfile:
8         print feature.properties
9         print feature.geometry.coordinates
10
11
Code Properties
Problems Console
Running script pygeoj_01.
{'marker-symbol': 'bar', 'address': 'u'1719 G St NW', 'name': 'u'The Exchange'}
[-77.039882, 38.898321]
{'marker-symbol': 'bar', 'address': 'u'1620 I St NW', 'name': 'u'Blackfin'}
[-77.038193, 38.901345]
{'note': 'u'Ask the bartender for the password', 'marker-symbol': 'bar', 'address': 'u'
[-77.03195, 38.907826]
{'marker-symbol': 'bar', 'address': 'u'1025 5th St NW', 'name': 'u'Busboys and Poets'}
[-77.018728, 38.902708]
{'marker-symbol': 'bar', 'address': 'u'2150 P St NW', 'name': 'u'SoHo Cafe'}
[-77.047745, 38.909637]
{'marker-symbol': 'bar', 'address': 'u'475 H St NW', 'name': 'u'Chinatown Coffee Compan
Line 10:0

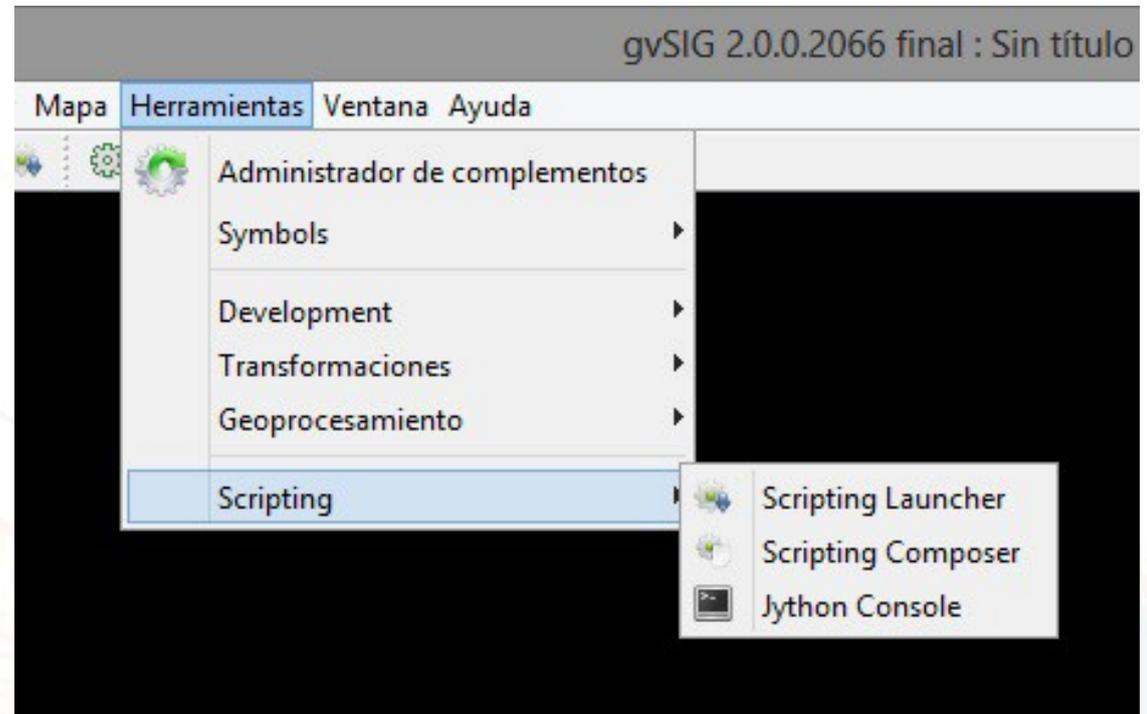
```

# Características

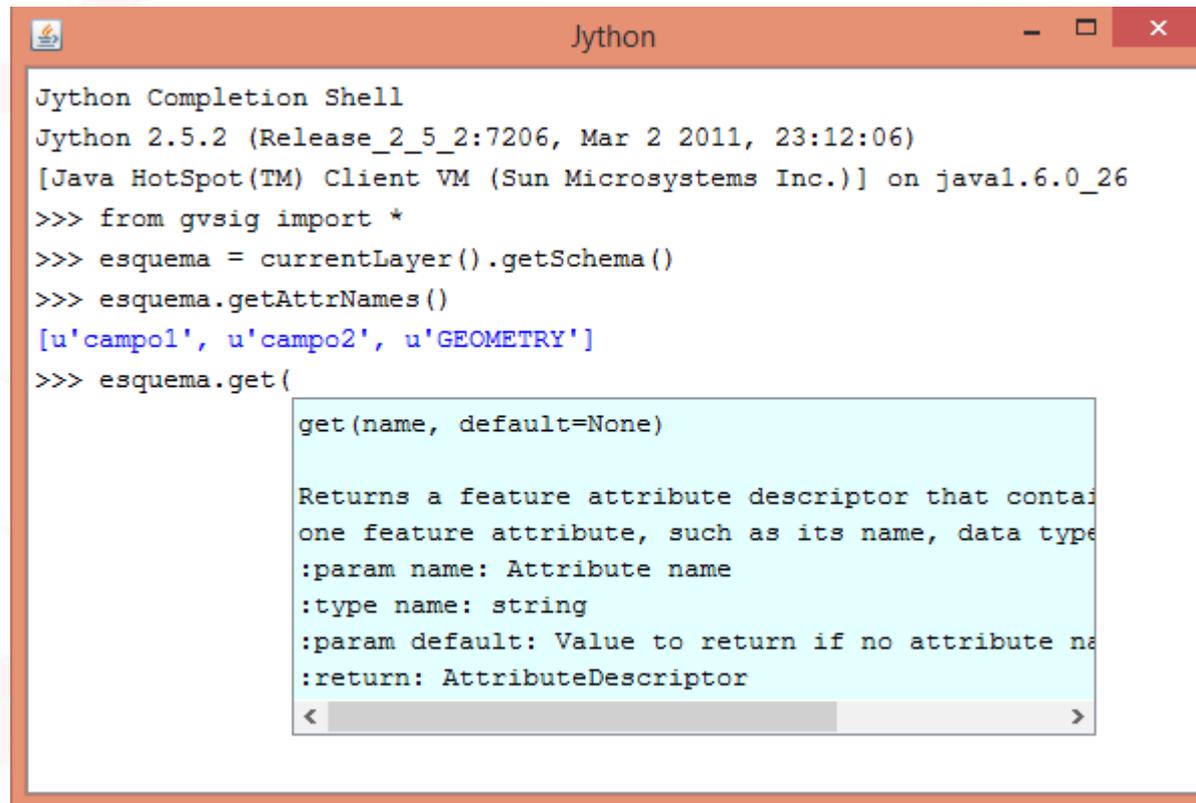
- Permite trabajar con nuestros datos
  - Librerías para el manejo de tablas, capas vectoriales, geometrías y raster
  - Lenguaje utilizado Jython (Python + Java)

# Herramientas

- Consola Jython
- Scripting composer
- Scripting launcher



# Consola Jython



```
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 java1.6.0_26
>>> from gvSIG import *
>>> esquema = currentLayer().getSchema()
>>> esquema.getAttrNames()
[u'campo1', u'campo2', u'GEOMETRY']
>>> esquema.get(
    get(name, default=None)

Returns a feature attribute descriptor that contains
one feature attribute, such as its name, data type
:param name: Attribute name
:type name: string
:param default: Value to return if no attribute name
:return: AttributeDescriptor
```

# Scripting Composer

The image displays the Scripting Composer application interface. The main window is titled "Scripting Composer" and features a menu bar (File, Edit, Tools, Help) and a toolbar. On the left, there is a file explorer showing a tree structure under "Usuario" and "Sistema" folders, with "baseCrearCapa" selected. The central area is a code editor with a yellow background, containing the following Python code:

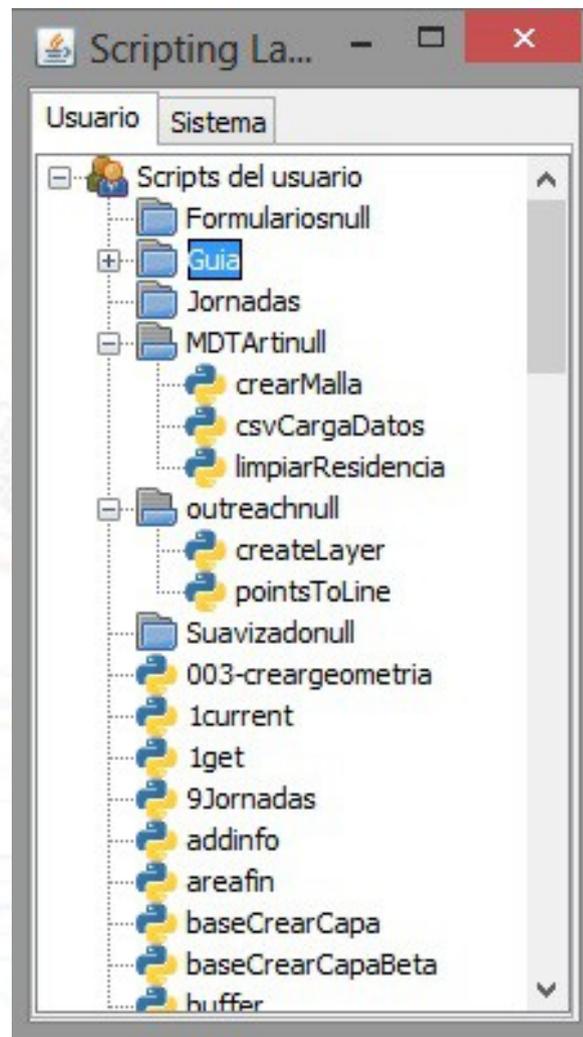
```
1  
2 from gvSIG import *  
3 from geom import *  
4  
5 def crearCapa():  
6     ruta = "C:/gvSIG/prjavageom6.sh  
7     CRS = currentProject().getProje  
8     schema = createSchema()
```

Below the code editor are tabs for "Code" and "Properties". A "Problems" and "Console" panel at the bottom shows the output: "Running script baseCrearCapa. Script baseCrearCapa terminated." A second window titled "formprueba1" is overlaid on the right, showing a dialog box with a "Lista de capas" dropdown menu, a "Borrar" button, and a text area containing "Borra el contenido del text area". The dialog box also has a "Property" table and a "Reset" button.

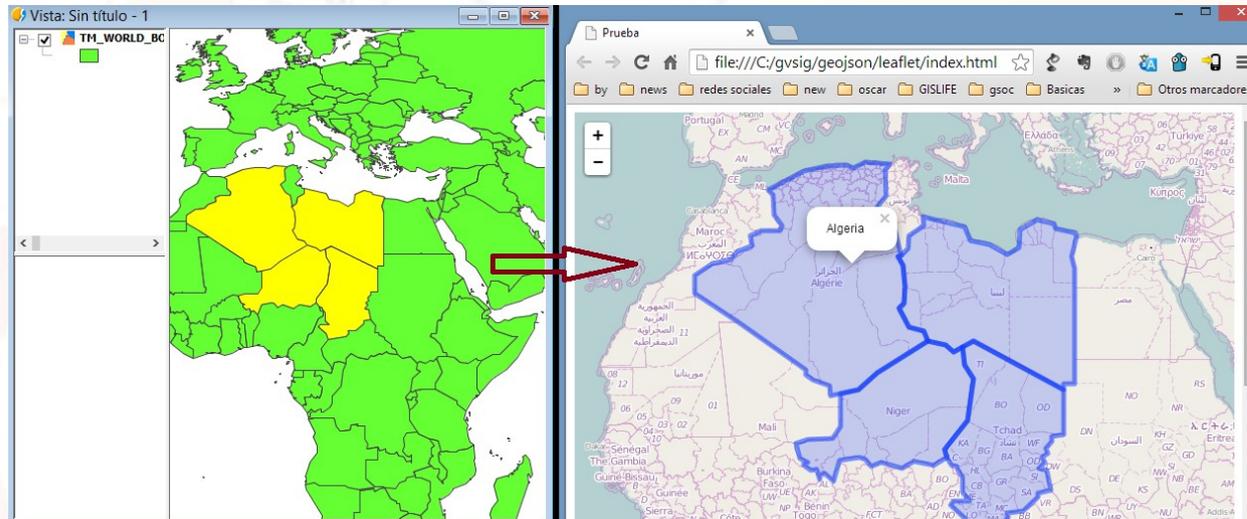
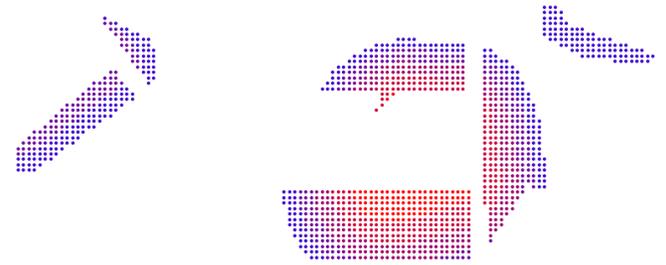
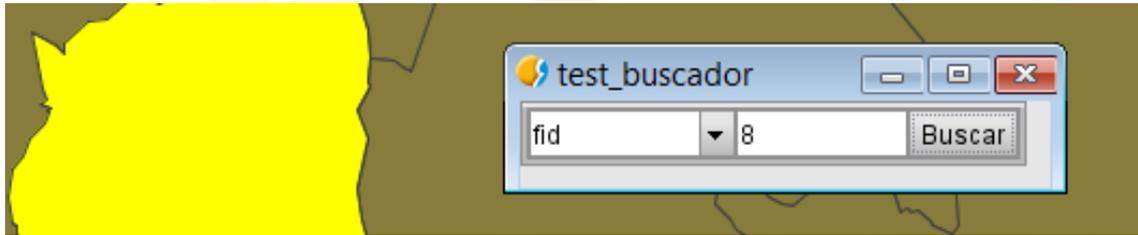
Property	Value
background	
border	false
bottom	0
colspan	0
columns	1

At the bottom of the dialog box, the following text is visible: "C:\Users\Oscar\gvSIG\plugins\org.gvsig.scripting.app.extension\scripts\Formularios\null\formprueba1.dlg loaded."

# Scripting Launcher



# Ejemplos



# Google Summer of Code

*“Proyecto gvpy:  
Acceso a los geoprocesos  
desde el Módulo de Scripting”*

<https://github.com/oscar9/gvpy>



# Objetivo del GSOC

- Aumentar la potencia del módulo
- Acceder a unas herramientas ya existentes y de uso común
- Función demandada por los usuarios

# Ejemplo de uso

```
gvpy.runalg("randomvector", COUNT=10, TYPE=0)
```

```
v1 = gvpy.runalg("randomvector", 100, 2)
```

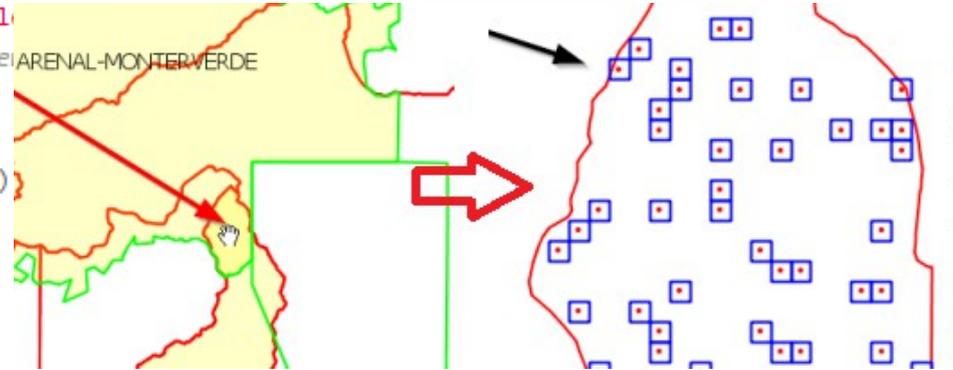
```
v2 = gvpy.runalg("perturbatepointslayer",  
    LAYER=v1,  
    MEAN=10,  
    STDDEV=10,  
    EXTENT=currentView(),  
    PATH="C://gvSIG//azpoints.shp"  
)
```

# Ejemplo: Áreas de estudio

```

44 gridSelect = gvpn.newLayer(grid, "C://gvSIG//grid_sel
45 #If reticula intersect with areaInterest, add to a new
46 for area in intersection[1].features():
47     for ret in grid.features():
48         if ret.geometry().intersects(area.geometry()):
49             gridSelect.append(ret.getValues())
50 gridSelect.commit()
51
52
53 #Random points inside shape "1" area of interest
54 fitpoints = gvpn.runalg("fitpointsinpolygon",intersection[1], 2, "74", 1, 0, PATH="C://gvSIG//fitpoints"+endit)
55
56 #New Layer with points inside grid
57 gridFinal = gvpn.newLayer(gridSelect, "C://gvSIG//grid_final"+endit)
58 for point in fitpoints.features():
59     for ret in gridSelect.features():
60         if ret.geometry().intersects(point.geometry()):
61             gridFinal.append(ret.getValues())
62 gridFinal.commit()
63
64 #Centroids
65 centroids = gvpn.runalg("centroids", gridFinal, PATH="C://gvSIG//centroids"+endit)
66 gvpn.removeField(centroids, "X")
67 gvpn.removeField(centroids, "Y")
68
69 #Pointcoordinates: add coordinates to the table
70 centroidsXY = gvpn.runalg("pointcoordinates", centroids, PATH="C://gvSIG//centroidsXY"+endit)
71 gvpn.modifyField(centroidsXY, "X", iType="FLOAT", newField="X_1")
72 gvpn.modifyField(centroidsXY, "Y", iType="FLOAT", newField="Y_1")

```



# Otras herramientas

- Librería

Común: `gvpy.removeField()`, `gvpy.copyLayer()`

Nuevos: `currentRaster()`

- Exportar modelos a script

`gvpy.model2script(pathModel, pathNewScript)`

Nombre:  Group:

```

graph TD
    A[+] --> B[Valor numérico1]
    B --> C[Capa vectorial con geometrías aleatorias]
    C --> D[Convertir geometrías en puntos]
        
```

**Modelizador**

```

<?xml version='1.0' encoding='ISO-8859-1' stan
<model name="test" group="Modelos">
  <algorithm key="PROC0" alg_cmd_line_na
    <assignment key="COUNT" assigned_t
    <assignment key="TYPE" assigned_to
  </algorithm>
  <algorithm key="PROC1" alg_cmd_line_na
    <assignment key="INPUT" assigned t
  </algorithm>
  <input name="INPUT0" description="Valo
    <attribute name="min" value="-2.14
    <attribute name="max" value="2.147
    <attribute name="default" value="5
    <attribute name="type" value="1" /
  </input>
        
```

**Modelo SEXTANTE**

```

1 # Modelo de SEXTANTE: test
2 import gvpy
3 import gvsig
4 import geom
5
6 def main(*args):
7     INNERPARAM0PROC0 = 0
8     INNERPARAM1PROC0 = 0
9     INPUT0 = 5.0
10
11 # "Resultado" de Capa vector:
12 RESULTPROC0 = gvpy.runalg("
13
14 # "Puntos" de Proceso 1: Con
15 POINTSPROC1 = gvpy.runalg("
        
```

**SCRIPTING**

# MOOC

## ***“Introducción a Scripting en gvSIG 2.1”***

*Curso en línea masivo y abierto*

gvSIG training  gvSIG  
association

*+550 participantes*

# MOOC

- Contenido totalmente gratuito
- Nuevos manuales y contenido audiovisual
- Foro de dudas y extras
- **Opcional:** obtención de certificación del curso con 30 créditos abonando 40€ habiendo superado el curso + proyecto final.

# MOOC

## Tema 1

Introducción a gvSIG, el uso básico del Módulo de Scripting y utilización de la librería commonsdialog para la creación de ventanas emergentes

- 1.1 Introducción el MOOC de Scripting
- Guía Semana 1
- 1.2 Módulo de Scripting
- 1.3 Ejemplo de uso: librería commonsdialog
- Script realizado en el vídeo del ejemplo 1.3
- Ejercicio 1.1
- Cuestionario fin de la semana 1

## Tema 2

- Guía semana 2.1 Acceso a datos
- 2.1 Acceso a la información
- Ejercicio 2.1
- Guía semana 2.2: Clase esquema
- 2.2 La clase Esquema
- Ejercicio 2.2 Esquema

Hasta el 7 de Diciembre la inscripción

Entrega final de ejercicios el día  
31 de Diciembre

¡Aún puedes apuntarte!

# ¡Gracias!

Óscar Martínez  
omartinez@gvsig.com  
@masquesig

