

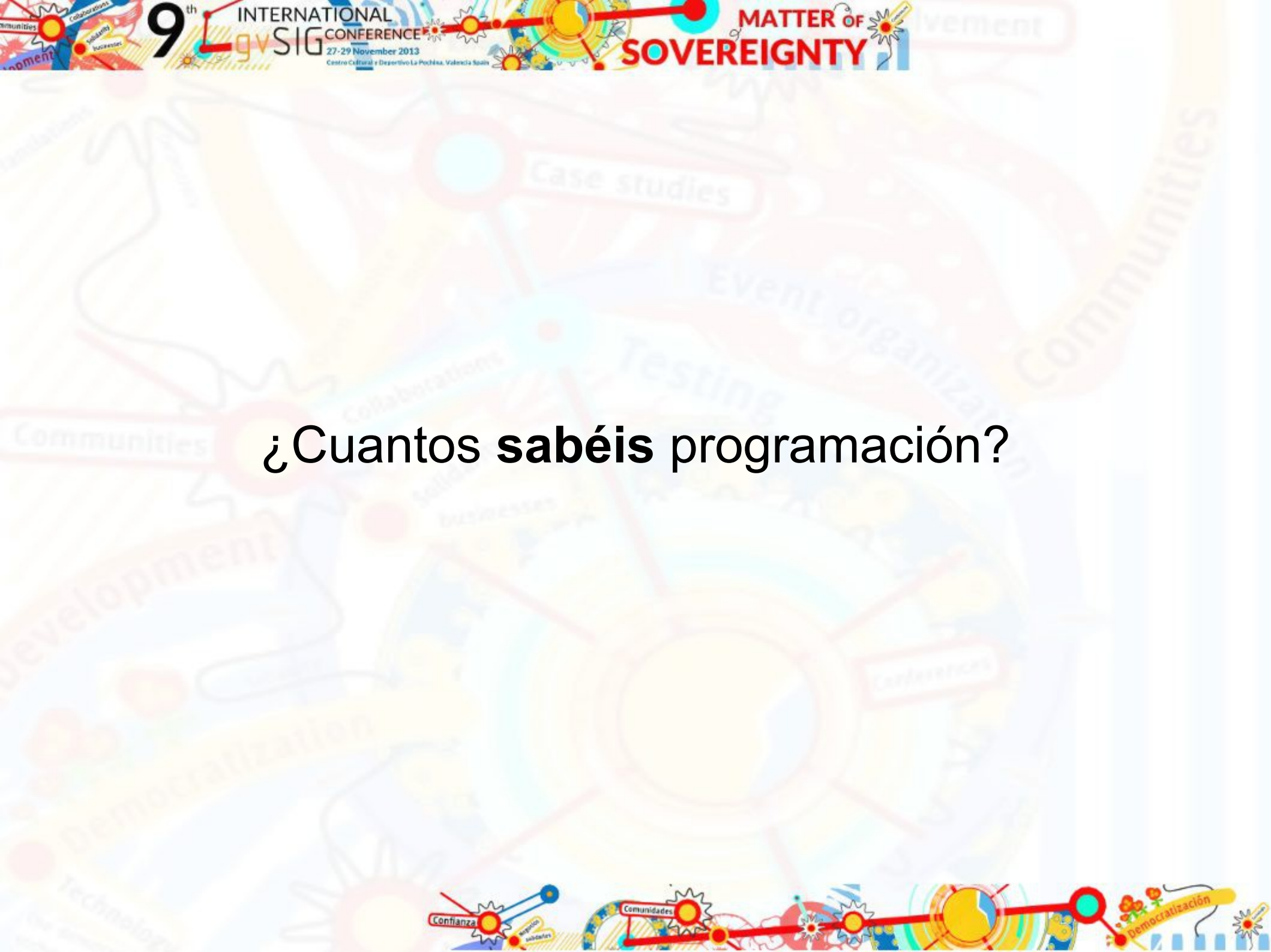
Scripting en gvSIG



Mi perfil inicial

- Estudiante de Topografía en la UPV
- Conocimientos básicos de programación
- Conocimientos nulos sobre programación SIG





9th

INTERNATIONAL
CONFERENCE
27-29 November 2013
Centro Cultural y Deportivo La Pechina, Valencia Spain

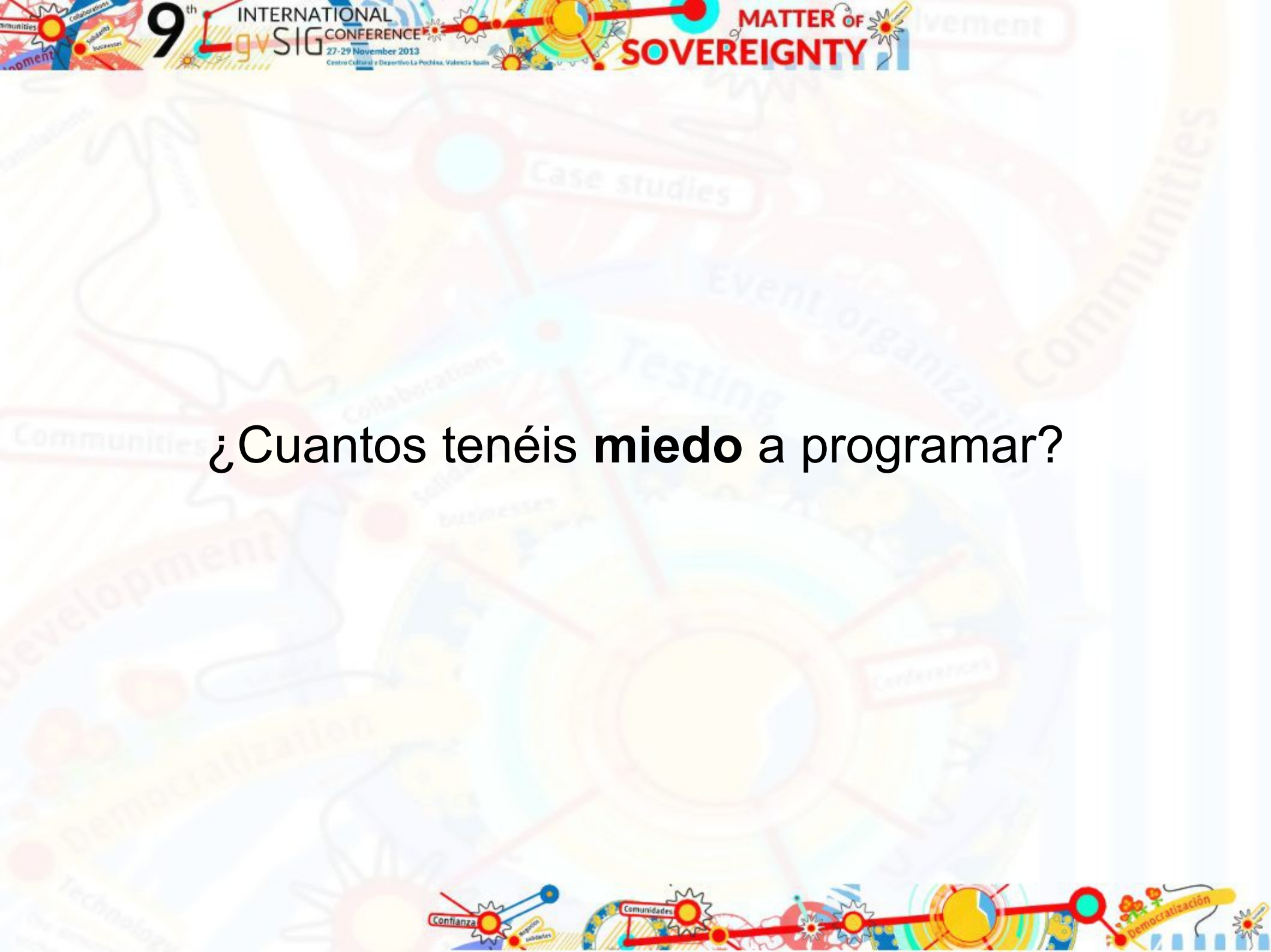


MATTER OF
SOVEREIGNTY



¿Cuántos **sabéis** programación?





9th

INTERNATIONAL
CONFERENCE
27-29 November 2013
Centro Cultural y Deportivo La Pechina, Valencia Spain



MATTER OF
SOVEREIGNTY



¿Cuántos tenéis **miedo** a programar?



Programar no es tan complicado

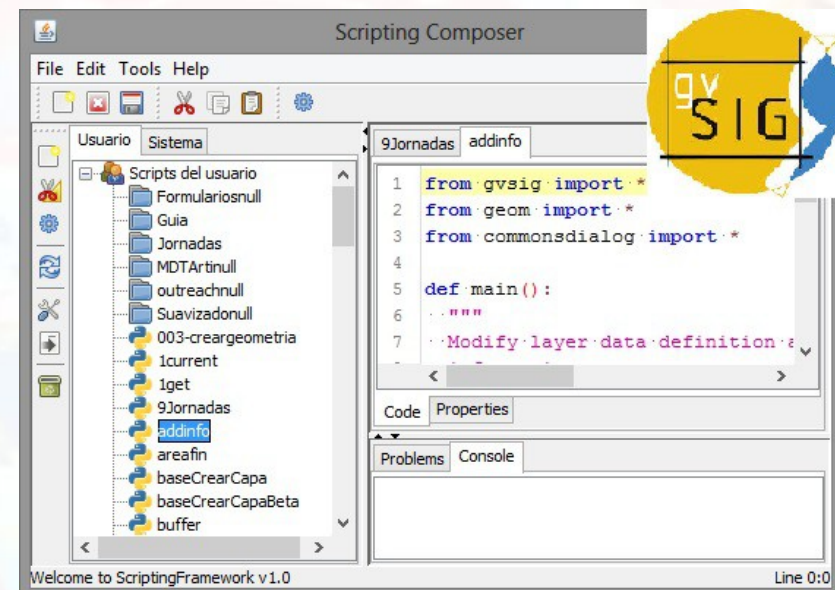
- Buscar una **película**
- Película **en la lista** de películas con 'nombre'
- 'Película' **en** 'Películas' **si** se llama 'nombre'
- **For** película **in** películas:
`if película.NAME == 'nombre': print "Encontrada!"`



Importante al comenzar

- Programación:
 - Lenguaje sencillo
 - Guía de los conceptos básicos
- Programación en SIG:
 - Enseñen su utilidad
 - Ejemplos muy visuales y prácticos





Python + Scripting

Módulo de Scripting

- Integrado en gvSIG
- Lenguaje Jython (Python + Java)
- Permite probar sin salir de gvSIG
- Generar nuestros propios scripts



Librerías

commonsdialog

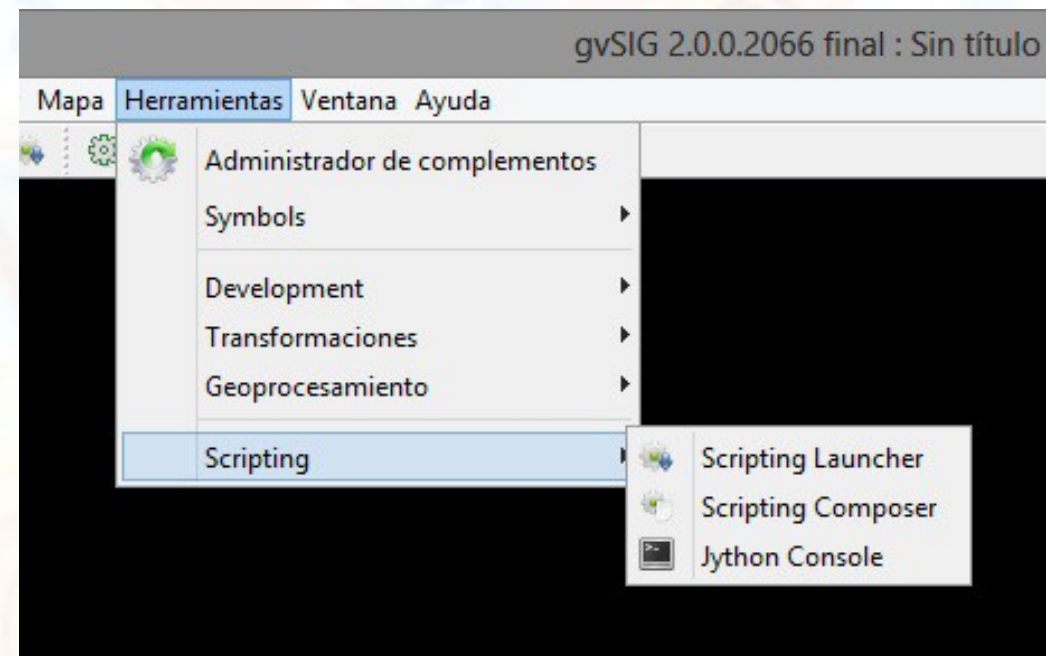
gvSIG

geom

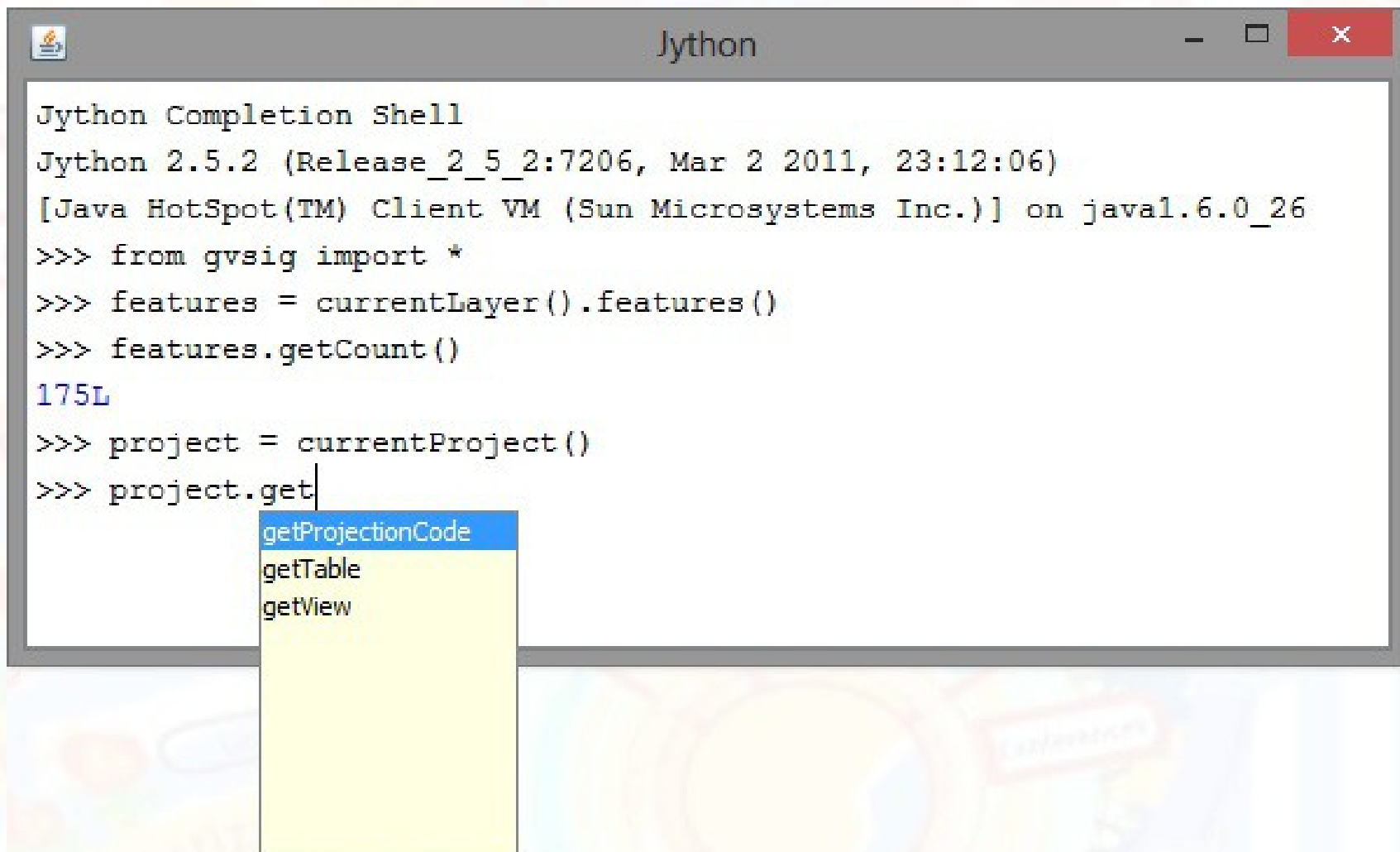
```
from gvSIG import *  
from geom import *  
from commonsdialog import *
```

Herramientas

- Consola Jython
- Scripting composer
- Scripting launcher

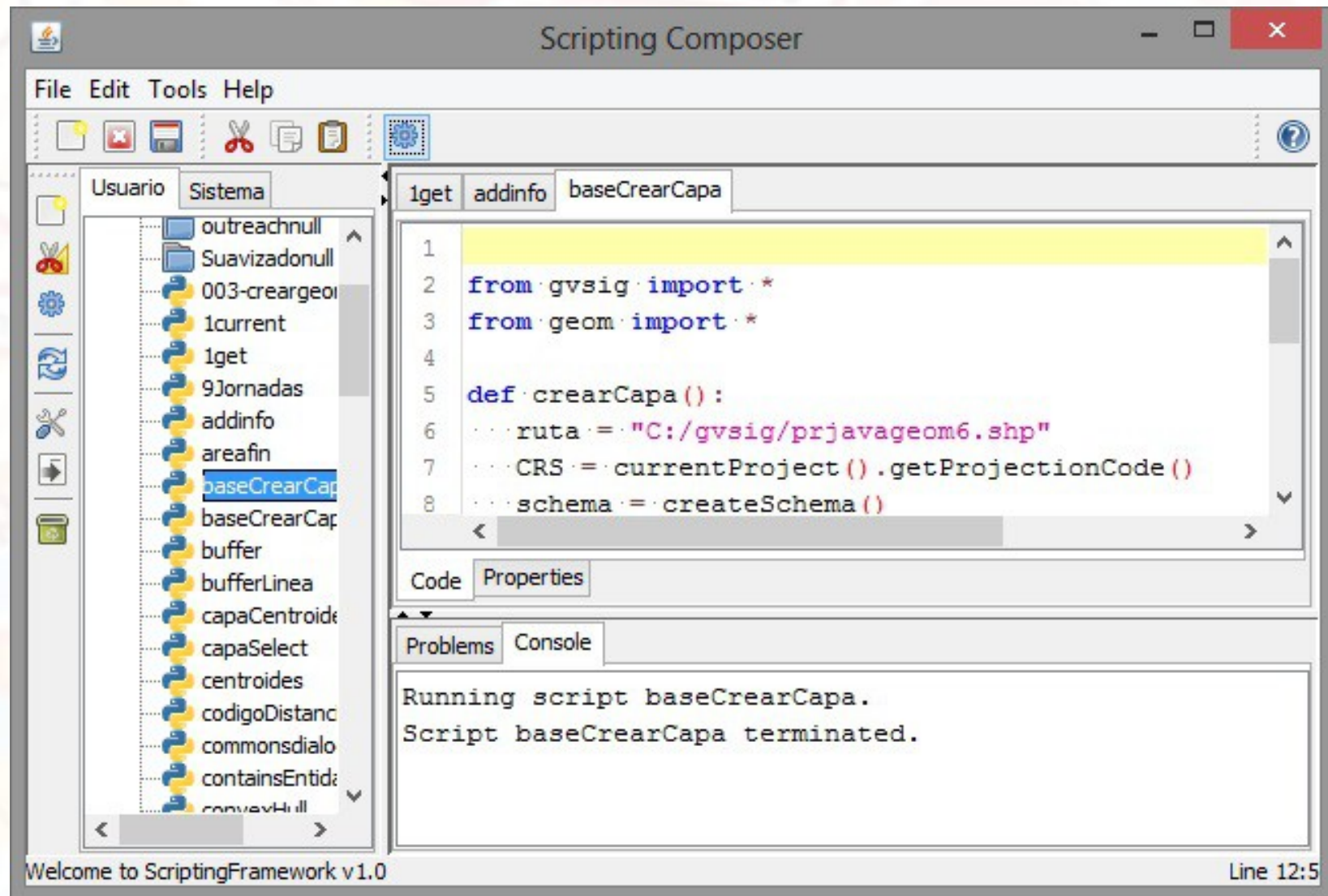


Consola Jython

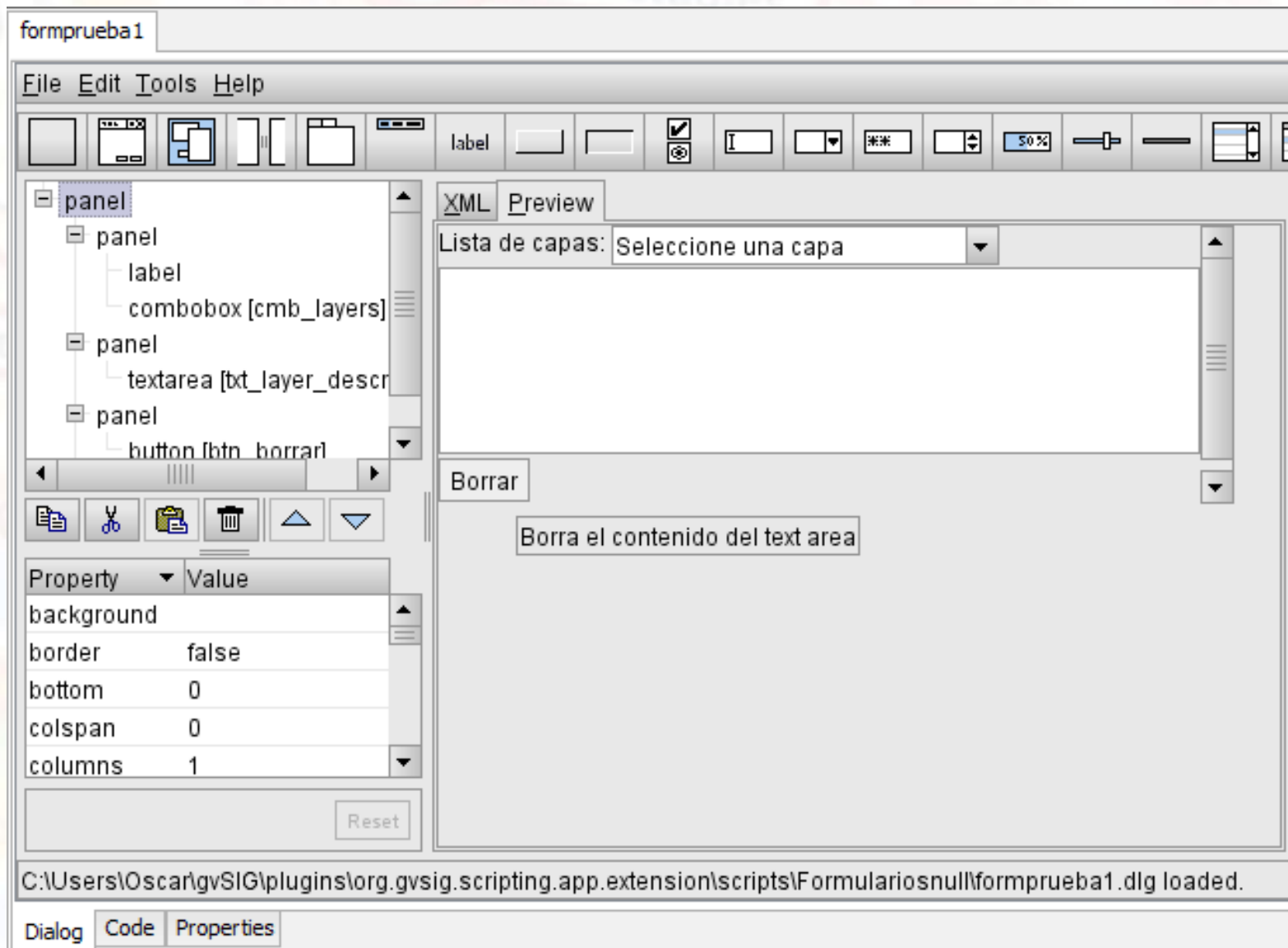


```
Jython Completion Shell
Jython 2.5.2 (Release_2_5_2:7206, Mar 2 2011, 23:12:06)
[Java HotSpot(TM) Client VM (Sun Microsystems Inc.)] on java1.6.0_26
>>> from gvsig import *
>>> features = currentLayer().features()
>>> features.getCount()
175L
>>> project = currentProject()
>>> project.get|
getProjectionCode
getTable
getView
```

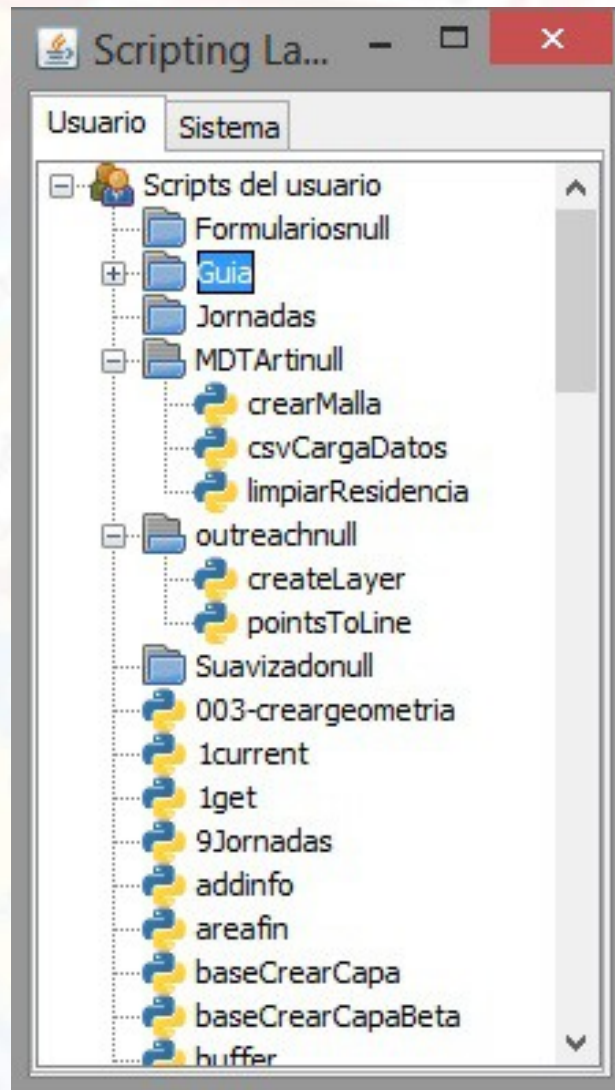
Scripting composer - Script



Scripting composer - Formulario



Scripting launcher



¿Qué nos permite un script?

- Automatizar tareas
- Administrar datos
- Analizar datos
- Crear nuestras propias herramientas



GRAN POTENCIAL

Entorno sencillo de utilizar y
fácil de programar





9th

INTERNATIONAL
CONFERENCE
27-29 November 2013
Centro Cultural y Deportivo La Pechina, Valencia Spain

MATTER OF
SOVEREIGNTY

Ejemplos de uso



Importar csv

The screenshot displays the gvSIG 2.0.0.2066 final interface. The main map window, titled 'Vista: Sin título', shows a map with several red points representing data from the 'csv_layer'. The 'Scripting Composer' window is open, showing a Python script named 'ejmm' that imports necessary modules and defines a main function. The 'Tornados: Bloc de notas' window shows a CSV file with 9 rows of data, including columns for ID, CODIGO, NOMBRE, FECHA, INTENSIDAD, VELOCIDAD, X, and Y. The status bar at the bottom indicates the current location and projection: '1: 160.693.481 Metros Lon = -7° 42' 44" Lat = 20° 6' 49" EPSG:4326'.

gvSIG 2.0.0.2066 final : Sin título

Archivo Selección Mostrar Capa Vista Mapa Herramientas Ventana Ayuda

Vista: Sin título

csv_layer

Tornados: Bloc de notas

ID	CODIGO	NOMBRE	FECHA	INTENSIDAD	VELOCIDAD	X	Y
1	X159	Gelir	23/06/09	5	30	0	0
2	X295	Ferla	24/08/09	4	20	10	15
3	X820	Vurla	25/10/10	7	35	23	44
4	X810	Bastir	26/04/11	5	24	25	38
5	X890	Cerstart	27/06/12	"3.5"	15	32	14
6	X183	Xer	28/06/11	"8.1"	35	65	73
7	X105	Gagmor	01/09/10	"7.5"	17	13	5
8	X109	Tristi	30/06/09	"6.4"	28	98	15
9	X852	Marme	20/07/11	"5.3"	20	83	45

Scripting Composer

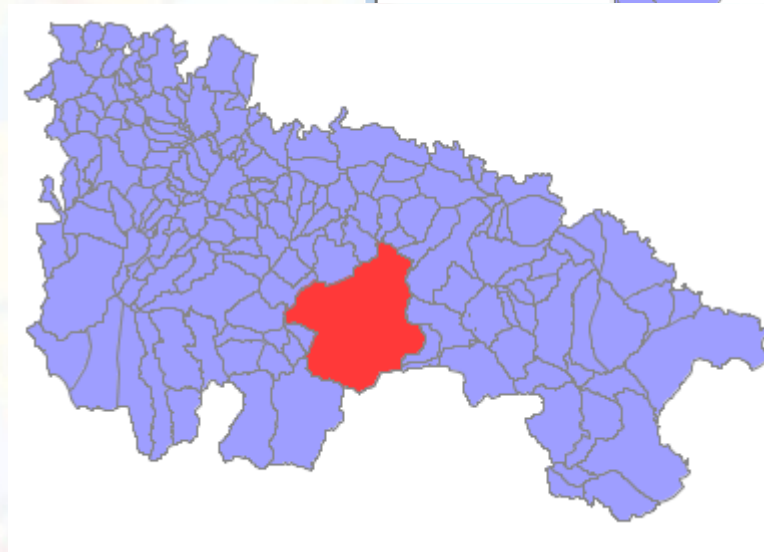
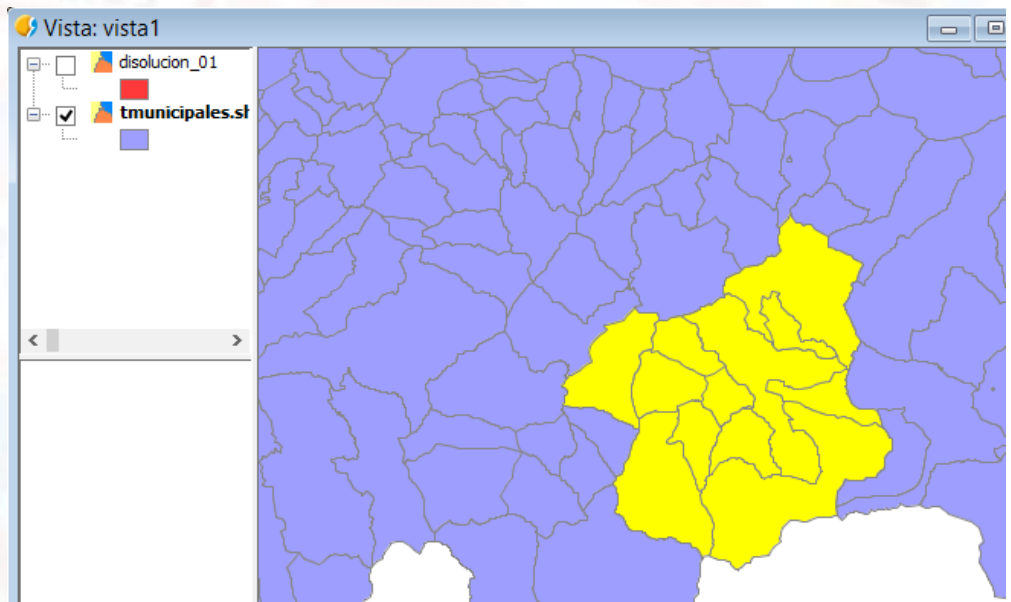
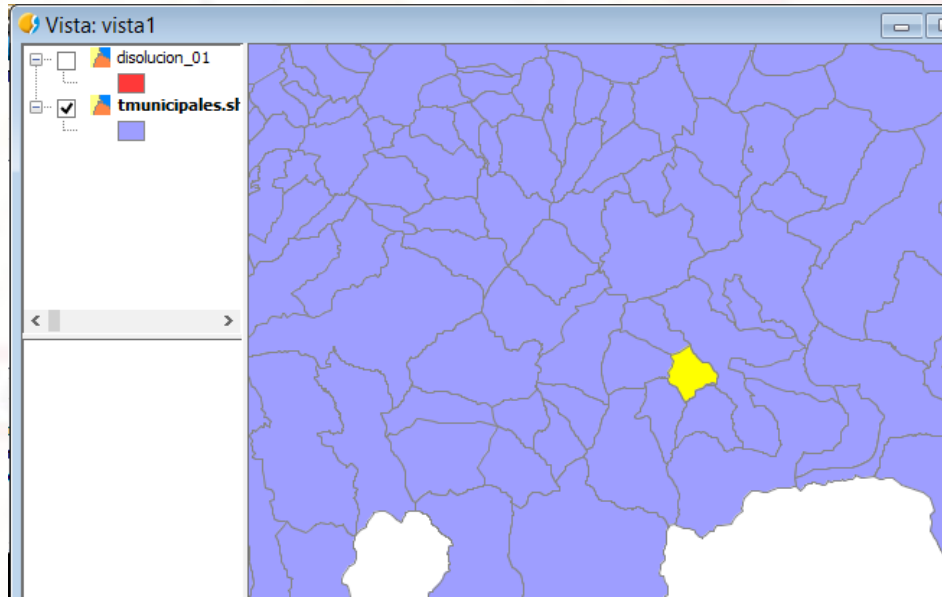
```
1 from gvsig import *
2 from commonsdialog import *
3 from geom import *
4 import csv
5 import os.path
6
7
8 def main():
9     """Crea Capa de puntos"""
10     # ... (script continues) ...
```

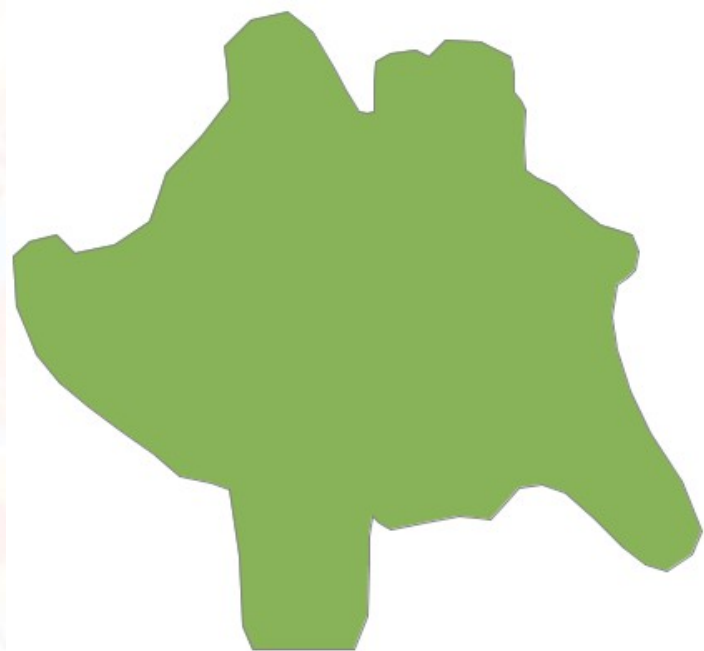
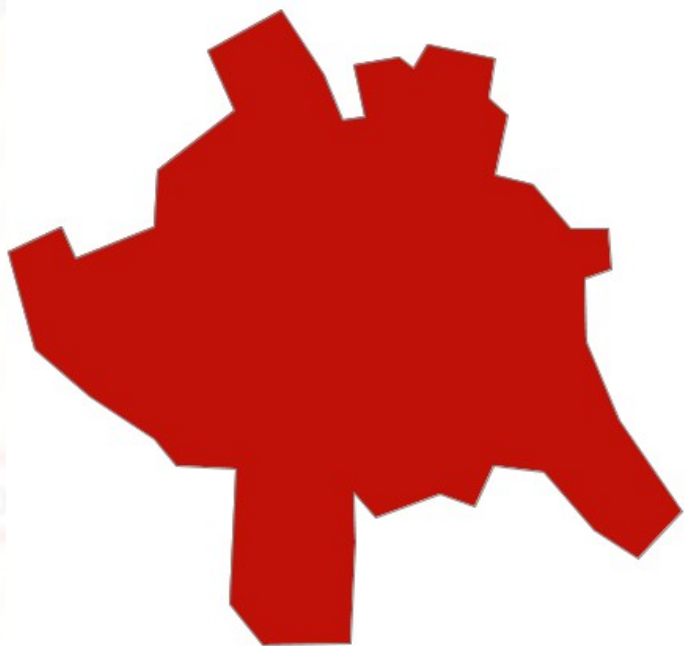
Running script ejmm.
Script ejmm terminated.
ActionEvent con INTERVAL_TY

Welcome to ScriptingFramework v1.0 Line 0:0

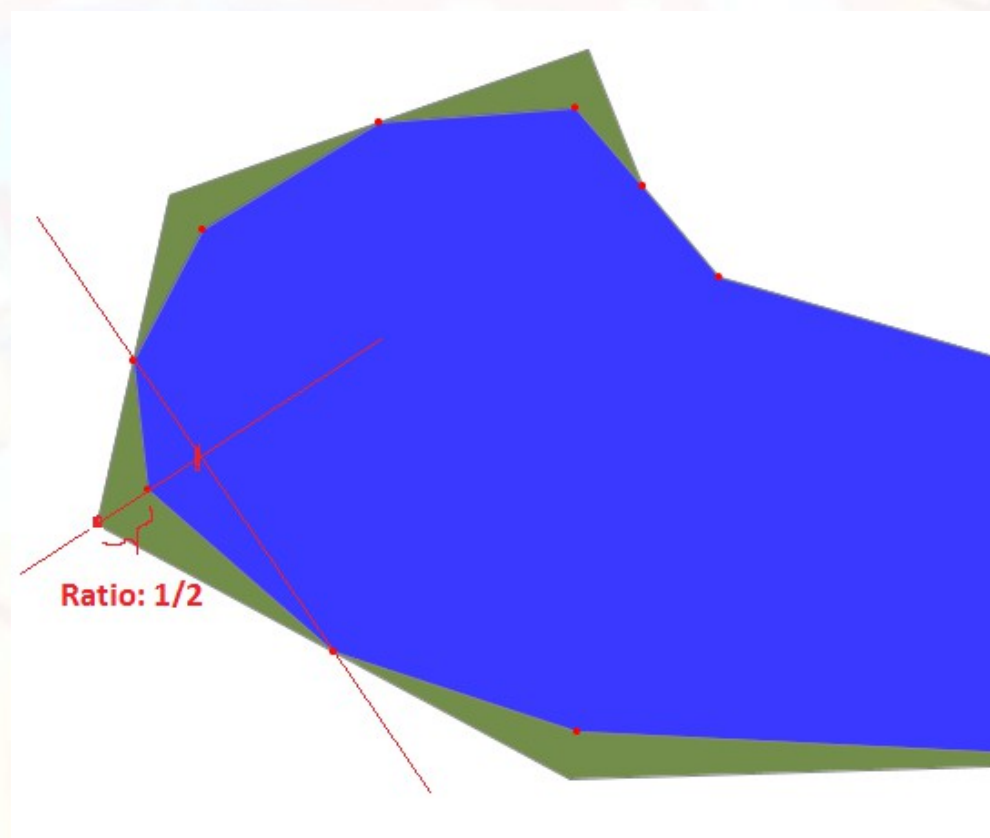
1: 160.693.481 Metros Lon = -7° 42' 44" Lat = 20° 6' 49" EPSG:4326

Disolver

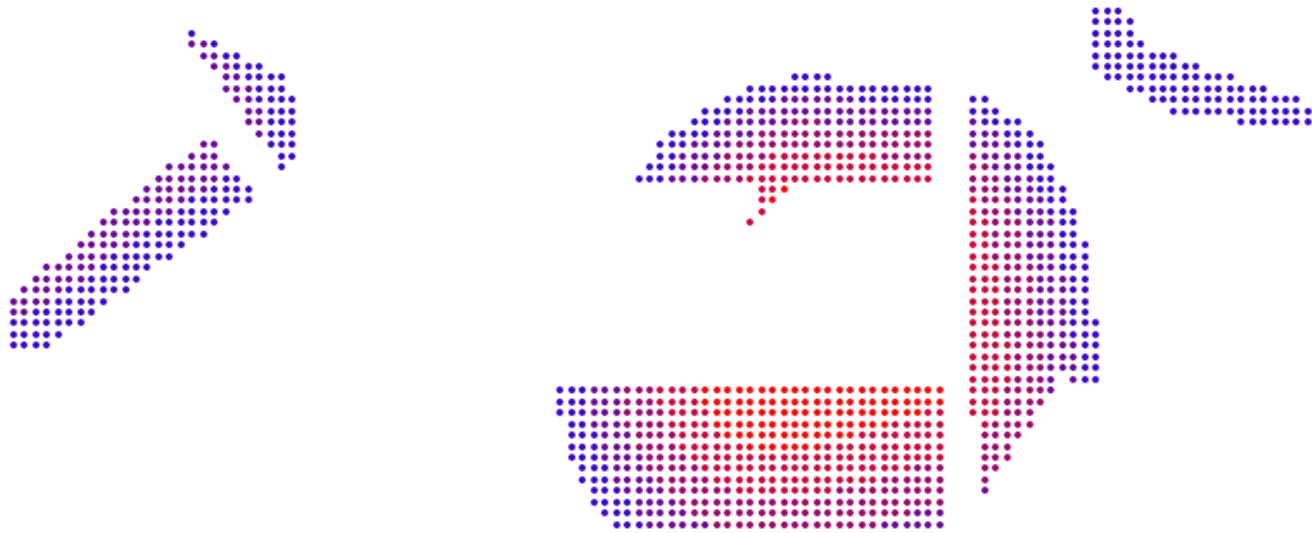




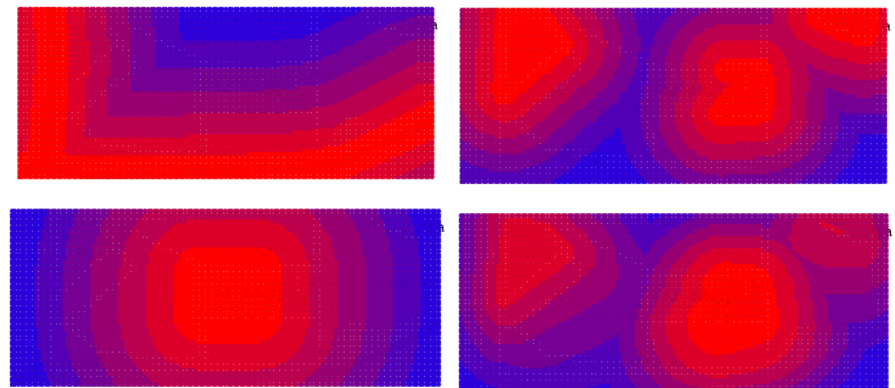
Suavizado



Análisis espacial



Ratios

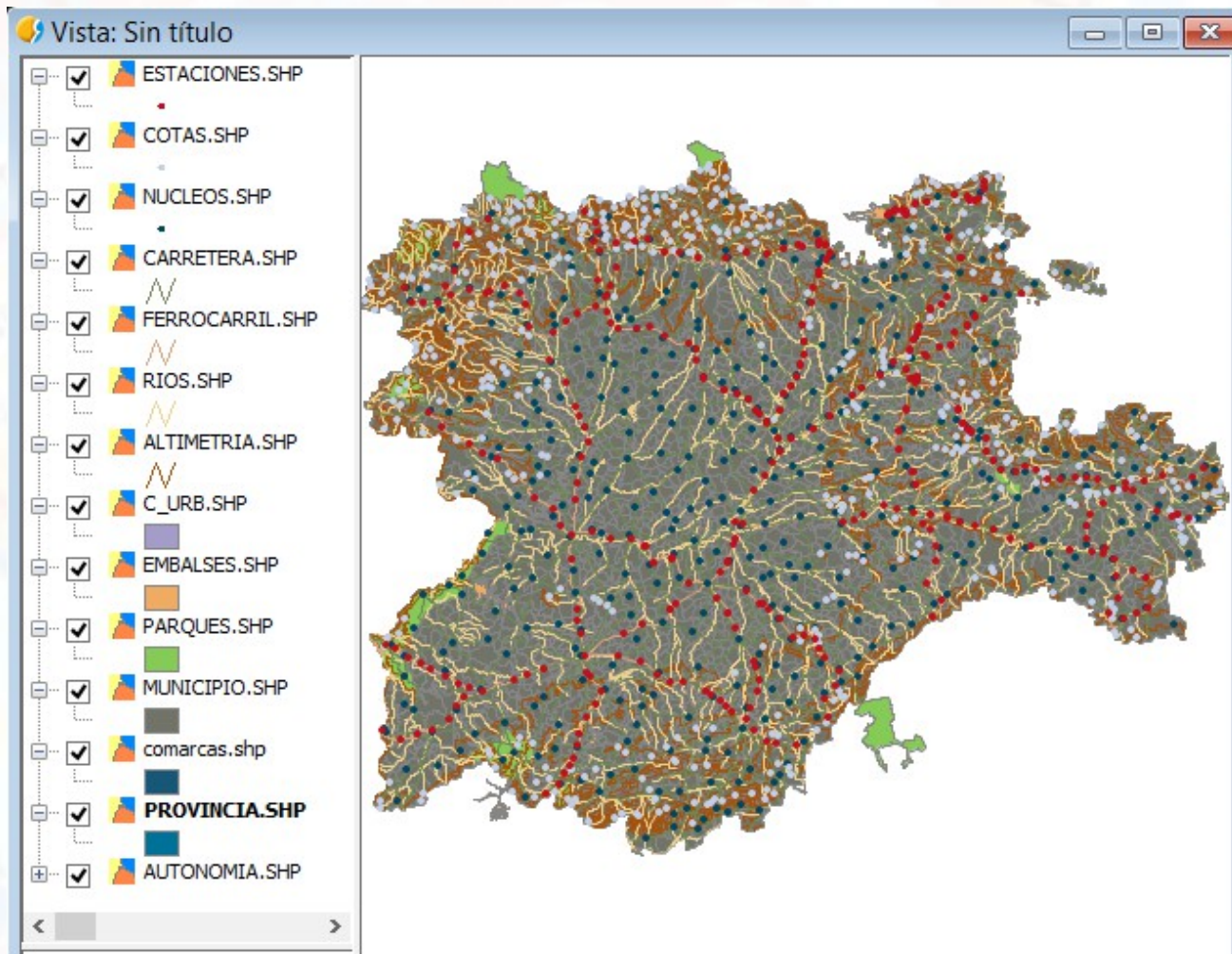


Auto Arrenge

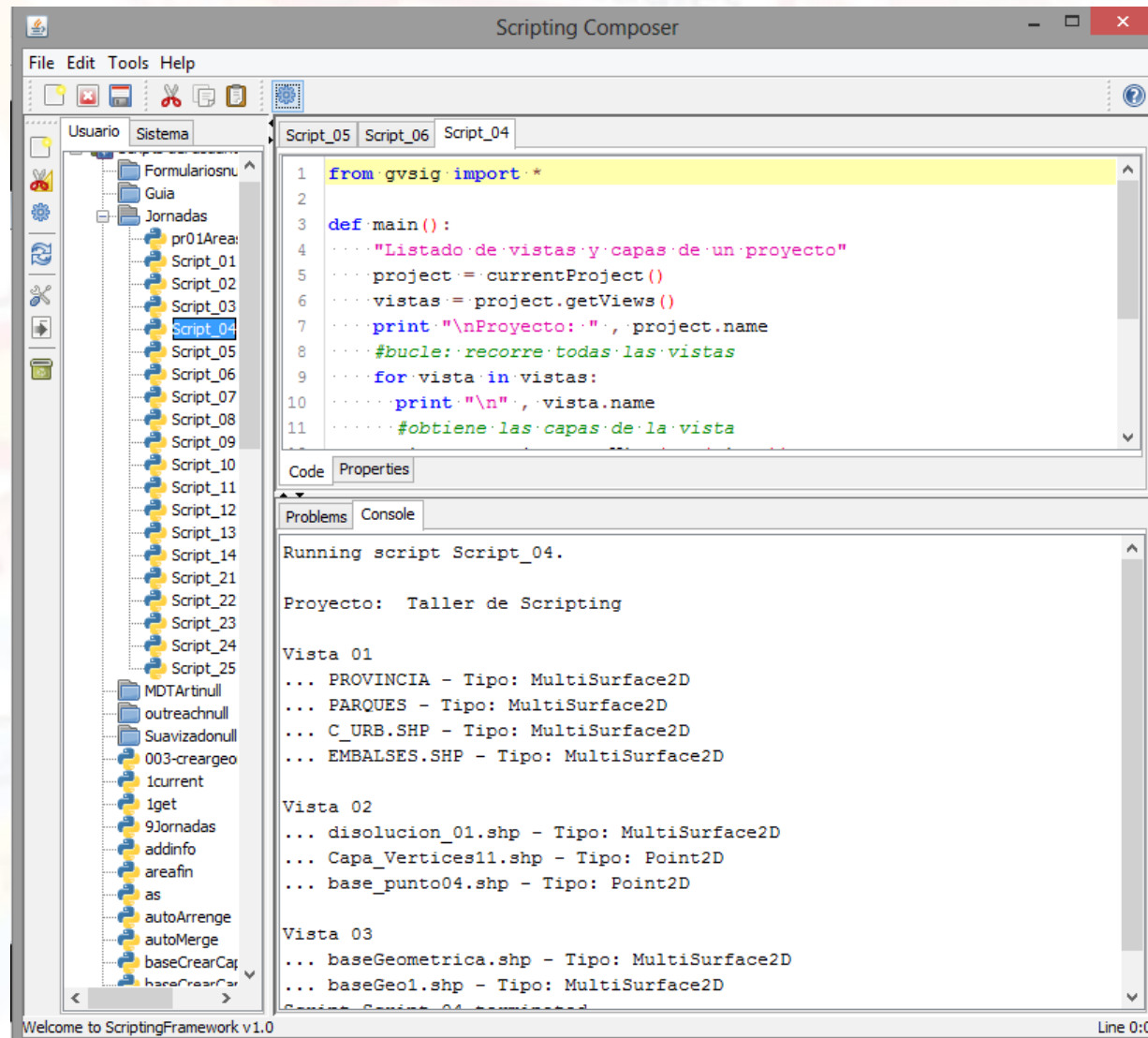
Problems Console

```
14 1 ESTACIONES.SHP 0 397
13 1 COTAS.SHP 0 450
12 1 NUCLEOS.SHP 0 459
11 2 CARRETERA.SHP 2465073.96034 1608
10 2 FERROCARRIL.SHP 3182134.08978 1422
9 2 RIOS.SHP 3186857.12785 1193
8 2 ALTIMETRIA.SHP 16783313.0288 573
7 3 C_URB.SHP 77813197.0 9
6 3 EMBALSES.SHP 506155973.984 51
5 3 PARQUES.SHP 3268015427.8 10
4 3 MUNICIPIO.SHP 94213025263.5 2305
3 3 comarcas.shp 94213025263.5 121
2 3 PROVINCIA.SHP 94222658154.9 9
1 3 AUTONOMIA.SHP 94222658154.9 1
```

Script autoArrenge terminated.



Listado de vistas y capas





PROYECTO:

“Scripting en gvSIG”



Partes

- Guía de referencia y Esquema
 - Traducida, sencilla y más completa
- Guía de inicio
 - Desde la instalación hasta tu script
- Blog de masquesig.com



Guía de referencia

Module commonsdialog reference

Contents

- [Module commonsdialog](#)
 - [Window message, msgbox](#)
 - [Confirm window, confirmDialog](#)
 - [Input window, inputbox](#)

Module commonsdialog

[Window message, msgbox](#)

Shows window message with accept button only.

`msgbox(message [, title, messageType])`

- `msgbox(message [, title, messageType])`
 - param message: Text to show
 - type message: string
 - param title (optional): Window title.
 - type title: string
 - messageType, integer [FORBIDDEN | IDEA | WARNING | QUESTION]: Show window icon to emphasize message. Recognized constant values
 - FORBIDDEN: Forbiden symbol
 - IDEA: Lamp symbol
 - WARNING: Warning symbol
 - QUESTION: question symbol

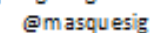
[Confirm window, confirmDialog](#)

Shows a message to the user to choose one option

`confirmDialog(message, [title, optionType, messageType])`

- `confirmDialog(message, [title, optionType, messageType])`
 - param message: Text to show
 - type message: string
 - param title (optional): Window title.
 - type title: string

Esquema



Guía de inicio

4.2.1 Esquema

Para crear un esquema nuevo lo haremos desde la función `createSchema`, con el formato de:

`createSchema([schema])`

- Devuelve un esquema válido, si el `parametro` esta vacío, se crea nuevo `schema`
- `param: schema (Schema)`

El parámetro `schema` es opcional y de clase `schema`. Aquí podríamos introducir un esquema ya creado con anterioridad, como el obtenido con el código anterior en la variable `schm1`. Si no, este esquema estará en blanco y tendremos que añadir los parámetros uno a uno de la siguiente forma:

```
schema = createSchema()  
schema.append("GEOMETRY", "GEOMETRY")  
schema.append("ID", "INTEGER", size = 7)  
schema.append("Codigo", "INTEGER", size = 5)  
schema.append("Valor", "STRING", size = 7)
```

Tabla de atributos: puntosAleat...

ID	Codigo	Valor
0	0	
1	0	
2	0	
3	0	
4	0	

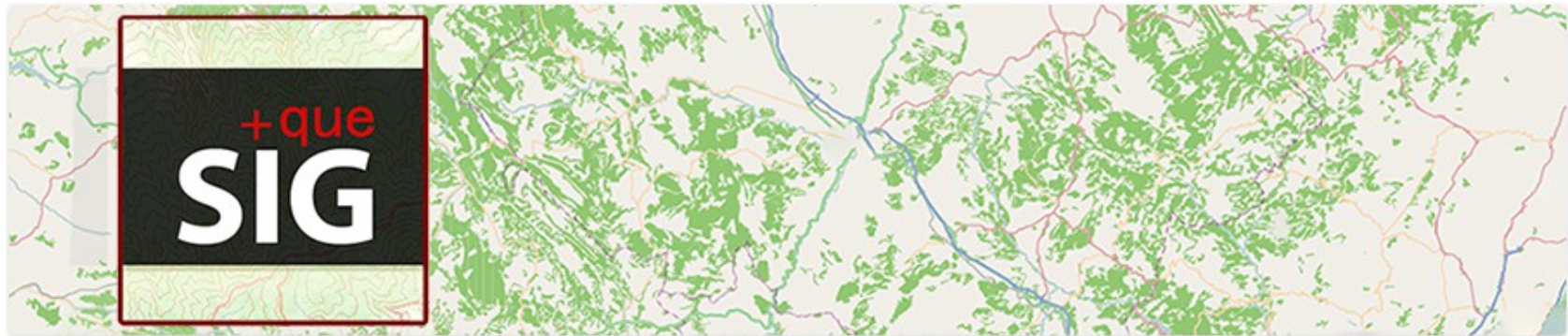
0 / 5 Total registros seleccionados.

Y el resultado después de que agregáramos este esquema a una capa (`shape`) sería el que aparece en la imagen.

Como podemos apreciar la columna

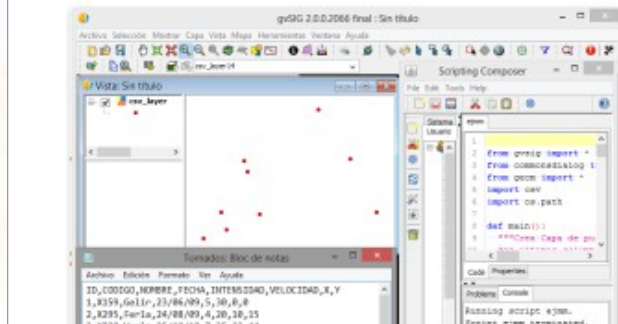
Blog de másqueSIG

INICIO ABOUT CHARLAS SCRIPTING



Scripting en gvSIG: Importar CSV como capa y crear geometría

[2 respuestas](#)



Teniendo un .csv con nuestros datos crearemos una nueva capa con estas entidades, y utilizando el valor de sus columnas X e Y crearemos una geometría que asociemos a cada entidad.

TALLER DE SCRIPTING



Mi perfil final

- Estudiante de Topografía en la UPV
- Mayor conocimiento de programación:
 - Python, Java
- Mayor conocimiento de programación SIG:
 - GDAL/OGR
 - ArcPy, QGIS, PostGIS



Conclusión

- **Animaros** a programar
- Comprender el **potencial** de este módulo
- La importancia de una **guía** de inicio
- La **difusión** de los datos



Contacto



www.masquesig.com

@masquesig

masquesig@gmail.com

Óscar Martínez

