

**Application: gvSIG desktop - gvSIG bugs #4639**  
**slow response / lenta respuesta**

08/23/2017 04:34 PM - Mario Fevre

<b>Status:</b> Outdated	<b>% Done:</b> 0%
<b>Priority:</b> Normal	<b>Spent time:</b> 0.00 hour
<b>Assignee:</b>	
<b>Category:</b>	
<b>Target version:</b>	
<b>Severity:</b> Minor	<b>Add-on version:</b>
<b>gvSIG version:</b> 2.4.0	<b>Add-on build:</b>
<b>gvSIG build:</b>	<b>Add-on resolve version:</b>
<b>Operative System:</b>	<b>Add-on resolve build:</b>
<b>Keywords:</b>	<b>Proyecto:</b>
<b>Has patch:</b>	<b>Hito:</b>
<b>Add-on name:</b> Unknown	

**Description**

A pesar de que gvSIG 2.4 (b2831) es mucho más ágil que 1.12 en muchos sentidos; identificamos algunos procesos que se tornan particularmente lentos, especialmente con capas grandes.

Aquí algunos resultados comparativos entre ambos programas en un mismo ordenador. Las mediciones las hice con un cronómetro de mano, pero la diferencia obtenida entre versiones, y la finalidad ilustrativa del ejercicio permite despreciar el margen de error.

capa utilizada: parcelas de la ciudad de buenos aires, sin reproyectar, 320000 poligonos, 11 campos,

tamaño 900mb

PC: Win 7 sp1. proc: intel i3-4170 cpu @ 3.70GHz 3.70GHz

RAM: 8. GB

sis: 64 bits

**gvSIG 1.12 portatil - sin reproyectar al vuelo.**

regenerar: 3.16 seg

info: cargar atributos de una parcela: 0.89 seg

abrir tabla: 0.5 se

añadir al listado de capas a cargar: 0.00 seg

seleccionar un elemento (+ su regeneración): 1.13 seg

regenerar: 5.71 seg

info: cargar atributos de una parcela:35.14 seg

abrir tabla: 0.66 seg

añadir al listado de capas a cargar: 4.63 seg

seleccionar un elemento (+ su regeneración): 9.79 seg

**History**

#1 - 08/23/2017 04:35 PM - Mario Fevre

**gvSIG 1.12 portatil - sin reproyectar al vuelo.**

regenerar: 3.16 seg

info: cargar atributos de una parcela: 0.89 seg

abrir tabla: 0.5 se

añadir al listado de capas a cargar: 0.00 seg

seleccionar un elemento (+ su regeneración): 1.13 seg

**gvSIG 2.4 b2831**

regenerar: 5.71 seg  
info: cargar atributos de una parcela:35.14 seg  
abrir tabla: 0.66 seg  
añadir al listado de capas a cargar: 4.63 seg  
seleccionar un elemento (+ su regeneración): 9.79 seg

## #2 - 08/24/2017 11:12 AM - Antonio Falciano

Hi Mario,

Mario Fevre wrote:

### ***gvSIG 1.12 portatil - sin reproyectar al vuelo.***

---

*regenerar: 3.16 seg  
info: cargar atributos de una parcela: 0.89 seg  
abrir tabla: 0.5 se  
añadir al listado de capas a cargar: 0.00 seg  
seleccionar un elemento (+ su regeneración): 1.13 seg*

### ***gvSIG 2.4 b2831***

---

*regenerar: 5.71 seg  
info: cargar atributos de una parcela:35.14 seg*

It seems that *"the more the layer geometries are complex (high number of vertices), the less the "Info by point" is quick in retrieving the alphanumerical info"* (see #2749). It's clear that reading also geometries together with the alphanumerical data has introduced an high inefficiency than in the past. I'd identify the geometries only on specific user demand.

*abrir tabla: 0.66 seg  
añadir al listado de capas a cargar: 4.63 seg*

The delay in appending layers to the "Add layer" list is clearly due to projection file recognition (missing in gvSIG 1.12). It can be more fast (5x-10x at least) than now and I've submitted a patch in order to improve the prj recognition speed and efficiency in #4531 a month ago, but it seems that it has not been evaluated by developers at the moment.

*seleccionar un elemento (+ su regeneración): 9.79 seg*

About view refresh (with or without selection), I suspect that reading geometries plays an important role here too.

Thanks to reporting this very interesting use case (big shapefile). It contains different issues, but very useful to improve the whole robustness of gvSIG application in the next future.

## #3 - 08/24/2017 12:29 PM - Antonio Falciano

In order to improve a bit the speed of view refresh, the user can choose an high refresh frequency in the *Preferences | View | Behaviour* settings. However it could be very useful to adopt an adequate geometry simplification technique in relation to the scale of representation. Less vertices to show would mean less computation time and so more refresh speed.

## #4 - 03/04/2020 02:50 PM - Álvaro Anguix

- Status changed from New to Outdated