

CONSTRUCCIÓN DE UN VISOR DE SHAPEFILES CON HERRAMIENTAS LIBRES

Germán Alonso Carrillo Romero
carrillo.german@gmail.com
<http://geotux.tuxfamily.org>

Introducción

Con el ánimo de construir mi propio visor de datos espaciales en formato vectorial, me di a la tarea de buscar herramientas de *software* que me lo permitieran de la manera más libre posible.

Para empezar, debo aclarar que no soy un buen programador, los programadores serios pueden ver el siguiente artículo como poco profundo. Sin embargo, teniendo en cuenta mis limitaciones, me parece fascinante poder integrar herramientas libres para obtener resultados prácticos en el ámbito de la geomática.

¿Para qué tener mi propio visor?

- Para poder acceder rápidamente a mis datos espaciales, sin tener que cargar los programas usuales, que pueden consumir recursos innecesarios en operaciones simples de consulta.
- Para aprender a emplear y a integrar las facilidades que presentan muchos de los proyectos de filosofía libre, demostrando así que el principal obstáculo en la implementación de estas herramientas es nuestro desconocimiento y no sus limitaciones funcionales.

¿Qué tipo de aplicación se va a generar?

Según Daniel P. Ames (quien está a cargo del desarrollo de componentes *GIS* en el proyecto *MapWindow*) existen tres tipos de escenarios para desarrollar *software GIS*:

1. Desarrollo de extensiones y *plug-ins* que agregan funcionalidades a *software GIS* existente en el escritorio.
2. Desarrollo de herramientas *web* de visualización y generación de mapas.
3. Desarrollo de aplicaciones independientes en el escritorio empleando componentes *GIS* programables.

Nuestro visor de datos espaciales en formato vectorial será una aplicación independiente en el escritorio, empleando un componente *GIS* programable, el *MapWinGis.ocx* de *MapWindow*.

¿Qué herramientas de software se requieren?

- Sistema Operativo: *Windows XP* (Bueno, por algo se empieza...) (Licencia: *Privativa*)
- Marco de Desarrollo de Software: *.NET Framework 2.0* (Licencia: *Freeware*)
- Entorno Integrado de Desarrollo (IDE): *SharpDevelop 2.1* (Licencia: *GNU/GPL*)
- Componente GIS: *MapWinGIS.ocx* (del proyecto [MapWindow Gis](#)) (Licencia: *Mozilla Public License*)

¿Qué es .NET Framework?

[.NET Framework](#) es un marco de desarrollo de *software* compuesto por lenguajes de programación, una biblioteca de clases ([BCL](#)) y un entorno de ejecución común para lenguajes ([CLR](#)). Implementa los estándares abiertos *ECMA* e *ISO/IEC* que propenden por una compatibilidad mínima entre los lenguajes que lo integran.

En otros términos, *.NET Framework* se encarga de proveer al programador los lenguajes, herramientas y servicios necesarios para distribuir su aplicación en múltiples plataformas de *hardware*.

¿Qué es SharpDevelop?

SharpDevelop es un *IDE OpenSource* que soporta el desarrollo de aplicaciones en lenguajes como *C#* y *VB.NET*; permite escribir código en *ASP.NET*, *ADO.NET*, *XML* y *HTML*, entre otros; tiene un depurador integrado; permite diseñar formularios para *C#* y *VB.NET*; está traducido parcialmente al español; soporta proyectos de instalación; tiene un generador de documentación embebido; soporta múltiples *Frameworks*: *.NET 1.1* y *2.0*, *Mono* y *Compact Framework*; y admite el protocolo *ActiveX* (de manera parcial).

Es un *IDE* muy completo que cubre fácilmente los requerimientos de un programador en *C#* o *VB.NET*. Particularmente, además de las necesidades básicas, necesitaba un entorno que permitiera el manejo de componentes *ActiveX* y *SharpDevelop* lo cumple.

¿Qué es el MapWinGIS.ocx?

El *MapWinGIS.ocx* es un componente *ActiveX* escrito en *C++* y desarrollado por el proyecto *MapWindow Gis*. Facilita el desarrollo de aplicaciones en varios lenguajes de programación (*C#*, *VC++*, *VB6*, *VB.NET*, *VBA* y *Delphi*). Esta compuesto por una Interfaz de Programación de Aplicaciones ([API](#)) que permite el acceso a objetos, funciones, propiedades y métodos relativos a la visualización y manipulación básica de información geográfica en formato *Shapefile*, *Grid* y *TIN*.

Para comenzar...

Para comenzar la construcción del visor, debemos instalar el software requerido.

Instalación del .NET Framework

El *.NET Framework 2.0* se puede descargar [aquí](#). Tiene un tamaño de 22,4 MB y requiere el *Service Pack 2* de *Windows* instalado.

Una vez descargado, debemos ejecutar el archivo y seguir normalmente los pasos de instalación.

Instalación del SharpDevelop 2.1

El instalador de *Sharpdevelop 2.1* puede obtenerse [aquí](#).

Contrario a lo que podríamos pensar sobre un tamaño exagerado del archivo de instalación (pues es un *IDE* muy completo), solamente ocupa 8,63 MB en el disco duro y después de instalado únicamente 34,26 MB, realmente sorprendente.

La instalación de *SharpDevelop* no presenta mayores inconvenientes.

Instalación del componente ActiveX MapWinGIS.ocx

Podemos obtener el componente *MapWinGIS.ocx* sin incluir sus dependencias [aquí](#) (archivo comprimido, 2,23 MB), o incluyendo sus dependencias [aquí](#) (archivo ejecutable 15 MB). El componente *ActiveX* requiere de varios archivos *dll* (*runtimes* de *C* y *C++ 7.0*) para funcionar correctamente, si se quiere estar seguro sobre el funcionamiento del mismo, se recomienda descargar la segunda opción, es decir, el archivo con dependencias.

1. Si se descargó el archivo sin dependencias:

- Descomprimir el archivo *MapWinGis43.zip*
- Copiar el archivo *MapWinGis.ocx* a una de las siguientes carpetas:

```
c:\Archivos de Programa\Archivos Comunes\MapWindow\  
c:\Archivos de Programa\Common Files\MapWindow\  

```

Debemos crear la carpeta si no existe. En realidad no interesa en qué carpeta se encuentre el archivo pero se recomienda una de las carpetas mencionadas para poderlo encontrar fácilmente si se planean realizar más aplicaciones que utilicen el componente.

2. Si se descargó el archivo con dependencias:

Ejecutar el archivo *MapWinGis43OCXOnly.exe* y seleccionar la opción “*No, I will restart the computer later*”.

El archivo *MapWinGis430CXOnly.exe* instala las dependencias del componente *ActiveX* y el archivo *MapWinGis.ocx* en la carpeta

`c:\Archivos de Programa\Common Files\MapWindow\`

Ahora debemos registrar el archivo *MapWinGis.ocx* en el sistema:

Abrir la ventana Ejecutar: Vamos a Inicio --> Ejecutar

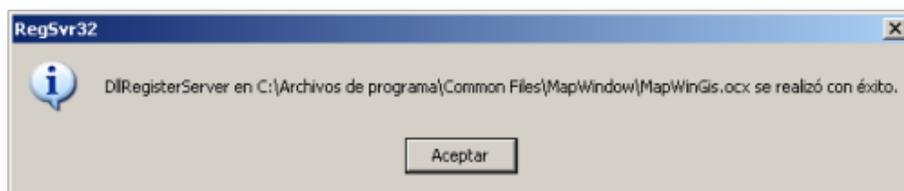
Correr el comando de registro de componentes:

En la caja de texto que aparece en la ventana *Ejecutar*, escribimos:

```
regsvr32.exe "C:\Archivos de programa\Archivos comunes\MapWindow\MapWinGIS.ocx" 0
```

```
regsvr32.exe "C:\Archivos de programa\Common Files\MapWindow\MapWinGIS.ocx"
```

dependiendo del directorio en el que se encuentre el componente. Obtenemos:



Lo cual nos indica que tenemos el *ActiveX* registrado.

Un obstáculo por superar

El *ActiveX MapWinGis.ocx* es un componente que emplea la tecnología *COM*, la cual no es soportada de manera directa por *.NET*; para poder utilizarlo en nuestra aplicación se deben crear un par de archivos interoperables con la tecnología *.NET*.

Estos archivos interoperables pueden ser generados empleando herramientas del *Kit de Desarrollo de Software* de *.NET* (*.NET SDK*), que se encuentra en *internet* con licencia *Freeware*. Sin embargo, el archivo de descarga tiene un tamaño superior a los 350 MB, por lo cual el proceso puede resultar inoportuno si se tiene en cuenta nuestro objetivo. Para conocer el procedimiento de obtención de los archivos interoperables podemos visitar [esta página web](#).

Como alternativa práctica, podemos optar por descargar directamente los archivos interoperables [aquí](#) (archivo comprimido, 46 KB). Tras descomprimir el archivo, obtenemos dos *dll*:

- AxInterop.MapWinGIS.dll
- Interop.MapWinGIS.dll

Estos archivos no requieren de registro en el sistema porque son componentes *.NET*; son importantes porque posibilitan el acceso a los objetos, funciones, propiedades y métodos del componente programable *MapWinGis* en nuestra aplicación.

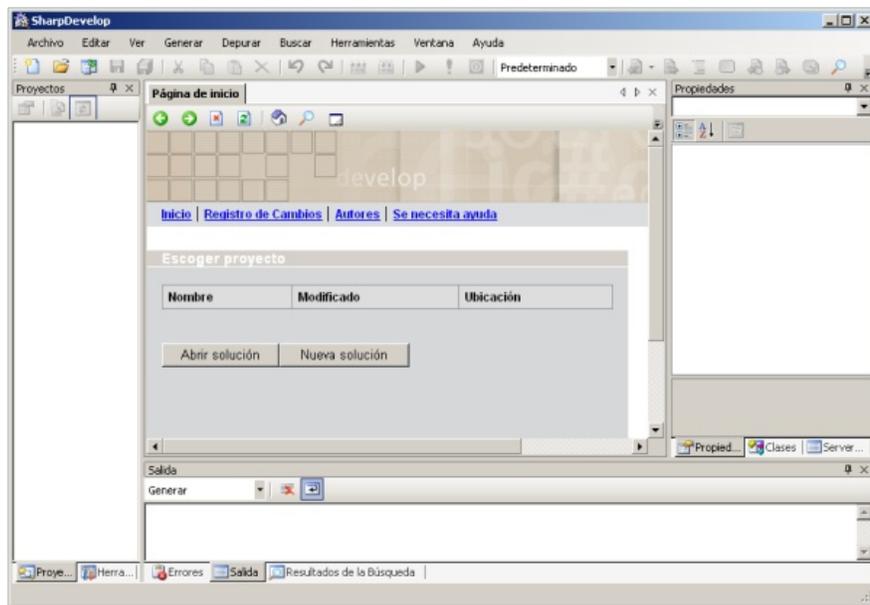
Ahora debemos copiar los dos archivos en la carpeta en la que se encuentra el componente *MapWinGis.ocx* (el cual debe dejarse allí pues su registro es necesario para la aplicación).

Hemos terminado con la instalación del software requerido para la construcción de nuestro propio visor, el siguiente paso es crear un proyecto *.NET* para escribir nuestro código allí.

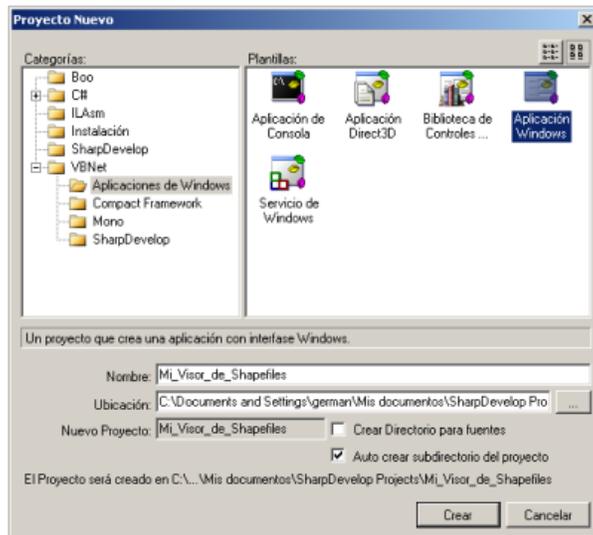
Creando el proyecto *.NET*



Empezaremos a utilizar el *Entorno Integrado de Desarrollo SharpDevelop*, cuya interfaz luce así:



Damos *click* en el botón *Nueva Solución* y obtenemos el siguiente diálogo:



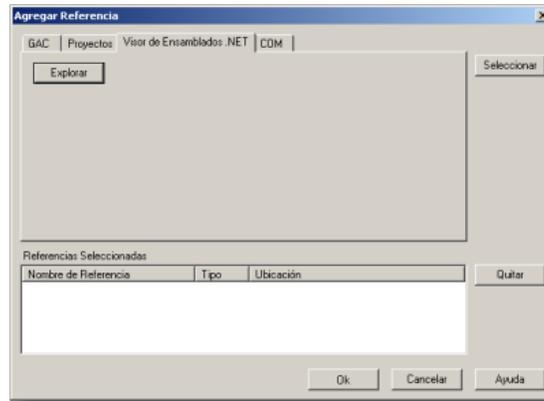
Escogemos en la parte izquierda la categoría *VBNET* y la subcategoría *Aplicaciones de Windows*. En la parte derecha seleccionamos *Aplicación Windows*. Digitamos en la caja de texto el nombre “*Mi_Visor_de_Shapefiles*” y seleccionamos la ubicación del proyecto. Damos *click* en *Crear* y tenemos la estructura de nuestro proyecto.

En la parte izquierda de la interfaz de *SharpDevelop* podemos encontrar que la pestaña *Proyectos* se encuentra activa. Observamos que nuestro proyecto tiene *Referencias*, un archivo de Información del ensamblado que compilaremos (*AssemblyInfo.vb*), una Forma Principal (*MainForm.vb*) y una clase que controla el comportamiento de nuestra aplicación (*Program.vb*). Estos archivos y referencias iniciales son creados por *SharpDevelop* facilitando el proceso de desarrollo de la aplicación.

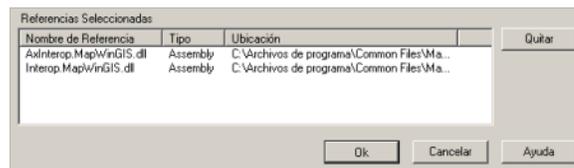


Agregando referencias

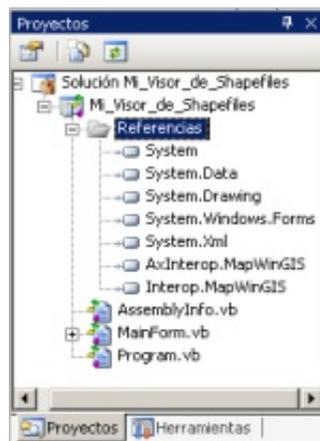
Debemos crear referencias a los archivos interoperables desde nuestro proyecto. Para ello, en la pestaña *Proyectos* damos *click derecho* a *Referencias* y seleccionamos *Agregar Referencias*.



Activamos la pestaña *Visor de Ensamblados .NET* y damos *click* en *Explorar*. Vamos al directorio en donde se encuentran los dos archivos interoperables (*AxInterop.MapWinGis.dll* e *Interop.MapWinGis.dll*) y los cargamos dando *click* en *Abrir*. Obtenemos lo siguiente:

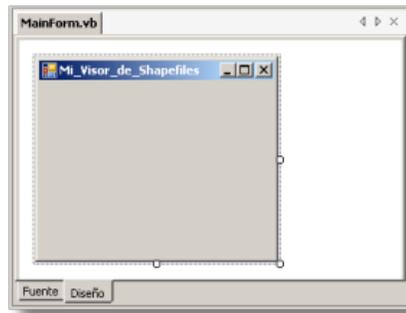


Damos *click* en *Ok* y ya tenemos las referencias a los archivos interoperables en nuestro proyecto.

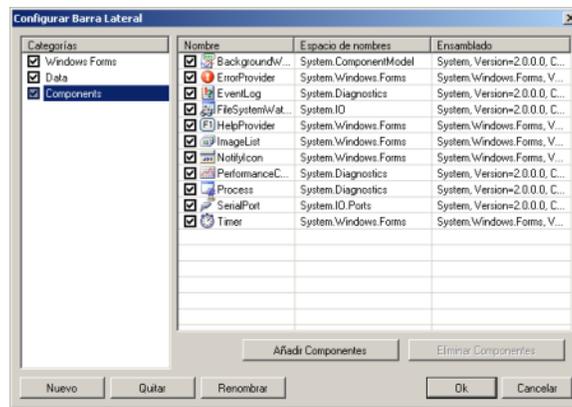


Agregando el control AxMap

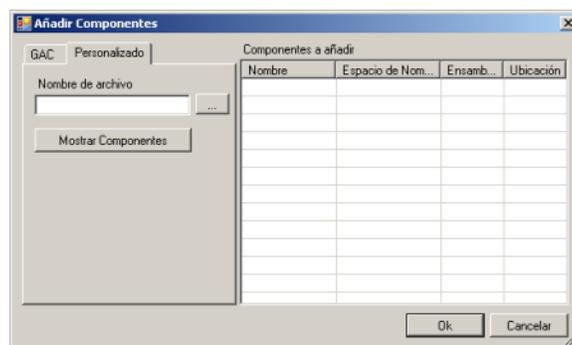
En la pestaña *Proyectos* damos *doble click* a la clase *MainForm.vb*, cuyo código se encuentra en la parte central de la interfaz. Activamos la pestaña *Diseño* y observamos la *Forma* sobre la cual ubicaremos varios controles posteriormente:



En la parte inferior izquierda de la interfaz activamos la pestaña *Herramientas* visualizando los diferentes controles que podemos agregar a la *Forma*. Damos *click derecho* en el área *Herramientas* que se acaba de desplegar y seleccionamos *Personalizar Barra Lateral...*



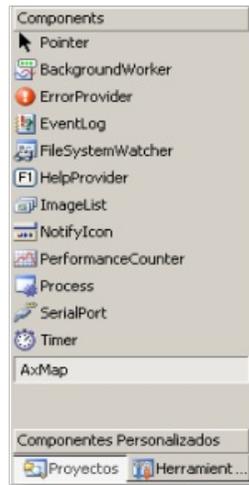
Seleccionamos la categoría *Componentes* (sin desactivarla) y damos *click* en el botón *Añadir Componentes*.



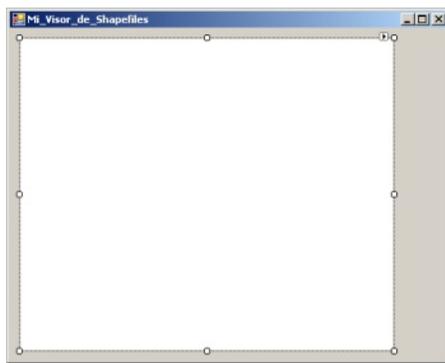
Activamos la pestaña *Personalizado* y damos *click* al botón con los puntos suspensivos, vamos a la carpeta que contiene los archivos interoperables y seleccionamos *AxInterop.MapWinGis.dll*, damos *click* en *Abrir* y vemos que la ruta al archivo se ha cargado en la caja de texto correspondiente. Damos *click* al botón *Mostrar Componentes* y observamos a la derecha el componente *AxMap*.



Damos *click* en *Ok* a las ventanas abiertas y con esto hemos habilitado el control que nos servirá para visualizar la información geográfica en la *Forma*.



En *Herramientas* damos *click* a la barra *Components* y a luego a *AxMap*. Llevamos el puntero hacia la *Forma* y dibujamos un rectángulo sobre la misma, allí se alojará el control *AxMap* que tiene un color blanco por defecto.



Diseñando la interfaz del visor

Continuando con el diseño de nuestro visor, en *Herramientas* damos *click* a la barra *Windows Forms* y seleccionamos *Button*. Llevamos el cursor a la *Forma* y dibujamos seis botones que corresponderán a Acercar (*Zoom In*), Alejar (*Zoom Out*), Zoom Previo, Zoom Completo, *Pan* y Agregar Capa.

A cada botón en la *Forma* debemos modificarle sus propiedades, visibles en una de las pestañas de la parte derecha de la interfaz de *SharpDevelop*, las configuramos así:

Botón	Propiedad Name	Propiedad Text	Propiedad Image
1	btnZoomIn		b_ZoomIn.png
2	btnZoomOut		b_ZoomOut.png
3	btnZoomPrevio		b_ZoomPrevio.png
4	btnZoomCompleto		b_ZoomCompleto.png
5	btnPan		b_Pan.png
6	btnAddLayer		b_AddLayer.png

Las imágenes que empleamos para los botones pueden descargarse [aquí](#) (archivo comprimido, 2 KB). La propiedad *Text* se pone nula para que no interfiera con la imagen del botón.

En este momento la *Forma* debe lucir así:



Agregamos ahora desde *Herramientas* en la barra *Windows Forms* el control *OpenFileDialog*, que nos va a permitir buscar los archivos *Shapefile* en nuestro disco duro para posteriormente cargarlos.

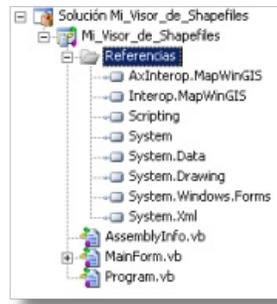
Ahora podemos agregar el código a la *Forma*.

Agregando el código de la aplicación

Antes de agregar el código de nuestro visor, debemos agregar una referencia al componente *Microsoft Scripting Runtime* para habilitar el *Objeto de Sistema de Archivos* que permite manipular directorios y archivos del sistema. Para ello, vamos a la pestaña *Proyectos*, damos *click derecho* a *Referencias* y seleccionamos *Agregar Referencia*. En la pestaña *COM* buscamos el componente mencionado, lo seleccionamos y lo cargamos.

Como se trata de un componente *COM*, *.NET* no lo soporta directamente; sin embargo, *SharpDevelop* genera de manera automática el archivo interoperable *Interop.Scripting.dll*. En el caso del componente *MapWinGis.ocx*, *SharpDevelop* no puede generar automáticamente el archivo *AxInterop.MapWinGis.dll* (aunque si el *Interop.MapWinGis.dll*) que contiene el control *AxMap*, importante en nuestra aplicación; por esta razón se prefiere suministrar los archivos interoperables ensamblados previamente.

Debemos poder visualizar la nueva referencia *Scripting*:



En la parte central de la interfaz de *SharpDevelop* (en donde estamos visualizando la *Forma*) activamos la pestaña *Fuente* que nos permite editar el código de la *Forma*.

Seleccionamos todo el texto que existe y lo borramos. Abrimos el archivo *codigo_visor.txt* que puede descargarse de [aquí](#) (archivo de texto, 5 KB) y copiamos todo su contenido en el espacio para el código de la *Forma*. Para comprender mejor el código podemos leer los comentarios realizados en cada sentencia.

Ahora debemos hacer que el código que hemos introducido se asocie a los botones de nuestra aplicación porque esto no ocurre de manera automática. Para conseguirlo, activamos la pestaña *Diseño* y damos *doble click* al primer botón de la *Forma*, lo cual nos envía al código y ubica el cursor justo en la sentencia que define su evento *click*. Hacemos lo mismo para cada uno de los cinco botones restantes.

Compilando la aplicación

Tecleamos *F5* para depurar el programa, si hemos seguido todos los pasos correctamente no deberían existir errores.

Damos *F9* para generar el ejecutable.

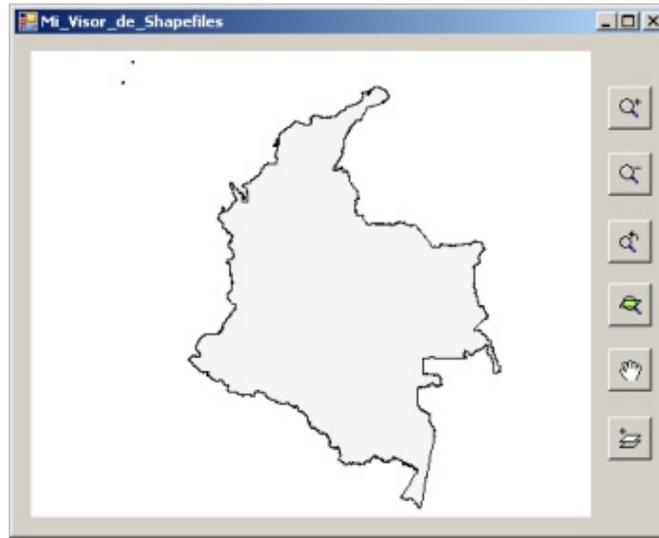
El archivo ejecutable se puede encontrar en:

```
directorio_aplicación/bin/Debug/Mi_Visor_de_Shapefiles.exe
```

donde *directorio_aplicación* es la ruta de la carpeta en la que se encuentra nuestro proyecto. Por ahora no podemos cambiarlo de directorio, pero podemos crear un acceso directo que nos facilite su búsqueda.

Podemos descargar un Shapefile de prueba [aquí](#) (archivo comprimido, 1.3 MB).

Con esto hemos generado nuestro propio Visor de *Shapefiles* empleando un *IDE OpenSource*, un *Marco de Desarrollo de Software Freeware* y un componente *GIS* libre.



Este artículo ha sido adaptado de *Getting Started With the MapWinGis ActiveX Control* (Daniel P. Ames. 2006)

Referencias consultadas:

- Ames, Daniel P. *Getting Started With the MapWinGis ActiveX Control*. 2006. Disponible en la URL: <http://www.mapwindow.org/doc/UsingMapWinGIS.pdf> (Última visita: Septiembre, 2007)
- Proyecto MapWindow GIS. URL <http://mapwindow.org> (Última visita: Septiembre, 2007)
- Proyecto SharpDevelop. URL: <http://www.icsharpcode.net/OpenSource/SD/> (Última visita: Septiembre, 2007)
- Wikipedia, la enciclopedia libre. URL: <http://es.wikipedia.org> (Última visita: Septiembre, 2007)