



# Quantum GIS

Guía de Usuario e  
Instalación

Versión 0.9.1 '*Ganymede*'

## Preámbulo

Este documento es la guía original de usuario e instalación del software Quantum GIS. El software y hardware mencionado en este documento son en la mayoría de los casos marcas registradas y están por tanto sujetas a requisitos legales. Quantum GIS está sujeto a la Licencia Pública GNU (GNU General Public License). Puedes encontrar más información en la web de Quantum GIS <http://www.qgis.org>.

Los detalles, datos, resultados, etc. que se ofrecen en este documento han sido escritos y verificados con el mayor conocimiento y responsabilidad de los editores. Sin embargo, los errores respecto al contenido son posibles. Por eso, todos los datos no están sujetos a ninguna función o garantía. Los editores y redactores no tienen ninguna responsabilidad o culpa por fallos y sus consecuencias. Siempre se es bienvenido para indicar posibles errores.

Este documento ha sido preparado con  $\text{\LaTeX}$ . Está disponible como código fuente  $\text{\LaTeX}$  y en línea como documentos HTML y PDF vía <http://www.qgis.org>.

También se pueden descargar versiones traducidas de este documento desde el área de documentación del proyecto QGIS. Más información sobre este documento y sobre su traducción está disponible en:

<http://wiki.qgis.org/qgiswiki/DocumentationWritersCorner>.

### **Redactores de la Guía de Usuario e Instalación:**

Gary E. Sherman

Tim Sutton

Radim Blazek

Stephan Holl

Otto Dassau

Tyler Mitchell

Brendan Morely

Lars Luthman

Godofredo Contreras

Magnus Homann

Martin Dobias

Con agradecimientos a Tisham Dhar por preparar el entorno inicial msys, a Tom Elwertowski y William Kyngesburye por ayudar en la Sección MAC OSX y a Tara Athan por sus revisiones.

---

**Copyright © 2004 - 2007 Quantum GIS Project**  
**Internet:** <http://www.qgis.org>

# Índice

<b>Título</b>	<b>I</b>
<b>Preamble</b>	<b>II</b>
<b>Índice</b>	<b>IV</b>
<b>Lista de figuras</b>	<b>XI</b>
<b>Lista de tablas</b>	<b>XIII</b>
<b>1. Prefacio</b>	<b>1</b>
1.1. Características . . . . .	1
1.2. Qué es nuevo en 0.9.0 . . . . .	2
<b>2. Introducción a los SIG</b>	<b>4</b>
2.1. ¿Por qué todo esto es tan nuevo? . . . . .	5
2.1.1. Datos ráster . . . . .	5
2.1.2. Datos vectoriales . . . . .	6
<b>3. Comenzar</b>	<b>7</b>
3.1. Instalación . . . . .	7
3.2. Datos de ejemplo . . . . .	7
3.3. Iniciar QGIS . . . . .	8
3.3.1. Opciones de línea de órdenes . . . . .	8
3.4. Interfaz Gráfica de Usuario (GUI) de QGIS . . . . .	10
3.4.1. Barra de menús . . . . .	11
3.4.2. Barras de herramientas . . . . .	12
3.4.3. Leyenda del mapa . . . . .	12
3.4.4. Vista del mapa . . . . .	14
3.4.5. Localizador del mapa . . . . .	14
3.4.6. Barra de estado . . . . .	15
3.5. Representación . . . . .	15
3.5.1. Representación dependiente de la escala . . . . .	15
3.5.2. Controlar la representación del mapa . . . . .	16
3.6. Measuring . . . . .	16
3.6.1. Measure length . . . . .	17
3.6.2. Measure areas . . . . .	17
3.7. Projects . . . . .	17
3.8. GUI Options . . . . .	18
3.9. Spatial Bookmarks . . . . .	20
3.9.1. Creating a Bookmark . . . . .	20

3.9.2. Working with Bookmarks . . . . .	20
3.9.3. Zooming to a Bookmark . . . . .	20
3.9.4. Deleting a Bookmark . . . . .	20
<b>4. Trabajar con datos vectoriales</b>	<b>21</b>
4.1. Archivos shape de ESRI . . . . .	21
4.1.1. Cargar un archivo Shape . . . . .	21
4.1.2. Mejorar el rendimiento . . . . .	23
4.1.3. Cargar una capa MapInfo . . . . .	24
4.1.4. Cargar una cobertura de ArcInfo . . . . .	24
4.2. Capas PostGIS . . . . .	24
4.2.1. Crear una conexión guardada . . . . .	24
4.2.2. Cargar una capa PostGIS . . . . .	25
4.2.3. Algunos detalles sobre las capas PostgreSQL . . . . .	26
4.2.4. Importar datos a PostgreSQL . . . . .	26
4.2.5. Mejorar el rendimiento . . . . .	28
4.3. El diálogo Propiedades de la capa . . . . .	29
4.3.1. Pestaña Simbología . . . . .	29
4.3.2. Pestaña General . . . . .	31
4.3.3. Pestaña Metadatos . . . . .	31
4.3.4. Pestaña Etiquetas . . . . .	31
4.3.5. Pestaña Acciones . . . . .	33
4.4. Edición . . . . .	36
4.4.1. Establecer la tolerancia de autoensamblado . . . . .	36
4.4.2. Editar una capa existente . . . . .	36
4.4.3. Crear una nueva capa . . . . .	42
4.5. Constructor de consultas . . . . .	43
4.5.1. Consultar capas PostGIS . . . . .	45
4.5.2. Consultar formatos OGR y archivos de GRASS . . . . .	45
<b>5. Working with Raster Data</b>	<b>46</b>
5.1. What is raster data? . . . . .	46
5.2. Raster formats supported in QGIS . . . . .	46
5.3. Loading raster data in QGIS . . . . .	46
5.4. Raster Properties Dialog . . . . .	47
5.4.1. Symbology Tab . . . . .	48
5.4.2. General Tab . . . . .	49
5.4.3. Metadata Tab . . . . .	50
5.4.4. Pyramids Tab . . . . .	50
5.4.5. Histogram Tab . . . . .	50
<b>6. Working with OGC Data</b>	<b>51</b>
6.1. What is OGC Data . . . . .	51

6.2. WMS Client . . . . .	51
6.2.1. Overview of WMS Support . . . . .	51
6.2.2. Selecting WMS Servers . . . . .	52
6.2.3. Loading WMS Layers . . . . .	53
6.2.4. Using the Identify Tool . . . . .	55
6.2.5. Viewing Properties . . . . .	55
6.2.6. WMS Client Limitations . . . . .	56
6.3. WFS Client . . . . .	57
6.3.1. Loading a WFS Layer . . . . .	57
<b>7. Trabajar con proyecciones</b>	<b>59</b>
7.1. Descripción del soporte para proyecciones . . . . .	59
7.2. Comenzar . . . . .	59
7.2.1. Especificar una proyección . . . . .	61
7.3. Proyecciones personalizadas . . . . .	61
<b>8. Integración de GRASS</b>	<b>63</b>
8.1. Iniciar QGIS con GRASS . . . . .	63
8.2. Cargar datos de GRASS . . . . .	64
8.3. Crear una localización . . . . .	64
8.4. Modelo de datos vectoriales . . . . .	66
8.5. Herramientas de digitalización y edición . . . . .	67
8.5.1. Barra de herramientas . . . . .	67
8.5.2. Pestaña Categoría . . . . .	67
8.5.3. Pestaña configuración . . . . .	68
8.5.4. Pestaña simbología . . . . .	69
8.5.5. Pestaña tabla . . . . .	69
8.6. Herramienta Región . . . . .	70
8.7. Caja de herramientas de GRASS . . . . .	70
8.7.1. Módulos dentro de la caja de herramientas . . . . .	71
8.7.2. Explorador de GRASS . . . . .	72
8.7.3. Personalizar la sección de los módulos . . . . .	72
8.8. Crear una nueva capa de GRASS . . . . .	73
<b>9. Making MapServer Map Files</b>	<b>75</b>
9.1. Creating the Project File . . . . .	75
9.2. Creating the Map File . . . . .	75
9.3. Testing the Map File . . . . .	77
<b>10. Diseñador de mapas</b>	<b>79</b>
10.1. Usar el diseñador de mapas . . . . .	79
10.1.1. Añadir un mapa al diseñador . . . . .	79
10.1.2. Añadir otros elementos al diseñador . . . . .	81

10.1.3. Otras funciones . . . . .	81
10.1.4. Crear la salida . . . . .	81
<b>11. Usar complementos</b>	<b>84</b>
11.1. Una introducción al uso de complementos . . . . .	84
11.1.1. Encontrar e instalar un complemento . . . . .	84
11.1.2. Administrar complementos . . . . .	84
11.1.3. Proveedores de datos . . . . .	84
11.1.4. Complementos integrados . . . . .	86
11.1.5. Complementos externos . . . . .	87
11.1.6. Plantillas de complementos . . . . .	87
11.2. Usar los complementos decorativos . . . . .	88
11.2.1. Complemento etiqueta de copyright . . . . .	88
11.2.2. Complemento flecha de Norte . . . . .	89
11.2.3. Complemento barra de escala . . . . .	89
11.3. Usar el complemento GPS . . . . .	91
11.3.1. ¿Qué es GPS? . . . . .	91
11.3.2. Cargar datos GPS de un archivo . . . . .	91
11.3.3. GPSTabel . . . . .	91
11.3.4. Importar datos de GPS . . . . .	92
11.3.5. Descargar datos de GPS desde un dispositivo . . . . .	92
11.3.6. Cargar datos de GPS a un dispositivo . . . . .	93
11.3.7. Definir nuevos tipos de dispositivo . . . . .	94
11.4. Usar el complemento de texto delimitado . . . . .	96
11.4.1. Requisitos . . . . .	96
11.4.2. Usar el complemento . . . . .	97
11.5. Usar el complemento generador de cuadrículas . . . . .	100
11.6. Usar el complemento georreferenciador . . . . .	102
11.7. Usar el complemento de Python . . . . .	106
11.7.1. Configurar la estructura . . . . .	106
11.7.2. Hacer reconocible el complemento . . . . .	107
11.7.3. Recursos . . . . .	107
11.7.4. Crear la interfaz gráfica de usuario (GUI) . . . . .	108
11.7.5. Crear el complemento . . . . .	108
11.7.6. Fallos y problemas . . . . .	112
11.7.7. Añadir retroalimentación . . . . .	113
11.7.8. Resumen . . . . .	113
<b>12. Crear aplicaciones</b>	<b>115</b>
12.1. Diseñar la GUI . . . . .	115
12.2. Crear la Ventana principal . . . . .	116
12.3. Finalizar . . . . .	121

12.4. Ejecutar la aplicación . . . . .	122
<b>13. Ayuda y soporte</b>	<b>124</b>
13.1. Listas de correo . . . . .	124
13.2. IRC . . . . .	125
13.3. Seguimiento de errores (BugTracker) . . . . .	125
13.4. Blog . . . . .	125
13.5. Wiki . . . . .	126
<b>A. Supported Data Formats</b>	<b>127</b>
A.1. Supported OGR Formats . . . . .	127
A.2. GDAL Raster Formats . . . . .	127
<b>B. Installation Guide</b>	<b>130</b>
B.1. General Build Notes . . . . .	130
B.2. An overview of the dependencies required for building . . . . .	130
<b>C. Building under windows using msys</b>	<b>131</b>
C.1. MSYS: . . . . .	131
C.2. Qt4.3 . . . . .	131
C.3. Flex and Bison . . . . .	132
C.4. Python stuff: (optional) . . . . .	132
C.4.1. Download and install Python - use Windows installer . . . . .	132
C.4.2. Download SIP and PyQt4 sources . . . . .	132
C.4.3. Compile SIP . . . . .	132
C.4.4. Compile PyQt . . . . .	133
C.4.5. Final python notes . . . . .	133
C.5. Subversion: . . . . .	133
C.6. CMake: . . . . .	133
C.7. QGIS: . . . . .	133
C.8. Compiling: . . . . .	134
C.9. Configuration . . . . .	134
C.10. Compilation and installation . . . . .	135
C.11. Run qgis.exe from the directory where it's installed (CMAKE_INSTALL_PREFIX) . . . . .	135
C.12. Create the installation package: (optional) . . . . .	135
<b>D. Building on Mac OSX using frameworks and cmake (QGIS &gt; 0.8)</b>	<b>135</b>
D.1. Install XCODE . . . . .	135
D.2. Install Qt4 from .dmg . . . . .	136
D.3. Install development frameworks for QGIS dependencies . . . . .	136
D.3.1. Additional Dependencies : GSL . . . . .	137
D.3.2. Additional Dependencies : Expat . . . . .	137
D.3.3. Additional Dependencies : SIP . . . . .	137



D.3.4. Additional Dependencies : PyQt . . . . .	138
D.3.5. Additional Dependencies : Bison . . . . .	138
D.4. Install CMAKE for OSX . . . . .	139
D.5. Install subversion for OSX . . . . .	139
D.6. Check out QGIS from SVN . . . . .	140
D.7. Configure the build . . . . .	141
D.8. GEOS Issues . . . . .	141
D.9. Building . . . . .	141
<b>E. Building on GNU/Linux . . . . .</b>	<b>142</b>
E.1. Building QGIS with Qt4.x . . . . .	142
E.2. Prepare apt . . . . .	142
E.3. Install Qt4 . . . . .	142
E.4. Install additional software dependencies required by QGIS . . . . .	143
E.5. GRASS Specific Steps . . . . .	143
E.6. Setup ccache (Optional) . . . . .	144
E.7. Prepare your development environment . . . . .	144
E.8. Check out the QGIS Source Code . . . . .	144
E.9. Starting the compile . . . . .	145
E.10. Running QGIS . . . . .	146
<b>F. Creation of MSYS environment for compilation of Quantum GIS . . . . .</b>	<b>146</b>
F.1. Initial setup . . . . .	146
F.1.1. MSYS . . . . .	146
F.1.2. MinGW . . . . .	146
F.1.3. Flex and Bison . . . . .	147
F.2. Installing dependencies . . . . .	147
F.2.1. Getting ready . . . . .	147
F.2.2. GDAL level one . . . . .	148
F.2.3. GRASS . . . . .	149
F.2.4. GDAL level two . . . . .	150
F.2.5. GEOS . . . . .	151
F.2.6. SQLITE . . . . .	151
F.2.7. GSL . . . . .	152
F.2.8. EXPAT . . . . .	152
F.2.9. POSTGRES . . . . .	152
F.3. Cleanup . . . . .	152
<b>G. Building with MS Visual Studio . . . . .</b>	<b>153</b>
G.1. Setup Visual Studio . . . . .	153
G.1.1. Express Edition . . . . .	153
G.1.2. All Editions . . . . .	153
G.2. Download/Install Dependencies . . . . .	154

---

G.2.1. Flex and Bison . . . . .	154
G.2.2. To include PostgreSQL support in Qt . . . . .	154
G.2.3. Qt . . . . .	155
G.2.4. Proj.4 . . . . .	155
G.2.5. GSL . . . . .	156
G.2.6. GEOS . . . . .	156
G.2.7. GDAL . . . . .	157
G.2.8. PostGIS . . . . .	157
G.2.9. Expat . . . . .	157
G.2.10.CMake . . . . .	157
G.3. Building QGIS with CMAKE . . . . .	158
<b>H. Building under Windows using MSVC Express</b>	<b>158</b>
H.1. System preparation . . . . .	159
H.2. Install the libraries archive . . . . .	159
H.3. Install Visual Studio Express 2005 . . . . .	159
H.4. Install Microsoft Platform SDK2 . . . . .	160
H.5. Edit your vsvars . . . . .	163
H.6. Environment Variables . . . . .	164
H.7. Install CMake . . . . .	167
H.8. Install Subversion . . . . .	167
H.9. Initial SVN Check out . . . . .	168
H.10.Create Makefiles using cmake-setup.exe . . . . .	169
H.11.Running and packaging . . . . .	169
<b>I. GNU General Public License</b>	<b>170</b>
I.1. Quantum GIS Qt exception for GPL . . . . .	175
<b>Literatura citada</b>	<b>176</b>

## Índice de figuras

1.	Ventana principal con datos de ejemplo de Alaska (GNU/Linux con KDE)	10
2.	Measure tools in action	17
3.	Diálogo Abrir una capa vectorial soportada por OGR	22
4.	QGIS con el archivo Shape de Alaska cargado	23
5.	Diálogo Propiedades de la capa	30
6.	Seleccionar objeto espacial y elegir acción	35
7.	Diálogo de captura de atributos de digitalización de vectoriales	39
8.	Diálogo de creación de un nuevo vectorial	43
9.	Constructor de consultas	44
10.	Raster context menu	47
11.	Raster Layers Properties Dialog	48
12.	Dialog for adding a WMS server, showing its available layers	54
13.	Adding a WFS layer	58
14.	Diálogo Proyección (GNU/Linux)	60
15.	Diálogo Proyección personalizada (OS X)	62
16.	Crear una localización de GRASS en QGIS	65
17.	Diálogo de edición de GRASS	68
18.	Caja de herramientas de GRASS	73
19.	Módulo generado mediante el análisis del archivo XML	74
20.	Export to MapServer map module in QGIS	77
21.	Diseñador de mapas	80
22.	Diseñador de mapas con la vista del mapa, leyenda, barra de escala y texto añadidos	82
23.	Administrador de complementos	85
24.	Complemento de copyright	88
25.	Complemento flecha de Norte	89
26.	Complemento barra de escala	90
27.	La ventana del diálogo <i>Herramientas GPS</i>	92
28.	Diálogo de selección de archivo de la herramienta de importación	93
29.	La herramienta descargar	94
30.	Diálogo de texto delimitado	97
31.	Archivo seleccionado	98
32.	Campos analizados del archivo de texto	99
33.	Seleccionar los campos X e Y	99
34.	Crear una capa de cuadrícula	100
35.	Seleccionar una imagen a georreferenciar	102
36.	Ajustar la ventana del complemento con la vista del mapa de QGIS	103
37.	Añadir puntos a la imagen ráster	104
38.	Mapa georreferenciado con carreteras superpuestas de la localización spearfish60	105
39.	Introducir nombre de la nueva tabla PostGIS	111
40.	Introducir nombres de campos para la nueva tabla PostGIS	111

41.	Introducir DSN para la conexión a la base de datos PostGIS . . . . .	112
42.	Cuadro de mensaje con los resultados del complemento . . . . .	113
43.	Iniciar la nueva aplicación de demostración . . . . .	122
44.	Añadir una capa a la aplicación de demostración . . . . .	123

## Índice de cuadros

1.	Parámetros de conexión PostGIS . . . . .	25
2.	WMS Connection Parameters . . . . .	52
3.	Example Public WMS URLs . . . . .	53
4.	Herramientas de digitalización de GRASS . . . . .	69
5.	Complementos integrados de QGIS . . . . .	86

---

## 1. Prefacio

¡Bienvenido al maravilloso mundo de los Sistemas de Información Geográfica(SIG)! Quantum GIS (QGIS) es un Sistema de Información Geográfica de Código Abierto. El proyecto nació en Mayo de 2002 y se estableció como un proyecto dentro de SourceForge en junio del mismo año. Hemos trabajado duro para hacer del software SIG (que tradicionalmente es software comercial caro) una posibilidad viable para cualquiera con un acceso básico a un ordenador personal. Actualmente QGIS corre en la mayoría de plataformas Unix, Windows y OS X. QGIS está desarrollado utilizando el Qt toolkit (<http://www.trolltech.com>) y C++. Esto hace que QGIS sea rápido y tenga una interfaz de usuario agradable y fácil de usar.

QGIS espera ser un SIG fácil de usar, proporcionando características y funciones comunes. El objetivo inicial fue proporcionar un visor de datos SIG. QGIS ha alcanzado este punto en su evolución y está siendo utilizado por muchos para sus necesidades diarias de visualización de datos SIG. QGIS soporta un buen número de formatos ráster y vectoriales, con nuevos soportes fácilmente añadidos utilizando su arquitectura de complementos (ver Apéndice A para consultar la lista completa de los formatos de datos soportados). QGIS se ha publicado bajo Licencia Pública (GNU General Public License) (GPL). Desarrollar QGIS bajo esta licencia quiere decir que se puede inspeccionar y modificar el código fuente y las garantías que se tienen, nuestros felices usuarios siempre tienen acceso a un programa SIG gratis y que puede ser libremente modificado. Debe tener una copia de la licencia con su copia de QGIS, y también se puede encontrar como Apéndice I.

**Note:** La última versión de este documento siempre se encuentra en  
<http://qgis.org/docs/userguide.pdf>

### 1.1. Características

QGIS tiene muchas funciones y características comunes a todos los SIG. Las características principales se enumeran aquí debajo, divididas en elementos del núcleo y complementos.

#### Elementos del núcleo

- Soporte ráster y vectorial mediante la librería OGR.
- Soporte para PostgreSQL con tablas espaciales utilizando PostGIS.
- Integración con GRASS, incluida visualización, edición y análisis.
- Digitalización GRASS y OGR/Shapefile.
- Diseño de Mapas.
- Soporte OGC.
- Panel de Vista General.

- Marcadores espaciales.
- Identificar/Seleccionar elementos.
- Editar/Visualizar/Buscar atributos.
- Etiquetado de elementos.
- Proyecciones al vuelo.
- Guardar y recuperar proyectos.
- Exportar ficheros map a Mapserver.
- Cambiar simbología vectorial y raster.
- Arquitectura extensible con complementos.

### **Complementos**

- Añadir capas WFS.
- Añadir capas de texto delimitado.
- Decoración (etiqueta de copyright, flecha de Norte y barra de escala)
- Georreferenciación.
- Herramientas GPS.
- GRASS.
- Generador de mallas.
- Funciones de geoprocesamiento PostgreSQL.
- Herramienta de importación de archivos shape a PostgreSQL/PostGIS (SPIT - Shapefile to PostgreSQL/PostGIS Import Tool)
- Consola de Python.
- openModeller.

## **1.2. Qué es nuevo en 0.9.0**

Como es habitual, la nueva versión 0.9.0 te ofrece muchas características interesantes.

- El lenguaje Python posibilita escribir complementos en Python y crear aplicaciones SIG que utilicen librerías de QGIS.
- Eliminado el sistema de compilación con automake build system - QGIS ahora necesita CMake para su compilación.
- Algunos módulos nuevos de GRASS añadidos a la barra de herramientas.
- Actualizaciones en el editor de Mapas.

- Correcciones en los ficheros shape 2.5D.
- Mejoras en la georeferenciación.
- Soporte de localización extendido a 26 idiomas.



---

## 2. Introducción a los SIG

Un Sistema de Información Geográfica (SIG)?<sup>1</sup> es una colección de software que permite crear, visualizar, consultar y analizar datos geoespaciales. Los datos geoespaciales se refieren a información sobre la localización geográfica de una entidad. Esto habitualmente implica el uso de una coordenada geográfica, como los valores latitud o longitud. Otros términos comunmente utilizados para datos espaciales son: datos geográficos, datos SIG/GIS, datos de mapa, datos de localización, coordenadas y datos geométricos espaciales.

Las aplicaciones que utilizan datos geoespaciales realizan gran variedad de funciones. La producción de mapas es la función más fácil de entender de las aplicaciones geoespaciales. Los programas de edición de mapas cogen datos geoespaciales y los transforman en datos visibles, normalmente sobre pantallas de ordenador o páginas impresas. Las aplicaciones pueden presentar mapas estáticos (una simple imagen) o mapas dinámicos que son personalizados por quien esté viendo el mapa a través de un programa de escritorio o una página web.

Mucha gente cree erroneamente que las aplicaciones geoespaciales sólo producen mapas, pero el análisis de datos geoespaciales es otra de las principales funciones de estas aplicaciones. Algunos tipos de análisis típicos incluyen computación:

1. distancias entre localizaciones geográficas.
2. la cantidad de área (p.ej., metros cuadrados) dentro de cierta región geográfica.
3. qué elementos geográficos solapan otros elementos.
4. la cantidad de solapes entre elementos.
5. el número de localizaciones dentro de cierta distancia.
6. y cosas así...

Esto puede parecer simplista, pero pueden ser aplicadas en todos los tipos de vías a través de muchas disciplinas. El resultado de un análisis se puede mostrar sobre un mapa, pero habitualmente se tabula en un informe para ayuda en la toma de decisiones.

El reciente fenómeno de los servicios basados en localización promete introducir todo tipo de características, pero muchas estarán basadas en una combinación de mapas y análisis. Por ejemplo, puedes tener un teléfono móvil que deje rastro de tu localización geográfica. Si tienes el software adecuado el teléfono puede decirte qué clase de restaurantes están a corta distancia. A pesar de que es una aplicación novedosa de tecnología geoespacial, básicamente está haciendo análisis geoespacial y lista los resultados para ti.

---

<sup>1</sup>Este capítulo es de Tyler Mitchell (<http://www.oreillynet.com/pub/wlg/7053>) y se utiliza bajo Licencia Creative Commons. Tyler es el autor de *Web Mapping Illustrated*, publicado por O'Reilly, 2005.

## 2.1. ¿Por qué todo esto es tan nuevo?

Bien, no todo. Hay muchos dispositivos hardware nuevos que están posibilitando los servicios móviles geoespaciales. También están disponibles muchas aplicaciones geoespaciales de código abierto, pero la existencia de hardware y software especializado en la industria no es nada nuevo. Los sistemas de posicionamiento global (GPS) son frecuentes, pero llevan utilizándose en la industria hace más de una década. También, las herramientas de edición de mapas y análisis han sido un importante mercado, principalmente orientado a industrias de gestión de recursos naturales.

Lo que es nuevo, es cómo el último hardware y software está siendo aplicado y quién lo está aplicando. Los usuarios tradicionales de herramientas de edición de mapas y análisis eran analistas SIG fuertemente entrenados o técnicos de edición de mapas entrenados para utilizar herramientas CAD. Ahora, las capacidades de procesamiento de un PC doméstico y los paquetes software de código abierto han descubierto un arma para aficionados, profesionales, desarrolladores web, etc. para interactuar con datos geoespaciales. La curva de aprendizaje está bajando. El costo está bajando. La cantidad de saturación de tecnología geoespacial se está incrementando.

¿Cómo se almacenan los datos geoespaciales? En pocas palabras, hay dos tipos de datos geoespaciales ampliamente utilizados hoy. En adición a los datos tabulares tradicionales que también son ampliamente utilizados por aplicaciones geoespaciales.

### 2.1.1. Datos ráster

Un tipo de dato geoespacial se llama dato ráster o simplemente "un ráster". La forma más fácilmente reconocible de un raster son las imágenes de satélite o fotos aéreas. Los modelos de elevación o de sombras también se representan típicamente como un ráster. Cualquier tipo de elemento de un mapa se puede representar como un ráster, pero tienen algunas limitaciones.

Un ráster es una malla regular formada por celdas, o en el caso de imágenes, píxeles. Tienen un número fijo de filas y columnas. Cada celda tiene un valor numérico y cierto tamaño geográfico (p.ej. un tamaño de 30x30 metros).

Se utilizan múltiples ráster superpuestos para representar imágenes con más colores (p.ej. un ráster por cada valor de rojo, verde y azul se combinan para crear una imagen a color). Las imágenes de satélite también representan datos en múltiples "bandas". Cada banda es esencialmente un ráster individual, espacialmente superpuesto, donde cada banda mantiene valores de cierta longitud de onda. Como puedes imaginar, un ráster grande, ocupa más espacio en disco. Un ráster con celdas más pequeñas proporciona más detalle, pero ocupa más espacio en disco. El truco es encontrar el balance correcto entre el tamaño de celda para almacenar y el tamaño de celda para análisis o mapas.

### 2.1.2. Datos vectoriales

Los datos vectoriales también se utilizan en aplicaciones geoespaciales. Si estuviste atento durante las clases de trigonometría y coordenadas geográficas, estarás familiarizado con algunas de las cualidades de los datos vectoriales. En su sentido más simple, los vectores son una vía para describir una localización utilizando un conjunto de coordenadas. Cada coordenada hace referencia a una localización geográfica mediante un sistema de valores x e y.

Esto puede parecerse a un plano cartesiano - ya sabes, los diagramas de la escuela que mostraban ejes x, y. Los puedes haber utilizado en gráficas para ver los ahorros de tu jubilación o para ver el incremento del interés de la hipoteca, pero el concepto es el mismo para el análisis espacial y la edición de mapas.

Hay formas obvias de representar estas coordenadas geográficas dependiendo de los propósitos. Esto es un área completa de estudio para otro día -proyecciones de mapas.

Los datos vectoriales son de tres formas:

1. Puntos - Una única coordenada (x y) representa una localización geográfica discreta.
2. Líneas - Múltiples coordenadas (x1 y1, x2 y2, x3 y4, ... xn yn) unidas en cierto orden. Como dibujar una línea de punto (x1 y1) a punto (x2 y2). Las partes entre cada punto se consideran segmentos. Estos tienen una longitud y la línea puede indicar la dirección base del orden de los puntos. Técnicamente, una línea es un par de coordenadas conectadas; mientras que un segmento múltiple son múltiples líneas conectadas juntas.
3. Polígonos - Cuando las líneas están unidas por más de dos puntos, con el último punto coincidiendo con el primero, lo podemos llamar polígono. Un triángulo, círculo, rectángulo, etc. son todos polígonos. La característica clave de los polígonos es que tienen un área dentro de ellos.

---

## 3. Comenzar

Este capítulo ofrece un vistazo rápido del funcionamiento de QGIS con datos disponibles en la página web de QGIS.

### 3.1. Instalación

La compilación de QGIS desde su código fuente está documentada en el Apéndice ?? para Windows, Apéndice ?? para Mac OSX y Apéndice ?? para GNU/Linux. Las instrucciones de instalación se distribuyen con el código fuente de QGIS y también están disponibles en <http://qgis.org>.

Los paquetes estándar de instalación están disponibles para Windows y Mac OS X. Se proporcionan los paquetes binarios para la mayoría de sabores de GNU/Linux. Obtenga la última información sobre los paquetes binarios en el sitio web QGIS <http://download.qgis.org>.

### 3.2. Datos de ejemplo

Si no tiene datos SIG a mano, puede obtener un conjunto de datos de Alaska de la web de QGIS <http://qgis.org>. La proyección para los datos de Alaska es Albers Equal Area con las unidades en metros:

```
PROJCS["NAD_1927_Albers",  
  GEOGCS["GCS_North_American_1927",  
    DATUM ["D_North_American_1927",  
      SPHEROID["Clarke_1866", 6378206.4,294.9786982]],  
      PRIMEM["Greenwich",0.0],  
      UNIT["Degree", 0.0174532925199433]],  
    PROJECTION["Albers"],  
    PARAMETER["False_Easting", 0.0],  
    PARAMETER["False_Northing",0.0],  
    PARAMETER["Central_Meridian",-154.0],  
    PARAMETER["Standard_Parallel_1", 55.0],  
    PARAMETER["Standard_Parallel_2",65.0],  
    PARAMETER ["Latitude_Of_Origin",50.0],  
    UNIT["Meter",1.0]]
```

Para utilizarlo con GRASS, se puede obtener una base de datos de GRASS de ejemplo (p.ej. Spearfish) de la web-SIG oficial de GRASS <http://grass.osgeo.org/download/data.php>. La proyección oficial del conjunto de datos de Spearfish es UTM Zone 13, Northern Hemisphere:

```
PROJCS["UTM Zone 13, Northern Hemisphere",
  GEOGCS["clark66",
    DATUM["North_American_Datum_1927",
      SPHEROID["clark66",6378206.4,294.9786982]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",-105],
  PARAMETER["scale_factor",0.9996],
  PARAMETER["false_easting",500000],
  PARAMETER["false_northing",0],
  UNIT["meter",1]]
```

Estos conjuntos de datos se utilizarán como base para muchos ejemplos y capturas de pantalla en este documento.

### 3.3. Iniciar QGIS

Suponiendo que tiene QGIS instalado en el PATH, puede iniciar QGIS tecleando: **qgis** en línea de órdenes o haciendo doble clic sobre el enlace a QGIS (o acceso directo) del escritorio. Bajo MS Windows, inicie QGIS a través del menú Inicio y bajo Mac OS X, haga doble clic en el icono de su carpeta de aplicaciones.

#### 3.3.1. Opciones de línea de órdenes

QGIS soporta un buen número de opciones cuando se inicia desde la línea de órdenes. Para obtener la lista de opciones, teclee `qgis --help` en la línea de órdenes. La sentencia de uso para QGIS es:

```
qgis --help
Quantum GIS - 0.9.0 'Ganymede'
Quantum GIS (QGIS) is a viewer for spatial data sets, including
raster and vector data.
Usage: qgis [options] [FILES]
  options:
    [--snapshot filename]  emit snapshot of loaded datasets to given file
    [--lang language]      use language for interface text
    [--project projectfile] load the given QGIS project
    [--extent xmin,ymin,xmax,ymax] set initial map extent
    [--help]                this text
```

**FILES:**

Files specified on the command line can include rasters, vectors, and QGIS project files (.qgs):

1. Rasters - Supported formats include GeoTiff, DEM and others supported by GDAL
2. Vectors - Supported formats include ESRI Shapefiles and others supported by OGR and PostgreSQL layers using the PostGIS extension

---

**Tip 1 EJEMPLO UTILIZANDO ARGUMENTOS EN LÍNEA DE ÓRDENES**

---

Puede iniciar QGIS especificando uno o más archivos de datos sobre la línea de órdenes. Por ejemplo, suponiendo que está en su directorio de datos, puede iniciar QGIS con dos archivos shape y un ráster preparados para cargarlos en el arranque utilizando la siguiente orden: `qgis ak_shade.tif alaska.shp majrivers.shp`

---

**Opción de línea de órdenes --snapshot**

Esta opción permite crear una captura de pantalla en formato PNG de la vista actual. Esto puede ser útil cuando se tienen muchos proyectos y se quieren generar capturas de los datos.

Actualmente esto genera un fichero PNG con 800x600 pixeles. Se puede añadir un nombre después de `--snapshot`.

**Opción de línea de órdenes --lang**

En base a su configuración local, QGIS selecciona la localización correcta. Si se quiere cambiar el idioma, se puede proporcionar otro idioma con esta opción.

**Opción de línea de órdenes --project**

También es posible iniciar QGIS con un proyecto existente. Simplemente añada la opción `-project` seguida del nombre de su proyecto y QGIS se abrirá con todas las capas cargadas definidas en el fichero dado.

**Opción de línea de órdenes --extent**

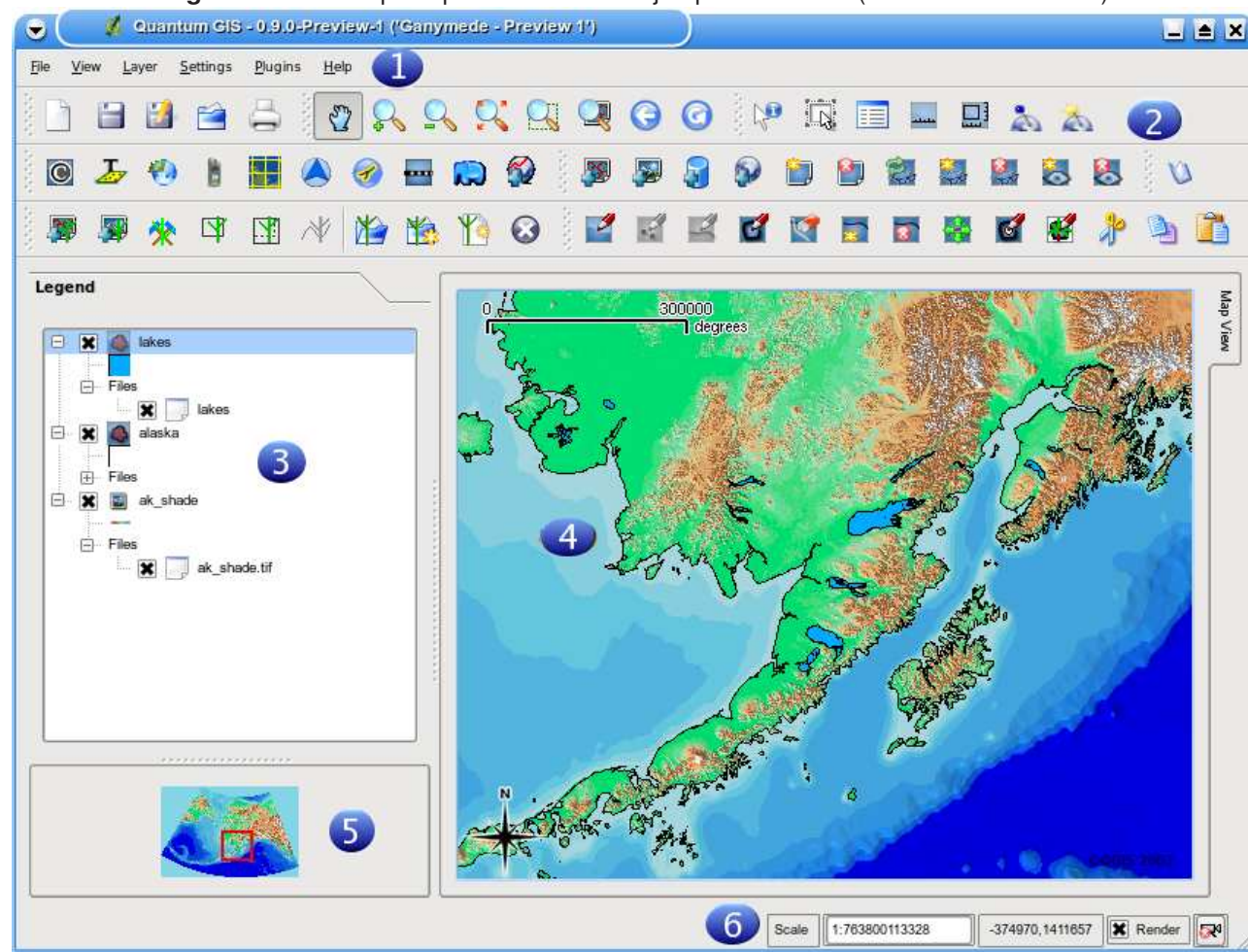
Para arrancar con una extensión específica del mapa utilice esta opción. Es necesario añadir los límites en el siguiente orden separados por comas:

```
--extent xmin,ymin,xmax,ymax
```

### 3.4. Interfaz Gráfica de Usuario (GUI) de QGIS

Cuando QGIS arranca, se encuentra con la GUI como se muestra abajo (los números del 1 hasta el 6 en ovalos azules señalan las seis áreas principales de la interfaz que se describen abajo):

**Figura 1:** Ventana principal con datos de ejemplo de Alaska (GNU/Linux con KDE)



**Nota:** La decoración de su ventana (barra de título, etc.) puede aparecer distinta dependiendo del sistema operativo y administrador de ventanas.

La Interfaz Gráfica de Usuario (GUI) de QGIS está dividida en seis áreas:

- |                          |                           |
|--------------------------|---------------------------|
| 1. Barra de menús        | 4. Vista del mapa         |
| 2. Barra de herramientas | 5. Vista general del mapa |
| 3. Leyenda del mapa      | 6. Barra de estado        |

Estos seis componentes de la interfaz de QGIS están descritos con más detalle en las siguientes



secciones.

### 3.4.1. Barra de menús

La barra de menús proporciona acceso a varias características de QGIS utilizando menús jerárquicos estándar. El nivel superior de los menús y un extracto de algunas funciones proporcionadas son:

- **Archivo**

- **Nuevo proyecto** - ver Sección 3.7
- **Abrir proyecto** - ver Sección 3.7
- **Abrir proyectos recientes** - ver Sección 3.7
- **Guardar proyecto** - ver Sección 3.7
- **Guardar proyecto como...** - ver Sección 3.7
- Guardar como imagen...
- Exportar a mapa de MapServer - ver Sección 9
- Imprimir - ver Sección 10
- Salir

- **Ver**

- Zum general
- Zum a Selección
- Zum a la capa
- Zum anterior
- Refrescar
- Mostrar marcadores
- Nuevo marcador...
- Mostrar todas las barras de herramientas posibles
- Ocultar todas las barras de herramientas posibles
- Visibilidad de barras de Herramientas

- **Capa**

- **Añadir una capa vectorial** - ver Sección 4
- **Añadir una capa ráster** - ver Sección 5
- **Añadir una capa de PostGIS** - ver Sección 4.2
- **Añadir una capa WMS** - ver Sección 6.2
- Eliminar capa
- Nueva capa vectorial - ver Sección 4.4.3
- Llevar al localizador
- Añadir todo al localizador



- Eliminar todo del localizador
- Ocultar todas las capas
- Mostrar todas las capas
- **Configuración**
  - **Propiedades del proyecto...** - ver Sección 3.7
  - **Proyección personalizada...** - ver Sección 7.3
  - **Opciones...** - ver Sección 3.8
- **Complementos** - (Posteriores elementos del menú serán añadidos por los complementos según se vayan activando.)
  - Administrador de complementos... - ver Sección 11.1.2
- **Ayuda**
  - Contenido sde la Ayuda
  - Página web de QGIS
  - Comprobar versión de QGIS
  - Acerca de

### 3.4.2. Barras de herramientas

Las barras de herramientas proporcionan acceso a la mayoría de las funciones de los menús, así como herramientas adicionales para interactuar con el mapa. Cada elemento de una barra de herramientas dispone de una ayuda emergente. Mantenga el ratón sobre el elemento y aparecerá una breve descripción sobre la función de la herramienta.

Cada barra de menú se puede mover de acuerdo con sus necesidades. Además, cada barra de menú se puede desactivar usando el menú contextual que aparece al pulsar con el botón derecho sobre la barra de herramientas.

---

**Tip 2 RESTAURAR BARRAS DE HERRAMIENTAS**

---

Si ha ocultado accidentalmente todas sus barras de herramientas, puede recuperarlas seleccionando Mostrar todas las barras de herramientas en el menú Ver.

---

### 3.4.3. Leyenda del mapa

El área de la leyenda del mapa se usa para establecer la visibilidad de las capas y su ordenación en altura. La ordenación en altura significa que las capas listadas más próximas a la parte superior de la leyenda se dibujan sobre aquellas listadas más abajo en la leyenda. La casilla de verificación de cada entrada de la leyenda se puede usar para mostrar u ocultar la capa.

Las capas se pueden agrupar en la ventana de la leyenda añadiendo un grupo de capas y arrastrando las capas dentro de él. Para hacerlo, vaya con el ratón a la ventana de la leyenda, pulse con el botón derecho y seleccione 'Añadir grupo'. Aparecerá una carpeta nueva. Ahora arrastre la capas al símbolo de la carpeta. Con esto será posible conmutar la visibilidad de todas las capas del grupo con un solo clic. Para sacar las capas de un grupo, vaya con el ratón al símbolo de la capa, pulse en él con el botón derecho y seleccione 'Subir el elemento al nivel superior'. Para dar a la carpeta un nombre nuevo, seleccione 'Cambiar nombre' en el menú del botón derecho pulsando sobre el grupo.

El contenido del menú contextual del botón derecho del ratón depende de si el elemento cargado en la leyenda sobre el que se pulsa es una capa ráster o vectorial. Para las capas vectoriales de GRASS no aparece la opción 'Conmutar edición'. Vea la sección 8.5 para información sobre la edición de capas vectoriales de GRASS.

■ **Menú del botón derecho para capas ráster**

- Zum a la extensión de la capa
- Zum a la mejor escala (100 %)
- Mostrar en el localizador
- Eliminar
- Propiedades
- Cambiar nombre
- Añadir grupo
- Expandir todo
- Comprimir todo
- Mostrar grupos de archivos

■ **Menú del botón derecho para capas vectoriales**

- Zum a la extensión de la capa
- Mostrar en el localizador
- Eliminar
- Abrir tabla de atributos
- Conmutar edición (no disponible para capas de GRASS)
- Guardar como archivo shape
- Guardar selección como archivo shape...
- Propiedades
- Cambiar nombre
- Añadir grupo
- Expandir todo
- Comprimir todo
- Mostrar grupos de archivos

■ **Menú del botón derecho para grupos de capas**

- Eliminar
- Cambiar nombre
- Añadir grupo
- Expandir todo
- Comprimir todo
- Mostrar grupos de archivos

Si varias fuentes de datos vectoriales tienen el mismo tipo vectorial y los mismos atributos se puede agrupar su simbolización. Esto significa que si la simbolización de una fuente de datos se cambia las otras tendrán la nueva simbolización de forma automática. Para agrupar las simbologías, abra el menú contextual en la ventana de la leyenda y seleccione «Mostrar grupos de archivos». Aparecerán los grupos de archivos de las capas. Ahora es posible arrastrar un archivo de un grupo de archivos dentro de otro. Si se hace esto las simbologías se agruparán. Tenga en cuenta que QGIS sólo permite arrastrar si las dos capas pueden compartir la simbología (mismo tipo vectorial y atributos).

#### 3.4.4. Vista del mapa

Este es el «final del negocio» de QGIS - ¡los mapas se muestran en este área! El mapa que se visualice en esta ventana dependerá de las capas vectoriales y ráster que haya seleccionado para cargar (vea las secciones que siguen para más información sobre cómo cargar capas). La vista del mapa se puede desplazar (moviendo el foco de visualización del mapa a otra región) y se puede acercar y alejar el zum. Se pueden realizar otras operaciones sobre el mapa como se indica en la descripción de las barras de herramientas más arriba. La vista del mapa y la leyenda están íntimamente relacionadas – los mapas de la vista reflejan los cambios que se hacen en el área de la leyenda.

---

**Tip 3 HACER ZUM SOBRE EL MAPA CON LA RUEDA DEL RATÓN**

---

Se puede usar la rueda del ratón para acercar o alejar el zum sobre el mapa. Sitúe el cursor del ratón dentro del área del mapa mueva la rueda hacia delante para acercar el zum y hacia atrás para alejarlo. El cursor del ratón es el centro sobre el que se hace zum. Puede personalizar el comportamiento de zum con la rueda del ratón usando la pestaña *Herramientas de mapa* bajo el menú *Configuración|Opciones*.

---

#### 3.4.5. Localizador del mapa

El área del localizador proporciona una vista a toda la extensión de las capas añadidas al mapa. Dentro de la vista hay un rectángulo que muestra la extensión actual del mapa. Esto permite determinar rápidamente que área del mapa se está viendo actualmente. Tenga en cuenta que las etiquetas no se visualizan en el localizador, incluso aunque las capas se hayan configurado para ser etiquetadas. Puede añadir una sola capa al localizador pulsando con el botón derecho sobre ella en la leyenda y seleccionando *Mostrar en el localizador*. También puede añadir o quitar todas las capas del localizador usando las herramientas del localizador en la barra de herramientas.

También puede coger el rectángulo rojo que muestra la extensión actual y desplazarlo alrededor; la vista del mapa se actualizará conforme se desplace.

#### 3.4.6. Barra de estado

La barra de estado muestra su posición actual en las coordenadas del mapa (ej.: metros o grados decimales) a medida que el puntero del ratón se mueve por la vista del mapa. La barra de estado también muestra la extensión de la vista del mapa mientras desplaza el mapa o modifica el nivel del zum. Una barra de progreso en la barra de estado muestra el progreso de la representación de cada capa que se está dibujando en la vista del mapa. En algunos casos, tales como la recopilación de estadísticas en capas ráster, la barra de progreso se usará para mostrar el estado de operaciones prolongadas. En la parte derecha de la barra de estado hay una pequeña casilla de verificación que se puede usar para evitar temporalmente que se representen en la vista del mapa (vea la Sección 3.5 más abajo). En el extremo derecho de la barra de estado hay un icono de un proyector. Pulsando en él se abren las propiedades de la proyección del proyecto actual.

### 3.5. Representación

Por omisión, QGIS representa todas las capas visibles cada vez que la vista del mapa se debe refrescar. Los eventos que desencadenan un refresco de la vista del mapa incluyen:

- Añadir una capa.
- Panorámica o zum.
- Redimensionar la ventana de QGIS.
- Cambiar la visibilidad de una o más capas.

QGIS permite controlar el proceso de representación de distintas formas.

#### 3.5.1. Representación dependiente de la escala

La representación dependiente de la escala permite especificar las escalas máxima y mínima a las que una capa será visible. Para establecer la representación dependiente de la escala, abra el diálogo de propiedades haciendo doble clic sobre una capa en la leyenda. En la pestaña *General*, marque la casilla *Utilizar representación dependiente de la escala* y luego establezca la escala mínima y máxima .

Puede determinar los valores de escala haciendo zum primero al nivel que quiera usar y anotando el valor de la escala mostrado en la barra de estado de QGIS.

### 3.5.2. Controlar la representación del mapa

La representación del mapa se puede controlar de las siguientes formas:

#### Suspender la representación

Para suspender la representación, marque la casilla *Representar* en la esquina inferior derecha de la barra de estado. Cuando la casilla *Representar* no está marcada, QGIS no redibuja la vista del mapa en respuesta a ninguno de los eventos descritos en la Sección 3.5. Entre los ejemplos de cuando puede desear suspender la representación se incluyen:

- Añadir muchas capas y simbolizarlas antes de que se dibujen.
- Añadir una o más capas grandes y establecer la dependencia de la escala antes de dibujarlas.
- Añadir una o más capas grandes y hacer zoom a una zona específica antes de dibujarla
- Cualquier combinación de las anteriores.

Marcar la casilla *Representar* activa la representación y desencadena el refresco inmediato de la vista del mapa.

#### Establecer opción al añadir capas

Se puede establecer una opción para que siempre se carguen las capas nuevas sin dibujarlas. Esto significa que las capas se añadirán al mapa, pero su casilla de verificación de visualización en la leyenda estará desmarcada por omisión. Para establecer esta opción, seleccione *Opciones* en el menú *Configuración* y pulse en la pestaña *Representación*. Desmarque la casilla *Por omisión, las nuevas casillas añadidas al mapa se deben visualizar*. Cualquier capa que añada al mapa estará desactivada (invisible) por omisión.

#### Updating the Map Display During Rendering

You can set an option to update the map display as features are drawn. By default, QGIS does not display any features for a layer until the entire layer has been rendered. To update the display as features are read from the datastore, choose *Options* from the *Settings* menu and click on the *Rendering* tab. Set the feature count to an appropriate value to update the display during rendering. Setting a value of 0 disables update during drawing (this is the default). Setting a value too low will result in poor performance as the map canvas is continually updated during the reading of the features. A suggested value to start with is 500.

### 3.6. Measuring

Measuring works within projected coordinate systems only (e.g., UTM). If the loaded map is defined with a geographic coordinate system (latitude/longitude), the results from line or area measurements will be incorrect. To fix this you need to set an appropriate map coordinate system.

#### 3.6.1. Measure length



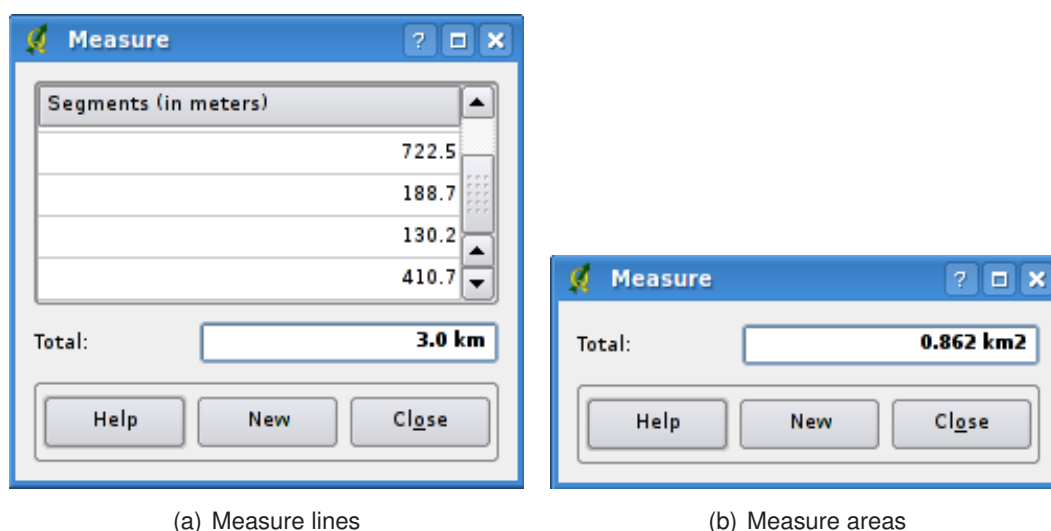
QGIS is also able to measure real distances between given points according to a defined ellipsoid. Therefore choose *Options* from the *Settings* menu, click on the *Map tools* tab and choose the appropriate ellipsoid. The tool then allows you to click points on the map. Each segment-length shows up in the measure-window and additionally the total length is printed. To stop measuring click your right mouse button.

#### 3.6.2. Measure areas



Also areas can be measured. The window only shows the accumulated area-size in the measure window (see figure 2).

Figura 2: Measure tools in action



### 3.7. Projects

The state of your QGIS session is considered a Project. QGIS works on one project at a time. Settings are either considered as being per-project, or as a default for new projects (see Section 3.8).

QGIS can save the state of your workspace into a project file using the menu option *File->Save Project*.

Loading saved projects is a similar process.

The kinds of information saved in a project file include:

- Layers added
- Layer properties, including symbolization
- Projection for the map view
- Last viewed extent

The project file is saved in XML format, so it is possible to edit the file outside QGIS if you know what you are doing.

The file format was updated several times compared to earlier QGIS versions. Project files from older QGIS versions may not work properly anymore.

### 3.8. GUI Options



Some basic options for QGIS can be selected using the Options dialog. Select the *Settings* entry from the menu and choose *Options* (Alt-O). The tabs where you can optimize your options are:

#### General Tab

- ask to save project changes when required

#### Appearance Tab

- Hide or show splash screen at startup
- Change the icon theme
- Change Selection and background Color
- Make layer names appear with Capitals

### Rendering Tab

- Update features during drawing or not until all features have been read.
- Set new layer visible or invisible when loaded
- Make lines appear less jagged at the expense of some drawing performance
- Fix problems with incorrectly filled polygons
- Continuously redraw when dragging the legend/map divider

### Map tools Tab

- Define Search Radius as a percentage of the map width
- Define Ellipsoid for distance calculations
- Set Rubberband Color for Measure Tool
- Define Mouse wheel action (Zoom, Zoom and recenter, Nothing)
- Set Zoom factor for wheel mouse

### Projection Tab

- Define what to do, when a layer is loaded without projection information
  - Prompt for projection
  - Project wide default projection will be used
  - Global default projection displayed below will be used

### Locale Tab

- Overwrite system locale and use defined locale instead
- Information about active system locale

### Help Browser Tab

- Define Browser to display help documents

You can modify the options according to your needs. Some of the changes may require a restart of QGIS before they will be effective.

On GNU/Linux everything is saved in:

```
$HOME/.config/QuantumGIS/qgis.conf
```

This is a normal text file consisting of blocks, where QGIS saves its current display options, PostGIS and WMS connections, and other settings.



On Windows settings are stored in the registry under:

```
\\HKEY_CURRENT_USER\\Software\\QuantumGIS\\qgis
```

On OS X you can find your settings in:

```
$HOME/Library/Preferences/org.qgis.qgis.plist
```

## 3.9. Spatial Bookmarks

Spatial Bookmarks allow you to “bookmark” a geographic location and return to it later.

### 3.9.1. Creating a Bookmark

To create a bookmark:

1. Zoom or pan to the area of interest.
2. Select the menu option *View->New Bookmark* or type Ctrl-B.
3. Enter a descriptive name for the bookmark (up to 255 characters).
4. Click *OK* to add the bookmark or *Cancel* to exit without adding the bookmark.

Note that you can have multiple bookmarks with the same name.

### 3.9.2. Working with Bookmarks

To use or manage bookmarks, select the menu option *View->Show Bookmarks*. The bookmarks dialog allows you to zoom to or delete a bookmark. You can not edit the bookmark name or coordinates.

### 3.9.3. Zooming to a Bookmark

From the Bookmarks dialog, select the desired bookmark by clicking on it, then click the *Zoom To* button. You can also zoom to a bookmark by double-clicking on it.

### 3.9.4. Deleting a Bookmark

To delete a bookmark from the Bookmarks dialog, click on it then click the *Delete* button. Confirm your choice by clicking *Yes* or cancel the delete by clicking *No*.

---

## 4. Trabajar con datos vectoriales

QGIS soporta datos vectoriales en distintos formatos, incluyendo aquellos soportados por el complemento del proveedor de datos de la biblioteca OGR, tales como los archivos shape de ESRI, MapInfo MIF (formato de intercambio) y MapInfo TAB (formato nativo). QGIS también soporta capas PostGIS en una base de datos PostgreSQL usando el complemento del proveedor de datos PostgreSQL. Complementos de proveedores de datos adicionales proporcionan soporte para tipos de datos adicionales (ej.: texto delimitado).

Esta sección describe cómo trabajar con dos formatos habituales: archivos shape de ESRI y capas PostGIS. Muchas de las funciones disponibles en QGIS funcionan igual cualquiera que sea la fuente de datos vectoriales. Esto es así por diseño e incluye las funciones identificar, seleccionar, etiquetar y atributos.

Trabajar con datos vectoriales de GRASS se describe en la sección 8.

### 4.1. Archivos shape de ESRI

El soporte para archivos shape de ESRI es proporcionado por una biblioteca de funciones conocida como OGR Simple Feature Library (<http://www.gdal.org/ogr>). Vea el Apéndice A.1 para una lista de todos los formatos soportados por OGR.

Un archivo shape en realidad consiste en al menos tres archivos:

- archivo .shp que contiene las geometrías del objeto espacial.
- archivo .dbf que contiene los atributos en formato dBase.
- archivo índice .shx.

Lo ideal es que haya otro archivo con extensión .prj, que contiene la información de la proyección del archivo shape.

Puede haber más archivos pertenecientes al conjunto de datos de un archivo shape. Para tener una visión más amplia sobre esto recomendamos las especificaciones técnicas del formato shape, que se pueden encontrar en <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

#### 4.1.1. Cargar un archivo Shape

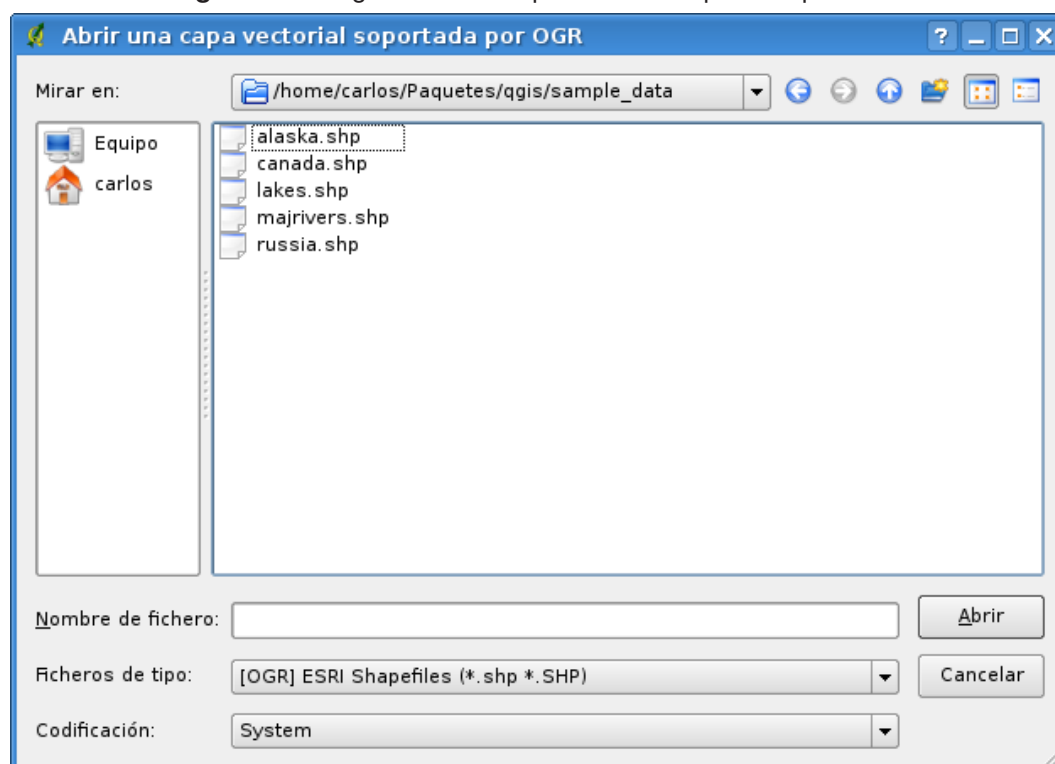


Para cargar un archivo shape, arranque QGIS y haga clic en el botón *Añadir una capa vectorial* de la barra de herramientas o simplemente pulse la tecla "V". Esta misma herramienta se puede usar para cargar cualquiera de los formatos soportados por la biblioteca OGR.

Al hacer clic en la herramienta aparece un diálogo estándar para abrir archivos (véase la Figura 3) que permite navegar por el sistema de archivos y cargar un archivo shape u otra fuente de datos soportada. El cuadro de selección *Archivos de tipo* permite preseleccionar algunos formatos de archivo soportados por OGR.

También puede seleccionar el tipo de codificación del archivo shape si lo desea.

**Figura 3:** Diálogo Abrir una capa vectorial soportada por OGR



Al seleccionar un archivo shape de la lista y hacer clic en Abrir, éste se carga en QGIS. La Figura 4 muestra QGIS después de cargar el archivo alaska.shp.

---

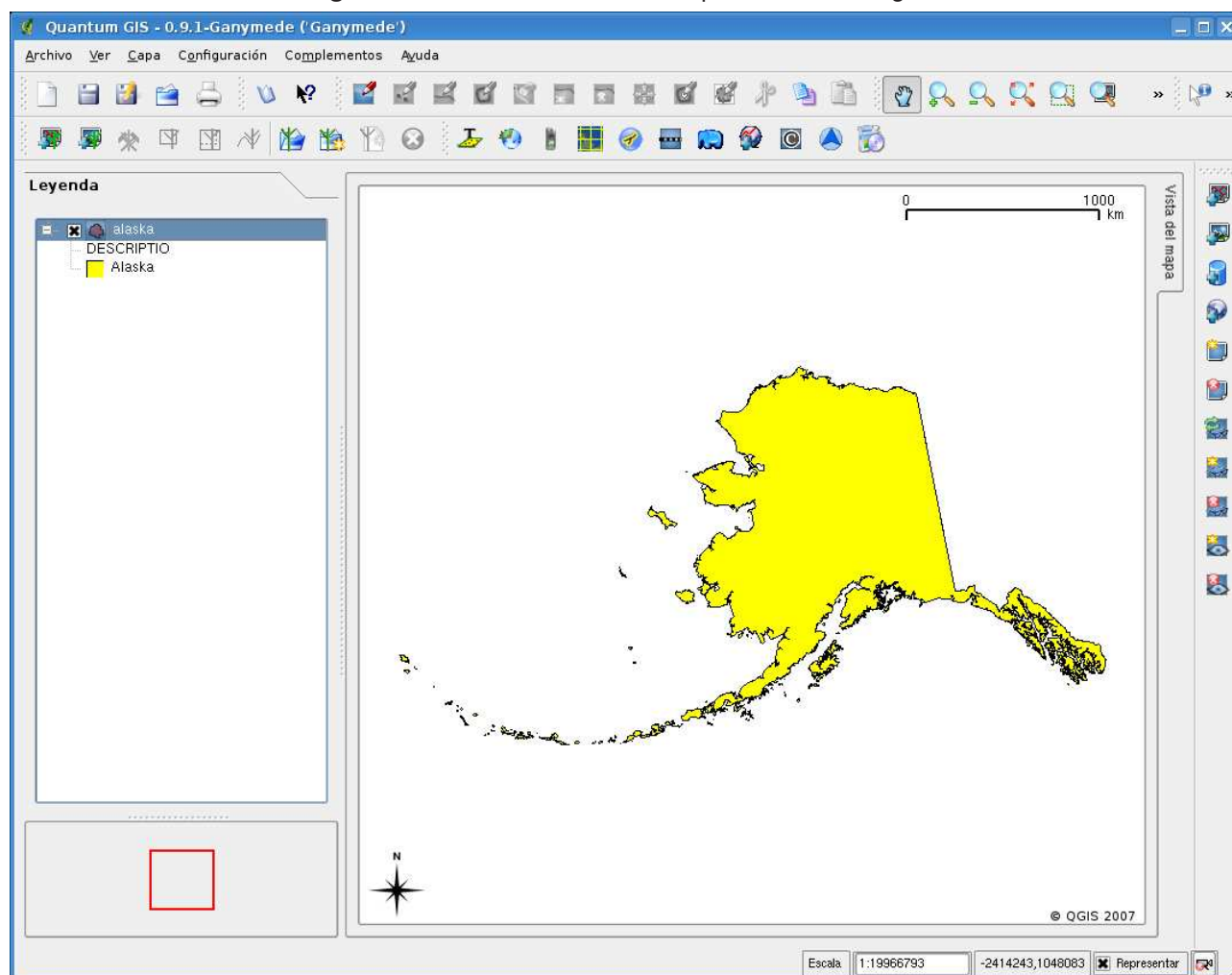
#### **Tip 4** COLORES DE CAPAS

Cuando añade una capa al mapa, se le asigna un color aleatorio. Cuando se añade más de una capa a la vez, se asignan colores diferentes a cada una.

---

Una vez cargada, se puede desplazar por el archivo shape usando las herramientas de navegación del mapa. Para cambiar la simbología de una capa, abra el diálogo de propiedades de la capa haciendo doble clic en el nombre de la capa en la leyenda o clic derecho y seleccionando *Propiedades* del menú emergente. Vea la Sección 4.3.1 para más información sobre la configuración de la simbología de capas vectoriales.

Figura 4: QGIS con el archivo Shape de Alaska cargado



#### 4.1.2. Mejorar el rendimiento

Para mejorar el rendimiento al dibujar un archivo shape, puede crear un índice espacial. Un índice espacial mejorará la velocidad tanto al hacer zoom como al desplazar la capa. Los índices espaciales usados por QGIS tienen la extensión *.qix*.

Siga estos pasos para crear el índice:

- Cargue un archivo shape.
- Abra el diálogo *Propiedades de la capa* haciendo doble clic en el nombre del archivo shape en la leyenda o mediante clic derecho y eligiendo *Propiedades* en el menú emergente.
- En la pestaña *General* haga clic en el botón *Crear* en el recuadro *Índice espacial*.

### 4.1.3. Cargar una capa MapInfo

Para cargar una capa MapInfo, haga clic en el botón *Añadir una capa vectorial* de la barra de herramientas o pulse la tecla "V", cambie el filtro de tipo de archivo a *MapInfo (\*.mif \*.tab \*.MIF \*.TAB)* y seleccione la capa que quiere cargar.

### 4.1.4. Cargar una cobertura de ArcInfo

Cargar una cobertura de ArcInfo se hace usando el mismo método que para archivos shape o capas MapInfo. Haga clic en el botón *Añadir una capa vectorial* de la barra de herramientas o pulse la tecla "V" para abrir el diálogo y cambie el filtro de tipo de archivo a *Todos los archivos (\*.\*)*. Navegue al directorio de coberturas y elija uno de los siguientes archivos (si están presentes en su cobertura):

- .lab - para cargar una capa de etiquetas (etiquetas de polígonos o puntos fijos).
- .cnt - para cargar una capa de centroides de polígono.
- .arc - para cargar una capa arc (línea).
- .pal - para cargar una capa de polígonos.

## 4.2. Capas PostGIS

Las capas PostGIS se guardan en una base de datos PostgreSQL. Las ventajas de PostGIS son el indexado espacial, filtrado y capacidad de consulta. Usar las funciones vectoriales de PostGIS, tales como seleccionar e identificar, es más preciso que con otras capas en QGIS.

Para usar capas PostGIS debe:

- Crear una conexión guardada en QGIS a la base de datos PostgreSQL (si no hay ya una definida).
- Conectarse a la base de datos.
- Seleccionar la capa a añadir al mapa.
- Opcionalmente, proporcionar una sentencia SQL *where* para definir qué objetos espaciales cargar de la capa.
- Cargar la capa.

### 4.2.1. Crear una conexión guardada



La primera vez que use una fuente de datos PostGIS, debe crear una conexión a la base de datos PostgreSQL que contenga los datos. Empiece por hacer clic en el botón *Añadir una capa*

de *PostGIS* de la barra de herramientas, seleccionar la opción *Añadir una capa de PostGIS...* del menú *Capa* o pulsar la tecla "D". Se mostrará el diálogo *Añadir tabla(s) PostGIS*. Para acceder al administrador de conexiones, pulse el botón *Nueva* para mostrar el diálogo *Crear una nueva conexión a PostGIS*. Los parámetros requeridos para una conexión se muestran en la Tabla 1.

**Cuadro 1:** Parámetros de conexión PostGIS

Nombre	Un nombre para esta conexión. Puede ser el mismo que el de la <i>Base de datos</i> .
Servidor	Nombre del servidor de la base de datos. Debe ser un nombre de servidor que se pueda resolver, lo mismo que si fuera a ser usado para abrir una conexión telnet o hacer ping al servidor.
Base de datos	Nombre de la base de datos.
Puerto	Número del puerto al que responde el servidor de la base de datos PostgreSQL. El puerto predeterminado es 5432.
Nombre de usuario	Nombre de usuario utilizado para conectarse a la base de datos.
Contraseña	Contraseña usada con el <i>Nombre de usuario</i> para conectarse a la base de datos.

Una vez que se han rellenado los parámetros, puede probar la conexión pulsando en el botón *Probar conexión*. Para guardar la contraseña con la información de la conexión, marque la opción *Guardar contraseña*.

---

#### **Tip 5** CONFIGURACIÓN DE USUARIO DE QGIS Y SEGURIDAD

Su configuración personal para QGIS se guarda en base al sistema operativo. En Linux/Unix, la configuración se guarda en su directorio personal en `.qt/qgisrc`. En Windows, la configuración se guarda en el registro. Dependiendo de su entorno de procesamiento, guardar contraseñas en su configuración de QGIS puede ser un riesgo de seguridad.

---

#### **4.2.2. Cargar una capa PostGIS**



Una vez que tiene una o más conexiones definidas, puede cargar capas de la base de datos PostgreSQL. Por supuesto, esto requiere tener datos en PostgreSQL. Vea la Sección 4.2.4 donde se trata la importación de datos a la base de datos.

Para cargar una capa desde PostGIS, realice los siguientes pasos:

- Si el diálogo de capas PostGIS no está aún abierto, pulse el botón *Añadir una capa de PostGIS* de la barra de herramientas.
- Seleccione la conexión de la lista desplegable y pulse *Conectar*.
- Encuentre la capa que desee añadir en la lista de capas disponibles.

- Selecciónela haciendo clic en ella. Puede seleccionar múltiples capas manteniendo pulsada la tecla Mayúsculas a la vez que hace clic. Vea la Sección 4.5 para información sobre el uso del constructor de consultas de PostgreSQL para definir mejor la capa.
- Haga clic en el botón *Añadir* para añadir la capa al mapa.

---

**Tip 6 CAPAS POSTGIS**

---

Normalmente una capa PostGIS está definida por una entrada en la tabla `geometry_columns`. Desde la versión 0.8.1 en adelante, QGIS puede cargar capas que no tienen una entrada en la tabla `geometry_columns`. Esto incluye tanto tablas como vistas. Definir una vista espacial proporciona un método potente de visualizar sus datos. Consulte su manual de PostgreSQL para información sobre la creación de vistas.

---

### 4.2.3. Algunos detalles sobre las capas PostgreSQL

Esta sección contiene algunos detalles sobre cómo accede QGIS a capas PostgreSQL. La mayoría de las veces QGIS debería simplemente proporcionarle una lista de tablas de base de datos que puede cargar y cargarlas según se le pida. Sin embargo, si tiene problemas para cargar una tabla de PostgreSQL en QGIS, la información que sigue le puede ayudar a entender cualquier mensaje de QGIS y darle una dirección en la que cambiar la tabla de PostgreSQL o la definición de la vista para permitir que QGIS la cargue.

QGIS requiere que las capas PostgreSQL contengan una columna que se pueda usar como clave única para la capa. Esto normalmente significa que la tabla necesita una clave primaria o tener una columna con una restricción única en ella. QGIS además necesita que esta columna sea de tipo `int4` (un entero con un tamaño de 4 bytes). Si una tabla carece de estos elementos, la columna `oid` se usará en su lugar. Las prestaciones mejorarán si la columna está indexada (tenga en cuenta que las claves primarias se indexan automáticamente en PostgreSQL).

Si la capa PostgreSQL es una vista existen los mismos requisitos, pero las vistas no tienen claves primarias o columnas con restricciones únicas. En este caso QGIS intentará encontrar una columna en la vista que sea derivada de una columna de la tabla que sea adecuada. Si no se puede encontrar una, QGIS no cargará la capa. Si ocurre esto, la solución es modificar la vista para que incluya una columna adecuada (de tipo `int4` y una clave primaria o con una restricción única, preferiblemente indexada).

### 4.2.4. Importar datos a PostgreSQL

#### **shp2pgsql**

Se pueden importar datos a PostgreSQL usando distintos métodos. PostGIS incluye una utilidad llamada *shp2pgsql* que se puede usar para importar archivos shape a bases de datos activadas Post-

GIS. Por ejemplo, para importar un archivo shape llamado lagos a una base de datos PostgreSQL llamada `datos_sig`, use el siguiente comando:

```
shp2pgsql -s 2964 lagos.shp lagos_nueva | psql datos_sig
```

Esto crea una nueva capa llamada `lagos_nueva` en la base de datos `datos_sig`. La nueva capa tendrá el mismo identificador de referencia espacial (SRID) de 2964. Vea la Sección 7 para más información sobre los sistemas de referencia espacial y las proyecciones.

---

### Tip 7 EXPORTAR CONJUNTOS DE DATOS DE POSTGIS

---

Al igual que la herramienta de importación *shp2pgsql* también hay una herramienta para exportar conjuntos de datos PostGIS a archivos shape: *pgsql2shp*. Ésta está incluida en su distribución PostGIS.

---

### Complemento SPIT



QGIS viene con un complemento llamado SPIT (Shapefile to PostGIS Import Tool->Herramienta de Importación de Archivos Shape a PostGIS). SPIT se puede usar para cargar múltiples archivos shape al mismo tiempo e incluye soporte para esquemas. Para usar SPIT, abra el Administrador de complementos desde el menú *Complementos*, marque la casilla junto al complemento SPIT y pulse Aceptar. El icono de SPIT se añadirá a la barra de herramientas de complementos.

Para importar un archivo shape, haga clic en la herramienta SPIT en la barra de herramientas para abrir el diálogo. Puede añadir uno o más archivos a la cola haciendo clic en el botón *Añadir*. Para procesar los archivos, haga clic en el botón *Importar*. El progreso de la importación y cualquier error/aviso se mostrará a medida que se procesa cada archivo shape.

---

### Tip 8 IMPORTAR ARCHIVOS SHAPE QUE CONTIENEN PALABRAS RESERVADAS DE POSTGRESQL

---

Si se añade un archivo shape a la cola que contiene campos que son palabras reservadas en una base de datos PostgreSQL aparecerá un diálogo mostrando el estado de cada campo. Puede editar los nombres de los campos antes de importar y cambiar aquellos que sean palabras reservadas (o cambiar cualquier otro nombre de campo si lo desea). Intentar importar un archivo shape con palabras reservadas como nombres de campos probablemente fallará.

---

### ogr2ogr

Además de *shp2pgsql* y *SPIT* hay otra herramienta para suministrar datos en PostGIS: *ogr2ogr*. Esto es parte de su instalación de GDAL. Para importar un archivo shape a PostGIS, haga lo siguiente:

```
ogr2ogr -f "PostgreSQL" PG:"dbname=postgis host=myhost.de user=postgres \
password=topsecret" alaska.shp
```

Esto importará el archivo shape `alaska.shp` a la base de datos PostGIS `postgis` usando el usuario `postgres` con la contraseña `topsecret` en el servidor `myhost.de`.



Tenga en cuenta que OGR debe estar compilado contra PostgreSQL para soportar PostGIS. Puede ver esto tecleando

```
ogrinfo --formats | grep -i post
```

#### 4.2.5. Mejorar el rendimiento

Recuperar objetos espaciales de una base de datos PostgreSQL puede llevar tiempo, especialmente a través de una red. Puede mejorar el rendimiento al dibujar capas de PostgreSQL asegurándose de que existe un índice espacial para cada capa de la base de datos. PostGIS soporta la creación de un índice GiST (Generalized Search Tree->Árbol de búsqueda generalizado) para acelerar las búsquedas espaciales de los datos.

La sintaxis para crear un índice GiST<sup>2</sup> es:

```
CREATE INDEX [nombreindice] ON [nombretabla]
    USING GIST ( [campogeometria] GIST_GEOMETRY_OPS );
```

Tenga en cuenta que para grandes tablas, la creación del índice puede llevar bastante tiempo. Una vez que se ha creado el índice, se debería realizar un *VACUUM ANALYZE*. Vea la documentación de PostGIS ? para más información.

El siguiente es un ejemplo de la creación de un índice GiST:

```
gsherman@madison:~/current$ psql gis_data
Welcome to psql 8.0.0, the PostgreSQL interactive terminal.
```

```
Type: \copyright for distribution terms
      \h for help with SQL commands
      \? for help with psql commands
      \g or terminate with semicolon to execute query
      \q to quit
```

```
gis_data=# CREATE INDEX sidx_lagos_alaska ON lagos_alaska
gis_data=# USING GIST (la_geometria GIST_GEOMETRY_OPS);
CREATE INDEX
gis_data=# VACUUM ANALYZE lagos_alaska;
VACUUM
gis_data=# \q
gsherman@madison:~/current$
```

<sup>2</sup>La información de los índices GiST se ha tomado de la documentación de PostGIS disponible en <http://postgis.refractory.net>

### 4.3. El diálogo Propiedades de la capa

El diálogo de propiedades de las capas vectoriales proporciona información sobre una capa, configuración de la simbología y opciones de etiquetado. Si su capa vectorial se ha cargado desde un almacén PostgreSQL / PostGIS, también puede modificar la consulta SQL subyacente para la capa - bien a mano, editando la SQL en la pestaña *General* o invocando el diálogo *Constructor de consultas* en la pestaña *General*. Para acceder al diálogo *Propiedades de la capa*, haga doble clic en una capa en la leyenda o clic derecho en la capa y seleccione *Propiedades* del menú emergente.

#### 4.3.1. Pestaña Simbología

QGIS soporta varios trazadores de simbología para controlar cómo se muestran los objetos espaciales vectoriales. Actualmente están disponibles los siguientes trazadores:

**Símbolo único** - se aplica un solo estilo a cada objeto de la capa.

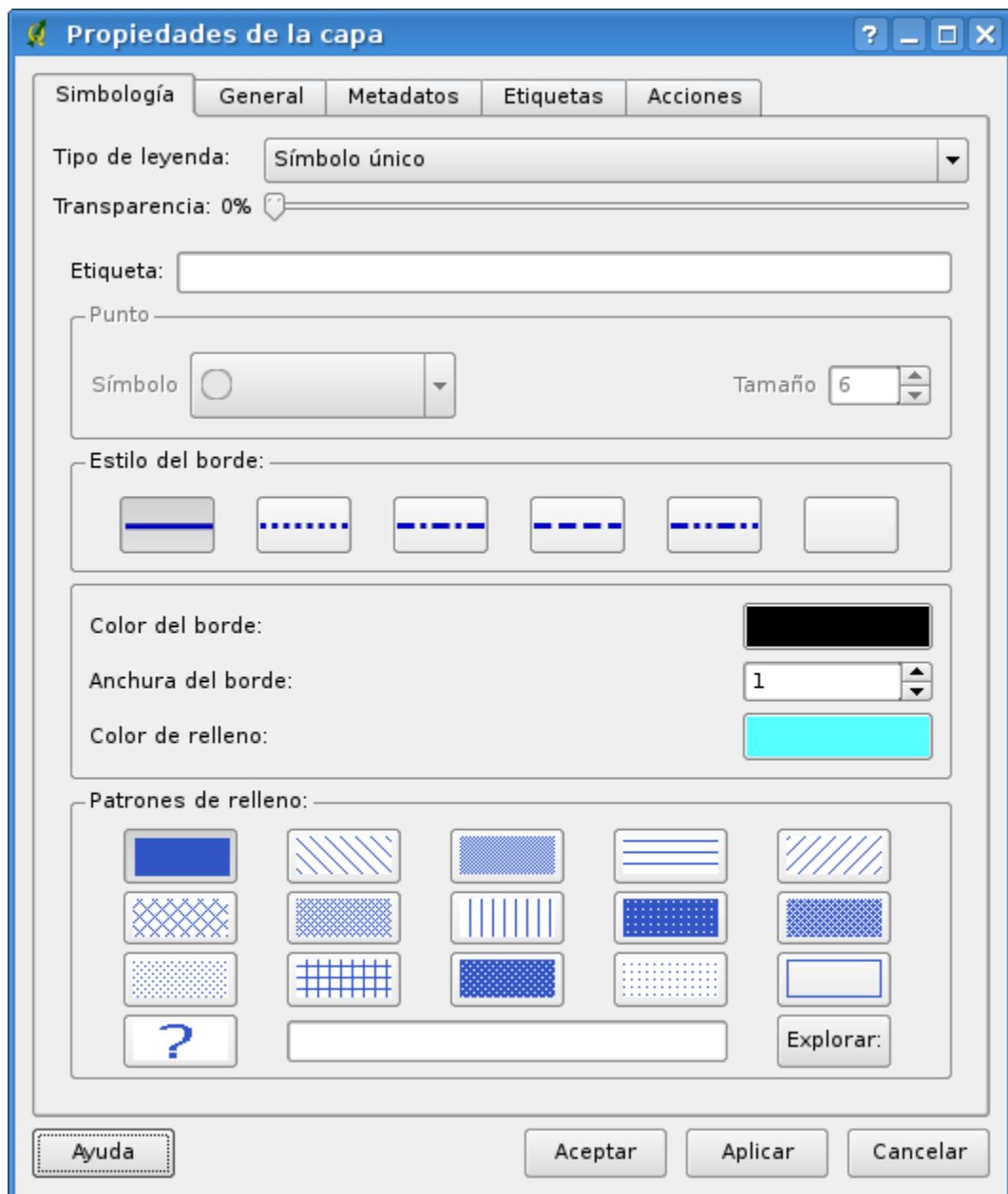
**Símbolo graduado** - los objetos de la capa se muestran con diferentes símbolos clasificados por los valores de un campo particular.

**Color continuo** - los objetos de la capa se muestran con un abanico de colores clasificados por el valor numérico de un campo especificado.

**Valor único** - los objetos se clasifican por el valor único contenido en un campo especificado, teniendo cada valor un símbolo diferente.

Para cambiar la simbología de una capa, simplemente haga doble clic en su entrada en la leyenda y se mostrará el diálogo *Propiedades de la capa*.

Figura 5: Diálogo Propiedades de la capa



**Nuevo en v0.9** hay una función para usar archivos de imagen guardados en su equipo como patrón de relleno de capas vectoriales.

### Transparencia de vectoriales

QGIS 0.9.1 permite establecer una transparencia para cada capa vectorial. Esto se puede hacer con del deslizador que hay justo debajo del tipo de leyenda (ver fig. 5). Esto es muy útil para superponer varias capas vectoriales.

#### 4.3.2. Pestaña General

La pestaña General es muy similar a la del diálogo de capas ráster. Permite cambiar el nombre que se muestra, establecer opciones de visualización dependientes de la escala, crear un índice espacial del archivo vectorial (sólo para formatos soportados por OGR y PostGIS) y ver o cambiar la proyección.

El botón *Constructor de consultas* permite crear un subconjunto de los objetos espaciales de la capa - pero este botón actualmente sólo esta disponible cuando abre la tabla de atributos y selecciona el botón *Avanzado...*

#### 4.3.3. Pestaña Metadatos

La pestaña metadatos contiene información sobre la capa, incluyendo especificaciones sobre el tipo y la localización, número de objetos espaciales, tipo de objetos espaciales y las posibilidades de edición. Los campos proyección y atributos y sus tipos de datos se muestran en esta pestaña. Esta es una manera rápida de obtener información sobre la capa.

#### 4.3.4. Pestaña Etiquetas

La pestaña etiquetas le permite activar el etiquetado de los objetos espaciales y controlar varias opciones relacionadas con el emplazamiento, estilo y márgenes.

Ilustraremos esto etiquetando el archivo shape lagos de los datos de ejemplo de qgis:

1. Cargue los archivos shape alaska.shp y lagos.shp en QGIS.
2. Acerque el zum un poco a su zona favorita con algunos lagos.
3. Haga la capa *lagos* activa.
4. Abra el diálogo de propiedades.
5. Pulse en la pestaña "Etiquetas".
6. Marque la casilla "Mostrar etiquetas" para activar el etiquetado.
7. Seleccione el campo con el que etiquetar. Usaremos *NAMES*.

8. Introduzca una etiqueta predeterminada para los lagos que no tienen nombre. La etiqueta predeterminada se usará cada vez que QGIS encuentre un lago sin valor en el campo *NAMES*.
9. Pulse *Aplicar*.

Ahora tenemos etiquetas. ¿Qué aspecto tienen? Probablemente son demasiado grandes o están mal situadas en relación con el símbolo marcador para los lagos.

Haga clic en la pestaña “Estilo de fuente” y use los botones *Fuente* y *Color* para establecer la fuente y el color.

Para cambiar la posición de la fuente en relación con el objeto espacial:

1. Haga clic en la pestaña “Alineación de fuente”.
2. Cambie la ubicación seleccionando uno de los botones circulares en el grupo “Ubicación”. Para colocar nuestras etiquetas, seleccione el botón circular “Derecha”.
3. Pulse *Aplicar* para ver sus cambios sin cerrar el diálogo.

Va mejorando, pero las etiquetas todavía están demasiado cerca del marcador. Para solucionarlo, podemos usar las opciones de la pestaña “Posición”. Aquí podemos añadir desplazamientos en las direcciones X e Y. Añadir un desplazamiento 5 en la X separará nuestras etiquetas del marcador y las hará más legibles. Por supuesto, si el símbolo o la fuente de su marcador son mayores, se requerirá un desplazamiento mayor.

El último ajuste que haremos será poner un margen a las etiquetas. Esto simplemente significa poner un fondo alrededor de las etiquetas para que destaquen más. Para poner un margen a las etiquetas de los lagos:

1. Pulse la pestaña “Margen”.
2. Marque la casilla “¿Poner margen a las etiquetas?” para activar los márgenes.
3. Seleccione un tamaño para el margen usando las flechas del cuadro tamaño.
4. Seleccione un color pulsando el botón *Color* y eligiendo su color favorito del selector de colores.
5. Haga clic en *Aplicar* para ver si le gustan los cambios.

Si no está conforme con los resultados, cambie los ajustes y pruebe otra vez pulsando *Aplicar*.

Un margen de 2 puntos parece dar buen resultado. Tenga en cuenta que también puede especificar el tamaño del margen en unidades del mapa si eso le conviene más.

El resto de las pestañas en la pestaña “Etiquetas” la permiten controlar el aspecto de las etiquetas usando atributos guardados en la capa. Las pestañas “Datos” le permiten establecer todos los parámetros de las etiquetas usando campos de la capa.

#### 4.3.5. Pestaña Acciones

QGIS proporciona la capacidad de realizar una acción en base a los atributos de un objeto espacial. Esto se puede usar para realizar cualquier número de acciones, por ejemplo, ejecutar un programa con argumentos elaborados a partir de los atributos de un objeto espacial o pasando parámetros a una herramienta de informes vía web.

Las acciones son útiles cuando con frecuencia se quiere ejecutar una aplicación externa o ver una página web basada en uno o más valores de su capa vectorial. Un ejemplo es realizar una búsqueda basada en el valor de un atributo. Este concepto se usa en la siguiente explicación.

##### Definir acciones

Las acciones de atributos se definen desde el diálogo de propiedades de capas vectoriales. Para definir una acción, abra el diálogo de propiedades de capas vectoriales y pulse la pestaña *Acciones*. Proporcione un nombre descriptivo para la acción. La propia acción debe contener el nombre de la aplicación que se ejecutará cuando se la invoque. Puede añadir uno o más valores de campos de atributos como argumentos para la aplicación. Cuando se invoca la acción cualquier conjunto de caracteres que empiece con un % seguido por el nombre de un campo se reemplazará por el valor de ese campo. Los caracteres especiales % % se sustituirán por el valor del campo que se seleccionó en los resultados de la identificación o en la tabla de atributos (vea Usar acciones más abajo). Las comillas dobles se pueden usar para agrupar texto en un solo argumento para el programa, script o comando. Las comillas dobles se ignorarán si van precedidas de una barra invertida.

A continuación se muestran dos ejemplos:

- `konqueror http://www.google.com/search?q=%nam`
- `konqueror http://www.google.com/search?q=% %`

En el primer ejemplo, se invoca al navegador web `konqueror` y se le pasa una URL para que la abra. La URL realiza una búsqueda en Google con el valor del campo *nam* de nuestra capa vectorial. Tenga en cuenta que la aplicación o script llamado por la acción debe estar en su variable de entorno `PATH` o si no debe proporcionar la ruta completa. Para asegurarse, podríamos rescribir el primer ejemplo como: `/opt/kde3/bin/konqueror http://www.google.com/search?q=%nam`. Esto asegurará que la aplicación `konqueror` se ejecute cuando se invoque la acción.

El segundo ejemplo usa la notación % % que no recae en un campo particular para su valor. Cuando se invoca la acción, % % será reemplazado por el valor del campo seleccionado en los resultados de la identificación o en la tabla de atributos.

##### Usar acciones

Se pueden invocar las acciones tanto desde el diálogo *Resultados de la identificación* como desde el diálogo *Tabla de atributos*. Para invocar una acción, haga clic derecho en el registro y seleccione

la acción del menú emergente. Las acciones se listan en el menú emergente por el nombre que les asignó al definir las acciones. Pulse en la acción que quiera invocar.

Si está invocando una acción que usa la notación `%%`, haga clic derecho en el valor del campo que quiera pasar a la aplicación o script en el diálogo *Resultados de la identificación* o en el diálogo *Tabla de atributos*.

Aquí hay otro ejemplo que extrae datos de una capa vectorial y los inserta en un archivo usando `bash` y el comando `'echo'` (así que sólo funcionará en GNU/Linux y quizás en Mac OS X). La capa en cuestión tiene campos para el nombre de una especie (`taxon_name`), la latitud (`lat`) y la longitud (`long`). Me gustaría poder hacer una selección espacial de unas localizaciones y exportar los valores de esos campos de los registros seleccionados (mostrados en amarillo en el área del mapa de QGIS) a un archivo de texto. Aquí está la acción para llevar esto a cabo:

```
bash -c "echo \"%taxon_name %lat %long\" >> /tmp/species_localities.txt"
```

Después de seleccionar unas cuantas localizaciones y ejecutar la acción en cada una de ellas, al abrir el archivo de salida se mostrará algo parecido a esto:

```
Acacia mearnsii -34.0800000000 150.0800000000
Acacia mearnsii -34.9000000000 150.1200000000
Acacia mearnsii -35.2200000000 149.9300000000
Acacia mearnsii -32.2700000000 150.4100000000
```

Como ejercicio podemos crear una acción que haga una búsqueda en Google sobre la capa *lagos*. Primero necesitamos determinar la URL que hace falta para realizar la búsqueda sobre una palabra clave. Esto se hace fácilmente simplemente yendo a Google, haciendo una búsqueda sencilla y tomando la URL de la barra de direcciones del navegador. Con este pequeño esfuerzo vemos que el formato es: `http://google.com/search?q=qgis`, donde *qgis* es el término de búsqueda. Con esta información podemos continuar:

- Asegúrese de que la capa *lagos* está cargada.
- Abra el diálogo de propiedades haciendo doble clic en la capa en la leyenda o clic derecho y seleccionando *Propiedades* del menú emergente.
- Haga clic en la pestaña “Acciones”.
- Introduzca un nombre para la acción, por ejemplo “Búsqueda Google”.
- Para la acción, necesitamos proporcionar el nombre de un programa externo a ejecutar. En este caso, podemos usar Firefox. Si el programa no está en su variable de entorno `PATH`, tiene que proporcionar la ruta completa.
- A continuación del nombre de la aplicación externa, añada la URL usada para hacer la búsqueda en Google, sin incluir el término de búsqueda: `http://google.com/search?q=`

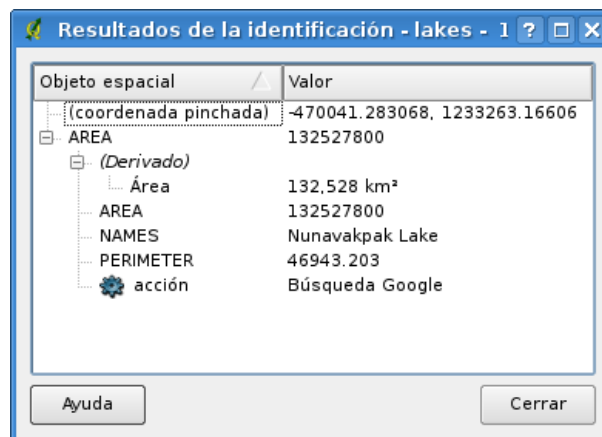
- El texto en el campo “Acción” debería ser como este:  
firefox http://google.com/search?q=
- Haga clic en el cuadro desplegable que contiene los nombre de los campos de la capa *lagos*. Se encuentra a la derecha del botón *Insertar campo*.
- Del cuadro desplegable, seleccione *NAMES* y haga clic en *Insertar campo*
- El texto de su acción ahora tiene este aspecto:  
firefox http://google.com/search?q= %NAMES

Esto completa la acción que ya está lista para usar. El texto final de la acción debería ser como este:

firefox http://google.com/search?q= %NAMES

Ahora podemos usar la acción. Cierre el diálogo de propiedades y acerque el zum a un área de su interés. Asegúrese de que la capa *lagos* está activa e identifique un lago. En el cuadro de resultados verá ahora visible nuestra acción:

**Figura 6:** Seleccionar objeto espacial y elegir acción



Cuando pulsamos sobre la acción, ésta abre Firefox y navega hasta la URL <http://www.google.com/search?q=Tustumena>. También es posible añadir más campos de atributos a la acción. Para ello puede añadir un “+” al final del texto de la acción, seleccionar otro campo y pulsar en *Insertar campo*. En este ejemplo no hay ningún otro campo disponible en el que tuviera sentido buscar.

Puede definir múltiples acciones para una capa y cada una aparecerá en el diálogo Resultados de la identificación. También puede invocar las acciones desde la tabla de atributos, seleccionando una fila, haciendo clic derecho y luego eligiendo la acción del menú emergente.

Se le pueden ocurrir toda clase de usos para las acciones. Por ejemplo, si tiene una capa de puntos que contiene localizaciones de imágenes o fotos junto con un nombre de archivo, podría crear una



acción para lanzar un visor para mostrar la imagen. También podría usar las acciones para lanzar informes basados en web para un campo de atributos o combinación de campos, especificándolos de la misma forma que lo hicimos para nuestro ejemplo de búsqueda en Google.

## 4.4. Edición

QGIS soporta capacidades básicas de edición de datos espaciales. Antes de continuar leyendo, debería tener en cuenta que en este nivel de desarrollo el soporte de edición todavía es preliminar. Antes de hacer cualquier edición, haga siempre una copia de respaldo de los datos que vaya a editar.

**Nota** – el procedimiento para editar capas de GRASS es diferente – vea la Sección 8.5 para más detalles.

### 4.4.1. Establecer la tolerancia de autoensamblado

Antes de poder editar vértices, necesitamos establecer la tolerancia de autoensamblado. Ésta es la distancia que QGIS utilizar para “buscar” el polígono y vértice que está intentando editar cuando pulsa en el mapa. Si no se encuentra dentro de la tolerancia de autoensamblado, QGIS no encontrará ni seleccionará el vértice para editarlo. La tolerancia se establece en las unidades del mapa, por lo que puede necesitar experimentar hasta conseguir establecerla correctamente. Si especifica una tolerancia excesiva, QGIS se puede autoensamblar al vértice incorrecto, especialmente si está trabajando con un gran número de vértices que se encuentran próximos. Si establece una tolerancia demasiado pequeña no encontrará nada y aparecerá un molesto aviso a tal efecto.

Para establecer la tolerancia de autoensamblado, seleccione *Propiedades del proyecto* del menú *Configuración* y pulse en la pestaña “General”. Recuerde que la tolerancia está en unidades de mapa. Para nuestro pequeño proyecto de digitalización, las unidades están en grados decimales. Sus resultados pueden variar, pero algo del orden de 0.05 a 0.1 debería ir bien.

### 4.4.2. Editar una capa existente

De forma predeterminada, QGIS carga las capas en modo de sólo lectura: esto es una medida de seguridad para evitar que se edite una capa de forma accidental si hay un descuido con el ratón. Sin embargo, puede seleccionar para editar cualquier capa con tal de que el proveedor de los datos tenga soporte para ella y la fuente de datos subyacente se pueda escribir (esto es, sus archivos no sean de sólo lectura).

La edición de capas es más versátil cuando se usa sobre fuentes de datos PostgreSQL/PostGIS.

Todas las sesiones de edición comienzan al seleccionar la opción *Conmutar edición*. Esta opción se puede encontrar en el menú emergente después de hacer clic derecho en la entrada de la leyenda.


---

**Tip 9 INTEGRIDAD DE LOS DATOS**

---

Por favor, considere hacer una copia de respaldo de sus datos originales antes de empezar a editar y también a intervalos regulares durante la edición. QGIS está todavía en estado de pre-versión 1.0 y por lo tanto puede que no sea capaz de proteger sus datos en todas las situaciones.

---

da para esa capa. De forma alternativa, puede usar el botón  *Conmutar edición* de la barra de herramientas para iniciar o parar el modo de edición.

---

**Tip 10 EDITAR UN MAPA ES DISTINTO DE EDITAR UNA TABLA DE ATRIBUTOS**

---

En esta versión de QGIS, el binomio *Iniciar edición/Detener edición* sobre la vista del mapa funciona por separado del binomio *Iniciar edición/Detener edición* en una tabla de atributos.

---



---

**Tip 11 GUARDAR REGULARMENTE**

---

Recuerde desactivar *Conmutar edición* o seleccionar el botón *Detener edición* regularmente. Esto permite guardar sus cambios hechos hasta ese momento y también confirma que su fuente de datos puede admitir todos sus cambios.

---

Una vez que la capa está en modo de edición, aparecerán marcadores en los vértices.

Puede realizar las siguientes funciones de edición:

- Añadir objetos espaciales (puntos, líneas y polígonos).
- Mover los objetos espaciales seleccionados.
- Dividir los objetos espaciales seleccionados.
- Borrar los objetos espaciales seleccionados.
- Añadir vértices a un objeto espacial.
- Borrar vértices de un objeto espacial.
- Mover vértices de un objeto espacial.
- Añadir anillos.
- Añadir islas.
- Cortar los objetos espaciales seleccionados.
- Copiar los objetos espaciales seleccionados.
- Pegar los objetos espaciales seleccionados.

**Añadir objetos espaciales**

Antes de empezar a añadir objetos espaciales, use las herramientas de desplazamiento y zum para encuadrar el área de interés.

Ahora puede usar los iconos *Capturar punto*, *Capturar línea* o *Capturar polígono* de la barra de herramientas para poner el cursor de QGIS en modo de digitalización.

**Tip 12 EDICIONES CONCURRENTES**

Esta versión de QGIS no controla si alguien más está editando un objeto espacial al mismo tiempo que usted. El último editor gana.

**Tip 13 ACERCAR ZUM ANTES DE EDITAR**

Antes de editar una capa, debería acercar el zum a su área de interés. Esto evita tener que esperar mientras todos los marcadores de vértices son representados a lo largo de toda la capa.

Para cada objeto espacial, primero se digitaliza la geometría y luego se introducen sus atributos.

Para digitalizar la geometría, pulse en el área del mapa para crear el primer punto de su nuevo objeto espacial.

Para líneas y polígonos, continúe pulsando con el botón izquierdo para cada punto adicional que quiera capturar. Cuando termine de añadir puntos, pulse con el botón derecho en cualquier lugar del área del mapa para confirmar que ha terminado de introducir la geometría de ese objeto espacial.

Aparecerá la ventana de atributos, permitiendo introducir la información del nuevo objeto espacial. La figura 7 muestra como se establecen los atributos para un nuevo río ficticio en Alaska.

**Editar vértices de un objeto espacial**

Se pueden editar los vértices de un objeto espacial, tanto para capas basadas en PostgreSQL/PostGIS como en archivos shape.

Los vértices se pueden editar directamente, esto es, no tiene que seleccionar el objeto espacial a editar antes de cambiar su geometría. En algunos casos, varios objetos espaciales pueden compartir el mismo vértice y entonces rigen las siguientes reglas cuando pulsa con el ratón cerca de los objetos espaciales del mapa:

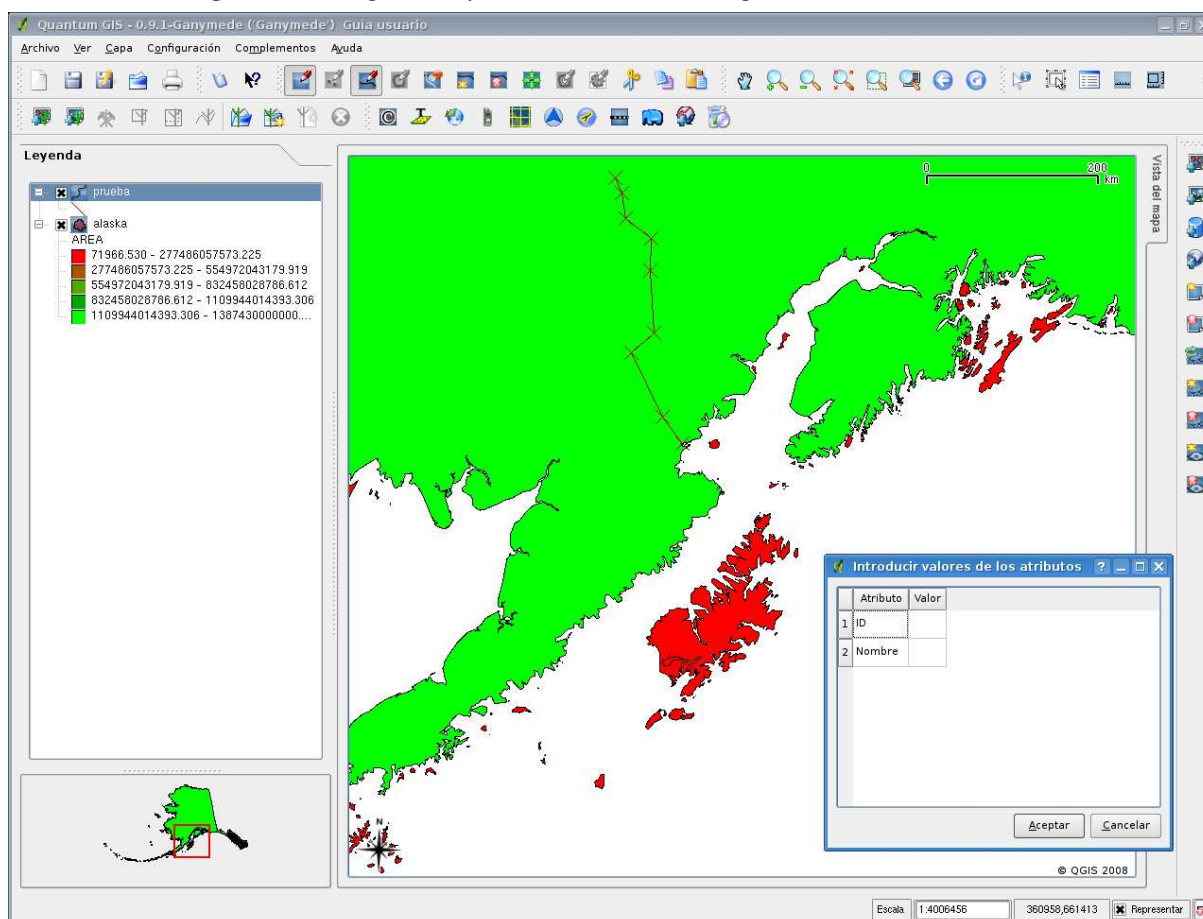
- **Líneas** - La línea más próxima a la posición del ratón se usa como el objeto espacial objetivo. Entonces (para mover y borrar un vértice) el vértice más próximo de esa línea será el objetivo de la edición.
- **Polígonos** - Si el ratón está dentro de un polígono, entonces este será el objeto espacial objetivo; de lo contrario, se usará el polígono más próximo. Entonces (para mover y borrar un vértice) el vértice más próximo de ese polígono será el objetivo de la edición.

Necesitará establecer la propiedad *Configuración->Propiedades del proyecto->General->Tolerancia de autoensamblado* a un número mayor que cero. De lo contrario QGIS no será capaz de decir qué objeto espacial se está editando.

**Tip 14 MARCADORES DE VÉRTICES**

Esta versión de QGIS no le permite cambiar los marcadores de vértices utilizados.

**Figura 7:** Diálogo de captura de atributos de digitalización de vectoriales

**Añadir vértices a un objeto espacial**

Puede añadir nuevos vértices a un objeto espacial usando el icono *Añadir vértice* de la barra de herramientas.

Nota: no tiene sentido añadir más vértices a objetos espaciales de tipo punto.

En esta versión de QGIS, los vértices sólo se pueden añadir a un segmento de línea *existente* de un objeto espacial de tipo línea. Si quiere extender una línea más allá de su punto final, tendrá que mover primero el vértice final y luego añadir un nuevo vértice donde anteriormente se encontraba el final.

---

**Tip 15 TIPOS DE VALORES DE ATRIBUTOS**

---

En la versión actual de QGIS, la casilla del diálogo de atributos no comprueba si los datos introducidos concuerdan con el tipo esperado (ej.: numérico vs. texto). Asegúrese de esto antes de pulsar *Aceptar*, de lo contrario puede encontrarse con que el error es detectado más tarde cuando intente guardar los cambios.

---

**Mover vértices de un objeto espacial**

Puede mover los vértices usando el icono *Mover vértice* de la barra de herramientas.

**Borrar vértices de un objeto espacial**

Puede borrar vértices usando el icono *Borrar vértice* de la barra de herramientas.

Nota: no tiene sentido borrar el vértice de un objeto espacial de tipo punto. Borre la totalidad del objeto espacial en su lugar.

De forma similar, una línea con un solo vértice o un polígono con dos vértices son bastante inútiles y darán lugar a resultados inesperados en cualquier operación de QGIS, así que no los haga.

**Aviso:** un vértice se identifica para borrar en el momento que pulsa con el ratón cerca de un objeto espacial susceptible de ser borrado. Para deshacer, necesitará desactivar el modo de edición y descartar los cambios (por supuesto, esto implicará que cualquier otro cambio no guardado se pierda).

**Añadir anillos****Nuevo en v0.9**

Puede crear polígonos anulares en QGIS. Esto significa que dentro de un área existente es posible digitalizar más polígonos, lo que ocurrirá como un «todo», de forma que el área entre los límites de los polígonos exterior e interior quedará como un polígono anular.

**Añadir islas****Nuevo en v0.9**

Puede añadir polígonos isla a un polígono múltiple seleccionado. El nuevo polígono isla tiene que digitalizarse fuera del polígono múltiple seleccionado.

**Cortar, copiar y pegar objetos espaciales**

Se pueden cortar, copiar y pegar los objetos espaciales seleccionados entre capas dentro del mismo proyecto de QGIS, con tal de que las capas de destino se hayan puesto en modo edición con anterioridad.

Los objetos espaciales también se pueden pegar en aplicaciones externas como texto: esto es,

los objetos espaciales se representan en formato CSV, apareciendo los datos de la geometría en formato OGC Well-Known Text (WKT).

Sin embargo, en esta versión de QGIS, los objetos espaciales de texto procedentes de fuera de QGIS no se pueden pegar en una capa dentro de QGIS.

¿Cuándo estarán disponibles las funciones de copiar y pegar? Bien, teniendo en cuenta que se puede editar más de una capa a la vez y copiar y pegar objetos espaciales entre ellas, ¿para qué queríamos hacer eso? Digamos que necesitamos trabajar en una nueva capa pero sólo necesitamos uno o dos lagos, no los 5.000 de nuestra capa *big\_lakes*. Podemos crear una nueva capa y usar copiar/pegar para añadirle los lagos que necesitamos.

Como ejemplo vamos a copiar algunos lagos a una capa nueva:

1. Cargar la capa de la que se quiere copiar (capa de origen).
2. Cargar o crear la capa en la que se quiere copiar (capa de destino).
3. Comenzar la edición de ambas capas.
4. Activar la capa origen pulsando en ella en la leyenda.
5. Usar la herramienta de selección para seleccionar los objetos espaciales de la capa de origen.
6. Pulsar en la herramienta *Copiar objetos espaciales*.
7. Activar la capa de destino pulsando en ella en la leyenda.
8. Pulsar en la herramienta *Pegar objetos espaciales*.
9. Detener la edición y guardar los cambios.

¿Qué pasa si las capas de origen y destino tienen distintos esquemas (los nombres y tipos de campos no son iguales)? QGIS rellena lo que sea igual e ignora el resto. Si no le interesa que se copien los atributos en la capa de destino, no importa cómo diseñe los campos y el tipo de datos. Si quiere asegurarse de que todo, objetos espaciales y atributos, se copie, asegúrese de que los esquemas concuerdan.

---

**Tip 16 CONGRUENCIA DE LOS OBJETOS ESPACIALES PEGADOS**

---

Si sus capas de origen y destino usan la misma proyección, entonces los objetos espaciales tendrán pegados una geometría idéntica a la de la capa de origen. Sin embargo, si la capa de destino tiene una proyección diferente, QGIS no puede garantizar que la geometría sea idéntica. Esto es sencillamente porque hay pequeños errores de redondeo cuando se convierte entre proyecciones.

---

**Borrar objetos espaciales seleccionados**

Si quiere borrar un polígono entero, puede hacerlo seleccionando primero el polígono usando la herramienta habitual *Seleccionar objetos espaciales*. Puede seleccionar múltiples objetos espaciales para borrarlos. Una vez que tenga la selección, use la herramienta *Borrar lo seleccionado* para borrar los objetos espaciales. No existe una función deshacer, pero recuerde que su capa no cambia

realmente hasta que detiene la edición y selecciona guardar los cambios. Por lo tanto, si comete un error, siempre puede cancelar el guardado.

La herramienta *Cortar objetos espaciales* de la barra de herramientas de digitalización también se puede usar para borrar objetos espaciales. Esto efectivamente borra el objeto espacial, pero lo coloca en un "portapapeles espacial". Podemos entonces usar la herramienta pegar para restaurarlo, dándonos la capacidad de deshacer una acción. Cortar, copiar y pegar funcionan en los objetos espaciales actualmente seleccionados, lo que significa que podemos operar en más de uno a la vez.

---

**Tip 17** SOPORTE PARA EL BORRADO DE OBJETOS ESPACIALES

---

Cuando se editan archivos shape de ESRI, el borrado de objetos espaciales sólo funciona si QGIS está enlazado a una versión de GDAL 1.3.2 o superior. Las versiones de QGIS para OS X y Window disponibles desde la página de descarga están compiladas usando GDAL 1.3.2 o superior.

---

### Modo de autoensamblado

QGIS permite que los vértices digitalizados se ensamblen automáticamente a otros vértices de la misma capa. Para establecer la tolerancia de autoensamblado, vaya a *Configuración->Propiedades del proyecto->General->Tolerancia de autoensamblado*. Tenga en cuenta que la tolerancia de autoensamblado está en unidades de mapa.

### Guardar las capas editadas

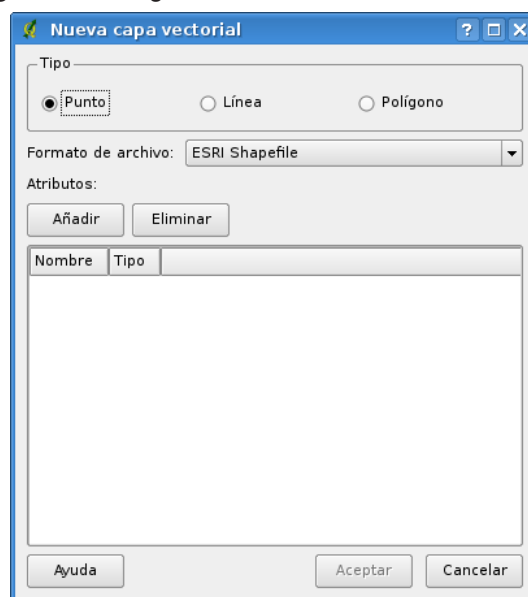
Cuando una capa está en modo edición, cualquier cambio permanece en la memoria de QGIS. Por lo tanto, no se transfieren/guardan de forma inmediata a la fuente de datos o al disco. Cuando desactiva el modo edición (o sale de QGIS para ello), se le pregunta si quiere guardar los cambios o descartarlos.

Si los cambios no se pueden guardar (por ejemplo, porque el disco este lleno o los atributos tienen valores que están fuera de intervalo), se conserva el estado de QGIS en memoria. Esto le permite ajustar sus ediciones y probar de nuevo.

#### 4.4.3. Crear una nueva capa

Para crear una capa nueva para editarla, seleccione *Nueva capa vectorial...* del menú *Capa*. El diálogo *Nueva capa vectorial* se mostrará como aparece en la Figura 8. Seleccione el tipo de capa (punto, línea o polígono).

Nota: QGIS aún no tiene soporte para la creación de objetos espaciales 2.5D (esto es, objetos espaciales con coordenadas X, Y y Z). En este momento sólo se pueden crear archivos shape. En una versión futura de QGIS, estará soportada la creación de cualquier tipo de capa OGR o PostgreSQL.

**Figura 8:** Diálogo de creación de un nuevo vectorial

La creación de capas de GRASS es posible desde el complemento de GRASS. Por favor vaya a la sección 8.8 para más información sobre la creación de capas vectoriales de GRASS.

Para completar la creación de una nueva capa, añada los atributos que desee pulsando el botón *Añadir* y especificando el nombre y tipo de atributos. Solo están soportados los atributos de tipo real, entero y cadera. Una vez que esté satisfecho con los atributos, pulse *Aceptar* y dele un nombre al archivo shape. QGIS añadirá automáticamente la extensión *.shp* al nombre que haya especificado. Una vez que la capa esté creada, se añadirá al mapa y la podrá editar de la misma forma descrita en la sección 4.4.2 anterior.

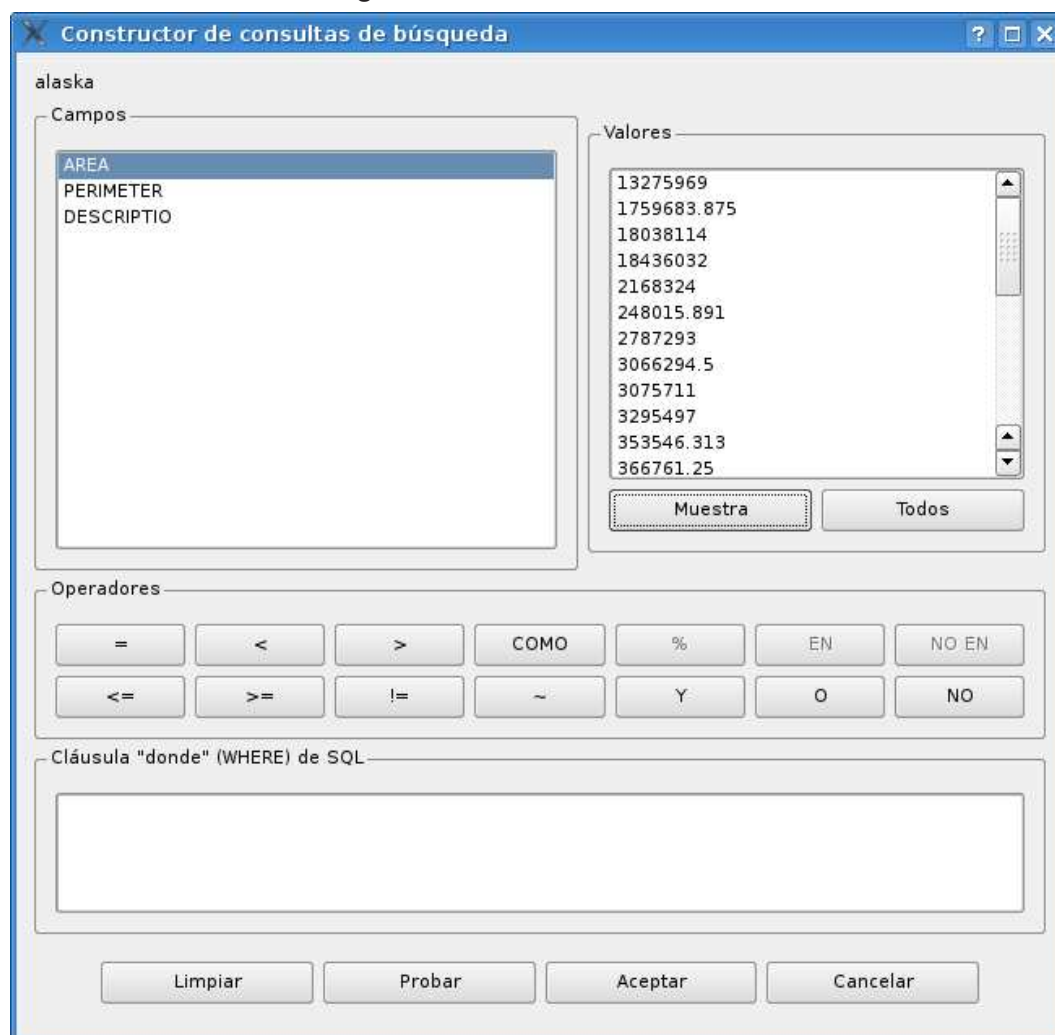
## 4.5. Constructor de consultas

El constructor de consultas le permite definir un subconjunto de una tabla y mostrarlo como una capa en QGIS. Se puede usar para todos los formatos soportados por OGR, archivos de GRASS y capas PostGIS. Por ejemplo, si tiene una capa de ciudades con un campo «población», podría seleccionar sólo las ciudades más grandes introduciendo *población >100000* en el cuadro de consulta SQL del constructor de consultas. La Figura 9 muestra un ejemplo del constructor de consultas relleno con datos de una capa PostGIS con atributos guardados en PostgreSQL.

El constructor de consultas muestra una lista de los campos que hay en la base de datos de la capa en el cuadro de lista de la izquierda. Puede obtener una muestra de los datos que contiene el campo resaltado pulsando en el botón *Muestra*. Esto recupera los primeros 25 valores distintos del campo de la base de datos. Para obtener una lista de todos los valores posibles de un campo, pulse en el



Figura 9: Constructor de consultas



botón *Todos*. Para añadir un campo o valor seleccionado a la consulta, haga doble clic en él. Puede usar los distintos botones para construir la consulta o puede simplemente escribirla en el cuadro SQL.

Para probar una consulta, pulse en el botón *Probar*. Esto devolverá la cuenta del número de registros que se incluirán en la capa. Cuando esté satisfecho con la consulta, pulse *Aceptar*. La sentencia SQL para la cláusula «donde» se mostrará en la columna SQL de la lista de capas.

#### **Tip 18** CAMBIAR LA DEFINICIÓN DE LA CAPA

Puede cambiar la definición de la capa después de que esté cargada alterando la consulta SQL usada para definir la capa. Para hacer esto, abra el diálogo de propiedades de la capa vectorial haciendo doble clic en la capa en la leyenda y pulse el botón *Constructor de consultas* en la pestaña *General*. Vea la Sección 4.3 para más información.

#### 4.5.1. Consultar capas PostGIS

Para consultar una capa PostGIS cargada hay dos opciones. La primera es pulsar en el botón *Abrir tabla* para abrir la tabla de atributos de la capa PostGIS. Luego pulse el botón *Avanzado...* en la parte inferior. Esto inicia el constructor de consultas que le permite definir un subconjunto de una tabla y mostrarlo como se describe en la Sección 4.5.

La segunda opción para consultar una capa PostGIS layer, es abrir el diálogo *Propiedades de la capa* haciendo doble clic en el nombre de la capa PostGIS en la leyenda o clic derecho y seleccionar *Propiedades* del menú emergente. En la pestaña *General* pulse el botón *Constructor de consultas* de la parte inferior.

#### 4.5.2. Consultar formatos OGR y archivos de GRASS

Para consultar un archivo de GRASS cargado o un formato soportado por OGR actualmente necesita pulsar el botón *Abrir tabla* para abrir la correspondiente tabla de atributos y pulsar el botón *Avanzado....* Esto inicia el constructor de consultas que le permite definir un subconjunto de una tabla y mostrarlo como se describe en la Sección 4.5.

La segunda opción para iniciar el constructor de consultas descrita en la sección 4.5.1 actualmente no está soportada para capas OGR y de GRASS.

---

## 5. Working with Raster Data

QGIS supports a number of raster data formats. This section describes how to work with raster data in QGIS.

### 5.1. What is raster data?

Raster data in GIS are matrices of discrete cells that represent features on, above or below the earth's surface. Each cell in the raster grid is the same size, and cells are usually rectangular (in QGIS they will always be rectangular). Typical raster datasets include remote sensing data such as aerial photography or satellite imagery and modelled data such as an elevation matrix.

Unlike vector data, raster data typically do not have an associated database record for each cell.

In GIS, a raster layer would have georeferencing data associated with it which will allow it to be positioned correctly in the map display to allow other vector and raster data to be overlaid with it. QGIS makes use of georeferenced rasters to properly display the data.

### 5.2. Raster formats supported in QGIS

QGIS supports a number of different raster formats. Currently tested formats include:

- Arc/Info Binary Grid
- Arc/Info ASCII Grid
- GRASS Raster
- GeoTIFF
- Spatial Data Transfer Standard Grids (with some limitations)
- USGS ASCII DEM
- Erdas Imagine

Because the raster implementation in QGIS is based on the GDAL library, other raster formats implemented in GDAL are also likely to work, but have not yet been tested. See Appendix A.2 for more details.

### 5.3. Loading raster data in QGIS



Raster layers are loaded either by clicking on the *Load Raster* icon or by selecting the *View-> Add Raster Layer* menu option. More than one layer can be loaded at the same time by holding down the

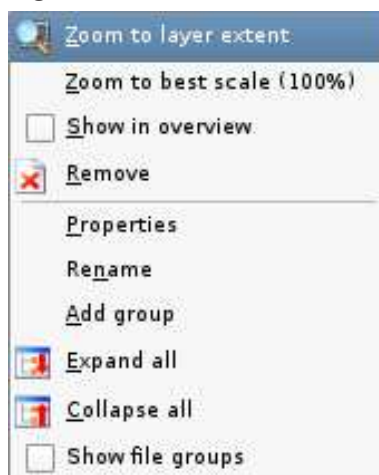
Control key and clicking on multiple items in the file dialog.

Please refer to section 8.2 if you intend to load GRASS rasterdata.

## 5.4. Raster Properties Dialog

To view and set the properties for a raster layer, right click on the layer name. This displays the raster layer context menu that includes a number of items that allow you to:

**Figura 10:** Raster context menu

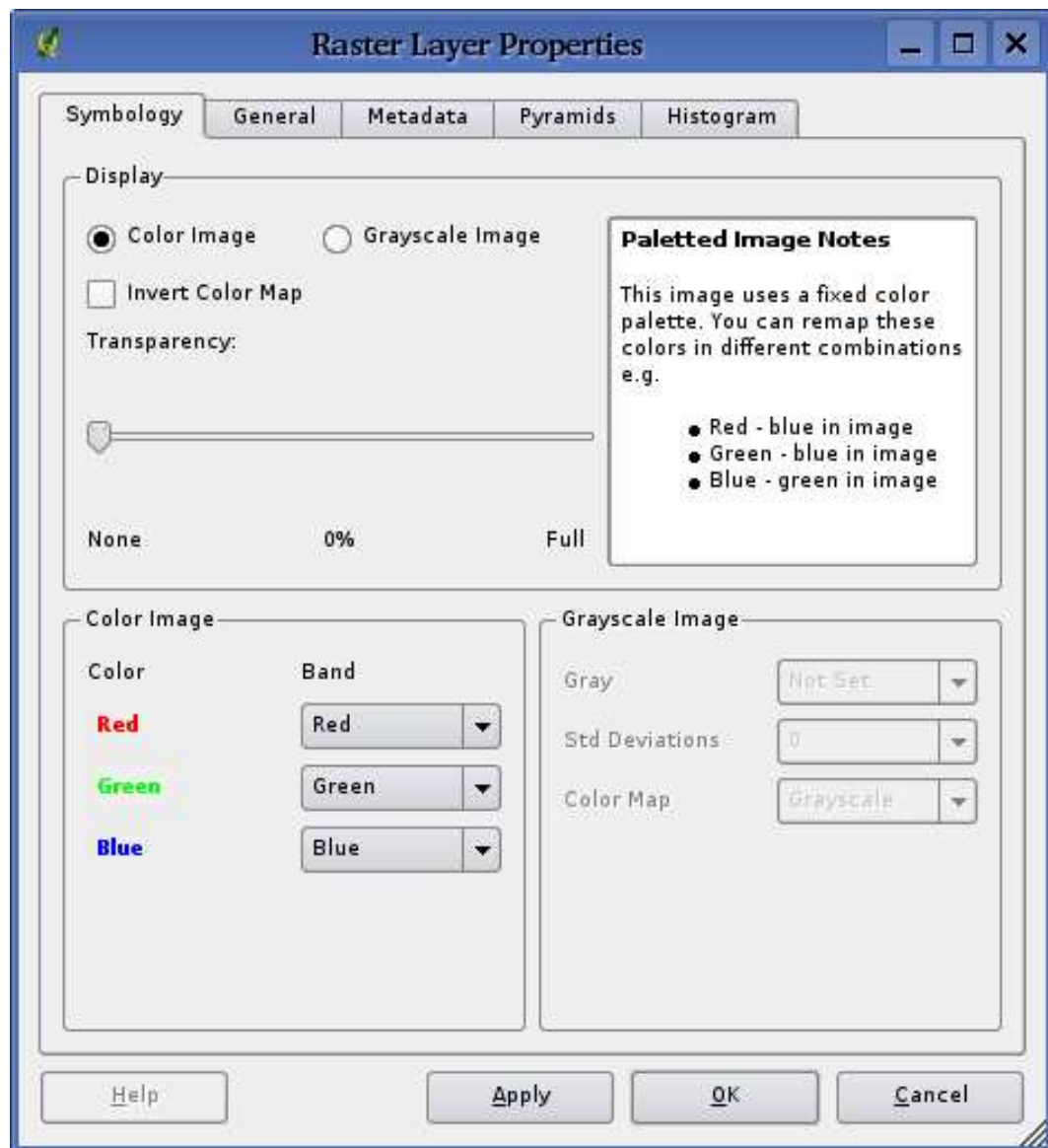


- Zoom to the full extent of the raster
- Zoom to the best scale of the raster
- Show the raster in the map overview window
- Remove the layer from the map
- Open the raster layers properties
- Rename the layer
- Add a layer group
- Expand legend tree view
- Collapse legend tree view
- Show file groups

Choose *Properties* from the context menu to open the raster properties dialog for the layer.

Figure 11 shows the properties dialog. There are five tabs on the dialog: *Symbology*, *General*, *Meta-data*, *Pyramids* and *Histogram*.

Figura 11: Raster Layers Properties Dialog



#### 5.4.1. Symbolygy Tab

QGIS supports three forms of raster layers:

- Single Band Grayscale Rasters
- Palette Based RGB Rasters
- Multiband RGB Rasters

From these three basic layer types, eight forms of symbolised raster display can be used:

- Single Band Grayscale
- Single Band Pseudocolor
- Paletted Grayscale (where only the red, green or blue component of the image is displayed)
- Paletted Pseudocolor (where only the red, green or blue component of the image is displayed, but using a pseudocolor algorithm)
- Paletted RGB
- Multiband Grayscale (using only one of the bands to display the image)
- Multiband Pseudocolor (using only one of the bands shown in pseudocolor)
- Multiband RGB (using any combination of three bands)

QGIS can invert the colors in a given layer so that light colors become dark (and dark colors become light). Use the *Invert Color Map* checkbox to enable / disable this behavior.

QGIS has the ability to display each raster layer at varying transparency levels. Use the transparency slider to indicate to what extent the underlying layers (if any) should be visible through the current raster layer.

QGIS can restrict the data displayed to only show cells whose values are within a given number of standard deviations of the mean for the layer. This is useful when you have one or two cells with abnormally high values in a raster grid that are having a negative impact on the rendering of the raster. This option is only available for pseudocolor images.

---

**Tip 19** VIEWING A SINGLE BAND OF A MULTIBAND RASTER

---

If you want to view a single band (for example Red) of a multiband image, you might think you would set the Green and Blue bands to “Not Set”. But this is not the correct way. To display the Red band, set the image type to grayscale, then select Red as the band to use for Gray.

---

#### 5.4.2. General Tab

The General tab displays basic information about the selected raster, including the layer source and display name in the legend (which can be modified). This tab also shows a thumbnail of the layer, its legend symbol, and the palette.

Additionally scale-dependent visibility can be set in this tab. You need to check the checkbox and set an appropriate scale where your data will be displayed in the map canvas.

Also the spatial reference system is printed here as a PROJ.4-string. This can be modified by hitting the *Change* button.

### 5.4.3. Metadata Tab

The Metadata tab displays a wealth of information about the raster layer, including statistics about each band in the current raster layer. Statistics are gathered on a 'need to know' basis, so it may well be that a given layer's statistics have not yet been collected.

---

**Tip 20** GATHERING RASTER STATISTICS

---

To gather statistics for a layer, select pseudocolor rendering and click the *Apply* button. Gathering statistics for a layer can be time consuming. Please be patient while QGIS examines your data!

---

### 5.4.4. Pyramids Tab

Large resolution raster layers can slow navigation in QGIS. By creating lower resolution copies of the data (pyramids), performance can be considerably improved as QGIS selects the most suitable resolution to use depending on the level of zoom.

You must have write access in the directory where the original data is stored to build pyramids. Several resampling methods can be used to calculate the pyramids:

- Average
- Nearest Neighbour
- Average Magphase

Please note that building pyramids may alter the original data file and once created they cannot be removed. If you wish to preserve a 'non-pyramided' version of your raster, make a backup copy prior to building pyramids.

### 5.4.5. Histogram Tab

The histogram tab allows you to view the distribution of the bands or colors in your raster. You must first generate the raster statistics by clicking the *Refresh* button. You can choose which bands to display by selecting them in the list box at the bottom right of the dialog. Two different chart types are allowed: Bar charts and Line graphs.

Once you view the histogram, you'll notice that the band statistics have been populated on the metadata tab.

---

## 6. Working with OGC Data

QGIS supports WMS and WFS as data sources. The support for WFS is preliminary at this time. WMS support is native; WFS is implemented using a plugin.

### 6.1. What is OGC Data

The Open Geospatial Consortium (OGC), is an international organization with more than 300 commercial, governmental, nonprofit and research organisations worldwide. Its members develop and implement standards for geospatial content and services, GIS data processing and exchange.

Describing a basic data model for geographic features an increasing number of specifications are developed to serve specific needs for interoperable location and geospatial technology, including GIS. Further information can be found under <http://www.opengeospatial.org/>.

Important OGC specifications are:

- **WMS** - Web Map Service
- **WFS** - Web Feature Service
- **WCS** - Web Coverage Service
- **CAT** - Web Catalog Service
- **SFS** - Simple Features for SQL
- **GML** - Geography Markup Language

OGC services are increasingly being used to exchange geospatial data between different GIS implementations and data stores. QGIS can now deal with three of the above specifications, being SFS (though support of the PostgreSQL / PostGIS data provider, see Section 4.2); WFS and WMS as a client.

### 6.2. WMS Client

#### 6.2.1. Overview of WMS Support



QGIS currently can act as a WMS client that understands WMS 1.1, 1.1.1 and 1.3 servers. It has particularly been tested against publicly accessible servers such as DEMIS and JPL OnEarth.

WMS servers act upon requests by the client (e.g. QGIS) for a raster map with a given extent, set of layers, symbolisation style, and transparency. The WMS server then consults its local data sources,



rasterizes the map, and sends it back to the client in a raster format. For QGIS this would typically be JPEG or PNG.

WMS is generically a REST (Representational State Transfer) service rather than a fully-blown Web Service. As such, you can actually take the URLs generated by QGIS and use them in a web browser to retrieve the same images that QGIS uses internally. This can be useful when troubleshooting problems, as there are several brands of WMS servers in the market and they all have their own interpretation of the WMS standard.

WMS layers can be added quite simply, as long as you know the URL to access the WMS server, you have a serviceable connection to that server, and the server understands HTTP as the data transport mechanism.

### 6.2.2. Selecting WMS Servers

The first time you use the WMS feature, there are no servers defined. You can begin by clicking the *Add WMS layer* button inside the toolbar, or through the Layer menu.

The dialog for adding layers from the WMS server pops up. Fortunately you can add some servers to play with by clicking the *Add default servers* button. This will add at least three WMS servers for you to use, including the NASA (JPL) WMS server. To define a new WMS server in the *Server Connections* section, select *New*. Then enter in the parameters to connect to your desired WMS server, as listed in table 2:

**Cuadro 2:** WMS Connection Parameters

Name	A name for this connection. This name will be used in the Server Connections drop-down box so that you can distinguish it from other WMS Servers.
URL	URL of the server providing the data. This must be a resolvable host name; the same format as you would use to open a telnet connection or ping a host.
Proxy Host	Network address or host name of the proxy server you would use to access this WMS server, or leave blank if no proxy is needed.
Proxy Port	Port number of the proxy server.
Proxy User	User name used to login to the proxy server.
Proxy Password	Password used to login to the proxy server.

At least *Name* and *URL* are required entries; the proxy entries can be left blank if you have a clear path to your WMS server.

Once the new WMS Server has been created, it will be preserved across future QGIS sessions.

Table 3 shows some example WMS URLs to get you started. These links were last checked in December 2006, but could change at any time:

**Tip 21 ON WMS SERVER URLS**

Be sure, when entering in the WMS server URL, that you have the base URL. For example, you shouldn't have fragments such as `request=GetCapabilities` or `version=1.0.0` in your URL.

**Cuadro 3:** Example Public WMS URLs

Name	URL
Atlas of Canada	<a href="http://atlas.gc.ca/cgi-bin/atlaswms_en?">http://atlas.gc.ca/cgi-bin/atlaswms_en?</a>
DEMIS	<a href="http://www2.demis.nl/wms/wms.asp?wms=WorldMap&amp;">http://www2.demis.nl/wms/wms.asp?wms=WorldMap&amp;</a>
Geoscience Australia	<a href="http://www.ga.gov.au/bin/getmap.pl?dataset=national">http://www.ga.gov.au/bin/getmap.pl?dataset=national</a>
NASA JPL OnEarth	<a href="http://wms.jpl.nasa.gov/wms.cgi?">http://wms.jpl.nasa.gov/wms.cgi?</a>
QGIS Users	<a href="http://qgis.org/cgi-bin/mapserv?map=/var/www/maps/main.map&amp;">http://qgis.org/cgi-bin/mapserv?map=/var/www/maps/main.map&amp;</a>

An exhaustive list of WMS servers can be found at <http://wms-sites.com>.

**6.2.3. Loading WMS Layers**

Once you have successfully filled in your parameters you can select the *Connect* button to retrieve the capabilities of the selected server. This includes the Image encoding, Layers, Layer Styles, and Projections. Since this is a network operation, the speed of the response depends on the quality of your network connection to the WMS server. While downloading data from the WMS server, the download progress is visualized in the left bottom of the WMS Plugin dialog

Your screen should now look a bit like Figure 12, which shows the response provided by the NASA JPL OnEarth WMS server.

**Image Encoding**

The *Image encoding* section now lists the formats that are supported by both the client and server. Choose one depending on your image accuracy requirements.

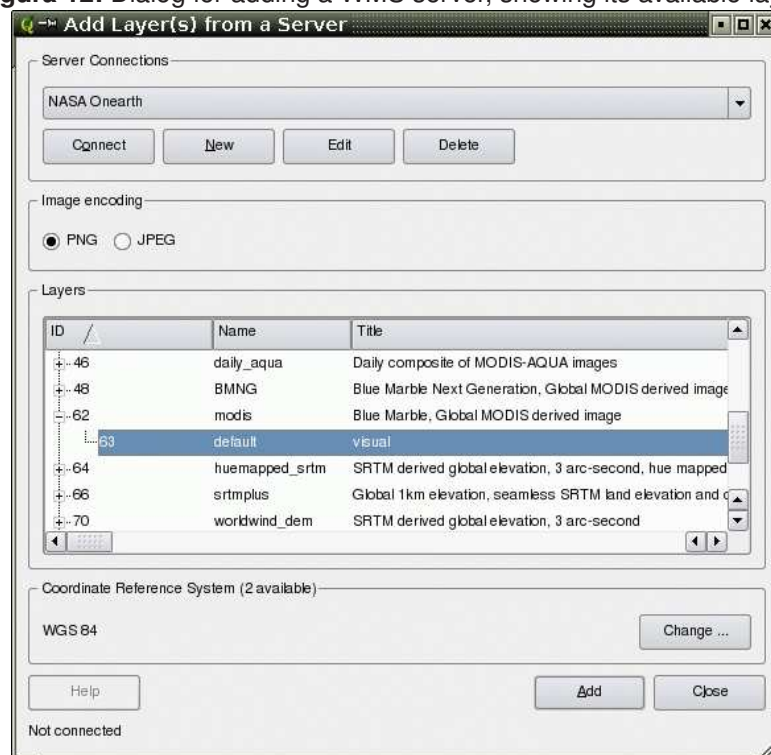
**Tip 22 IMAGE ENCODING**

You will typically find that a WMS server offers you the choice of JPEG or PNG image encoding. JPEG is a lossy compression format, whereas PNG faithfully reproduces the raw raster data.

Use JPEG if you expect the WMS data to be photographic in nature and/or you don't mind some loss in picture quality. This trade-off typically reduces by 5 times the data transfer requirement compared to PNG. Use PNG if you want precise representations of the original data, and you don't mind the increased data transfer requirements.

**Layers**

The *Layers* section lists the layers available from the selected WMS server. You may notice that some layers are expandible, this means that the layer can be displayed in a choice of image styles.

**Figura 12:** Dialog for adding a WMS server, showing its available layers

You can select several layers at once, but only one image style per layer. When several layers are selected, they will be combined at the WMS Server and transmitted to QGIS in one go.

---

### Tip 23 WMS LAYER ORDERING

In this version of QGIS, WMS layers rendered by a server are overlaid in the order listed in the Layers section, from top to bottom of the list. If you want to overlay layers in the opposite order, then you can select *Add WMS layer* a second time, choose the same server again, and select the second group of layers that you want to overlay the first group.

---

### Transparency

In this version of QGIS, the transparency setting is hard-coded to be always on, where available. Therefore no option for it exists on-screen.

This, in theory, allows you to overlay WMS layers on other layers (raster, vector or WMS) and still see through to those lower layers.

---

### Tip 24 WMS LAYER TRANSPARENCY

The availability of WMS image transparency depends on the image encoding used: PNG and GIF support transparency, whilst JPEG leaves it unsupported.

---

## Coordinate Reference System

A Coordinate Reference System is the OGC terminology for a QGIS Projection.

Each WMS Layer can be presented in multiple CRSs, depending on the capability of the WMS server. You may notice that the *x* changes in the *Coordinate Reference System (x available)* header as you select and deselect layers from the *Layers* section.

To choose a CRS, select *Change...* and a screen similar to Figure 14 in Section 7.2 will appear. The main difference with the WMS version of the screen is that only those CRSs supported by the WMS Server will be shown.

---

### Tip 25 WMS PROJECTIONS

For best results, make the WMS layer the first layer you add in the project. This allows the project projection to inherit the CRS you used to render the WMS layer. On-the-fly projection (see Section ??) can then be used to fit any subsequent vector layers to the project projection. In this version of QGIS, if you add a WMS layer later, and give it a different CRS to the current project projection, unpredictable results can occur.

---

#### 6.2.4. Using the Identify Tool

Once you have added a WMS server, and if any layer from a WMS server is queryable, you can then use the *Identify* tool to select a pixel on the map canvas. A query is made to the WMS server for each selection made.

The results of the query are returned in plain text. The formatting of this text is dependent on the particular WMS server used.

#### 6.2.5. Viewing Properties

Once you have added a WMS server, you can view its properties by right-clicking on it in the legend, and selecting *Properties*.

### Metadata Tab

The Metadata tab displays a wealth of information about the WMS server, generally collected from the Capabilities statement returned from that server.

Many definitions can be gleaned by reading the WMS standards ?, ?, but here are a few handy definitions:

- **Server Properties**

- **WMS Version** - The WMS version supported by the server.

- **Image Formats** - The list of MIME-types the server can respond with when drawing the map. QGIS supports whatever formats the underlying Qt libraries were built with, which is typically at least `image/png` and `image/jpeg`.
- **Identity Formats** - The list of MIME-types the server can respond with when you use the Identify tool. Currently QGIS supports the `text-plain` type.

#### ■ Layer Properties

- **Selected** - Whether or not this layer was selected when its server was added to this project.
- **Visible** - Whether or not this layer is selected as visible in the legend. (Not yet used in this version of QGIS.)
- **Can Identify** - Whether or not this layer will return any results when the Identify tool is used on it.
- **Can be Transparent** - Whether or not this layer can be rendered with transparency. This version of QGIS will always use transparency if this is `Yes` and the image encoding supports transparency.
- **Can Zoom In** - Whether or not this layer can be zoomed in by the server. This version of QGIS assumes all WMS layers have this set to `Yes`. Deficient layers may be rendered strangely.
- **Cascade Count** - WMS servers can act as a proxy to other WMS servers to get the raster data for a layer. This entry shows how many times the request for this layer is forwarded to peer WMS servers for a result.
- **Fixed Width, Fixed Height** - Whether or not this layer has fixed source pixel dimensions. This version of QGIS assumes all WMS layers have this set to nothing. Deficient layers may be rendered strangely.
- **WGS 84 Bounding Box** - The bounding box of the layer, in WGS 84 coordinates. Some WMS servers do not set this correctly (e.g. UTM coordinates are used instead). If this is the case, then the initial view of this layer may be rendered with a very “zoomed-out” appearance by QGIS. The WMS webmaster should be informed of this error, which they may know as the WMS XML elements `LatLonBoundingBox`, `EX_GeographicBoundingBox` or the `CRS:84 BoundingBox`.
- **Available in CRS** - The projections that this layer can be rendered in by the WMS server. These are listed in the WMS-native format.
- **Available in style** - The image styles that this layer can be rendered in by the WMS server.

#### 6.2.6. WMS Client Limitations

Not all possible WMS Client functionality had been included in this version of QGIS. Some of the more notable exceptions follow:

### Editing WMS Layer Settings

Once you've completed the *Add WMS layer* procedure, there is no ability to change the settings.

A workaround is to delete the layer completely and start again.

### WMS Servers Requiring Authentication

Only public WMS servers are accessible. There is no ability to apply a user name and password combination as an authentication to the WMS server.

## 6.3. WFS Client

In QGIS, a WFS layer behaves pretty much like any other vector layer. You can identify and select features and view the attribute table. The WFS plugin doesn't support editing at this time.

Adding a WFS layer is very similar to the procedure used with WMS. The difference is there are no default servers defined, so we have to add our own.

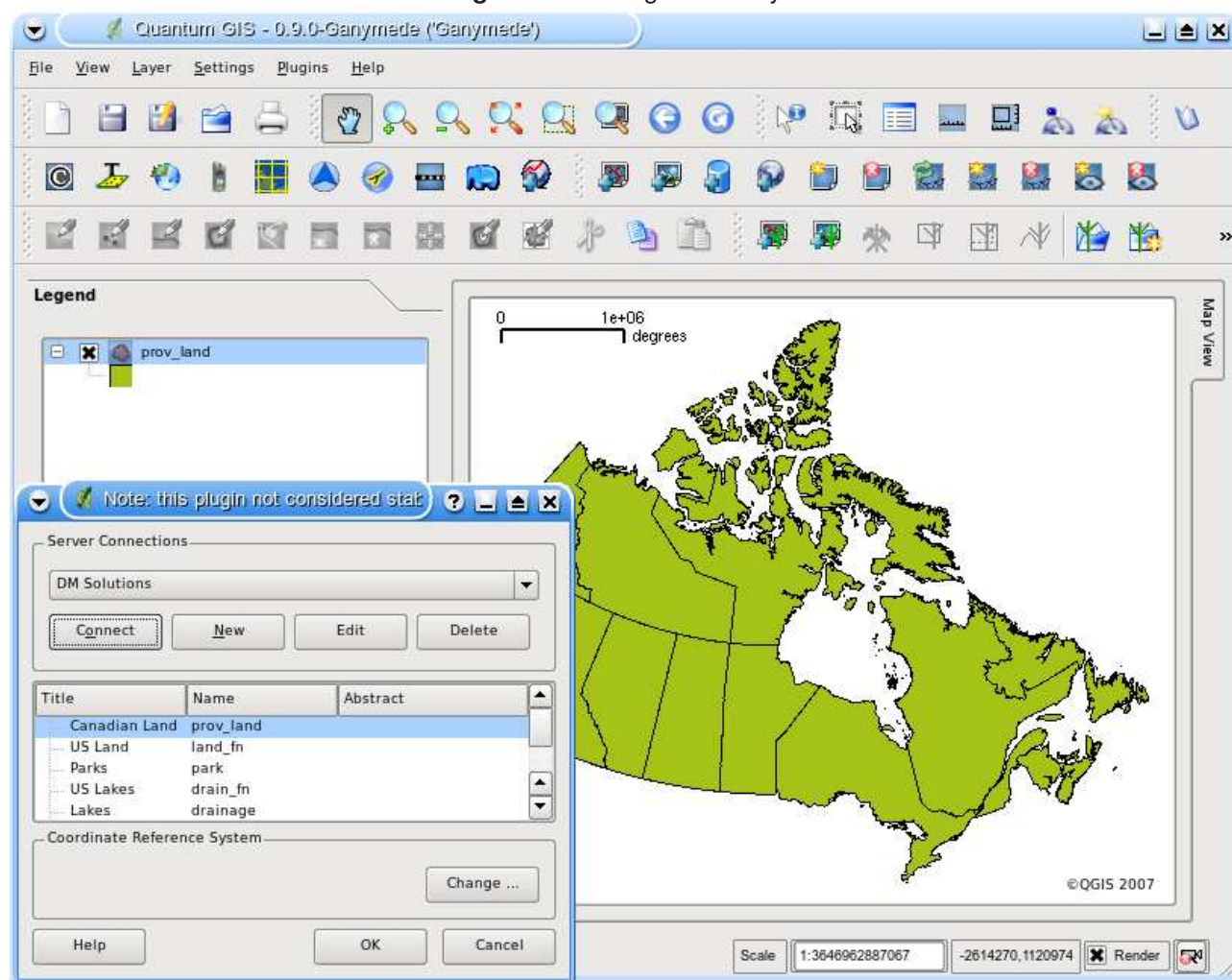
### 6.3.1. Loading a WFS Layer

As an example we use the DM Solutions WFS server and display a layer. The URL is:

```
http://www2.dmsolutions.ca/cgi-bin/mswfs_gmap?VERSION=1.0.0&SERVICE=
wfs&REQUEST=GetCapabilities
```

1. Make sure the WFS plugin is loaded; if not, open the Plugin Manager and load it
2. Click on the *Add WFS Layer* tool on the plugins toolbar
3. Click on *New*
4. Enter "DM Solutions" as the name
5. Enter the URL (see previous page)
6. Click *OK*
7. Choose "DM Solutions" from the drop-down box
8. Click *Connect*
9. Wait for the list of layers to be populated
10. Click on the "Canadian Land" layer
11. Click *Add* to add the layer to the map
12. Wait patiently for the features to appear

Figura 13: Adding a WFS layer



You'll notice the download progress is visualized in the left bottom of the QGIS main window. Once the layer is loaded, you can identify and select a province or two and view the attribute table.

Remember this plugin is still experimental. You might also experience random behavior and crashes. You can look forward to improvements in a future version of the plugin.

### Tip 26 FINDING WMS AND WFS SERVERS

You can find additional WMS and WFS servers by using Google or your favorite search engine. There are a number of lists, some of them maintained and some not, that list public servers you can use.



---

## 7. Trabajar con proyecciones

QGIS soporta la proyección al vuelo de capas vectoriales. Esta función permite visualizar capas con diferente sistema de coordenadas y superponerlas de forma adecuada.

### 7.1. Descripción del soporte para proyecciones

QGIS tiene soporte para aproximadamente 2.700 proyecciones conocidas. Las proyecciones se guardan en una base de datos de Sqlite que se instala con QGIS. Normalmente no se necesita manipular la base de datos directamente. De hecho, hacerlo puede ocasionar que falle el soporte de proyecciones. Las proyecciones personalizadas se guardan en una base de datos del usuario. Ver la sección 7.3 para información sobre la gestión de sus proyecciones personalizadas.

Las proyecciones disponibles en QGIS están basadas en aquellas definidas por EPSG y están ampliamente resumidas de la tabla de referencias\_espaciales de PostGIS versión 1.x. Note que los identificadores usados en QGIS no se corresponden con los de EPSG o PostGIS. Los identificadores EPSG y PostGIS están presentes en la base de datos y se pueden usar para especificar una proyección en QGIS.

Para usar una proyección al vuelo, sus datos deben contener información sobre su sistema de coordenadas. Para capas PostGIS QGIS usa el identificador de referencia espacial que se especificó cuando se creó la capa. Para datos soportados por OGR, QGIS se base en la presencia de un medio específico del formato para definir el sistema de coordenadas. En el caso archivos shape, esto significa un archivo que contenga la especificación WKT (Well Known Text) del sistema de coordenadas. El archivo de proyección tiene el mismo nombre base que el archivo shape y una extensión prj. Por ejemplo, un archivo shape que se llame lagos.shp tendrá un archivo de proyección correspondiente llamado lagos.prj.

### 7.2. Comenzar

Al inicio, QGIS no tiene la proyección al vuelo habilitada. Para usar la proyección al vuelo, debe abrir el diálogo *Propiedades del proyecto*, seleccionar una proyección para el mapa y activar la proyección al vuelo. Hay dos formas de abrir el diálogo *Propiedades del proyecto*:

1. Seleccionar *Propiedades del proyecto* del menú *Configuración*.
2. Hacer clic en el icono del proyector en la esquina inferior derecha de la barra de estado.

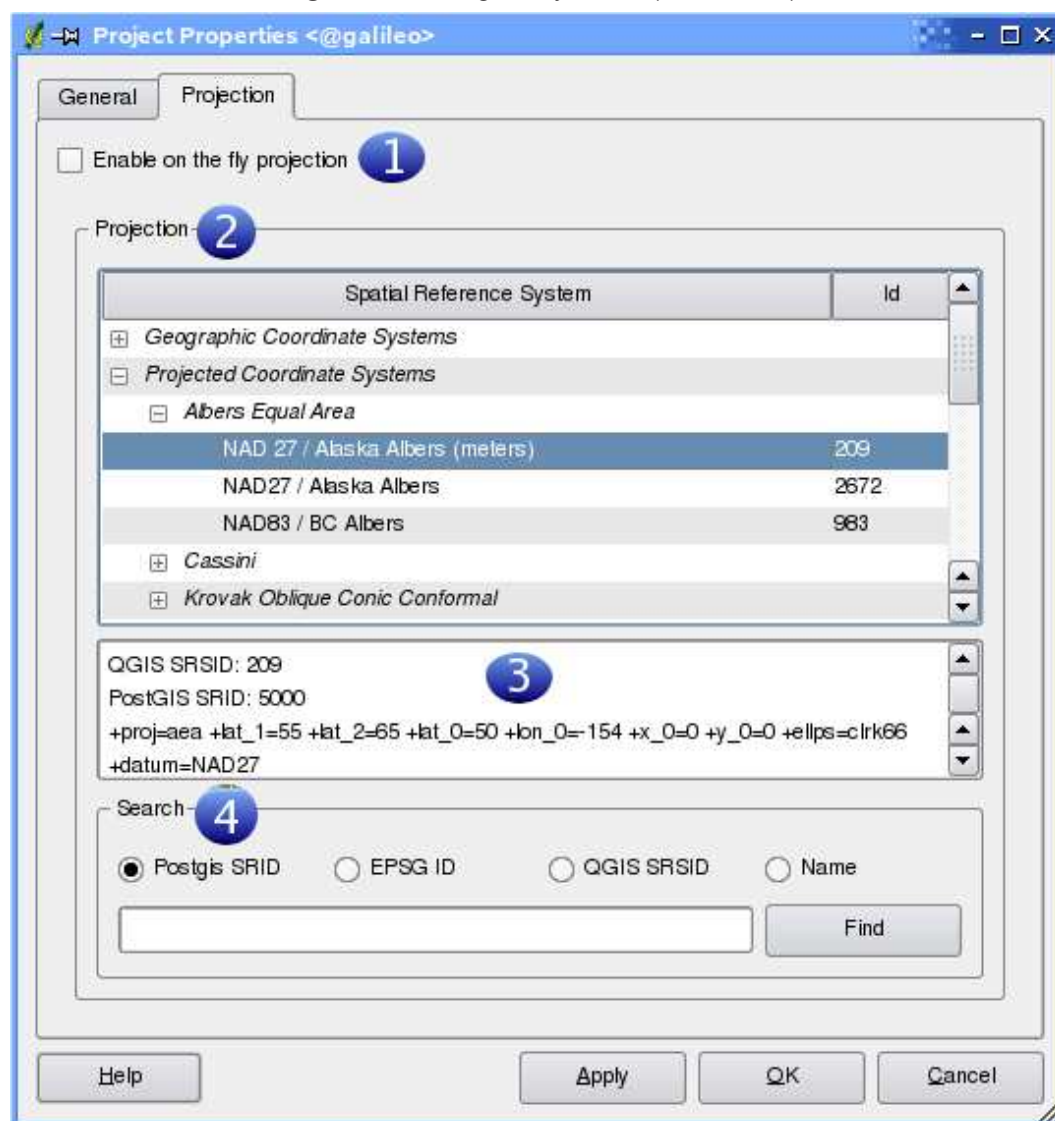
El diálogo Proyección contiene cuatro componentes importantes tal como se numeran en la Figura 14 y se describe a continuación.

1. **Activar proyección al vuelo** - esta casilla de verificación se usa para activar o desactivar la



**Tip 27** DIÁLOGO PROPIEDADES DEL PROYECTO

Si abre el diálogo *Propiedades del proyecto* desde el menú *Configuración*, debe hacer clic en la pestaña *Proyección* para ver la configuración de la proyección. Abriendo el diálogo desde el icono del proyector llevará directamente la pestaña *Proyección* al frente.

**Figura 14:** Diálogo Proyección (GNU/Linux)

proyección al vuelo. Cuando no está marcada, no se proyecta nada y cada capa se dibuja usando las coordenadas leídas de la fuente de datos. Cuando está marcada, las coordenadas de cada capa se proyectan al sistema de coordenadas de la vista del mapa.

2. **Proyecciones** - esta es una lista de todas las proyecciones soportadas por QGIS, incluyendo sistemas de coordenadas geográficas, proyectadas y personalizadas. Para usar un sistema

de coordenadas, selecciónelo de la lista expandiendo la rama adecuada y seleccionando la proyección. La proyección activa está preseleccionada.

3. **Texto Proj4** - esta es la cadena de la proyección usada por el motor de proyección Proj4. Este texto es de sólo lectura y se proporciona con finalidad informativa.
4. **Buscar** - si conoce el identificador PostGIS, EPSG o QGIS SRSID o el nombre de una proyección, puede usar la función de búsqueda para encontrarlo. Introduzca el identificador y pulse *Encontrar*.

### 7.2.1. Especificar una proyección

QGIS establece automáticamente la proyección del mapa al sistema de coordenadas de la primera capa cargada. Una forma de especificar la proyección del mapa es cargar primero una capa con la proyección que quiera para todo el mapa. A continuación abra el diálogo *Propiedades del proyecto* y marque la casilla *Activar proyección al vuelo*. Ahora puede cerrar el diálogo *Propiedades del proyecto* y añadir capas adicionales al mapa.

Si ya ha añadido capas y quiere activar la proyección al vuelo, abra el diálogo *Propiedades del proyecto* y busque el sistema de coordenadas proyectadas o geográficas que quiera usar en la lista de proyecciones. Alternativamente puede usar la función de búsqueda como se describe en la sección anterior.

## 7.3. Proyecciones personalizadas

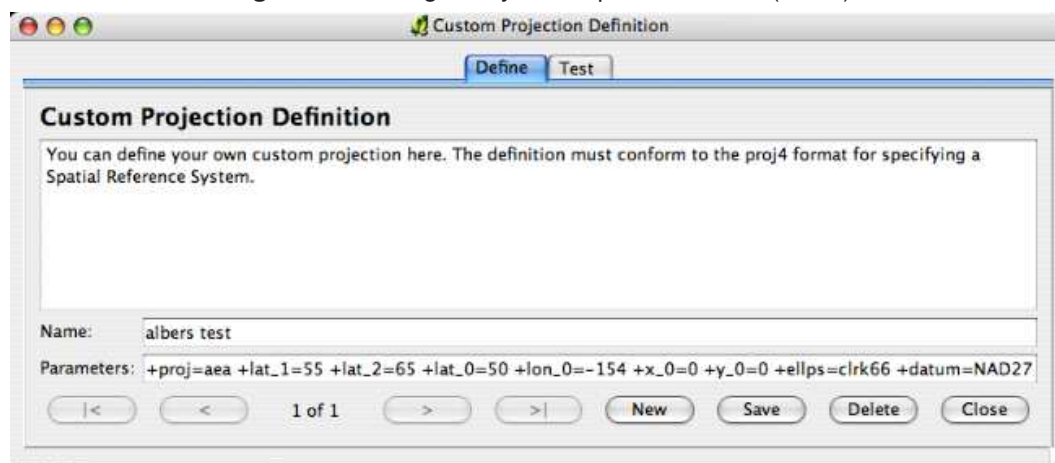
Si QGIS no tiene la proyección que necesita, puede definir una proyección personalizada. Para definir una proyección, seleccione *Proyección personalizada* del menú *Configuración*. Las proyecciones personalizadas se guardan en su base de datos de usuario de QGIS. Además de sus proyecciones, esta base de datos contiene sus marcadores espaciales y otros datos de usuario.

En la versión 0.9.1 de QGIS, definir una proyección personalizada requiere un buen conocimiento de la biblioteca de proyecciones Proj.4. Para empezar, consulte el documento *Cartographic Projection Procedures for the UNIX Environment - A User's Manual* by Gerald I. Evenden, U.S. Geological Survey Open-File Report 90-284, 1990 (disponible en <ftp://ftp.remotesensing.org/proj/OF90-284.pdf>). Este manual describe el uso del comando *proj* y utilidades relacionadas de línea de comandos. Los parámetros cartográficos usados con *proj* y descritos en el manual son los mismos que usa QGIS.

El diálogo *Proyecciones personalizadas* sólo requiere dos parámetros para definir una proyección de usuario:

1. un nombre descriptivo y
2. los parámetros cartográficos.

Figura 15: Diálogo Proyección personalizada (OS X)



Para crear una nueva proyección, pulse el botón *Nueva* e introduzca un nombre descriptivo y los parámetros de la proyección. La figura 15 muestra el diálogo con una proyección de ejemplo. Los parámetros mostrados fueron introducidos en base al conocimiento de la proyección y la información encontrada en OF90-284.









Puede probar los parámetros de su proyección para ver si dan buen resultado haciendo clic en la pestaña *Probar* y pegando los parámetros de su proyección en el campo *Parámetros*. Introduzca a continuación valores conocidos de latitud y longitud WGS 84 en los campos Norte y Este respectivamente. Pulse en *Calcular* y compare los resultados con los valores conocidos en su sistema de coordenadas proyectadas.

---

## 8. Integración de GRASS

El complemento de GRASS <sup>3</sup> proporciona acceso a GRASS desde dentro de QGIS. Esto incluye la capacidad de ver, editar y crear datos, así como realizar análisis usando los módulos de geoprocesamiento de GRASS.

En este capítulo presentaremos el complemento y alguna de las formas que se pueden usar para trabajar con datos de GRASS. El complemento de GRASS proporciona las siguientes funciones:

-  Añadir capas vectoriales de GRASS.
-  Añadir capas ráster de GRASS.
-  Caja de herramientas de GRASS.
-  Cambiar la región de GRASS.
-  Digitalización de capas vectoriales.
-  Abrir directorios de mapas existentes.
-  Crear nuevos directorios de mapas.
-  Crear nuevas capas vectoriales de GRASS.
-  Cerrar directorios de mapas de GRASS.

### 8.1. Iniciar QGIS con GRASS

Para usar las funciones de GRASS desde dentro de QGIS, debe cargar el complemento de GRASS con el administrador de complementos (vea la Sección 11.1.2) como todos los complementos de QGIS. Después de cargarlo, aparecerá una nueva barra de herramientas en la interfaz de usuario.<sup>3</sup>

Después de cargar el complemento, inmediatamente puede cargar un conjunto de datos vectoriales y ráster de GRASS existente (vea la Sección 8.2) o puede crear una nueva localización de GRASS con QGIS (vea la Sección 8.3).

---

<sup>3</sup>El complemento de GRASS es único en cuanto que crea su propia barra de herramientas

## 8.2. Cargar datos de GRASS

Con el complemento de GRASS, puede cargar capas vectoriales o ráster usando los botones adecuados de la barra de herramientas. Como ejemplo usaremos la localización spearfish de muestra en proyección UTM (vea la Sección 3.2).

1. Descargue el archivo `spearfish_grass60data-0.3.zip`.
2. Cree una carpeta nueva `grassdata` y descomprima en ella el archivo `spearfish_grass60data-0.3.zip`.
3. Inicie QGIS.
4. En la barra de herramientas de GRASS, pulse el icono *Abrir directorio de mapas* para iniciar el asistente *Seleccionar directorio de mapas de GRASS*.
5. Para la *Base de datos* explore e introduzca la ruta a la carpeta recién creada `grassdata`.
6. Ahora debería poder seleccionar la localización `spearfish60` y los directorios de mapas *PERMANENT* o *user1*.
7. Pulse *Aceptar*. Note como algunas de las herramientas de la barra de herramientas de GRASS que estaban desactivadas ahora están activadas.
8. Pulse en *Añadir capa ráster de GRASS*, seleccione el *Nombre del mapa* `geology` y pulse *Aceptar*. El mapa `geology` se visualizará.
9. Pulse en *Añadir capa vectorial de GRASS*, seleccione el *Nombre del mapa* `roads` (carreteras) y pulse *Aceptar*. Ahora el mapa `roads` se superpondrá encima de la geología.

Como puede ver, es muy sencillo cargar capas ráster y vectoriales de GRASS en QGIS. Vea las siguientes secciones para editar datos de GRASS y crear nuevas localizaciones.

---

**Tip 28 CARGAR DATOS DE GRASS**

---

Si tiene problemas para cargar datos o QGIS termina de forma anormal, asegúrese de que ha cargado el complemento de GRASS correctamente como se describe en la Sección 8.1.

---

## 8.3. Crear una localización

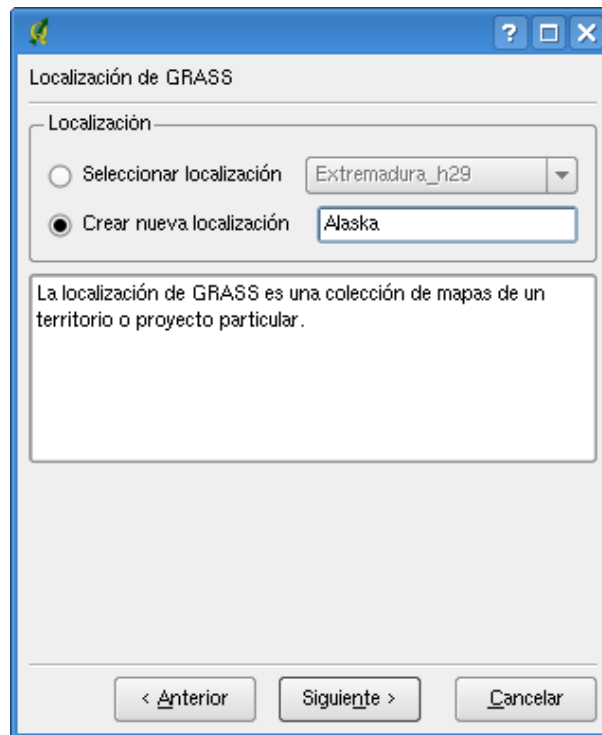
GRASS guarda los datos en una “localización” que representa un área específica con un sistema de coordenadas específico. Para usar datos de GRASS, debemos importarlos a una *localización*.<sup>4</sup>

Aquí tiene un ejemplo de cómo crear una localización de GRASS en la proyección Albers Equal Area con unidades en metros para los datos de ejemplo de QGIS (vea la Sección 3.2).

1. Inicie QGIS.

---

<sup>4</sup>Esto no es estrictamente cierto, se pueden ver conjuntos de datos externos sin importarlos

**Figura 16:** Crear una localización de GRASS en QGIS

2. Asegúrese de que el complemento de GRASS está cargado.
3. Cargue el archivo shape *alaska.shp* (vea la Sección 4.1.1).
4. En la barra de herramientas de GRASS, pulse en la herramienta *Nuevo directorio de mapas* para llamar al asistente directorio de mapas.
5. Cada localización se guarda en un directorio. Seleccione un directorio de datos existente o cree uno nuevo para guardar la localización.
6. Pulse *Siguiente*.
7. Podemos usar el asistente para crear un nuevo directorio de mapas dentro de una localización existente o crear una nueva localización todo junto. Marque el botón circular “Crear nueva localización”.
8. Introduzca un nombre para la localización (usaremos Alaska).
9. Pulse *Siguiente*.
10. Defina la proyección pulsando el botón circular “Proyección” para activar la lista de proyecciones.
11. Estamos usando la proyección Albers Equal Area Alaska (metros). Como sabemos que su SRID de PostGIS es 5000, lo introducimos en la casilla de búsqueda. (Si quiere repetir este

proceso para otra capa y no recuerda el SRID de PostGIS, pulse en el icono del proyector en la esquina inferior derecha de la barra de estado (vea la Sección 7.2).)

12. Pulse *Encontrar* para seleccionar la proyección.
13. Pulse *Siguiente*.
14. Para definir la región predeterminada, tenemos que introducir los límites en dirección Norte, Sur, Este y Oeste. Aquí simplemente pulsaremos el botón *Establecer la extensión actual de QGIS*.
15. Pulse *Siguiente*.
16. Tenemos que definir un directorio de mapas dentro de nuestra nueva localización. Póngale el nombre que prefiera (su nombre de usuario es una buena opción).
17. Compruebe el resumen para asegurarse que es correcto.
18. Pulse *Terminar*
19. El directorio de mapas y la localización son creados y se abren como el conjunto de trabajo actual.
20. Vea como algunas de las herramientas de la barra de herramientas de GRASS que estaban desactivadas ahora están activadas para su uso.

Si le parecieron muchos pasos, ésto no es tan malo y sí una forma muy rápida de crear una localización. Nuestra localización ahora está lista para usar. Para ver la región predeterminada, aleje el zum. Pulsando la herramienta *Mostrar región actual de Grass* se activa/desactiva la visualización de la región.

## 8.4. Modelo de datos vectoriales

Es importante entender el modelo de datos vectoriales de GRASS antes de digitalizar. En general, GRASS usa un modelo vectorial topológico. Esto significa que las áreas no se representan como polígonos cerrados, sino por uno o más contornos. Un contorno entre dos áreas adyacentes sólo se digitaliza una vez, y es compartido por ambas áreas. Los contornos deben estar conectados sin saltos. Un área es identificada (etiquetada) por los centroides del área.

Además de contornos y centroides, un para vectorial puede contener puntos y líneas. Todos estos elementos geométricos pueden mezclarse en un vectorial y se representarán en las llamadas «capas» en QGIS.

Es posible guardar más «capas» en un conjunto de datos vectorial. Por ejemplo, se pueden guardar campos, bosques y lagos en un vectorial. Los bosques y lagos adyacentes pueden compartir el mismo contorno, pero tiene tablas de atributos separadas. También es posible adjuntar atributos a los contornos. Por ejemplo, el contorno entre lago y bosque es una carretera, por lo que puede tener una tabla de atributos diferente.

La «capa» de los objetos espaciales se define por la «capa» dentro de GRASS. «Capa» es un número que define si hay más de una capa dentro del conjunto de datos, por ejemplo, si la geometría es bosque o lago. De momento, puede ser sólo un número, en el futuro GRASS también soportará nombre como campos en la interfaz de usuario.

Los atributos se pueden guardar en bases de datos externas, por ejemplo DBF, PostgreSQL, MySQL, SQLite3, etc.

Los atributos en las tablas de las bases de datos se enlazan a los elementos geométricos usando la «categoría». La «categoría» (clave, ID) es un entero adjunto a los primitivos de la geometría y se usa como el enlace a una columna de la tabla de la base de datos.

---

#### **Tip 29** APRENDER EL MODELO VECTORIAL DE GRASS

---

La mejor forma de aprender el modelo vectorial de GRASS y sus capacidades es descargar uno de los muchos manuales de GRASS, donde se describe el modelo vectorial con más detalle. Vea <http://grass.osgeo.org/gdp/manuals.php> para más información, libros y manuales en varios idiomas.

---

## **8.5. Herramientas de digitalización y edición**



Las herramientas de digitalización para las capas vectoriales de GRASS son accesibles usando la herramienta *Editar capa vectorial de GRASS* de la barra de herramientas. Asegúrese de que ha cargado un vectorial de GRASS y que éste es la capa seleccionada en la leyenda antes de pulsar la herramienta de edición. Si quiere crear un nuevo vectorial de GRASS, necesita usar la entrada de menú Complementos->GRASS->Crear nuevo vectorial de GRASS o el icono de la barra de herramientas de Grass.

La figura 17 muestra el diálogo de edición de GRASS que se muestra cuando se pulsa la herramienta de edición.

Las herramientas y la configuración se tratan en las siguientes secciones.

### **8.5.1. Barra de herramientas**

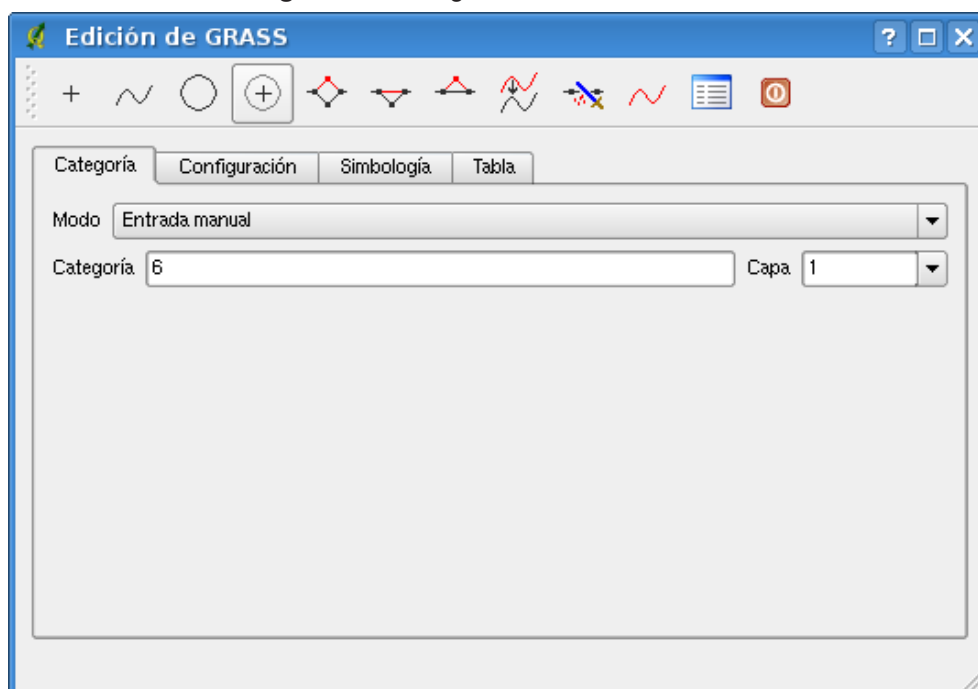
La tabla 4 lista las herramientas de digitalización proporcionadas por el complemento de GRASS. Éstas corresponden a los botones de herramientas en la barra de herramientas de la parte superior del diálogo.

### **8.5.2. Pestaña Categoría**

Esta pestaña le permite establecer la forma en que se asignará la categoría a cada nuevo objeto espacial y/o asignar una categoría a un objeto espacial.



Figura 17: Diálogo de edición de GRASS



- **Modo:** qué categoría se debería adjuntar a la geometría.
  - La siguiente sin usar: la siguiente categoría que aún no está usada en el archivo vectorial.
  - Entrada manual: definir la categoría en el campo de entrada «Categoría».
  - Ninguna categoría: digitalizar la geometría sin introducir ninguna categoría.
- **Categoría:** un número (ID) que se adjunta al objeto espacial digitalizado.
- **Capa:** identificación del objeto espacial (tabla de atributos).

---

**Tip 30** CREAR «CAPAS» ADICIONALES CON QGIS

Si quiere añadir más capas a su conjunto de datos, simplemente añada un nuevo número en la casilla «Capa» y pulse Intro. En la pestaña Tabla puede crear una nueva tabla conectada a su nueva capa.

---

### 8.5.3. Pestaña configuración

Esta pestaña le permite establecer el autoensamblado en píxeles de la pantalla. Esto es el umbral en píxeles en el que los nuevos puntos o finales de líneas son ensamblados de forma automática a nodos existentes. Esto ayuda a evitar saltos o balanceos entre contornos. El valor predeterminado es 10 píxeles.

**Cuadro 4:** Herramientas de digitalización de GRASS

Icono	Herramienta	Propósito
	Nuevo punto	Digitalizar un punto nuevo
	Nueva línea	Digitalizar una línea nueva (finaliza al seleccionar una herramienta nueva)
	Nuevo contorno	Digitalizar un contorno nuevo (finaliza al seleccionar una herramienta nueva)
	Nuevo centroide	Digitalizar un centroide nuevo (etiquetar un área existente)
	Mover vértice	Seleccionar un vértice de una línea o contorno existente e identificar una nueva posición
	Añadir vértice	Añadir un vértice nuevo a una línea existente
	Borrar vértice	Borrar un vértice de una línea existente (confirmar el vértice seleccionado con otra pulsación)
	Mover elemento	Seleccionar un elemento existente y pulsar en la nueva posición
	Dividir línea	Dividir una línea existente en dos partes
	Borrar elemento	Borrar un elemento existente (confirmar el elemento seleccionado con otra pulsación)
	Editar atributos	Editar los atributos de un elemento existente (tenga en cuenta que un elemento puede representar a más objetos espaciales, vea arriba)
	Cerrar	Cerrar la sesión de digitalización (reconstruye la topología a continuación)

#### 8.5.4. Pestaña simbología

Esta pestaña le permite ver y establecer la simbología y la configuración del color para varios tipos de geometría y su estado topológico (ej.: contorno cerrado / abierto).

#### 8.5.5. Pestaña tabla

Esta pestaña proporciona información sobre la tabla de la base de datos de una «capa» dada. Aquí puede añadir, modificar o crear nuevas tablas de base de datos para la capa actual.

**Tip 31 PERMISOS DE EDICIÓN DE GRASS**

Tiene que ser el propietario del directorio de mapas de GRASS que quiera editar. Es imposible editar vectoriales en directorios de mapas que no sean suyos, incluso si tiene permiso de escritura.

**8.6. Herramienta Región**

La región actual (ventana) en GRASS es muy importante para todos los módulos ráster. Todos los ráster de nueva creación tienen la extensión y resolución de la región actual, no importa cuál sea su región original. La región se guarda en el archivo \$LOCATION/\$MAPSET/WIND, y define el Norte, Sur, Este, Oeste, número de columnas y filas y la resolución espacial horizontal y vertical.

Es posible encender/apagar la región de GRASS en la vista del mapa de QGIS usando el botón *Mostrar región actual de GRASS*.

Con el botón *Editar la región actual de GRASS* puede abrir una herramienta en la que puede cambiar la región actual y la simbología del rectángulo de la región de GRASS en la vista del mapa de QGIS. Cuando se está ejecutando la herramienta también es posible seleccionar una nueva región de forma interactiva con el ratón sobre el lienzo de QGIS.

**8.7. Caja de herramientas de GRASS**

La caja de herramientas de GRASS proporciona funciones analíticas de GRASS dentro de QGIS. Para usar la caja de herramientas de GRASS necesita tener abierto un directorio de mapas en el que tenga permiso de escritura. Esto es necesario porque QGIS con toda probabilidad creará nuevos conjuntos de datos que necesitarán escribirse en un directorio de mapas válido.

Por lo tanto necesita iniciar QGIS desde dentro de una sesión de GRASS. Así su directorio de mapas actual se abrirá para escritura.

Otra opción para abrir un directorio de mapas para escritura la proporciona la entrada del complemento de GRASS. Use Complementos->GRASS->Abrir directorio de mapas.

Si tiene el botón de la caja de herramientas de GRASS atenuado, asegúrese de que abrió un directorio de mapas válido para escritura, puesto que el complemento de GRASS necesita un directorio de mapas para guardar sus resultados.

La caja de herramientas también proporciona un explorador de datos muy útil para navegar por su localización actual y los directorios de mapas que contiene.

### 8.7.1. Módulos dentro de la caja de herramientas

La caja de herramientas de GRASS proporciona una colección de módulos de GRASS que se pueden usar desde dentro de QGIS. Están agrupados en bloques temáticos que se pueden definir por el usuario (vea la Sección 8.7.3).

Cuando pulse en un módulo se añadirá una nueva pestaña a su caja de herramientas que proporciona tres nuevas subpestañas:

1. Opciones
2. Salida
3. Manual

#### Opciones

Esta pestaña proporciona un campo de entrada muy simplificado en el que tiene que seleccionar los mapas necesarios e introducir los parámetros para ejecutar el módulo seleccionado. Tenga en cuenta que estas opciones se mantienen lo más simples posible, con el fin de mantener clara la estructura. Si necesita más opciones del módulo, siéntase libre de usar la consola de GRASS para ejecutar el módulo.

#### Salida

Esta pestaña proporciona la salida generada por el módulo que se está ejecutando. Después de pulsar el botón «Ejecutar», el módulo pasa a la pestaña Salida y verá información sobre el proceso. Si todo va bien, verá `Finalizado correctamente` al final.

#### Manual

Esta pestaña muestra la página de ayuda de cada módulo de GRASS. Puede echar un vistazo a la página del manual si quiere tener un conocimiento mayor sobre el objetivo del módulo. Puede que se haya dado cuenta de que algunos módulos tienen más opciones y parámetros que los que aparecen en la pestaña *Opciones*. Esto es correcto y hecho así por diseño. Para mantener la interfaz de usuario lo más simple posible, sólo se ponen las opciones y parámetros necesarios en la pestaña Opciones. Pero siempre puede usar la consola de GRASS para ejecutar el módulo con todos sus parámetros.

---

**Tip 32** MOSTRAR RESULTADOS INMEDIATAMENTE

---

Si quiere mostrar los resultados de sus cálculos inmediatamente en el lienzo de su mapa, puede usar el botón «Ver salida» de la parte inferior de la pestaña del módulo.

---







### 8.7.2. Explorador de GRASS

Otra función útil es el explorador de GRASS. En la Figura 19(a) puede ver la localización actual con su directorio de mapas.

El explorador de la izquierda le permite navegar por todos sus directorios de mapas dentro de la localización seleccionada.

La parte derecha de la ventana del explorador muestra alguna metainformación del conjunto de datos seleccionado, por ejemplo la resolución, límites exteriores, fuente de datos, tabla de atributos para datos vectoriales...

La barra de herramientas que hay dentro de la pestaña Explorador le proporciona las siguientes herramientas para el conjunto de datos seleccionado:

-  Añadir el mapa seleccionado a la vista del mapa.
-  Copiar el mapa seleccionado.
-  Cambiar el nombre al mapa seleccionado.
-  Borrar el mapa seleccionado.
-  Establecer la región actual al mapa seleccionado.
-  Actualizar la ventana del explorador.

Los botones «Cambiar nombre» y «Borrar» sólo están disponibles en su directorio de mapas actual. Todas las demás herramientas también funcionan en mapas de otros directorios de mapas.

### 8.7.3. Personalizar la sección de los módulos

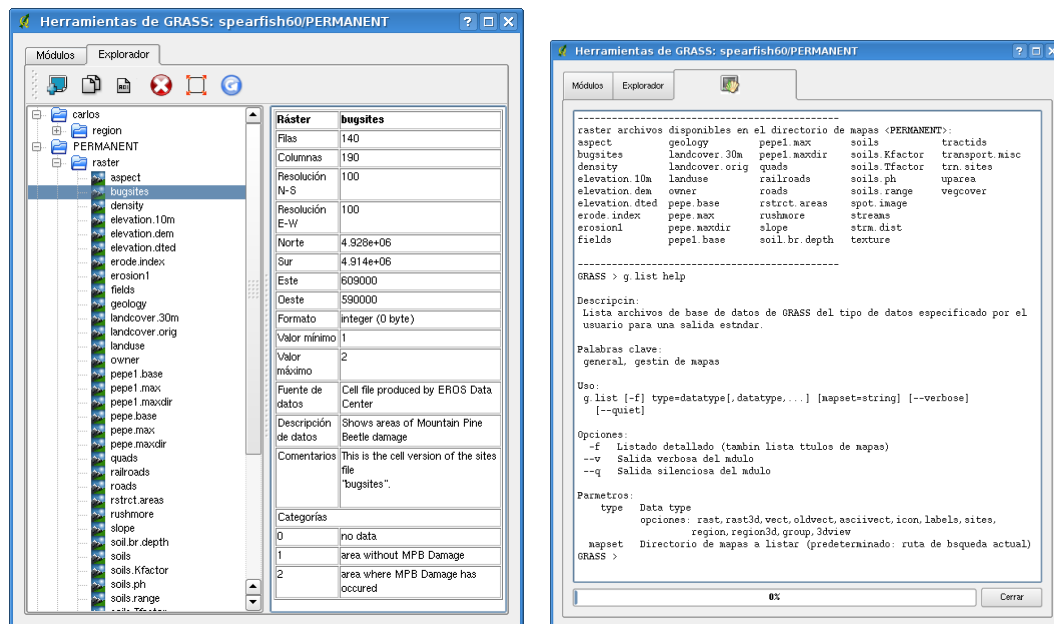
Casi todos los módulos de GRASS se pueden añadir a la caja de herramientas de GRASS. Se proporciona una interfaz XML para analizar los sencillos archivos XML que configuran los módulos dentro de la caja de herramientas.

Se puede encontrar una breve descripción sobre cómo añadir nuevos módulos, cambiar los grupos de módulos, etc. en el wiki de QGIS en [http://wiki.qgis.org/qgiswiki/Adding\\_New\\_Tools\\_to\\_the\\_GRASS\\_Toolbox](http://wiki.qgis.org/qgiswiki/Adding_New_Tools_to_the_GRASS_Toolbox).

Un ejemplo de archivo XML para generar el módulo `v.buffer` (`v.buffer.qgm`) tiene este aspecto:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE qgisgrassmodule SYSTEM "http://mrcc.com/qgisgrassmodule.dtd">
```

Figura 18: Caja de herramientas de GRASS



```
<qgisgrassmodule label="Vector buffer" module="v.buffer">
  <option key="input" typeoption="type" layeroption="layer" />
  <option key="buffer"/>
  <option key="output" />
</qgisgrassmodule>
```

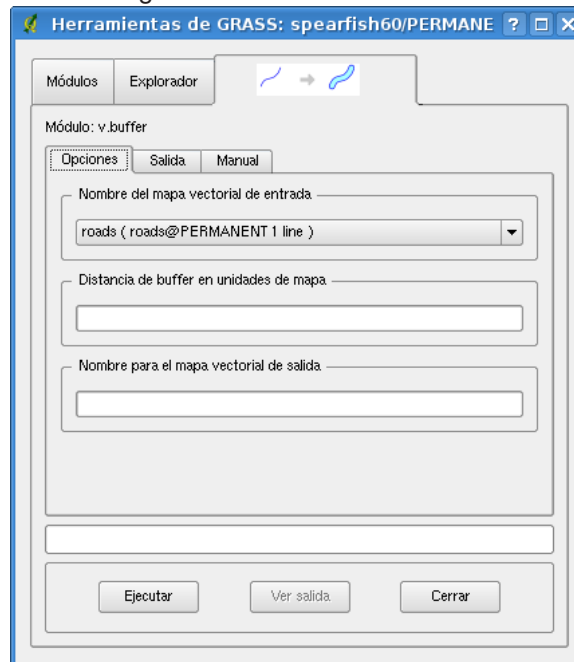
El analizador lee esta definición y crea una pestaña nueva dentro de la caja de herramientas cuando selecciona el módulo:

## 8.8. Crear una nueva capa de GRASS

Con esta versión de QGIS también es posible crear nuevos vectoriales desde dentro de GRASS muy fácilmente.

Simplemente seleccione *Complementos->GRASS->Crear nuevo vectorial de GRASS* de la barra de herramientas, dele un nombre nuevo en el cuadro de texto y comience a digitalizar. Si encuentra el botón atenuado, asegúrese de que tiene activado un directorio de mapas de trabajo. Si olvidó cómo activar un directorio de mapas eche un vistazo a la Sección 8.2.

Puesto que GRASS es capaz de organizar todo tipo de geometrías en una capa, no hay necesidad

**Figura 19:** Módulo generado mediante el análisis del archivo XML

de seleccionar la geometría. Ésto sólo se aplica a la creación de archivos shape (vea la sección 4.4.3).

Algunos consejos para hacer su digitalización más útil:

- Asegúrese de crear una tabla de atributos con las columnas necesarias antes de empezar a digitalizar si quiere asignar atributos a sus objetos digitalizados. Vaya a la pestaña Tabla dentro de la ventana de digitalización.
- Si planea crear una capa de polígonos, considere establecer el modo a *Sin categoría*. A continuación empiece a digitalizar los contornos que realmente no necesitan una entrada en la tabla de atributos. Si ha hecho esto, vuelva a cambiar a *La siguiente sin usar* y comience a digitalizar los centroides, que contienen la información de los atributos de un polígono.

---

## 9. Making MapServer Map Files

QGIS can be used to create map files for MapServer. You use QGIS to “compose” your map by adding and arranging layers, symbolizing them, and customizing the colors.

In order to use the MapServer exporter, you must have Python on your system and QGIS must have been compiled with support for it.

### 9.1. Creating the Project File

To create a MapServer map file:

1. Add your layers to QGIS
2. Symbolize your layers, setting the renderer and colors
3. Arrange the layers in the order you want them to appear in MapServer
4. Save your work to a QGIS project file

This gets us to the point where we are ready to create the map file.

---

#### **Tip 33** MAPSERVER EXPORT REQUIRES A QGIS PROJECT FILE

This has been a source of confusion for a number of people. The MapServer export tool operates on a saved QGIS project file, **not** the current contents of the map canvas and legend. When using the tool, you need to specify a QGIS project file as input.

---

### 9.2. Creating the Map File

The exporter tool (*msexport*) is installed in your QGIS binary directory and can be used independently of QGIS.

From QGIS you can start the exporter by choosing *Export to MapServer Map...* from the *File* menu.

Here is a summary of the input fields:

#### **Map file**

Enter the name for the map file to be created. You can use the button at the right to browse for the directory where you want the map file created.

#### **Qgis project file**

Enter the full path to the QGIS project file (.qgs) you want to export. You can use the button at the right to browse for the QGIS project file.

#### **Map Name**

A name for the map. This name is prefixed to all images generated by the mapserver.



**Map Width**

Width of the output image in pixels.

**Map Height**

Height of the output image in pixels.

**Map Units**

Units of measure used for output

**Image type**

Format for the output image generated by MapServer

**Web Template**

Full path to the MapServer template file to be used with the map file

**Web Header**

Full path to the MapServer header file to be used with the map file

**Web Footer**

Full path to the MapServer footer file to be used with the map file

Only the Map file and QGIS project file inputs are required to create a map file, however you may end up with a non-functional map file, depending on your intended use. Although QGIS is good at creating a map file from your project file, it may require some tweaking to get the results you want - but it's still way better than writing a map file from scratch.

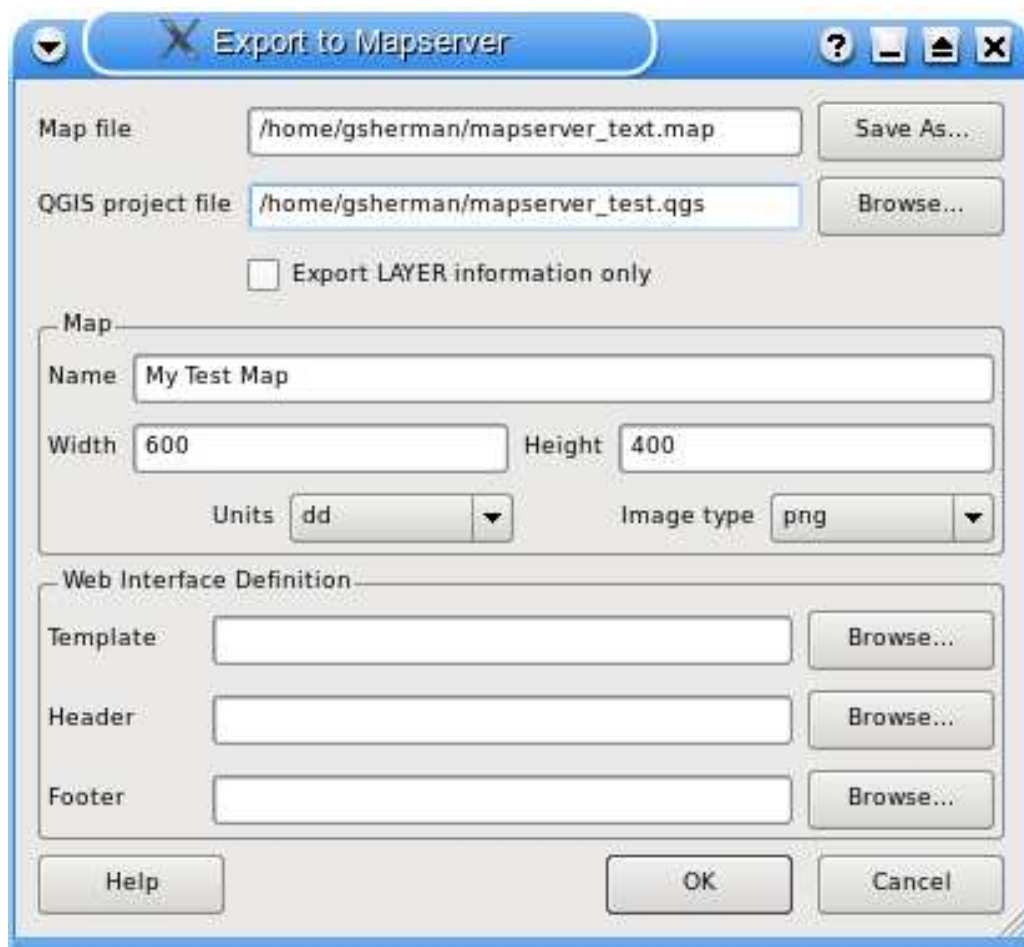
**Creating a Map File**

Let's create a map file using the shape files *alaska*, *lakes* and *rivers* layers from the *qgis\_sample\_data*:

1. Load the *alaska*, *rivers* and *lakes* layers into QGIS
2. Change the colors and symbolize the data as you like
3. Save the project using *Save Project* from the *File* menu
4. Open the exporter by clicking on *Export to MapServer Map...* in the *File* menu
5. Enter a name for your new map file
6. Browse and find the project file you just saved
7. Enter a name for the map
8. Enter 600 for the width and 400 for the height
9. Our layers are in decimal degrees so we don't need to change the units
10. Choose "png" for the image type
11. Click *OK* to generate the map file

You'll notice there is no feedback on the success of your efforts. This is an enhancement scheduled for the next version.

Figura 20: Export to MapServer map module in QGIS



You can view the map file in an editor or using *less*. If you take a look, you'll notice that the export tool adds the metadata needed to enable our map file for WMS.

### 9.3. Testing the Map File

Let's test our work by using the *shp2img* command to create an image from the map file. The *shp2img* utility is part of MapServer, but is also distributed with FWTools. To create an image from our map:

- Open a terminal window
- If you didn't save your map file in your home directory, change to the directory where you saved it
- Run *shp2img*
- View the created image

Assuming our map file was named *mapserver\_test.map*, the `shp2img` command is:

```
shp2img -m mapserver_test.map -o mapserver_test.png
```

This creates a PNG for us to view, containing all the layers that were on when we saved the QGIS project. In addition, the extent of the PNG will be the same as when we saved the project.

If you plan to use the map file to serve WMS requests, you probably don't have to tweak anything. If you plan to use it with a mapping template or a custom interface, you may have a bit of manual work to do. To see how easy it is to go from QGIS to serving maps on the web, take a look at Christopher Schmidt's 5 minute flash video. <sup>5</sup>

---

<sup>5</sup><http://openlayers.org/presentations/mappingyourdata/>

---

## 10. Diseñador de mapas

El diseñador de mapas es una función que proporciona capacidades limitadas de trazado e impresión. El diseñador le permite añadir elementos tales como la vista del mapa de QGIS, leyenda, barra de escala, imágenes y texto. Puede modificar el tamaño y la posición de cada elemento y ajustar las propiedades para crear su composición. El resultado se puede imprimir, exportar como imagen o exportarse a SVG.

Para acceder al diseñador de mapas, haga clic en el botón *Imprimir* de la barra de herramientas o seleccione *Imprimir* del menú *Archivo*.

### 10.1. Usar el diseñador de mapas

Para usar el diseñador de mapas, primero añada las capas que quiera imprimir a QGIS. Debería representar y simbolizar las capas a su gusto antes de diseñar el mapa.

Abrir el diseñador de mapas le proporciona un lienzo en blanco al que puede añadir la vista del mapa actual, leyenda, barra de escala y texto. La figura 21 muestra la vista inicial del diseñador de mapas antes de que se añada ningún elemento.

El diseñador de mapas tiene dos pestañas: *General* y *Elemento*. La pestaña *General* permite establecer el tamaño del papel, la orientación y la resolución del mapa. La pestaña *Elemento* muestra las propiedades del elemento seleccionado actualmente en el mapa. Seleccionando un elemento en el mapa (ej. leyenda, barra de escala, texto, etc.) y haciendo clic en la pestaña *Elemento*, puede personalizar la configuración.

Puede añadir múltiples elementos al diseñador. Esto permite tener más de una vista de mapa y leyenda en el diseñador. Cada elemento tiene sus propias propiedades y en el caso del mapa, su propia extensión.

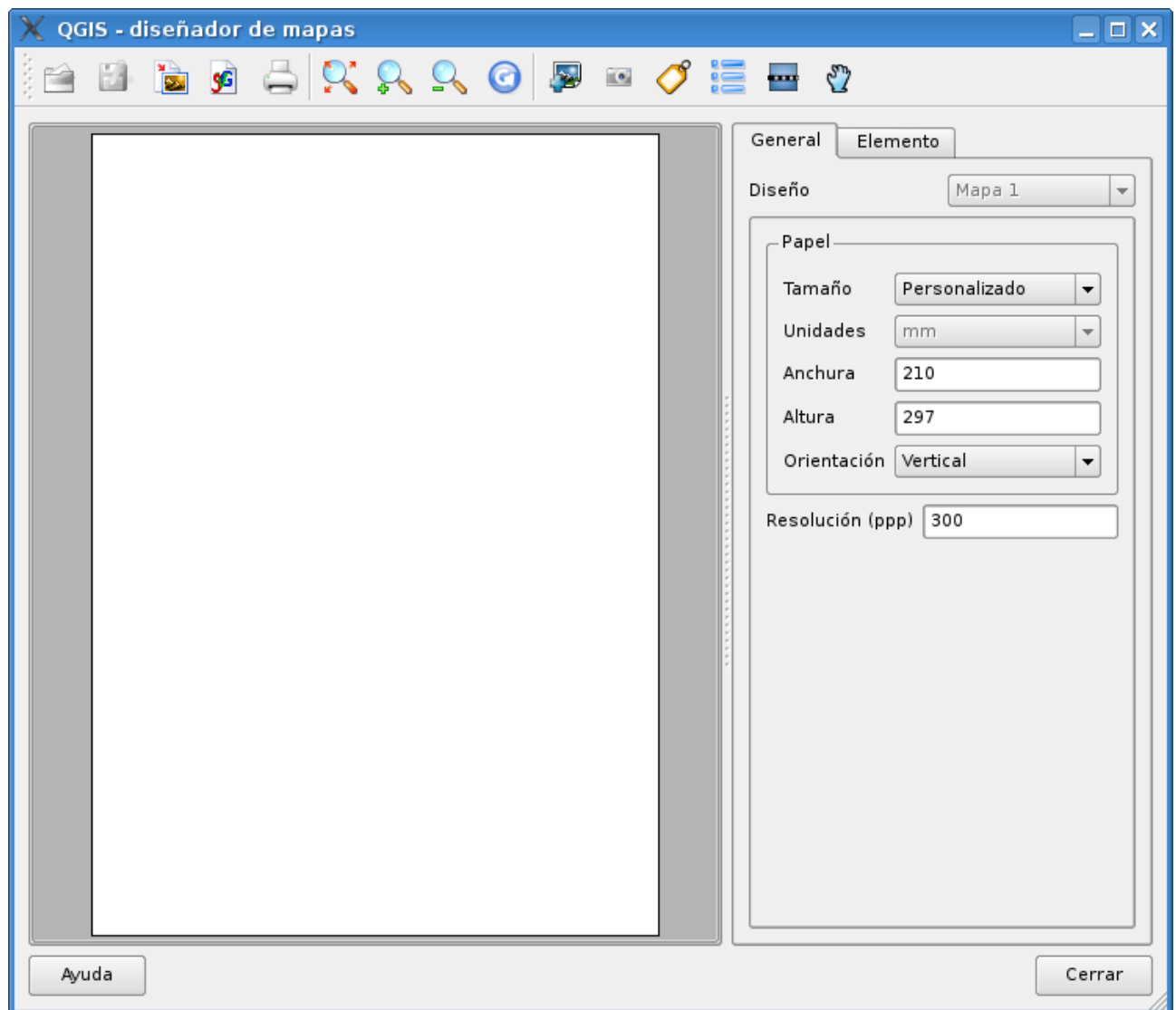
#### 10.1.1. Añadir un mapa al diseñador



Para añadir la vista del mapa de QGIS al diseñador de mapas, haga clic en el botón *Añadir mapa nuevo* en la barra de herramientas. Arrastre un rectángulo en el lienzo del diseñador para añadir el mapa. Puede redimensionar el mapa más tarde haciendo clic en el botón *Seleccionar/Mover elemento*, haciendo clic en el mapa y arrastrando una de las asas de las esquinas del mapa. Con el mapa seleccionado, también puede redimensionarlo especificando la anchura y la altura en la pestaña de propiedades del elemento.

El mapa está enlazado con la vista del mapa de QGIS. Si cambia la vista en el lienzo del mapa haciendo zoom o desplando, puede actualizar la vista del diseñador de mapas haciendo clic en el

Figura 21: Diseñador de mapas



botón *Actualizar vista*. También puede cambiar la vista del diseñador especificando una escala de mapa. Para establecer la vista a una escala determinada:

1. Seleccione *Escala (calcular extensión)* del cuadro de lista *Establecer*.
2. Introduzca el denominador de la escala en el cuadro de escala.
3. Pulse Intro.

### 10.1.2. Añadir otros elementos al diseñador



Se pueden usar plantillas existentes de QGIS para cargar y adaptar fácilmente disposiciones de mapa. Para abrir una plantilla existente, haga clic en el botón *Abrir plantilla*. Seleccione una plantilla y personalice su apariencia.



Para añadir un logotipo, flecha de Norte o cualquier clase de imagen al diseñador, haga clic en el botón *Añadir imagen*. La imagen se situará en el lienzo del diseñador y la podrá mover donde desee.



Se puede añadir una leyenda al lienzo del diseñador y personalizarla para mostrar sólo las capas deseadas. Para añadir una leyenda, haga clic en el botón *Añadir nueva leyenda vectorizada*. La leyenda se situará en el lienzo del diseñador y la podrá mover donde desee. Haga clic en la pestaña *Elemento* para personalizar el aspecto de la leyenda, incluyendo qué capas se muestran.



Para añadir una barra de escala al diseñador, haga clic en el botón *Añadir nueva barra de escala*. Use la pestaña *Elemento* para personalizar el tamaño y número de segmentos y las unidades, tamaño y fuente de la escala.



Puede añadir etiquetas de texto al diseñador haciendo clic en el botón *Añadir etiqueta nueva*. Use la pestaña *Elemento* mientras el texto está seleccionado para personalizar la configuración o cambiar el texto predeterminado.

La figura 22 muestra el diseñador de mapas después de añadir cada tipo de elemento del mapa.

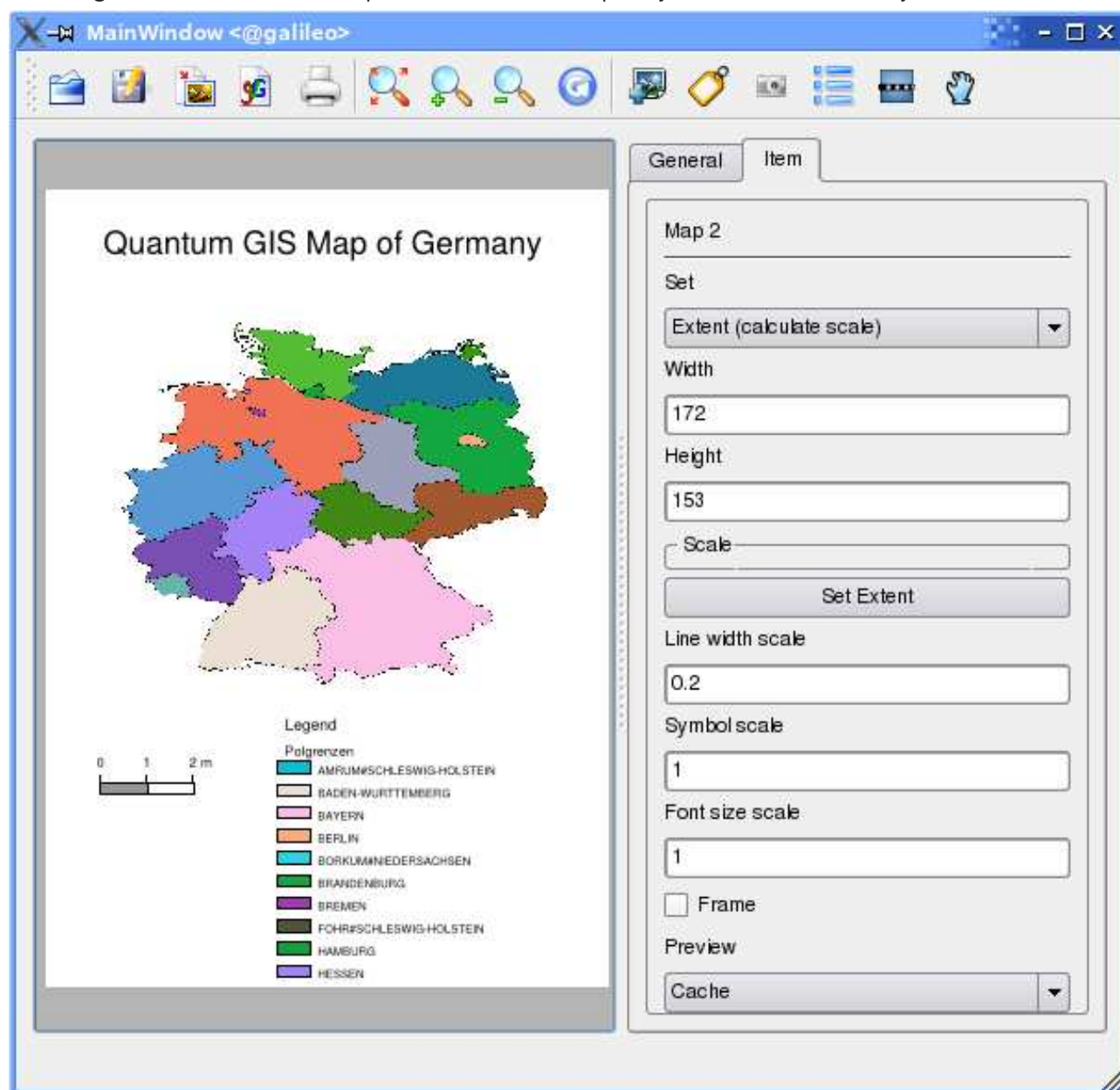
### 10.1.3. Otras funciones


El diseñador de mapas tiene herramientas de navegación para acercar y alejar el zum. Para acercar el zum, haga clic en la herramienta *Acercar zum*. El lienzo del diseñador de mapas se ampliará en un factor de 2. Use las barras de desplazamiento para ajustar la vista al área de interés. Alejar con zum funciona de forma similar.


Si encuentra la vista en un estado inconsistente, puede usar el botón *Actualizar vista* para volver a dibujar el lienzo del diseñador.

### 10.1.4. Crear la salida

El diseñador de mapas le permite imprimir el mapa en una impresora, exportarlo a PNG o a SVG. Cada una de estas funciones está disponible desde la barra de herramientas del diseñador.

**Figura 22:** Diseñador de mapas con la vista del mapa, leyenda, barra de escala y texto añadidos

 Para guardar el lienzo del diseñador como plantilla, haga clic en el botón *Guardar plantilla como*. Busque el directorio que desee y guarde la plantilla para usarla de nuevo para otro mapa.

 Es posible exportar el resultado como una imagen haciendo clic en el botón *Exportar como imagen*.



Para exportar el lienzo del diseñador como un SVG (Gráfico vectorial escalable), haga clic en el botón *Exportar como SVG*. **Nota:** Actualmente la salida SVG es muy básica. Esto no es un problema de QGIS, sino de la biblioteca subyacente Qt. Esto se solucionará en versiones futuras.



---

## 11. Usar complementos

### 11.1. Una introducción al uso de complementos

QGIS se ha diseñado con una arquitectura de complementos. Esto permite que se añadan nuevas funciones a la aplicación. Muchas de las funciones actuales de QGIS están en realidad implementadas como complementos.

Hay dos tipos de complementos en QGIS: integrados o aportados por usuarios. Un complemento integrado es mantenido por el equipo de desarrollo de QGIS y forma parte de cada distribución de QGIS. Un complemento aportado por usuarios es un complemento externo que es mantenido por el autor individual. La web del SVN de QGIS (<http://svn.qgis.org>) sirve algunos complementos aportados por usuarios.

#### 11.1.1. Encontrar e instalar un complemento

Cuando instala QGIS, todos los complementos integrados están incluidos (vea el capítulo 11.1.4).

De forma típica, los complementos aportado por usuarios se distribuyen en forma de código fuente y hay que compilarlos. Para instrucciones sobre la compilación e instalación de un complemento aportado por usuarios, vea la documentación incluida con el complemento.

#### 11.1.2. Administrar complementos

La administración de complementos consiste en cargarlos o descargarlos desde QGIS. Los complementos cargados se «recuerdan» cuando sale de la aplicación y son restaurados la siguiente vez que ejecuta QGIS.

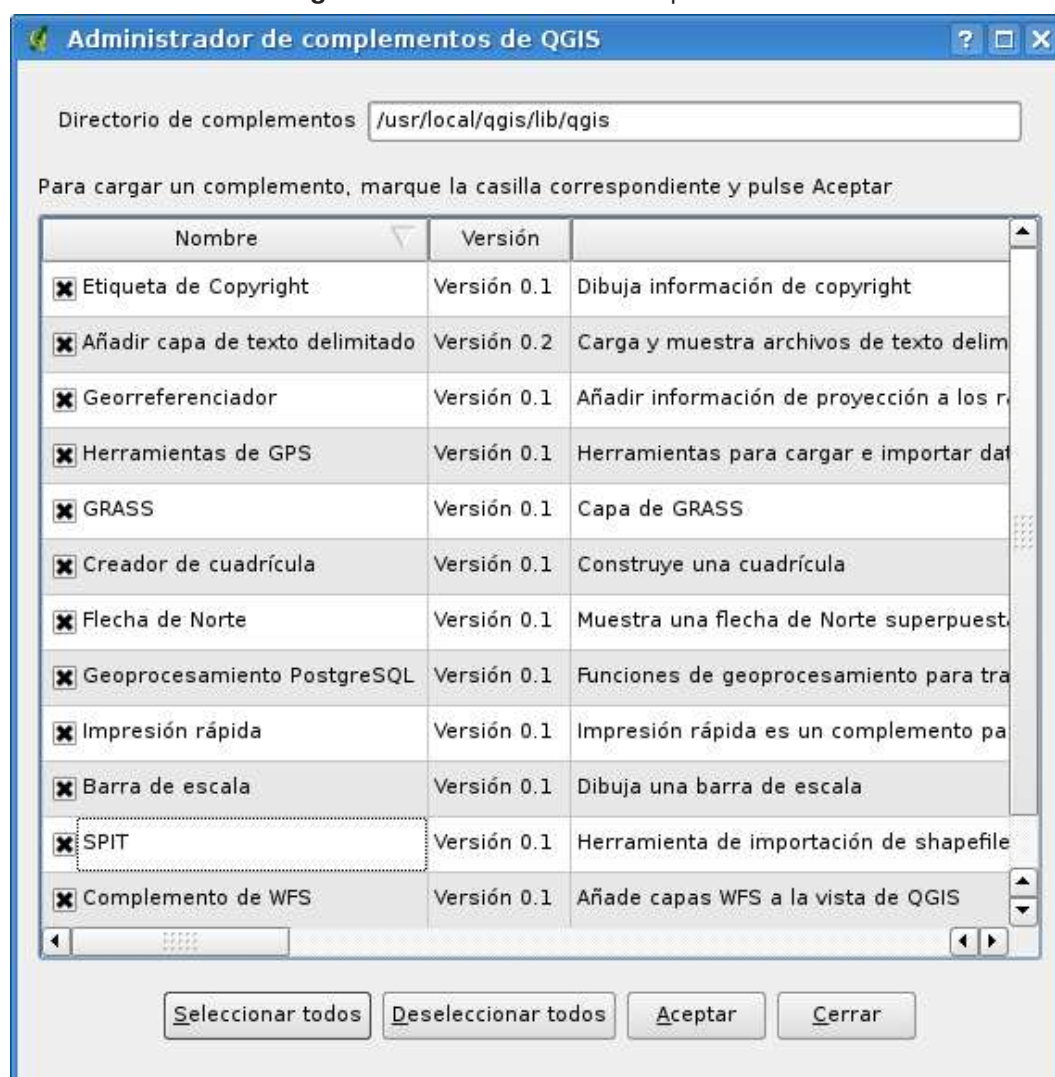
Para administrar complementos, abra el *Administrador de complementos* desde el menú *Complementos*. El Administrador de complementos muestra todos los complementos disponibles y su estado (cargados o no cargados). La Figura 23 muestra el diálogo del Administrador de complementos.

De forma típica todos los complementos de QGIS se instalan en la misma ubicación. Esta localización se muestra en el campo de texto Directorio de complementos. Le puede decir a QGIS que cargue complementos desde otra localización especificando un directorio distinto.

#### 11.1.3. Proveedores de datos

Los proveedores de datos son complementos «especiales» que proporcionan acceso a un almacén de datos. De forma predeterminada, QGIS soporta capas PostGIS y almacenes de datos basados

Figura 23: Administrador de complementos



en disco soportados por la biblioteca GDAL/OGR (Apéndice A.1). Un complemento de proveedor de datos amplía la capacidad de QGIS para usar otras fuentes de datos.

Los complementos de proveedores de datos son registrados automáticamente por QGIS al iniciarse. No son gestionados por el Administrador de complementos, pero se usan sin notarlo cuando el tipo de datos correspondiente se añade como capa en QGIS.











**Tip 34 COMPLEMENTOS QUE SE CUELGAN**

Si nota que QGIS se cuelga al iniciar, puede que haya un complemento que está dando problemas. Puede detener todos los complementos para que no se carguen editando su archivo de configuración guardado (vea 3.8 para su localización). Localice la configuración de los complementos y cambie todos los valores a false para evitar que se carguen. Por ejemplo, para evitar que se cargue el complemento Texto delimitado, la entrada en \$HOME/.config/QuantumGIS/qgis.conf en Linux debería ser como esta: `Add Delimited Text Layer=false`. Haga esto con cada complemento en la sección [Plugins]. Luego puede arrancar QGIS y añadir los complementos de uno en uno desde el Administrador de complementos para determinar cuál está ocasionando los problemas.

**11.1.4. Complementos integrados**

Actualmente QGIS contiene 9 complementos integrados que se pueden cargar usando el Administrador de complementos. La tabla 5 lista cada uno de los complementos integrados junto con una descripción de su propósito y el icono de la barra de herramientas. Note que el complemento de GRASS no está incluido abajo, porque éste instala su propia barra de herramientas (vea la sección 8 para ver en detalle las funciones disponibles en el complemento de GRASS).

**Cuadro 5:** Complementos integrados de QGIS

Icono	Complemento	Descripción
	Etiqueta de Copyright	Muestra una etiqueta de copyright sobre el lienzo del mapa
	Texto delimitado	Carga un archivo de texto delimitado que contenga coordenadas X e Y como una capa de puntos
	Herramientas de GPS	Carga y muestra datos de GPS
	Creador de cuadrículas	Crea una cuadrícula latitud/longitud y la guarda como archivo shape
	Barra de escala	Añade una barra de escala a la vista del mapa
	Flecha de Norte	Añade una flecha de Norte a la vista del mapa
	Geoprocesamiento PostgreSQL	Crea un área de influencia de una capa PostGIS
	SPIT	Herramienta de importación de archivos Shape a PostGIS - importa archivos shape a PostgreSQL
	Georreferenciador <sup>a</sup>	Georreferenciar capas ráster
	WFS	Carga y muestra capas WFS

<sup>a</sup>El complemento Georreferenciador sólo está disponible si tiene instaladas las bibliotecas y cabeceras de gsl durante la compilación. Por favor, revise el capítulo de instalación ?? para más detalles.

**Tip 35 CONFIGURACIÓN DE COMPLEMENTOS GUARDADA EN PROYECTOS**

Cuando guarda un proyecto .qgs, cualquier cambio que haya hecho en los complementos flecha de Norte, barra de escala y etiqueta de copyright se guardará en el proyecto y se restaurará la próxima vez que cargue el proyecto.

---

**11.1.5. Complementos externos**

QGIS también viene con algunos complementos desarrollados de forma externa. Éstos no están incluidos con la distribución predeterminada. Sin embargo, se pueden compilar y usar en QGIS.

Actualmente los complementos externos sólo están disponibles directamente desde SVN. Para comprobar todos los complementos externos disponibles haga lo siguiente:

```
svn co https://svn.osgeo.org/qgis/trunk/external_plugins external_qgis_plugins
```

Esto creará la carpeta `external_qgis_plugins` en su carpeta actual. Cada subdirectorio tiene sus propias instrucciones de compilación e instalación. Léalas detenidamente para instalar el complemento.

**11.1.6. Plantillas de complementos**

Si quiere desarrollar su propio complemento de QGIS las fuentes principales incluyen un buen script que le guía a través del proceso de crear su propia estructura de directorios de plantillas dentro del árbol de las fuentes de QGIS. El script se encuentra en `QGIS/src/plugins/plugin_builder.pl`.

Lo único que hay que hacer es poner el código de sus funciones dentro del complemento (y por supuesto aportar su complemento al equipo de desarrollo de QGIS).

Además de esto, el wiki de QGIS (<http://wiki.qgis.org>) y el blog de QGIS (<http://blog.qgis.org>) también proporcionan artículos útiles sobre la escritura de su propio complemento. ¡Visite las páginas web para detalles!

## 11.2. Usar los complementos decorativos

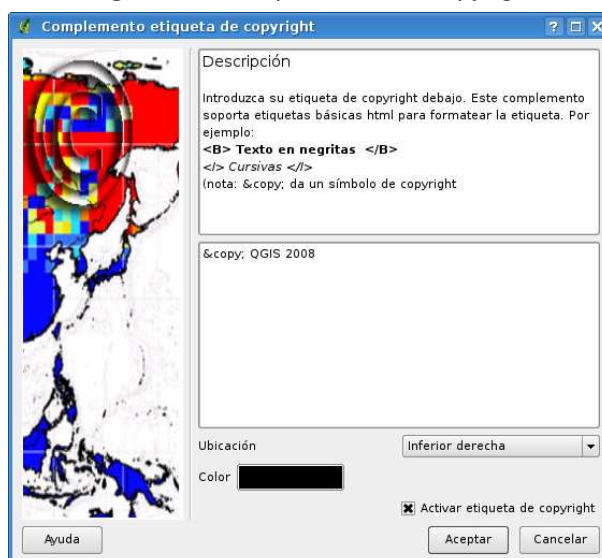
Los complementos “decorativos” incluyen lo siguiente:

- Complemento etiqueta de copyright.
- Complemento flecha de Norte.
- Complemento barra de escala.

“Decoran” el mapa añadiendo elementos cartográficos.

### 11.2.1. Complemento etiqueta de copyright

**Figura 24:** Complemento de copyright



El título de este complemento puede dar lugar a confusión, ya que puede añadir cualquier texto aleatorio al mapa.

1. Asegúrese de que el complemento está cargado.
2. Pulse en la herramienta *Etiqueta de Copyright* en la barra de herramientas de complementos.
3. Introduzca el texto que desee colocar en el mapa. Puede usar HTML como en el ejemplo mostrado.
4. Seleccione el emplazamiento de la etiqueta en la casilla desplegable.
5. Asegúrese de que la casilla de verificación “Activar etiqueta de copyright” está marcada.
6. Pulse *Aceptar*

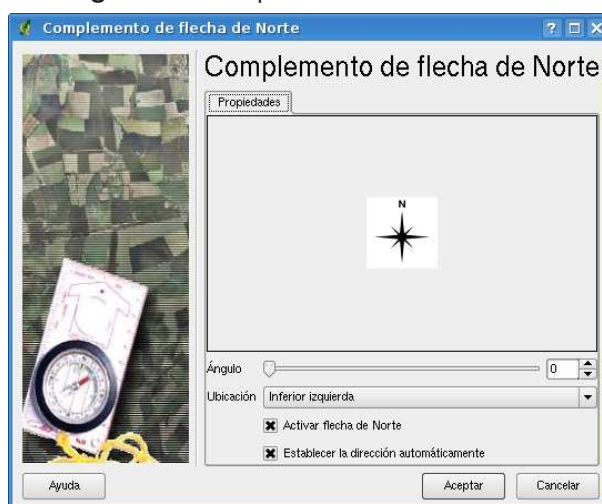
En el ejemplo anterior, la primera línea está en negrita, la segunda (creada usando <br>) contiene un símbolo de copyright, seguido por el nombre de nuestra compañía en cursiva.

### 11.2.2. Complemento flecha de Norte

El complemento flecha de Norte coloca una sencilla flecha de Norte sobre la vista del mapa. Actualmente sólo hay un estilo disponible. Puede ajustar el ángulo de la flecha o dejar que QGIS establezca la dirección automáticamente. Si elige dejar que QGIS determine la dirección, averiguará lo mejor posible cómo se debe orientar la flecha.

En cuanto a la ubicación de la flecha, tiene cuatro opciones, correspondientes a las cuatro esquinas de la vista del mapa.

**Figura 25:** Complemento flecha de Norte



### 11.2.3. Complemento barra de escala

El complemento barra de escala añade una barra de escala sencilla a la vista del mapa. Puede controlar el estilo y la ubicación, así como el etiquetado de la barra.

QGIS sólo puede mostrar la escala en las mismas unidades que tenga el mapa. Por tanto, si sus capas están en metros no puede crear una barra de escala en pies. Del mismo modo, si está usando grados decimales no puede crear una barra de escala para mostrar las distancias en metros.

Para añadir una barra de escala:

1. Abra el diálogo del complemento pulsando en la herramienta *Barra de escala* en la barra de herramientas de complementos.

2. Seleccione la ubicación en la lista desplegable.
3. Seleccione el estilo.
4. Seleccione el color de la barra o use el negro predeterminado.
5. Establezca el tamaño de la barra y su etiqueta.
6. Asegúrese de que la casilla de verificación “Activar barra de escala” está marcada.
7. Opcionalmente puede elegir redondear a un número exacto cuando se redimensiona la vista del mapa.
8. Pulse *Aceptar*

**Figura 26:** Complemento barra de escala





## 11.3. Usar el complemento GPS

### 11.3.1. ¿Qué es GPS?

GPS, el Sistema de Posicionamiento Global, es un sistema basado en satélites que permite a cualquiera con un receptor GPS conocer su posición exacta en cualquier parte del mundo. Se usa como ayuda en navegación, por ejemplo para aviones, en barcos y por excursionistas. El receptor GPS utiliza la señal de los satélites para calcular su latitud, longitud y (en ocasiones) altura. La mayoría de los receptores tiene también la capacidad de guardar localizaciones (conocidas como *waypoints*), secuencias de localizaciones que forman una *ruta* planeada y un registro de recorridos o *track* de los movimientos del receptor a lo largo del tiempo. Waypoints, rutas y tracks son los tres tipos de elementos básicos en los datos de GPS. QGIS muestra los waypoints en capas de puntos, mientras que las rutas y tracks se muestran en capas de líneas.

### 11.3.2. Cargar datos GPS de un archivo

Existen docenas de formatos de archivo diferentes para guardar datos de GPS. El formato utilizado por QGIS se llama GPX (GPS eXchange format-Formato de intercambio GPS), que es un formato estándar de intercambio que puede almacenar cualquier número de waypoints, rutas y tracks en el mismo archivo.



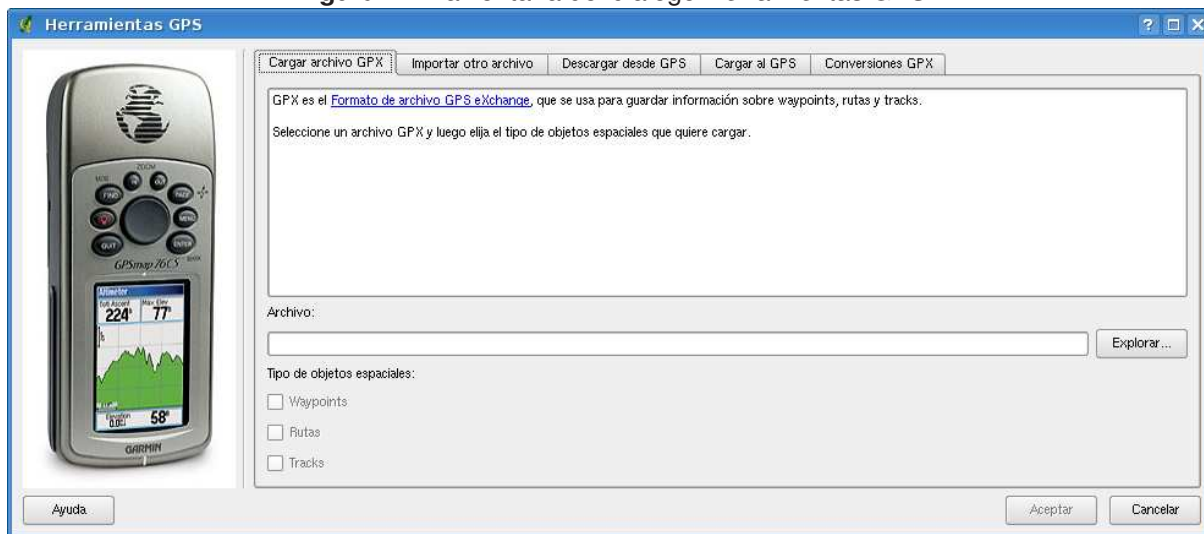
Para cargar un archivo GPX necesita usar el complemento *Herramientas GPS*. Cuando se carga este complemento aparece un botón con un pequeño dispositivo GPS de mano en la barra de herramientas (el dispositivo se parece un poco a un teléfono móvil). Pulsando en este botón se abrirá el diálogo *Herramientas GPS* (vea la Figura 27).

Utilice el botón [Explorar...] para seleccionar el archivo GPX y luego use las casillas de verificación para seleccionar el tipo de objeto espacial que quiere cargar de ese archivo GPX. Cada tipo de objeto espacial se cargará en una capa diferente cuando pulse Aceptar.

### 11.3.3. GPSTools

Puesto que QGIS utiliza archivos GPX, necesita una forma de convertir otros formatos de archivo de GPS a GPX. Esto lo puede hacer para muchos formatos usando el programa libre GPSTools, que está disponible en <http://www.gpsbabel.org>. Este programa también puede transferir datos de GPS entre su equipo y un dispositivo GPS. QGIS utiliza GPSTools para hacer estas cosas, por lo que se recomienda que lo instale. Sin embargo, si solamente quiere cargar datos de GPS desde archivos GPX no lo necesitará. La versión 1.2.3 de GPSTools se sabe que funciona con QGIS, pero debería poder usar versiones posteriores sin problemas.



Figura 27: La ventana del diálogo *Herramientas GPS*

#### 11.3.4. Importar datos de GPS

Para importar datos de GPS de un archivo que no sea GPX se usa la herramienta *Importar otro archivo* del diálogo *Herramientas GPS*. Aquí seleccione el archivo que quiera importar, el tipo de objeto espacial que quiera importar de él, dónde quiere guardar el archivo convertido a GPX y cuál debe ser el nombre de la nueva capa.

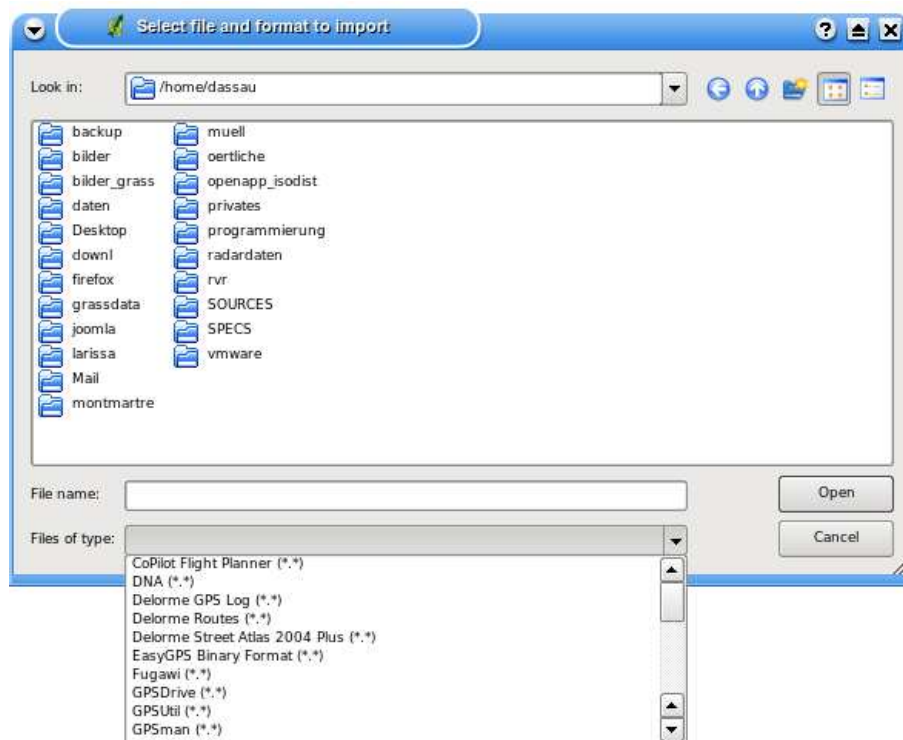
Cuando seleccione el archivo a importar también debe seleccionar el formato del archivo usando el menú del diálogo de selección de archivo (vea la figura 28). No todos los formatos soportan los tres tipos de objetos espaciales, por lo que para muchos formatos sólo podrá seleccionar uno o dos tipos.

#### 11.3.5. Descargar datos de GPS desde un dispositivo

QGIS puede usar GPSTabel para descargar datos desde un receptor GPS directamente a capas vectoriales. Para ello utilice la herramienta *Descargar desde GPS* (vea la Figura 29), donde se selecciona el tipo de dispositivo GPS, el puerto al que está conectado, el tipo de objeto espacial que quiere descargar, el archivo GPX donde se deben guardar los datos y el nombre de la nueva capa.

El tipo de dispositivo que seleccione en el menú de receptores GPS determina la forma en la que GPSTabel intenta comunicarse con él. Si no funciona ninguno de los tipos de dispositivo con su receptor GPS puede crear un nuevo tipo (vea la sección 11.3.7).

El puerto es un nombre de archivo u otro nombre que utilice su sistema operativo como referencia del puerto físico de su ordenador al que está conectado el dispositivo GPS. En Linux esto es algo como `/dev/ttyS0` o `/dev/ttyS1` y en Windows es COM1 o COM2.

**Figura 28:** Diálogo de selección de archivo de la herramienta de importación

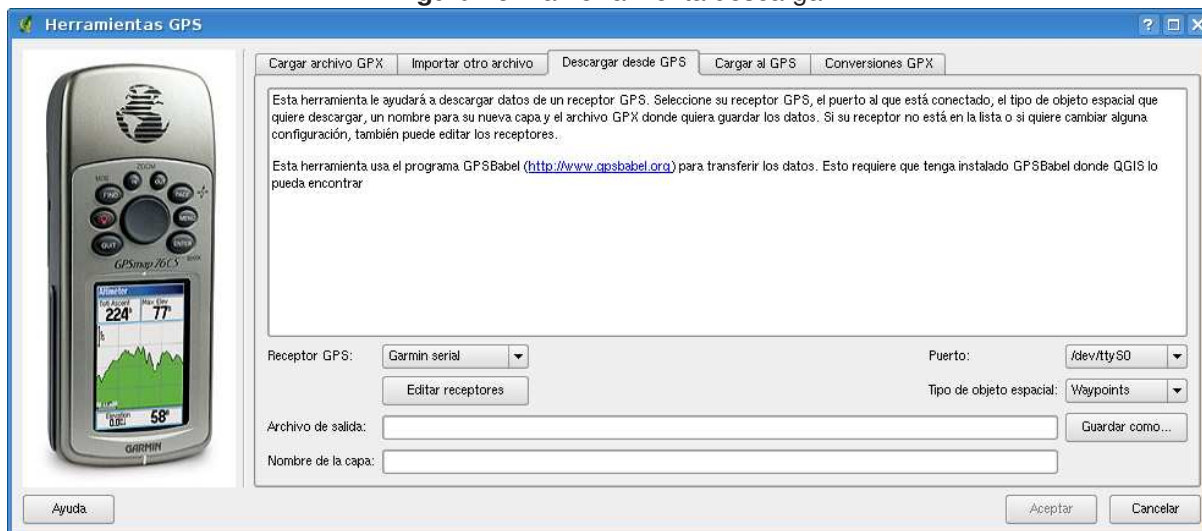
Cuando pulse Aceptar los datos se descargarán desde el receptor y aparecerán como una capa en QGIS.

### 11.3.6. Cargar datos de GPS a un dispositivo

También puede cargar datos directamente desde una capa vectorial de QGIS a un dispositivo GPS, usando la herramienta *Cargar a GPS*. La capa debe ser una capa GPX. Para hacer esto simplemente seleccione la capa que quiera cargar, el tipo de su dispositivo GPS y el puerto al que esté conectado. Igual que con la herramienta descargar, puede especificar nuevos tipos de dispositivo si el suyo no está en la lista.

Esta herramienta es muy útil junto con las capacidades de edición de capas vectoriales de QGIS. Puede cargar una mapa, crear algunos waypoints y rutas y luego cargarlos y usarlos en su dispositivo GPS.

Figura 29: La herramienta descargar



### 11.3.7. Definir nuevos tipos de dispositivo

Existen numerosos tipos distintos de receptores GPS. Los desarrolladores de QGIS no puede probarlos todos, por lo que si tiene uno que no funciona con ninguno de los tipos listados en las herramientas de descarga y carga puede definir su propio tipo. Esto se hace usando el *editor de receptores GPS*, que puede iniciar pulsando el botón *Editar receptores* en las ventanas de descarga o carga.

Para definir un receptor nuevo simplemente pulse el botón *Nuevo receptor*, introduzca un nombre, una orden de descarga y una orden de carga para su dispositivo y pulse el botón *Actualizar receptor*. El nombre, que puede ser cualquier cadena, aparecerá en la lista de receptores en las ventanas de carga y descarga.

La orden de descarga es la que se usa para descargar datos desde un receptor a un archivo GPX. Probablemente será una orden de GPSTools, pero puede usar cualquier otro programa de línea de órdenes que pueda crear un archivo GPX. QGIS sustituirá las palabras clave *%type*, *%in* y *%out* cuando ejecute la orden.

*%type* se sustituirá por “-w” si está descargando waypoints, por “-r” si está descargando rutas y por “-t” si está descargando tracks. Estas son opciones de línea de órdenes que le dicen a GPSTools qué tipos de objetos espaciales descargar.

*%in* se sustituirá por el nombre del puerto que elija en la ventana de descarga y *%out* se sustituirá por el nombre que elija para el archivo GPX en el que se guardarán los datos descargados. Así, si crea un tipo de dispositivo con la orden de descarga “gpsbabel *%type* -i garmin -o gpx *%in* *%out*” (esta es en realidad la orden de descarga para el tipo de receptor predefinido “Garmin serie”) y lo usa para descargar waypoints del puerto “/dev/ttyS0” al archivo “output.gpx”, QGIS sustituirá las palabras

clave y ejecutará la orden “gpsbabel -w -i garmin -o gpx /dev/ttyS0 output.gpx”.

La orden de carga es la que se usa para cargar datos al receptor. Se utilizan las mismas palabras clave, pero *%in* ahora se sustituye por el nombre del archivo GPX de la capa que se está cargando y *%out* se sustituye por el nombre del puerto. Puede aprender más sobre GPSTools y sus opciones de línea de órdenes en <http://www.gpsbabel.org>

Una vez que haya creado un tipo de receptor nuevo, aparecerá en la lista de dispositivos de las herramientas de descarga y carga.

## 11.4. Usar el complemento de texto delimitado

El complemento de texto delimitado le permite cargar un archivo de texto delimitado como una capa en QGIS.

### 11.4.1. Requisitos

Para ver un archivo de texto delimitado como una capa, el archivo de texto debe contener:

1. Una fila de encabezado delimitado con los nombres de los campos. Debe ser la primera línea del archivo de texto.
2. La fila de encabezado debe contener un campo X e Y. Estos campos se pueden llamar de cualquier forma.
3. Las coordenadas X e Y deben estar especificadas como números. El sistema de coordenadas no es importante.

Un ejemplo de un archivo de texto válido tendría este aspecto:

```
nombre|latdec|longdec|celda|
196 mile creek|61.89806|-150.0775|tyonek d-1 ne|
197 1/2 mile creek|61.89472|-150.09972|tyonek d-1 ne|
a b mountain|59.52889|-135.28333|skagway c-1 sw|
apw dam number 2|60.53|-145.75167|cordova c-5 sw|
apw reservoir|60.53167|-145.75333|cordova c-5 sw|
apw reservoir|60.53|-145.75167|cordova c-5 sw|
aaron creek|56.37861|-131.96556|bradfield canal b-6|
aaron island|58.43778|-134.81944|juneau b-3 ne|
aats bay|55.905|-134.24639|craig d-7|
```

Algunos elementos a tener en cuenta sobre los archivos de texto son:

1. El archivo de texto de ejemplo usa | como delimitador. Se puede usar cualquier carácter para delimitar los campos.
2. La primera fila es la fila de encabezado. Contiene los campos nombre, latdec, longdec y celda.
3. No se usan comillas (") para delimitar campos de texto.
4. Las coordenadas X están en el campo *longdec*.
5. Las coordenadas Y están en el campo *latdec*.

### 11.4.2. Usar el complemento

Para usar el complemento debe estar ejecutando QGIS y usar el Administrador de complementos para cargarlo:

Inicie QGIS y abra el Administrador de complementos seleccionando el menú *Complementos|Administrador de complementos*. El Administrador de complementos muestra una lista de todos los complementos disponibles. Aquellos que se encuentren ya cargados tendrán una marca a la izquierda del nombre. Marque la casilla que hay a la izquierda del complemento *Añadir capa de texto delimitado* y pulse Aceptar para cargarlo como se describe en la Sección 11.1.2.


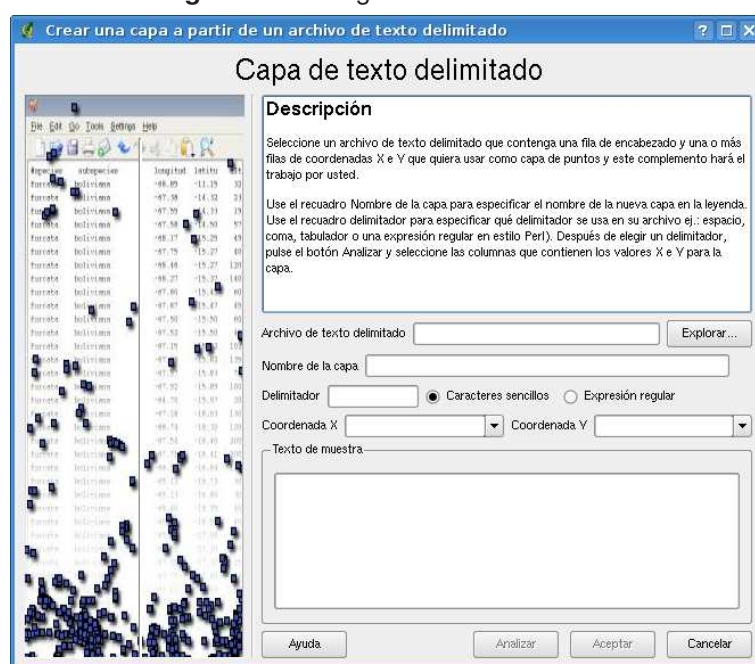
Ahora habrá un nuevo icono en la barra de herramientas:  Pulse el icono para abrir el diálogo de texto delimitado, tal como aparece en la Figura 30.

Figura 30: Diálogo de texto delimitado

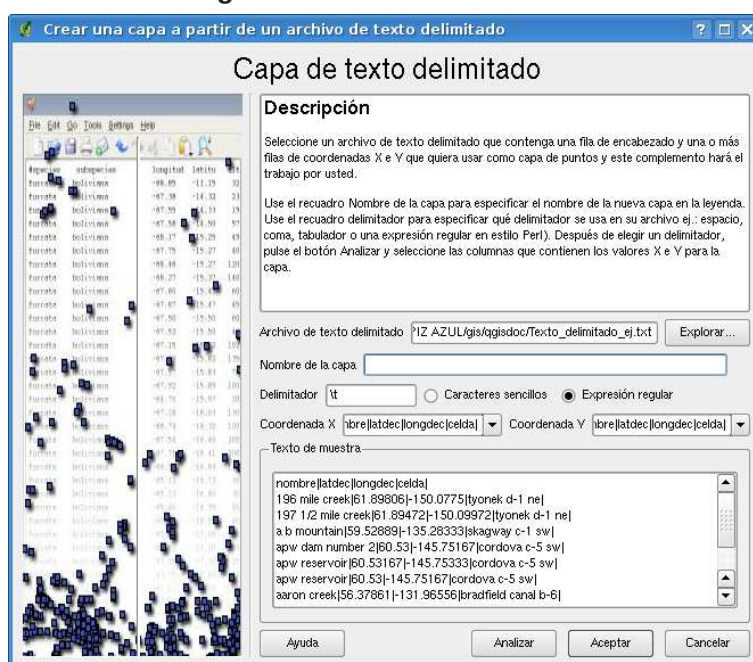


Primero seleccione el archivo a importar pulsando en el botón *Explorar....* Seleccione el archivo de texto deseado en el diálogo de archivos. Una vez seleccionado, el complemento intenta analizar el archivo usando el último delimitador empleado, en este caso | (vea la Figura 31).

En este caso el delimitador | no es correcto para el archivo, que en realidad está delimitado con tabulador. Vea que los campos desplegables X e Y no contienen nombre de campo válidos.

Para analizar correctamente el archivo, cambie el delimitador a tabulador, usando \t (esta es una expresión regular para el carácter tabulador). Después de cambiar el delimitador, pulse *Analizar*. Las

Figura 31: Archivo seleccionado



casillas desplegables ahora contendrá los campos analizados correctamente como se muestra en la Figura 32.

Seleccione los campos X e Y de los cuadros desplegables e introduzca un nombre de capa como se muestra en la Figura 33. Para añadir la capa al mapa pulse *Aceptar*. El archivo de texto delimitado ahora funcionará como cualquier otra capa de mapas en QGIS.



Figura 32: Campos analizados del archivo de texto

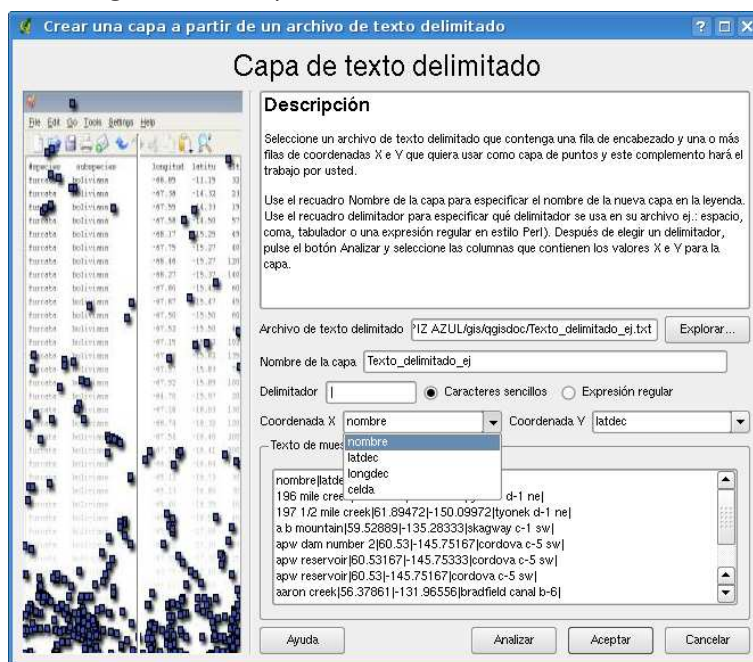
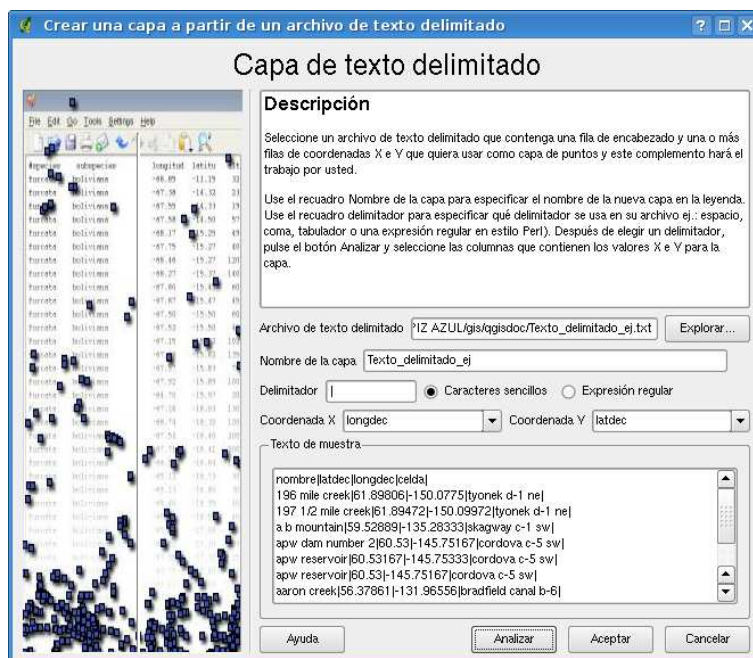


Figura 33: Seleccionar los campos X e Y





## 11.5. Usar el complemento generador de cuadrículas

El generador de cuadrículas permite crear una «malla» de puntos, líneas o polígonos que cubra nuestro área de interés. Se deben introducir todas las unidades en grados decimales. La salida es un archivo shape que se puede proyectar al vuelo para ajustarlo a sus otros datos.

**Figura 34:** Crear una capa de cuadrícula



Aquí hay un ejemplo de cómo crear una cuadrícula:

1. Asegúrese de que el complemento está cargado.
2. Pulse en la herramienta *Generador de cuadrículas* en la barra de herramientas de complementos.
3. Seleccione el tipo de cuadrícula que quiera crear: punto, línea o polígono.
4. Introduzca la latitud y la longitud para las esquinas inferior izquierda y superior derecha de la cuadrícula.
5. Introduzca el intervalo a usar para construir la cuadrícula. Puede introducir valores diferentes para las direcciones X e Y (longitud, latitud).
6. Seleccione el nombre y la localización del archivo shape a crear.

7. Pulse *Aceptar* para crear la cuadrícula y añadirla a la vista del mapa.

## 11.6. Usar el complemento georreferenciador

El complemento georreferenciador permite generar archivos de referenciación (world files) para rás-  
ters. Para ello se seleccionan puntos en el ráster y se añaden sus coordenadas y el complemento  
procesa los parámetros del archivo de referenciación. Cuantas más coordenadas se proporcionen  
mejor será el resultado.

Como ejemplo generaremos un archivo de referenciación para una hoja topográfica de Dakota del  
Sur desde SDGS. Más tarde se puede visualizar junto con los datos de la localización spearfish60  
de GRASS. Puede descargar la hoja topográfica aquí:

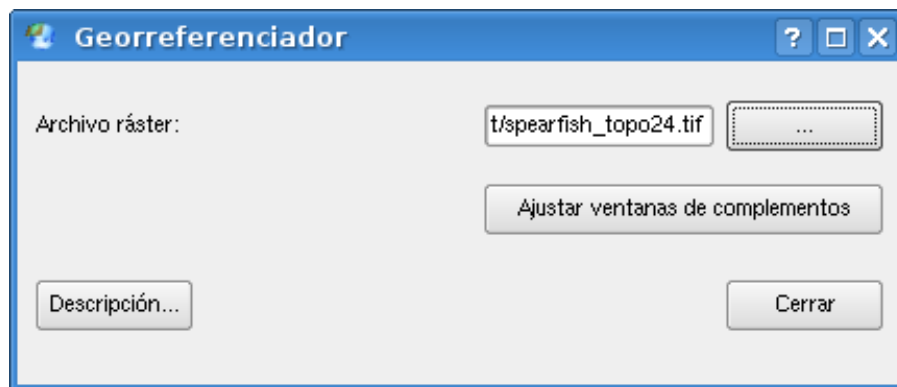
[http://grass.osgeo.org/sampleddata/spearfish\\_toposheet.tar.gz](http://grass.osgeo.org/sampleddata/spearfish_toposheet.tar.gz)

Como primer paso descargamos el archivo y los descomprimos.

```
wget http://grass.osgeo.org/sampleddata/spearfish_toposheet.tar.gz
tar xvzf spearfish_toposheet.tar.gz
cd spearfish_toposheet
```

El siguiente paso es iniciar QGIS, cargar el complemento georreferenciador y seleccionar el archivo  
spearfish\_topo24.tif.

**Figura 35:** Seleccionar una imagen a georreferenciar

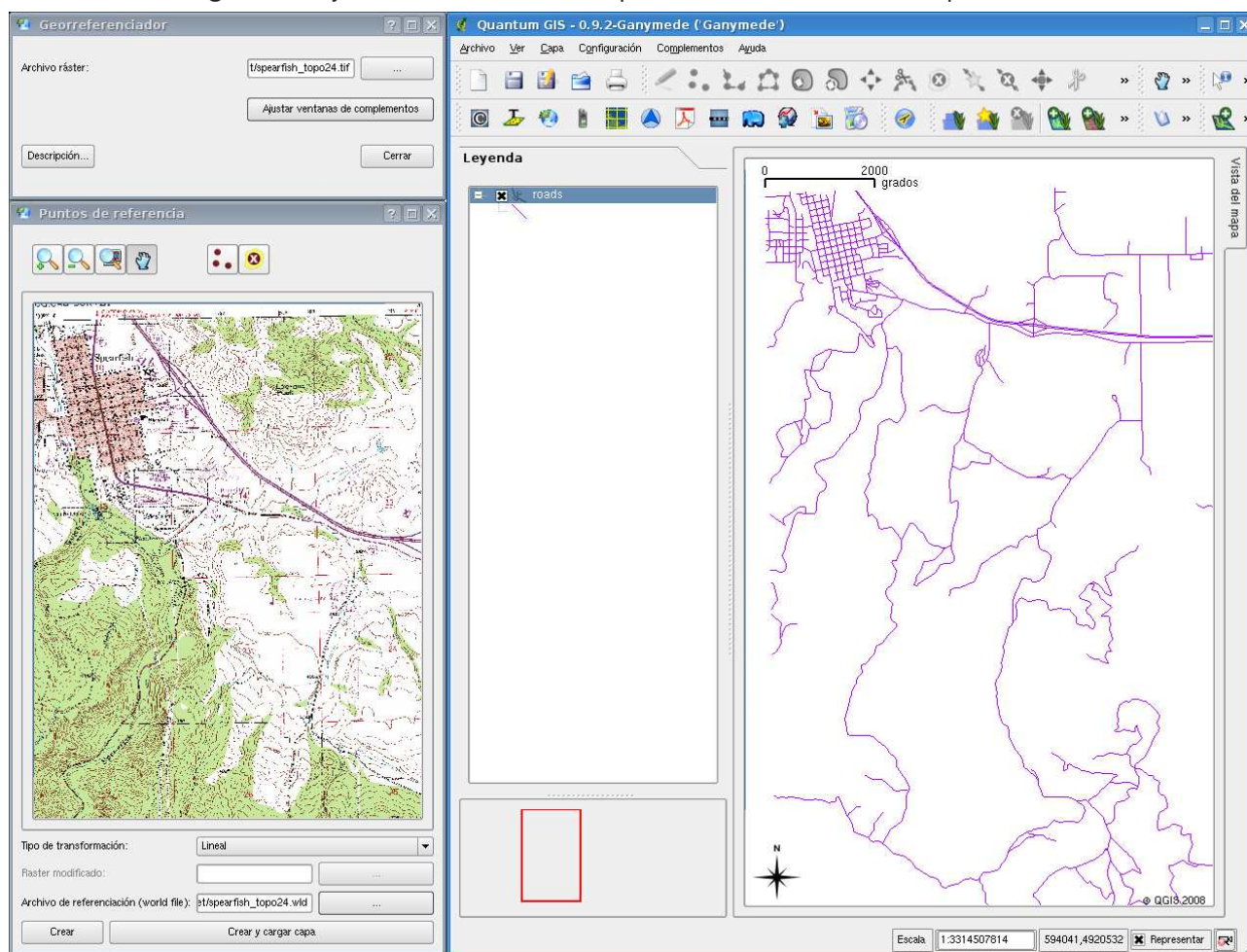


Ahora pulse el botón *Ajustar ventanas de complementos* para abrir la imagen en el georreferenciador  
y ajustarla en su escritorio con la vista del mapa de referencia en QGIS.

Con el botón *Añadir punto* puede comenzar a añadir puntos en la imagen ráster e introducir sus  
coordenadas y el complemento procesará los parámetros del archivo de referenciación (vea la Fi-  
gura 37). Cuantas más coordenadas proporcione mejor será el resultado. Hay dos opciones para el  
procedimiento:

1. Pulse en un punto del mapa ráster e introduzca las coordenadas X e Y manualmente.

Figura 36: Ajustar la ventana del complemento con la vista del mapa de QGIS



2. Pulse en un punto en el mapa ráster y seleccione el botón *de la vista del mapa* para añadir las coordenadas X e Y con la ayuda de un mapa georreferenciado ya cargado en QGIS.

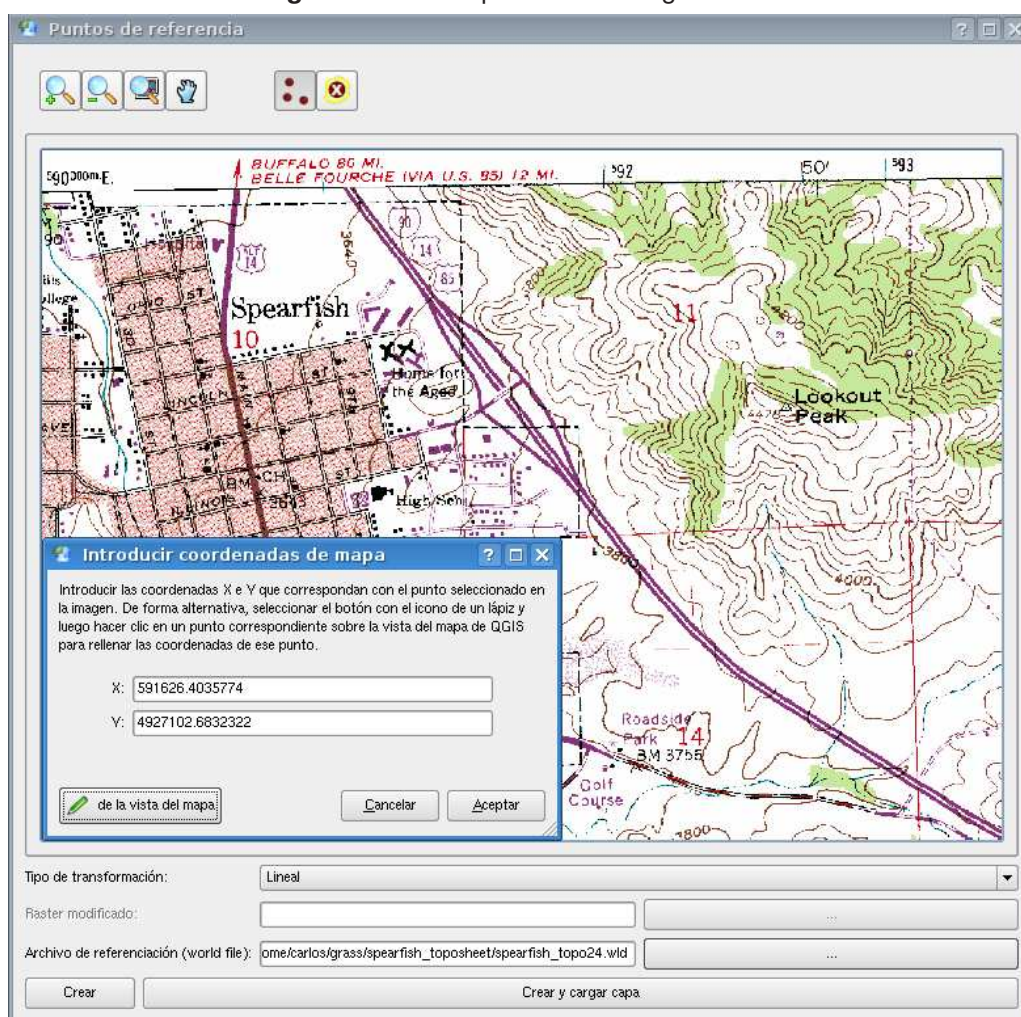
Para este ejemplo usamos la segunda opción e introducimos las coordenadas de los puntos seleccionados con la ayuda del mapa *roads* proporcionado con la localización *spearfish60* de: [http://grass.osgeo.org/sampled/data/spearfish\\_grass60data-0.3.tar.gz](http://grass.osgeo.org/sampled/data/spearfish_grass60data-0.3.tar.gz)

Si no sabe como integrar la localización *spearfish60* con el complemento de GRASS, se da información en la Sección 8.

Como puede ver en la Figura 37, el georreferenciador proporciona botones para hacer *zum*, *panorámica*, *añadir* y *borrar* puntos en la imagen.

Después de añadir suficientes puntos a la imagen necesita seleccionar el tipo de transformación para el proceso de georreferenciación y guardar el archivo de referenciación resultante junto con el

Figura 37: Añadir puntos a la imagen ráster



Tiff. En nuestro ejemplo elegimos transformación lineal, aunque una transformación Helmert también podría valer.

### Tip 36 SELECCIONAR EL TIPO DE TRANSFORMACIÓN

La transformación lineal (afín) es una transformación de primer orden y se usa para el escalado, translación y rotación de imágenes geoméricamente correctas. Con la transformación Helmert simplemente se añade información de coordenadas a la imagen como geocódigo. Si su imagen está retorcida necesitará usar software que proporcione transformaciones polinomiales de segundo y tercer orden, por ejemplo GRASS GIS.

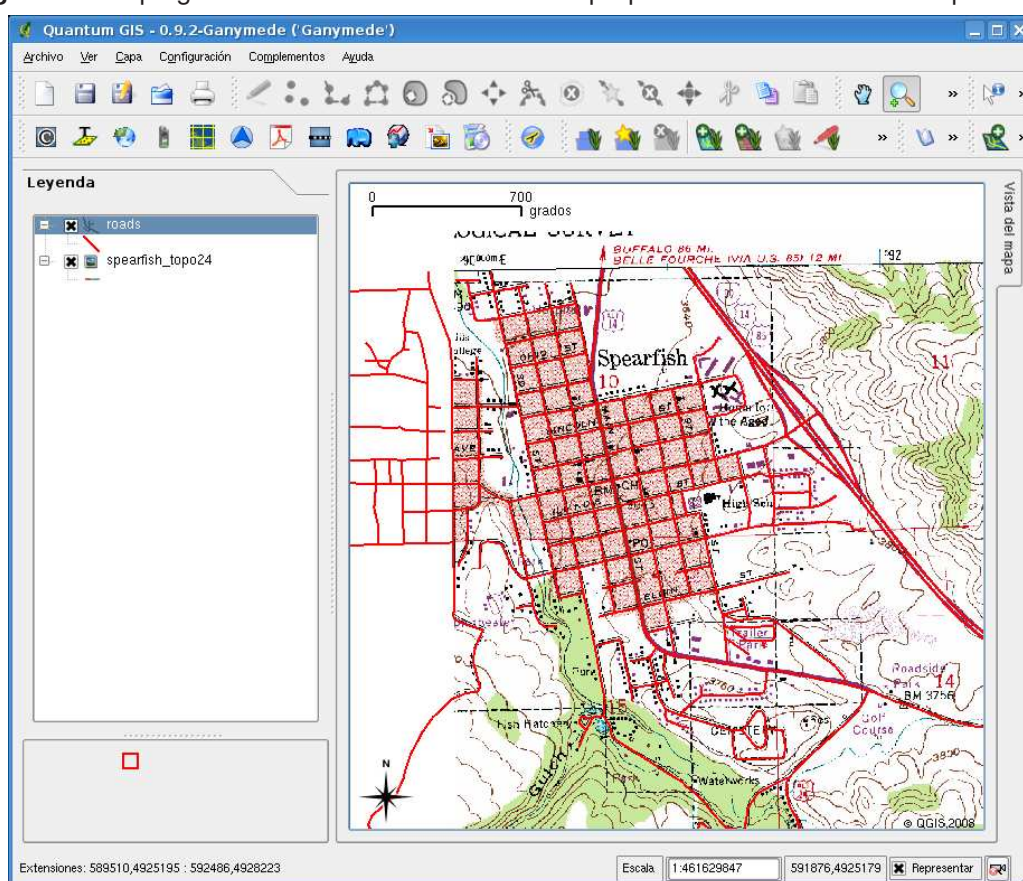
Los puntos que añadimos al mapa se guardarán en un archivo *spearfish\_topo24.tif.points* junto con la imagen ráster. Esto nos permite volver a abrir el complemento de georreferenciación y añadir puntos nuevos o borrar existentes para optimizar el resultado. El archivo *spearfish\_topo24.tif.points* de este ejemplo muestra los puntos:



mapX	mapY	pixelX	pixelY
591630.196867999969982	4927104.309682800434530	591647	4.9271e+06
608453.589164100005291	4924878.995150799863040	608458	4.92487e+06
602554.903929700027220	4915579.220743400044739	602549	4.91556e+06
591511.138448899961077	4915952.302661700174212	591563	4.91593e+06
602649.526155399973504	4919088.353569299913943	602618	4.91907e+06

Hemos usado 5 puntos para georreferenciar la imagen ráster. Para conseguir resultados correctos es importante separar los puntos de forma regular por la imagen. Finalmente comprobamos el resultado y cargamos el nuevo mapa georreferenciado *spearfish\_topo24.tif* y lo superponemos con el mapa *roads* de la localización spearfish60.

**Figura 38:** Mapa georreferenciado con carreteras superpuestas de la localización spearfish60



## 11.7. Usar el complemento de Python

Escribir complementos en Python es mucho más sencillo que usar C++. Para crear un complemento de PyQGIS, necesita QGIS 0.9, Python, PyQt y las herramientas de desarrollo de Qt ?.

Cuando QGIS arranca escanea ciertos directorios en busca de complementos tanto de C++ como de Python. Para que un archivo (biblioteca compartida, DLL o script de python) sea reconocido como complemento tiene que tener una firma específica. Para los scripts de Python es bastante sencillo. QGIS busca en las siguientes localizaciones dentro del directorio de instalación:

- **Linux y otros Unix:** `./share/qgis/python/plugins`
- **Mac OS X:** `./Contents/MacOS/share/qgis/python/plugins`
- **Windows:** `.\share\QGIS\python\plugins`

Cada complemento de Python está contenido en su propio directorio. Cuando QGIS arranca busca en cada subdirectorio en *share/qgis/python/plugins* e inicializa cada complemento que encuentra. Una vez que esto está hecho, el complemento se mostrará en el administrador de complementos.

Vamos a crear un complemento para rellenar un hueco en la interfaz de QGIS. Esta complemento nos permitirá crear una nueva capa PostGIS para que la digitalicemos. Será un complemento sencillo y bastante burdo, pero ilustrará como iniciarse para escribir sus propios complementos de PyQGIS.

### 11.7.1. Configurar la estructura

Lo primero que necesitamos hacer es configurar la estructura de nuestro complemento. En este ejemplo desarrollaremos nuestro complemento en Linux, pero el método es el mismo, solo que adaptado a las órdenes del sistema de archivos adecuados a su plataforma. QGIS está instalado en un directorio llamado *qgis\_09* dentro de su directorio personal. Vamos a crear el directorio para el complemento.

```
mkdir ~/qgis_09/share/qgis/python/plugins/capa_nueva
```

Para empezar, necesitamos crear los siguientes archivos en el directorio *capa\_nueva* (necesitaremos algunos archivos adicionales dentro de poco):

```
__init__.py
recursos.py
recursos.qrc
capanueva.py
```

### 11.7.2. Hacer reconocible el complemento

La inicialización del complemento se hace en el script `__init__.py`. Para nuestro complemento *Capa Nueva* el script contiene:

```
1 # Cargar la clase CapaNueva desde el archivo capanueva.py
2 from capanueva import CapaNueva
3 def name():
4     return "Nueva capa PostGIS"
5 def description():
6     return "Crea una capa nueva Postgis vacía"
7 def version():
8     return "Version 0.1"
9 def classFactory(iface):
10     return CapaNueva(iface)
```

Las cosas que un complemento debe devolver de forma imperativa son un nombre, descripción y versión, todo lo cual está implementado en nuestro script de arriba. Cada método simplemente devuelve una cadena con la información adecuada. El otro requisito es el método *classFactory* que debe devolver una referencia al propio complemento (línea 10), después de recibir el objeto *iface* como argumento. Con este sencillo código QGIS reconocerá nuestro script como un complemento.

### 11.7.3. Recursos

Para poder tener un bonito icono para nuestro complemento, necesitamos un archivo de recursos al que llamaremos *recursos.qrc*. Se trata de un sencillo archivo XML que define el recurso del icono:

```
<RCC>
  <qresource prefix="/plugins/capanueva">
    <file>icono.png</file>
  </qresource>
</RCC>
```

El archivo de recursos usa un prefijo para evitar conflictos con los nombres de otros complementos. Usar el nombre del complemento normalmente es suficiente. El archivo *icono.png* es simplemente una imagen PNG que se usará en la barra de herramientas cuando se active el complemento. Puede usar cualquier imagen con tal de que tenga 22x22 píxeles (para que se ajuste a la barra de herramientas).

Para convertir el archivo de recursos en algo que el complemento pueda usar, se debe compilar usando el compilador de recursos de PyQt:



```
pyrcc4 -o recursos.py recursos.qrc
```

El modificador `-o` se usa para especificar el archivo de salida. Ahora que tenemos recursos, necesitamos una forma de recoger la información necesaria para crear una capa nueva.

#### 11.7.4. Crear la interfaz gráfica de usuario (GUI)

Normalmente usaríamos la misma herramienta que usan los desarrolladores de C++ para crear una GUI: Qt Designer. Se trata de una herramienta de diseño visual que le permite crear ventanas de diálogos y principales cogiendo y arrastrando herramientas y definiendo sus propiedades.

Para diseñar nuestro complemento CapaNueva podríamos conseguir herramientas bastante entretenidas e integradas para los tipos de campo y otras opciones. Sin embargo, puesto que nuestro tiempo es limitado, usaremos otros medios para recopilar la información que necesitamos para crear la tabla. Esto ilustrará los conceptos y luego se podrá profundizar más usando los manuales del blog de QGIS.

Para recopilar la entrada del usuario, usaremos la clase *QInputDialog* de la biblioteca Qt. Esta pedirá al usuario una línea simple de entrada. Aunque esto hará nuestro complemento un poco crudo, servirá para ilustrar los conceptos.

Todo lo que tenemos que escribir ahora es el código de Python para recopilar la entrada y crear la tabla.

#### 11.7.5. Crear el complemento

Una vez que tenemos los preliminares preparados podemos centrarnos en escribir el código que hará el trabajo real. Empezemos por mirar las cosas que necesitamos importar y la inicialización del complemento en *capanueva.py*.

```
1 # Importar las bibliotecas de PyQt y QGIS
2 from PyQt4.QtCore import *
3 from PyQt4.QtGui import *
4 from qgis.core import *
5 import psychopg
6 # Inicializar los recursos de Qt del archivo recursos.py
7 import recursos
8
9 # Nuestra clase principal para el complemento
10 class CapaNueva:
11
```

```

12 def __init__(self, iface):
13     # Guardar la referencia para la interfaz de QGIS
14     self.iface = iface
15
16 def initGui(self):
17     # Crear la acción que iniciará la configuración del complemento
18     self.action = QAction(QIcon(":/plugins/capa_nueva/icono.png"),\
19         "Nueva capa PostGIS", self.iface.getMainWindow())
20     QObject.connect(self.action, SIGNAL("activated()"), self.run)
21
22     # Añadir botón a la barra de herramientas y entrada del menú
23     self.iface.addToolBarIcon(self.action)
24     self.iface.addPluginMenu("&Nueva capa PostGIS...", self.action)
25
26 def unload(self):
27     # Eliminar la entrada del menú y el icono del complemento
28     self.iface.removePluginMenu("&Nueva capa PostGIS...",self.action)
29     self.iface.removeToolBarIcon(self.action)

```

En las líneas 2 a 7 importamos las bibliotecas necesarias para el complemento. Esto incluye las bibliotecas de PyQt, la biblioteca principal de QGIS y la biblioteca de Python PostgreSQL `psycopg`. Cada script de Python que use las bibliotecas de QGIS y PyQt necesita importar las bibliotecas de `QtCore` y `QtGui`, así como la biblioteca principal de QGIS. Esto nos da acceso a los PyQt wrappers para nuestros objetos de Qt (como nuestro diálogo de entrada) y la biblioteca principal de QGIS. También necesitamos importar el archivo `recursos.py` que creamos con la definición del icono.

En la línea 10 declaramos la clase **CapaNueva**. En el método `__init__` (líneas 12 a 14) se inicializa nuestra clase y se pasa el objeto **iface** de QGIS via el método `classFactory` en la línea 10 de `__init__.py`. Guardamos **iface** como una variable de miembro, de forma que la podamos usar después.

En las líneas 16 a 24 inicializamos los elementos de la GUI para el complemento. En Qt se usa una **QAction** para crear una acción de la interfaz de usuario que se puede usar para crear tanto un elemento de menú como de la barra de herramientas. En nuestro complemento lo usamos para ambas cosas. En la línea 18 creamos la acción usando nuestro recurso de icono (observe el prefijo que hemos especificado en `recursos.qrc`). También proporcionamos algo de texto que aparecerá cuando se use en un menú o al pasar el ratón por encima y finalmente necesitamos especificar el “padre”. En un complemento, el padre es la ventana principal de QGIS. El objeto **iface** que guardamos durante la inicialización nos permite obtener la referencia a la ventana principal en la línea 19.

Una vez que la acción está creada, podemos añadirla tanto a la barra de herramientas como al menú de *Complementos* (líneas 23 y 24). Esto se encarga de inicializar la GUI para el complemento. La otra cosa que necesitamos es hacer limpieza detrás nuestra cuando se descarga el complemento. El método `unload` se encarga de esto eliminando la entrada del menú y la herramienta de la barra

de herramientas (líneas 28 y 29).

Esto se encarga del proceso de inicialización y de que nuestro complemento se cargue y descargue correctamente. Miremos ahora al código que hace el verdadero trabajo. Todo se encuentra en el método *run*.

```

30 def run(self):
31     # Obtener la entrada del usuario, comenzando por el nombre de la tabla
32     table_name = QDialog.getText(None, "¿Nombre de la tabla?", \
33         "Nombre para la nueva capa PostGIS")
34     if table_name[0].length() > 0:
35         # Obtener los nombres y tipos de los campos
36         fields = QDialog.getText(None, "Nombres de los campos", \
37             "Campos (separados por coma)")
38         parts = fields[0].split(',')
39         # Crear la sentencia SQL
40         sql = "create table " + table_name[0] + " (id int4 primary key, "
41         for fld in parts:
42             sql += fld + " varchar(10), "
43         sql = sql[0:-2]
44         sql += ")"
45         # Conectar con la base de datos
46         # Primero obtener la DSN
47         dsn = QDialog.getText(None, "DSN de la base de datos", \
48             "Introducir la DSN para conectar con la base de datos (dbname=db user=user)")
49         if dsn[0].length() > 0:
50             con = psycopg.connect(str(dsn[0]))
51             curs = con.cursor()
52             curs.execute(str(sql))
53             con.commit()
54             # añadir la columna de la geometría
55             curs.execute("select AddGeometryColumn(' " + str(table_name[0]) + \
56                 "', 'the_geom', 4326, 'POLYGON', 2)")
57             con.commit()
58             # crear el índice GIST
59             curs.execute("create index idx_" + str(table_name[0]) + " on " + \
60                 str(table_name[0]) + " USING GIST(the_geom GIST_GEOMETRY_OPS)")
61             con.commit()

```

Lo primero que tenemos que hacer es usar **QInputDialog** para obtener el nombre de la tabla a crear. Esto se hace en la línea 32, en la que se pide el nombre.

En la línea 34 comprobamos que el usuario realmente ha introducido algo antes de continuar.

**Figura 39:** Introducir nombre de la nueva tabla PostGIS

A continuación necesitamos obtener los nombres de los campos. Para este ejemplo lo vamos a hacer muy simple. Cada campo será un varchar(10), lo que significa que puede almacenar hasta 10 caracteres. Si realmente queremos hacer útil este complemento, necesitaremos proporcionar una forma de que el usuario especifique el tipo. En la línea 36 pedimos al usuario que introduzca una lista de nombres de campos separados por comas.

**Figura 40:** Introducir nombres de campos para la nueva tabla PostGIS

Luego dividimos esta lista en sus componentes para usarla en la construcción de la sentencia (línea 38).

La línea 40 contiene la primera parte de la sentencia SQL. Observe que estamos creando la tabla con un campo ID entero que será la clave primaria. A continuación iteramos por la lista de campos, añadiendo el código adecuado a la sentencia SQL (línea 41).

Una vez que tenemos todos los campos añadidos a la sentencia SQL, truncamos los caracteres de trailing que no queremos (línea 43) y luego añadimos el paréntesis de cierre para completar la sentencia (línea 44).

Ahora ya estamos preparados para conectar a la base de datos y crear la tabla. Para acceder a la base de datos usamos pycopg (<http://www.initd.org>). Para poder conectar a la base de datos tenemos que especificar el nombre de la fuente de datos (DSN-Data Source Name) con el nombre de la base de datos, usuario y contraseña si es necesario. Si estamos ejecutando tanto QGIS como PostgreSQL en la misma máquina normalmente no es necesario especificar una contraseña. En este

caso, el DSN tendrá un aspecto parecido a esto:

*dbname=gis\_data user=gsherman*

Para obtener el DSN, preguntamos a usuario con un **QInputDialog** en la línea 47.

**Figura 41:** Introducir DSN para la conexión a la base de datos PostGIS



Si el usuario introduce un DSN podemos continuar con la conexión a la base de datos en la línea 50. Obtenemos el cursor desde la conexión en la línea 51 y luego ejecutamos la sentencia SQL para crear la tabla y remitir el cambio en las líneas 52 a 53. Esto crea la tabla, pero para que ésta sea una capa válida y lista para usarla, necesita un par de cosas más.

Primero necesita una columna de geometría. No hemos introducido una a propósito cuando creamos la tabla para poder usar la función *AddGeometryColumn* para crearla. Esta función añade una columna de geometría y luego pone una entrada en la tabla *geometry\_columns* por nosotros. En la línea 55 especificamos el nombre de la tabla, el nombre que queremos para la columna de geometría, el SRID, tipo de objeto espacial y sus dimensiones.

Lo último que hay que hacer es crear el índice espacial en la tabla, de forma que obtengamos un funcionamiento óptimo cuando hagamos búsquedas espaciales y mostremos los datos en QGIS. En la línea 59 hemos apañado junta la SQL para crear el índice. La sentencia real es la siguiente:

```
create index sidx_park_land on park_land
  USING GIST(the_geom GIST_GEOMETRY_OPS);
```

### 11.7.6. Fallos y problemas

Nuestro complemento ya está completo. Veamos ahora algunas cosas que están mal en él o que podemos mejorar:

- Podríamos usar una GUI mejorada, una que permita al usuario introducir toda la información necesaria en un diálogo.

- El usuario no puede especificar tipos de campos.
- La comprobación de errores es limitada en el diálogo.
  - Si no se introduce ningún campo el complemento falla.
  - No hay comprobación de errores en ninguna de las operaciones de base de datos.
- No hay retroalimentación desde el complemento una vez que finaliza.

A pesar de todos estos fallos, aún sirve como un complemento primordial que ilustra el proceso y ayuda a iniciarse en el desarrollo de complementos.

### 11.7.7. Añadir retroalimentación

Vamos a solucionar uno de los pequeños problemas añadiendo algo de retroalimentación al final del proceso. Sólo añadiremos un cuadro de mensaje para indicar al usuario que todo está hecho y para comprobar la base de datos para asegurarnos de que se creó la tabla.

Para hacer esto, simplemente añadiremos el siguiente código después de la línea 61:

```
# mostrar al usuario lo que ha ocurrido
QMessageBox.information(None, "Resultados", "La tabla " + str(table_name[0]) + \
" se ha creado. Compruebe su base de datos para confirmar.")
```

Cuando la tabla se cree el usuario verá esto:

**Figura 42:** Cuadro de mensaje con los resultados del complemento



### 11.7.8. Resumen

Escribir un complemento de QGIS en Python es bastante sencillo. Algunos complementos no requerirán siquiera una GUI. Por ejemplo, podría escribir un complemento que devuelva las coordenadas del mapa para el punto del mapa en el que pulse. Tal complemento no requeriría ninguna entrada por parte del usuario y podría usar un **QMessageBox** estándar de Qt para mostrar el resultado.

También puede escribir complementos para QGIS en C++, pero eso es otra historia. Puede encontrar manuales sobre la escritura de complementos de QGIS tanto en C++ como en Python en el blog de QGIS en:

<http://blog.qgis.org>

---

## 12. Crear aplicaciones

Uno de los objetivos de QGIS es proporcionar no sólo una aplicación, sino un conjunto de bibliotecas que se puedan usar para crear nuevas aplicaciones. Este objetivo se ha realizado con la reconstrucción de bibliotecas que tuvo lugar después del lanzamiento de la versión 0.8. Con el lanzamiento de la 0.9 es posible el desarrollo de aplicaciones independientes que usen bien C++ o Python.

En este capítulo echaremos una breve ojeada al proceso de crear aplicaciones independientes en Python. El blog de QGIS tiene varios ejemplos de creación de aplicaciones en PyQGIS<sup>6</sup>. Usaremos uno de ellos como punto de partida para tener una idea de cómo crear una aplicación.

Las funciones que queremos en la aplicación son:

- Cargar una capa vectorial.
- Panorámica.
- Acercar y alejar zum.
- Zum a la extensión completa de la capa.
- Estalecer colores personalizados al cargar la capa.

Se trata de un conjunto mínimo de funciones. Comencemos por diseñar la GUI usando Qt Designer.

### 12.1. Diseñar la GUI

Puesto que estamos creando una aplicación mínima, usaremos la misma aproximación con la GUI. Usando Qt Designer, creamos una `MainWindow` sencilla sin menús ni barras de herramientas. Esto nos da una ventana en blanco con la que trabajar. Para crear la `MainWindow`:

1. Crear un directorio para desarrollar la aplicación y cambiar a él.
2. Ejecutar Qt Designer.
3. Debe aparecer el diálogo "Formulario nuevo". Si no lo hace, seleccionar *Formulario nuevo...* del menú *Archivo*.
4. Seleccionar "Ventana principal" de la lista de plantillas/formularios.
5. Pulsar *Crear*.
6. Redimensionar la nueva ventana a algo manejable.
7. Buscar el control Marco (Frame) en la lista (bajo Contenedores) y arrastrarlo a la ventana principal que acabamos de crear.
8. Pulsar fuera del marco para seleccionar el área de la ventana principal.

---

<sup>6</sup>An application created using Python and the QGIS bindings



9. Pulsar en la herramienta *Lay Out in a Grid*. Al hacerlo el marco se expandirá hasta ocupar totalmente la ventana principal.
10. Guardar el formulario como *mainwindow.ui*.
11. Salir de Qt Designer.

Ahora compile el formulario usando el compilador de la interfaz de PyQt:

```
pyuic4 -o mainwindow_ui.py mainwindow.ui
```

Esto crea la fuente de Python para la ventana principal de la GUI. Lo siguiente que necesitamos es crear el código de la aplicación para rellenar la ventana en blanco con algunas herramientas que podamos usar.

## 12.2. Crear la Ventana principal

Ahora estamos listos para escribir la clase **MainWindow** que hará el trabajo real. Puesto que esté llevará unas cuantas líneas, lo veremos por partes; comenzaremos con la sección de importación y la configuración del entorno:

```
1 # Basado libremente en:
2 #   C++ Tutorial 2 original por Tim Sutton
3 #   migrado a Python por Martin Dobias
4 #   con mejoras por Gary Sherman para FOSS4G2007
5 # Licenciado bajo los términos de la GNU GPL 2
6
7 from PyQt4.QtCore import *
8 from PyQt4.QtGui import *
9 from qgis.core import *
10 from qgis.gui import *
11 import sys
12 import os
13 # Importar nuestra GUI
14 from mainwindow_ui import Ui_MainWindow
15 # Importar nuestros recursos (iconos)
16 import resources
17
18 # La variable de entorno QGISHOME se debe establecer al directorio de instalación
19 # de la versión 0.9 antes de ejecutar esta aplicación
20 qgis_prefix = os.getenv("QGISHOME")
```

Parte de esto debería resultar familiar de nuestro complemento, especialmente las importaciones de PyQt4 y QGIS. Algunas cosas específicas que destacar son la importación de nuestra GUI en la línea 14 y la importación de nuestro archivo de recursos en la línea 16.

Nuestra aplicación necesita saber dónde encontrar la instalación de QGIS. Por eso, establecemos la variable de entorno QGISHOME para que apunte al directorio de instalación de QGIS 0.9. En la línea 20 guardamos este valor del entorno para usarla después.

Lo siguiente que necesitamos es crear nuestra clase **MainWindow** que contendrá toda la lógica de nuestra aplicación.

```

21 class MainWindow(QMainWindow, Ui_MainWindow):
22
23     def __init__(self):
24         QMainWindow.__init__(self)
25
26         # Requerido por Qt4 para inicializar la UI
27         self.setupUi(self)
28
29         # Establecer el título de la aplicación
30         self.setWindowTitle("Aplicación de demostración FOSS4G2007")
31
32         # Crear el lienzo del mapa
33         self.canvas = QgsMapCanvas()
34         # Establecer el color de fondo a azul claro
35         self.canvas.setCanvasColor(QColor(200,200,255))
36         self.canvas.enableAntiAliasing(True)
37         self.canvas.useQImageToRender(False)
38         self.canvas.show()
39
40         # Disponer nuestros controles en la ventana principal usando una
41         # disposición de caja vertical
42         self.layout = QVBoxLayout(self.frame)
43         self.layout.addWidget(self.canvas)
44
45         # Crear las acciones para nuestras herramientas y conectar cada una con el método
46         # adecuado
47         self.actionAddLayer = QAction(QIcon(":/foss4g2007/mActionAddLayer.png"),
48         \
49             "Add Layer", self.frame)
50         self.connect(self.actionAddLayer, SIGNAL("activated()"), self.addLayer)
51         self.actionZoomIn = QAction(QIcon(":/foss4g2007/mActionZoomIn.png"), \
52             "Zoom In", self.frame)

```

```

53     self.connect(self.actionZoomIn, SIGNAL("activated()"), self.zoomIn)
54     self.actionZoomOut = QAction(QIcon(":/foss4g2007/mActionZoomOut.png"), \
55         "Zoom Out", self.frame)
56     self.connect(self.actionZoomOut, SIGNAL("activated()"), self.zoomOut)
57     self.actionPan = QAction(QIcon(":/foss4g2007/mActionPan.png"), \
58         "Pan", self.frame)
59     self.connect(self.actionPan, SIGNAL("activated()"), self.pan)
60     self.actionZoomFull = QAction(QIcon(":/foss4g2007/mActionZoomFullExtent.png"), \
61         "Zoom Full Extent", self.frame)
62     self.connect(self.actionZoomFull, SIGNAL("activated()"),
63         self.zoomFull)
64
65     # Crear una barra de herramientas
66     self.toolbar = self.addToolBar("Map")
67     # Añadir las acciones a la barra de herramientas
68     self.toolbar.addAction(self.actionAddLayer)
69     self.toolbar.addAction(self.actionZoomIn)
70     self.toolbar.addAction(self.actionZoomOut);
71     self.toolbar.addAction(self.actionPan);
72     self.toolbar.addAction(self.actionZoomFull);
73
74     # Crear las herramientas de mapa
75     self.toolPan = QgsMapToolPan(self.canvas)
76     self.toolZoomIn = QgsMapToolZoom(self.canvas, False) # false = in
77     self.toolZoomOut = QgsMapToolZoom(self.canvas, True) # true = out

```

Las líneas 21 a 27 son la declaración básica y la inicialización de la **MainWindow** y la configuración de la interfaz de usuario usando el método *setupUi*. Esto hace falta para todas las aplicaciones.

A continuación ponemos el título de la aplicación de forma que diga algo más interesante que 'MainWindow' (línea 30). Una vez que esto está hecho, estamos listos para completar la interfaz de usuario. Cuando la creamos en el Designer, la dejamos muy escasa—sólo una ventana principal y un marco. Se podría haber añadido un menú y la barra de herramientas usando el Designer, sin embargo lo haremos con Python.

En las líneas 33 a 38 configuramos la vista del mapa, establecemos el color de fondo a azul claro y habilitamos antialiasing. También le decimos que no use un QImage para renderizar (confiar en mí en esto) y hacemos visible el lienzo del mapa llamando al método *show*.

A continuación configuramos la capa para que use una disposición de caja vertical dentro del marco y le añadimos la vista del mapa en la línea 43.

Las líneas 48 a 63 configuran las acciones y conexiones de las herramientas de nuestra barra de herramientas. Para cada herramienta creamos una **QAction** usando el icono que definimos en nuestro archivo de recursos. Luego conectamos la señal *activated* de la herramienta al método de nuestra clase que manejará la acción. Esto es similar a cómo configuramos las cosas en el ejemplo de complemento.

Una vez que tenemos las acciones y las conexiones, necesitamos añadirlas a la barra de herramientas. En las líneas 66 a 72 creamos la barra de herramientas y le añadimos cada herramienta.

Por último creamos las tres herramientas de mapa para la aplicación (líneas 75 a 77). Usaremos las herramientas en un momento cuando definamos los métodos para hacer funcional la aplicación. Veamos los métodos para las herramientas de mapa.

```

78  # Establecer la herramienta de mapa para acercar zum
79  def zoomIn(self):
80      self.canvas.setMapTool(self.toolZoomIn)
81
82  # Establecer la herramienta de mapa para alejar zum
83  def zoomOut(self):
84      self.canvas.setMapTool(self.toolZoomOut)
85
86  # Establecer la herramienta de mapa para panorámica
87  def pan(self):
88      self.canvas.setMapTool(self.toolPan)
89
90  # Zum a la extensión de la capa
91  def zoomFull(self):
92      self.canvas.zoomFullExtent()
```

Para cada herramienta de mapa necesitamos un método que corresponda a la conexión que hemos hecho para cada acción. En las líneas 79 a 88 establecemos el método para cada una de las tres herramientas que interaccionan con el mapa. Cuando se activa una herramienta pulsando en ella en la barra de herramientas, se llama al método correspondiente que “le dice” a la vista del mapa que esa es la herramienta activa, la cual gobierna lo que pasa cuando se pulsa el ratón sobre la vista del mapa.

La herramienta zum a toda la extensión no es una herramienta de mapa—hace su trabajo sin que se requiera una pulsación en el mapa. Cuando se activa llamamos al método *zoomFullExtent* de la vista del mapa (línea 92). Esto completa la implementación de todas nuestras herramientas menos una—la herramienta añadir capa. Veámosla a continuación:

```

93  # Añadir una capa OGR al mapa
```

```

94 def addLayer(self):
95     file = QFileDialog.getOpenFileName(self, "Abrir archivo Shape", ".", "Archivos shape
96     (*.shp)")
97     fileInfo = QFileInfo(file)
98
99     # Añadir la capa
100     layer = QgsVectorLayer(file, fileInfo.fileName(), "ogr")
101
102     if not layer.isValid():
103         return
104
105     # Cambiar el color de la capa a gris
106     symbols = layer.renderer().symbols()
107     symbol = symbols[0]
108     symbol.setFill(QColor.fromRgb(192,192,192))
109
110     # Añadir capa al registro
111     QgsMapLayerRegistry.instance().addMapLayer(layer);
112
113     # Establecer extensión a la de nuestra capa
114     self.canvas.setExtent(layer.extent())
115
116     # Establecer el conjunto de capas de la vista del mapa
117     cl = QgsMapCanvasLayer(layer)
118     layers = [cl]
119     self.canvas.setLayerSet(layers)

```

En el método *addLayer* usamos **QFileDialog** para obtener el nombre del archivo shape a cargar. Esto se hace en la línea 96. Observe que especificamos un “filtro” para que el diálogo sólo muestre los archivos de tipo *.shp*.

A continuación en la línea 97 creamos un objeto **QFileInfo** a partir de la ruta del archivo shape. Ahora la capa está lista para crearse en la línea 100. Usando el objeto **QFileInfo** para obtener el nombre del archivo de la ruta lo especificamos para el nombre de la capa cuando se crea. Para asegurarnos de que la capa es válida y no dará problemas cuando se cargue, la comprobamos en la línea 102. Si no es válida, nos libramos de ella y no la añadimos a la vista del mapa.

Normalmente las capas se añaden con un color aleatorio. Aquí queremos ajustar los colores de la capa para una visualización más agradable. Además sabemos que vamos a añadir la capa *world\_borders* a la vista del mapa y esto hará que se vea bien sobre nuestro fondo azul. Para cambiar el color, necesitamos obtener el símbolo usado para renderizar y usarlo para establecer un color de relleno nuevo. Esto se hace en las líneas 106 a 108.

Todo lo que queda es añadir realmente la capa al registro y unos pocos elementos de mantenimiento más (líneas 111 a 119). Este proceso es estándar para añadir una capa y el resultado final son los bordes del mundo sobre un fondo azul claro. La única cosa que puede no querer hacer es establecer la extensión a la capa, si va a añadir más de una capa a su aplicación.

Este es el corazón de la aplicación y completa la clase **MainWindow**.

### 12.3. Finalizar

El resto del código mostrado abajo crea el objeto **QgsApplication**, establece la ruta a la instalación de QGIS, configura el método *main* y a continuación inicia la aplicación. La única cosa a destacar es que movemos la ventana de la aplicación a la esquina superior izquierda del monitor. Podríamos echarle imaginación y usar el API de Qt API para centrarla en la pantalla.

```

120 def main(argv):
121     # crear aplicación Qt
122     app = QApplication(argv)
123
124     # Inicializar las bibliotecas de qgis
125     QgsApplication.setPrefixPath(qgis_prefix, True)
126     QgsApplication.initQgis()
127
128     # crear la ventana principal
129     wnd = MainWindow()
130     # Mover la ventana de la aplicación arriba a la izquierda
131     wnd.move(100,100)
132     wnd.show()
133
134     # ¡ejecutar!
135     retval = app.exec_()
136
137     # salir
138     QgsApplication.exitQgis()
139     sys.exit(retval)
140
141
142 if __name__ == "__main__":
143     main(sys.argv)

```

## 12.4. Ejecutar la aplicación

Ahora podemos ejecutar la aplicación y ver qué pasa. Por supuesto si se es como la mayoría de desarrolladores, se habrá ido probando a medida que se avanzaba.

Antes de que podamos ejecutar la aplicación, tenemos que establecer algunas variables de entorno. En Linux u OS X:

```
export LD_LIBRARY_PATH=$HOME/qgis_09/lib
export PYTHONPATH=$HOME/qgis_09/share/qgis/python
export QGISHOME=$HOME/qgis_09
```

Para Windows:

```
set PATH=C:\qgis;%PATH%
set PYTHONPATH=C:\qgis\python
set QGISHOME=C:\qgis
```

En el caso de Linux u OS X, asumimos que QGIS está instalado en su directorio personal en *qgis\_09*. Para Windows, QGIS está instalado en *C:\qgis*.

Cuando la aplicación arranca tiene este aspecto:

**Figura 43:** Iniciar la nueva aplicación de demostración



Para añadir la capa *world\_borders*, pulse en la herramienta *Añadir capa* y navegue al directorio de datos. Seleccione el archivo shape y pulse *Abrir* para añadirla al mapa. Se aplicará nuestro color personalizado y el resultado es:

Crear una aplicación de PyQGIS es realmente muy sencillo. En menos de 150 líneas de código tenemos una aplicación que puede cargar un archivo shape y navegar por el mapa. Si juega un poco con el mapa, notará que algunas de las funciones incrustadas del lienzo también funcionan, incluido el desplazamiento con la rueda del ratón y la panorámica manteniendo pulsada la barra *Espacio* y moviendo el ratón.

**Figura 44:** Añadir una capa a la aplicación de demostración

Algunas aplicaciones sofisticadas se han creado usando PyQGIS y más están en camino. Esto es bastante impresionante, considerando que este desarrollo ha tenido lugar incluso antes del lanzamiento oficial de QGIS 0.9.

---

**Tip 37** DOCUMENTACIÓN PARA PYQGIS

---

Tanto si está escribiendo un complemento o una aplicación en PyQGIS, va a necesitar consultar tanto la documentación de la API de QGIS (<http://qgis.org>) como la Guía de referencia de enlaces Python PyQt (PyQt Python Bindings Reference Guide) (<http://www.riverbankcomputing.com/Docs/PyQt4/pyqt4ref.html>). Estos documentos proporcionan información sobre las clases y métodos que usará para dar vida a su creación de Python.

---



---

## 13. Ayuda y soporte

### 13.1. Listas de correo

QGIS aún está bajo un desarrollo activo y como tal, no siempre funcionará como se espera. El modo preferible para obtener ayuda es apuntarse a la lista de correo de usuarios de qgis (qgis-users).

#### **qgis-users**

Sus preguntas llegarán a una audiencia mayor y las respuestas beneficiarán a otros. Puede suscribirse a la lista de correo de usuarios de qgis (qgis-users) visitando la siguiente URL:  
<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-user>

#### **qgis-developer**

Si es un desarrollador con problemas de carácter más técnico, quizá quiera unirse a la lista de correo de desarrolladores (qgis-developer) aquí:  
<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-developer>

#### **qgis-commit**

Cada vez que se hace un envío al repositorio del código de QGIS se envía un correo a esta lista. Si quiere estar al día con cada cambio en el actual código base, se puede suscribir a esta lista en:  
<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-commit>

#### **qgis-trac**

Esta lista proporciona notificaciones por correo electrónico relacionadas con la administración del proyecto, incluyendo informes de errores, tareas y solicitudes de funciones. Puede suscribirse a esta lista en:  
<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-trac>

#### **qgis-doc**

Esta lista trata asuntos como la documentación, ayuda contextual, guía de usuario y esfuerzos de traducción. Si también quiere trabajar en la guía de usuario, esta lista es un buen punto de inicio para hacer sus preguntas. Puede suscribirse a esta lista en:  
<http://lists.qgis.org/cgi-bin/mailman/listinfo/qgis-doc>

#### **qgis-psc**

Esta lista se usa para debatir asuntos del Comité de Dirección (Steering Committee) relacionados con la administración general de Quantum GIS. Puede suscribirse a esta lista en:

<http://mrcc.com/cgi-bin/mailman/listinfo/qgis-psc>

Será bienvenido a cualquiera de las listas. Por favor, recuerde contribuir a la lista contestando preguntas y compartiendo sus experiencias. Tenga en cuenta que las listas qgis-commit y qgis-trac están diseñadas para la notificación solamente y no para correos de los usuarios.

## 13.2. IRC

También mantenemos una presencia en IRC – visítenos uniéndose al canal #qgis en [irc.freenode.net](http://irc.freenode.net). Por favor, espere un poco a las respuestas a sus preguntas, ya que muchos colegas en el canal están haciendo otras cosas y puede llevar un rato hasta que noten su pregunta. También hay disponible soporte comercial para QGIS. Compruebe la página web <http://qgis.org/content/view/90/91> para más información.

¡Si se has perdido un debate en IRC, no hay problema! Registramos todos los debates, así que puede recuperarlo fácilmente. Simplemente vaya a <http://logs.qgis.org> y lea los registros del IRC.

## 13.3. Seguimiento de errores (BugTracker)

Mientras que la lista de correo de usuarios de qgis es útil para preguntas del tipo ¿Cómo hago xyz en QGIS?, puede que desee notificarnos un error en QGIS. Puede enviar informes de errores usando el seguidor de errores de QGIS en <http://svn.qgis.org/trac>. Cuando cree un nuevo registro para un error, por favor proporcione una dirección de correo electrónico en la que podamos solicitar información adicional.

Por favor, tenga en mente que su error no siempre tendrá la prioridad que pueda esperar (dependiendo de la gravedad). Algunos errores pueden requerir un esfuerzo significativo para remediarlos y no siempre se dispone de los recursos humanos para ello.

Las solicitudes de funciones se pueden enviar usando el mismo sistema que para los errores. Por favor, asegúrese de seleccionar el tipo **enhancement** (mejora).

Si ha encontrado un error y lo soluciona usted mismo, puede enviar también el parche. El sistema de seguimiento en <http://svn.qgis.org/trac> tiene también este tipo. Seleccione **patch** (parche) en el menú tipo. Alguno de los desarrolladores lo revisará y lo aplicará a QGIS. No se alarme si su parche no se aplica inmediatamente, los desarrolladores pueden estar desbordados con otros envíos.

## 13.4. Blog

La comunidad de QGIS también lleva un blog web (BLOG) en <http://blog.qgis.org> el cual tiene algunos artículos interesantes para usuarios y desarrolladores. Le invitamos a contribuir al blog después

de registrarse.

### **13.5. Wiki**

Por último, mantenemos un WIKI en la web en <http://wiki.qgis.org> donde puede encontrar información útil variada relacionada con el desarrollo de QGIS, planes de lanzamientos, enlaces a sitios de descarga, apuntes sobre la traducción de mensajes, etc. ¡Compruébelo, hay buenas cosas dentro!

---

## A. Supported Data Formats

### A.1. Supported OGR Formats

At the date of this document, the following formats are supported by the OGR library. Formats known to work in QGIS are indicated in **bold**.

- **Arc/Info Binary Coverage**
- Comma Separated Value (.csv)
- DODS/OPeNDAP
- **ESRI Shapefile**
- FMEObjects Gateway
- GML
- IHO S-57 (ENC)
- **Mapinfo File**
- Microstation DGN
- OGDl Vectors
- ODBC
- Oracle Spatial
- PostgreSQL<sup>7</sup>
- **SDTS**
- SQLite
- UK .NTF
- U.S. Census TIGER/Line
- VRT - Virtual Datasource

### A.2. GDAL Raster Formats

At the date of this document, the following formats are supported by the GDAL library. Note that not all of these format may work in QGIS for various reasons. For example, some require external commercial libraries. Only those formats that have been well tested will appear in the list of file types when loading a raster into QGIS. Other untested formats can be loaded by selecting the *All other files* (\*) filter. Formats known to work in QGIS are indicated in **bold**.

---

<sup>7</sup>QGIS implements its own PostgreSQL functions. OGR should be built without PostgreSQL support

- **Arc/Info ASCII Grid**
- **Arc/Info Binary Grid (.adf)**
- Microsoft Windows Device Independent Bitmap (.bmp)
- BSB Nautical Chart Format (.kap)
- VTP Binary Terrain Format (.bt)
- CEOS (Spot for instance)
- First Generation USGS DOQ (.doq)
- New Labelled USGS DOQ (.doq)
- Military Elevation Data (.dt0, .dt1)
- ERMapper Compressed Wavelets (.ecw)
- ESRI .hdr Labelled
- ENVI .hdr Labelled Raster
- Envisat Image Product (.n1)
- EOSAT FAST Format
- FITS (.fits)
- Graphics Interchange Format (.gif)
- **GRASS Rasters<sup>8</sup>**
- **TIFF / GeoTIFF (.tif)**
- Hierarchical Data Format Release 4 (HDF4)
- **Erdas Imagine (.img)**
- Atlantis MFF2e
- Japanese DEM (.mem)
- **JPEG JFIF (.jpg)**
- JPEG2000 (.jp2, .j2k)
- JPEG2000 (.jp2, .j2k)
- NOAA Polar Orbiter Level 1b Data Set (AVHRR)
- Erdas 7.x .LAN and .GIS
- In Memory Raster
- Atlantis MFF
- Multi-resolution Seamless Image Database MrSID
- NITF

---

<sup>8</sup>GRASS raster support is supplied by the QGIS GRASS data provider plugin

- NetCDF
- OGD Bridge
- PCI .aux Labelled
- PCI Geomatics Database File
- Portable Network Graphics (.png)
- Netpbm (.ppm,.pgm)
- **USGS SDTS DEM (\*CATD.DDF)**
- SAR CEOS
- **USGS ASCII DEM (.dem)**
- X11 Pixmap (.xpm)

---

## B. Installation Guide

The following chapters provide build and installation information for QGIS Version 0.9.1. This document corresponds almost to a  $\text{\LaTeX}$  conversion of the INSTALL.t2t file coming with the QGIS sources from November, 29th 2007.

A current version is also available at the wiki, see: <http://wiki.qgis.org/qgiswiki/BuildingFromSource>

### B.1. General Build Notes

At version 0.8.1 QGIS no longer uses the autotools for building. QGIS, like a number of major projects (eg. KDE 4.0), now uses cmake for building from source. The configure script in this directory simply checks for the existence of cmake and provides some clues to build QGIS.

For complete information, see the wiki at: [http://wiki.qgis.org/qgiswiki/Building\\_with\\_CMake](http://wiki.qgis.org/qgiswiki/Building_with_CMake)

### B.2. An overview of the dependencies required for building

#### Required build deps:

- CMake  $\geq 2.4.3$
- Flex, Bison

#### Required runtime deps:

- Qt  $\geq 4.2.0$
- Proj  $\geq ?$  (known to work with 4.4.x)
- GEOS  $\geq 2.2$  (3.0 is supported, maybe 2.1.x works too)
- Sqlite3  $\geq ?$  (probably 3.0.0)
- GDAL/OGR  $\geq ?$  (1.2.x should work)

#### Optional dependencies:

- for GRASS plugin - GRASS  $\geq 6.0.0$
- for georeferencer - GSL  $\geq ?$  (works with 1.8)
- for postgis support and SPIT plugin - PostgreSQL  $\geq ?$
- for gps plugin - expat  $\geq ?$  (1.95 is OK)
- for mapserver export and PyQGIS - Python  $\geq ?$  (probably 2.3)
- for PyQGIS - SIP  $\geq 4.5$ , PyQt  $\geq 4.1$

---

**Recommended runtime deps:**

- for gps plugin - gpsbabel

## **C. Building under windows using msys**

### **C.1. MSYS:**

MSYS provides a unix style build environment under windows. We have created a zip archive that contains just about all dependencies.

Get this:

<http://qgis.org/uploadfiles/msys/msys.zip>

and unpack to c:\msys

If you wish to prepare your msys environment yourself rather than using our pre-made one, detailed instructions are provided elsewhere in this document.

### **C.2. Qt4.3**

Download qt4.3 opensource precompiled edition exe and install (including the download and install of mingw) from here:

<http://www.trolltech.com/developer/downloads/qt/windows>

When the installer will ask for MinGW, you don't need to download and install it, just point the installer to c:\msys\mingw

When Qt installation is complete:

Edit C:\Qt\4.3.0\bin\qtvars.bat and add the following lines:

```
set PATH=%PATH%;C:\msys\local\bin;c:\msys\local\lib
set PATH=%PATH%; "C:\Program Files\Subversion\bin"
```

I suggest you also add C:\Qt\4.3.0\bin\ to your Environment Variables Path in the windows system preferences.

If you plan to do some debugging, you'll need to compile debug version of Qt:  
C:\Qt\4.3.0\bin\qtvars.bat compile\_debug



Note: there is a problem when compiling debug version of Qt 4.3, the script ends with this message "mingw32-make: \*\*\* No rule to make target 'debug'. Stop.". To compile the debug version you have to go out of src directory and execute the following command:

```
c:\Qt\4.3.0 make
```

### C.3. Flex and Bison

\*\*\* Note I think this section can be removed as it should be installed int the msys image already. TS

Get Flex [http://sourceforge.net/project/showfiles.php?group\\_id=23617&package\\_id=16424](http://sourceforge.net/project/showfiles.php?group_id=23617&package_id=16424) (the zip bin) and extract it into c:\msys\mingw\bin

### C.4. Python stuff: (optional)

Follow this section in case you would like to use Python bindings for QGIS. To be able to compile bindings, you need to compile SIP and PyQt4 from sources as their installer doesn't include some development files which are necessary.

#### C.4.1. Download and install Python - use Windows installer

(It doesn't matter to what folder you'll install it)

<http://python.org/download/>

#### C.4.2. Download SIP and PyQt4 sources

```
\htmladdnormallink{http://www.riverbankcomputing.com/Downloads/sip4/}  
\htmladdnormallink{http://www.riverbankcomputing.com/Downloads/PyQt4/GPL/}
```

Extract each of the above zip files in a temporary directory. Make sure to get versions that match your current Qt installed version.

#### C.4.3. Compile SIP

```
c:\Qt\4.3.0\bin\qtvars.bat  
python configure.py -p win32-g++
```

```
make
make install
```

#### C.4.4. Compile PyQt

```
c:\Qt\4.3.0\bin\qtvars.bat
python configure.py
make
make install
```

#### C.4.5. Final python notes

!!\ You can delete the directories with unpacked SIP and PyQt4 sources after a successfull install, they're not needed anymore.

### C.5. Subversion:

In order to check out QGIS sources from the repository, you need Subversion client. This installer should work fine:

<http://subversion.tigris.org/files/documents/15/36797/svn-1.4.3-setup.exe>

### C.6. CMake:

CMake is build system used by Quantum GIS. Download it from here:

<http://www.cmake.org/files/v2.4/cmake-2.4.6-win32-x86.exe>

### C.7. QGIS:

Start a cmd.exe window ( Start -> Run -> cmd.exe ) Create development directory and move into it

```
md c:\dev\cpp
cd c:\dev\cpp
```

Check out sources from SVN For svn head:

---

```
svn co https://svn.qgis.org/repos/qgis/trunk/qgis
```

For svn 0.8 branch

```
svn co https://svn.qgis.org/repos/qgis/branches/Release-0_8_0 qgis0.8
```

## C.8. Compiling:

As a background read the generic building with CMake notes at the end of this document.

Start a cmd.exe window ( Start -> Run -> cmd.exe ) if you don't have one already. Add paths to compiler and our MSYS environment:

```
c:\Qt\4.3.0\bin\qtvars.bat
```

For ease of use add c:\Qt\4.3.0\bin\ to your system path in system properties so you can just type qtvars.bat when you open the cmd console. Create build directory and set it as current directory:

```
cd c:\dev\cpp\qgis
md build
cd build
```

## C.9. Configuration

```
cmakesetup ..
```

**NOTE:** You must include the '..'above.

Click 'Configure'button. When asked, you should choose 'MinGW Makefiles'generator.

There's a problem with MinGW Makefiles on Win2K. If you're compiling on this platform, use 'MSYS Makefiles'generator instead.

All dependencies should be picked up automatically, if you have set up the Paths correctly. The only thing you need to change is the installation destination (CMAKE\_INSTALL\_PREFIX) and/or set 'Debug'.

For compatibility with NSIS packaging cripts I recommend to leave the install prefix to its default c:\program files\

When configuration is done, click ÓK'to exit the setup utility.

## C.10. Compilation and installation

```
make  
make install
```

### C.11. Run qgis.exe from the directory where it's installed (CMAKE\_INSTALL\_PREFIX)

Make sure to copy all .dll:s needed to the same directory as the qgis.exe binary is installed to, if not already done so, otherwise QGIS will complain about missing libraries when started.

The best way to do this is to download both the QGIS current release installer package from <http://qgis.org/uploadfiles/testbuilds/> and install it. Now copy the installation dir from C:\Program Files\Quantum GIS into c:\Program Files\qgis-0.8.1 (or whatever the current version is. The name should strictly match the version no.) After making this copy you can uninstall the release version of QGIS from your c:\Program Files directory using the provided uninstaller. Double check that the Quantum GIS dir is completely gone under program files afterwards.

Another possibility is to run qgis.exe when your path contains c:\msys\local\bin and c:\msys\local\lib directories, so the DLLs will be used from that place.

### C.12. Create the installation package: (optional)

Download and install NSIS from ([http://nsis.sourceforge.net/Main\\_Page](http://nsis.sourceforge.net/Main_Page))

Now using windows explorer, enter the win\_build directory in your QGIS source tree. Read the READMEfile there and follow the instructions. Next right click on qgis.nsi and choose the option 'Compile NSIS Script'.

## D. Building on Mac OSX using frameworks and cmake (QGIS > 0.8)

In this approach I will try to avoid as much as possible building dependencies from source and rather use frameworks wherever possible.

### D.1. Install XCODE

I recommend to get the latest xcode dmg from the Apple XDC Web site. Install XCODE after the ~941mb download is complete.

## D.2. Install Qt4 from .dmg

You need a minimum of Qt4.2. I suggest getting the latest (at time of writing).

```
ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.3.2.dmg
```

If you want debug libs, Qt also provide a dmg with these:

```
ftp://ftp.trolltech.com/qt/source/qt-mac-opensource-4.3.2-debug-libs.dmg
```

I am going to proceed using only release libs at this stage as the download for the debug dmg is substantially bigger. If you plan to do any debugging though you probably want to get the debug libs dmg. Once downloaded open the dmg and run the installer. Note you need admin access to install.

After installing you need to make two small changes:

First edit `/Library/Frameworks/QtCore.framework/Headers/qconfig.h` and change

```
/*! Note this doesnt seem to be needed since version 4.2.3
```

```
QT_EDITION_UNKNOWN to QT_EDITION_OPENSOURCE
```

Second change the default mkspec symlink so that it points to macx-g++:

```
cd /usr/local/Qt4.3/mkspecs/ sudo rm default sudo ln -sf macx-g++ default
```

## D.3. Install development frameworks for QGIS dependencies

Download William Kyngesburye's excellent all in one framework that includes proj, gdal, sqlite3 etc

```
http://www.kyngchaos.com/files/software/unixport/AllFrameworks.dmg
```

Once downloaded, open and install the frameworks.

William provides an additional installer package for PostgreSQL/PostGIS. Its available here:

```
http://www.kyngchaos.com/software/unixport/postgres
```

There are some additional dependencies that at the time of writing are not provided as frameworks so we will need to build these from source.

### D.3.1. Additional Dependencies : GSL

Retrieve the Gnu Scientific Library from

```
curl -O ftp://ftp.gnu.org/gnu/gsl/gsl-1.8.tar.gz
```

Then extract it and build it to a prefix of /usr/local:

```
tar xvfz gsl-1.8.tar.gz
cd gsl-1.8
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

### D.3.2. Additional Dependencies : Expat

Get the expat sources:

```
http://sourceforge.net/project/showfiles.php?group\_id=10127
```

```
tar xvfz expat-2.0.0.tar.gz
cd expat-2.0.0
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

### D.3.3. Additional Dependencies : SIP

Retrieve the python bindings toolkit SIP from

```
http://www.riverbankcomputing.com/Downloads/sip4/
```

Then extract and build it to a prefix of /usr/local:

```
tar xvfz sip-<version number>.tar.gz
```

```
cd sip-<version number>
python configure.py
make
sudo make install
cd ..
```

#### D.3.4. Additional Dependencies : PyQt

Make sure you have the latest python fom

<http://www.python.org/download/mac/>

If you encounter problems compiling PyQt using the instructions below you can also try adding python from your frameworks dir explicitly to your path e.g.

```
export PATH=/Library/Frameworks/Python.framework/Versions/Current/bin:$PATH$
```

Retrieve the python bindings toolkit for Qt from

<http://www.riverbankcomputing.com/Downloads/PyQt4/GPL/>

Then extract and build it to a prefix of /usr/local:

```
tar xvfz PyQt-mac<version number here>
cd PyQt-mac<version number here>
python configure.py
yes
make
sudo make install
cd ..
```

#### D.3.5. Additional Dependencies : Bison

The version of bison available by default on Mac OSX is too old so you need to get a more recent one on your system. Download if from:

```
curl -O http://ftp.gnu.org/gnu/bison/bison-2.3.tar.gz
```

Now build and install it to a prefix of /usr/local :

```
tar xvfz bison-2.3.tar.gz
cd bison-2.3
./configure --prefix=/usr/local
make
sudo make install
cd ..
```

## D.4. Install CMAKE for OSX

Get the latest release from here:

<http://www.cmake.org/HTML/Download.html>

At the time of writing the file I grabbed was:

```
curl -O http://www.cmake.org/files/v2.4/cmake-2.4.6-Darwin-universal.dmg
```

Once downloaded open the dmg and run the installer

## D.5. Install subversion for OSX

The <http://sourceforge.net/projects/macsvn/> project has a downloadable build of svn. If you are a GUI inclined person you may want to grab their gui client too. Get the command line client here:

```
curl -O http://ufpr.dl.sourceforge.net/sourceforge/macsvn/Subversion_1.4.2.zip
```

Once downloaded open the zip file and run the installer.

You also need to install BerkleyDB available from the same <http://sourceforge.net/projects/macsvn/>. At the time of writing the file was here:

```
curl -O http://ufpr.dl.sourceforge.net/sourceforge/macsvn/Berkeley_DB_4.5.20.zip
```

Once again unzip this and run the installer therein.

Lastly we need to ensure that the svn commandline executable is in the path. Add the following line to the end of /etc/bashrc using sudo:



```
sudo vim /etc/bashrc
```

And add this line to the bottom before saving and quitting:

```
export PATH=/usr/local/bin:$PATH:/usr/local/pgsql/bin
```

/usr/local/bin needs to be first in the path so that the newer bison (that will be built from source further down) is found before the bison (which is very old) that is installed by MacOSX

Now close and reopen your shell to get the updated vars.

## D.6. Check out QGIS from SVN

Now we are going to check out the sources for QGIS. First we will create a directory for working in:

```
mkdir -p ~/dev/cpp cd ~/dev/cpp
```

Now we check out the sources:

Trunk:

```
svn co https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

For svn 0.8 branch

```
svn co https://svn.qgis.org/repos/qgis/branches/Release-0_8_0 qgis0.8
```

For svn 0.9 branch

```
svn co https://svn.qgis.org/repos/qgis/branches/Release-0_9_0 qgis0.9
```

The first time you check out QGIS sources you will probably get a message like this:

```
Error validating server certificate for 'https://svn.qgis.org:443':
- The certificate is not issued by a trusted authority. Use the fingerprint to
  validate the certificate manually! Certificate information:
- Hostname: svn.qgis.org
- Valid: from Apr  1 00:30:47 2006 GMT until Mar 21 00:30:47 2008 GMT
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
- Fingerprint: 2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b
  (R)ectect, accept (t)emporarily or accept (p)ermanently?
```

I suggest you press 'p'to accept the key permanently.

## D.7. Configure the build

CMake supports out of source build so we will create a 'build'dir for the build process . By convention I build my software into a dir called apps in my home directory. If you have the correct permissions you may want to build straight into your /Applications folder (although personally I don't really recommend this). The instructions below assume you are building into a pre-existing \${HOME}/apps directory ...

```
cd qgis
mkdir build
cd build
cmake -D CMAKE_INSTALL_PREFIX=${HOME}/apps/ -D CMAKE_BUILD_TYPE=Release ..
```

To use a specific GRASS version, You can optionally use the following cmake invocation (with modifications to suite your system (thanks William Kyngesburye for this hint):

```
cmake -D CMAKE_INSTALL_PREFIX=${HOME}/apps/ \
      -D GRASS_INCLUDE_DIR=/Applications/GRASS-6.3.app/Contents/Resources/include \
      -D GRASS_PREFIX=/Applications/GRASS-6.3.app/Contents/Resources \
      -D CMAKE_BUILD_TYPE=Release \
      ..
```

## D.8. GEOS Issues

I had some issues with GEOS headers so I made the following edits:

In file /Library/Frameworks/GEOS.framework/Headers/io.h, comment out line 61

In file /Library/Frameworks/GEOS.framework/Headers/geom.h, comment out line 145

## D.9. Building

Now we can start the build process:

```
make
```

If all built without errors you can then install it:

```
make install
```

---

## E. Building on GNU/Linux

### E.1. Building QGIS with Qt4.x

**\*Requires:\*** Ubuntu Edgy / Debian derived distro

These notes are for if you want to build QGIS from source. One of the major aims here is to show how this can be done using binary packages for **\*all\*** dependencies - building only the core QGIS stuff from source. I prefer this approach because it means we can leave the business of managing system packages to apt and only concern ourselves with coding QGIS!

This document assumes you have made a fresh install and have a 'clean' system. These instructions should work fine if this is a system that has already been in use for a while, you may need to just skip those steps which are irrelevant to you.

### E.2. Prepare apt

The packages qgis depends on to build are available in the universe component of Ubuntu. This is not activated by default, so you need to activate it:

1. Edit your /etc/apt/sources.list file. 2. Uncomment the all the lines starting with "deb"

Also you will need to be running (K)Ubuntu edgy or higher in order for all dependencies to be met.

Now update your local sources database:

```
sudo apt-get update
```

### E.3. Install Qt4

```
sudo apt-get install libqt4-core libqt4-debug \
libqt4-dev libqt4-gui libqt4-qt3support libqt4-sql lsb-qt4 qt4-designer \
qt4-dev-tools qt4-doc qt4-qtconfig uim-qt gcc libapt-pkg-perl resolvconf
```

!/  
**\*A Special Note:\*** If you are following this set of instructions on a system where you already have Qt3 development tools installed, there will be a conflict between Qt3 tools and Qt4 tools. For example, qmake will point to the Qt3 version not the Qt4. Ubuntu Qt4 and Qt3 packages are designed to live alongside each other. This means that for example if you have them both installed you will have three qmake exe's:

```
/usr/bin/qmake -> /etc/alternatives/qmake
```

```
/usr/bin/qmake-qt3  
/usr/bin/qmake-qt4
```

The same applies to all other Qt binaries. You will notice above that the canonical 'qmake' is managed by apt alternatives, so before we start to build QGIS, we need to make Qt4 the default. To return Qt3 to default later you can use this same process.

You can use apt alternatives to correct this so that the Qt4 version of applications is used in all cases:

```
sudo update-alternatives --config qmake  
sudo update-alternatives --config uic  
sudo update-alternatives --config designer  
sudo update-alternatives --config assistant  
sudo update-alternatives --config qtconfig  
sudo update-alternatives --config moc  
sudo update-alternatives --config lupdate  
sudo update-alternatives --config lrelease  
sudo update-alternatives --config linguist
```

Use the simple command line dialog that appears after running each of the above commands to select the Qt4 version of the relevant applications.

## E.4. Install additional software dependencies required by QGIS

```
sudo apt-get install gdal-bin libgdal1-dev libgeos-dev proj \  
libgdal-doc libhdf4g-dev libhdf4g-run python-dev \  
libgs10-dev g++ libjasper-1.701-dev libtiff4-dev subversion \  
libsqlite3-dev sqlite3 ccache make libpq-dev flex bison cmake txt2tags \  
python-qt4 python-qt4-dev python-sip4 sip4 python-sip4-dev
```

#!/ Debian users should use libgdal-dev above rather

#!/ **\*Note:** For python language bindings SIP >= 4.5 and PyQt4 >= 4.1 is required! Some stable GNU/Linux distributions (e.g. Debian or SuSE) only provide SIP < 4.5 and PyQt4 < 4.1. To include support for python language bindings you may need to build and install those packages from source.

## E.5. GRASS Specific Steps

#!/ **\*Note:** If you don't need to build with GRASS support, you can skip this section.

Now you can install grass from dapper:

```
sudo apt-get install grass libgrass-dev libgdal1-grass
```

#!/ You may need to explicitly state your grass version e.g. libgdal1-1.3.2-grass

## E.6. Setup ccache (Optional)

You should also setup ccache to speed up compile times:

```
cd /usr/local/bin
sudo ln -s /usr/bin/ccache gcc
sudo ln -s /usr/bin/ccache g++
```

## E.7. Prepare your development environment

As a convention I do all my development work in \$HOME/dev/<language>, so in this case we will create a work environment for C++ development work like this:

```
mkdir -p ${HOME}/dev/cpp
cd ${HOME}/dev/cpp
```

This directory path will be assumed for all instructions that follow.

## E.8. Check out the QGIS Source Code

There are two ways the source can be checked out. Use the anonymous method if you do not have edit privileges for the QGIS source repository, or use the developer checkout if you have permissions to commit source code changes.

### 1. Anonymous Checkout

```
cd ${HOME}/dev/cpp
svn co https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

### 2. Developer Checkout

```
cd ${HOME}/dev/cpp
svn co --username <yourusername> https://svn.qgis.org/repos/qgis/trunk/qgis qgis
```

The first time you check out the source you will be prompted to accept the qgis.org certificate. Press 'p' to accept it permanently:

```
Error validating server certificate for 'https://svn.qgis.org:443':
- The certificate is not issued by a trusted authority. Use the
  fingerprint to validate the certificate manually! Certificate
  information:
- Hostname: svn.qgis.org
- Valid: from Apr  1 00:30:47 2006 GMT until Mar 21 00:30:47 2008 GMT
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US
- Fingerprint:
  2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b (R) eject,
  accept (t)emporarily or accept (p)ermanently?
```

## E.9. Starting the compile

I compile my development version of QGIS into my ~/apps directory to avoid conflicts with Ubuntu packages that may be under /usr. This way for example you can use the binary packages of QGIS on your system along side with your development version. I suggest you do something similar:

```
mkdir -p ${HOME}/apps
```

Now we create a build directory and run cmake:

```
cd qgis
mkdir build
cd build
cmake ..
```

When you run cmake (note the .. is required!), a menu will appear where you can configure various aspects of the build. If you do not have root access or do not want to overwrite existing QGIS installs (by your packagemanager for example), set the CMAKE\_BUILD\_PREFIX to somewhere you have write access to (I usually use /home/timlinux/apps). Now press 'c' to configure, 'é' to dismiss any error messages that may appear. and 'g' to generate the make files. Note that sometimes 'c' needs to be pressed several times before the 'g' option becomes available. After the 'g' generation is complete, press 'q' to exit the cmake interactive dialog.

Now on with the build:

```
make
make install
```

It may take a little while to build depending on your platform.

## **E.10. Running QGIS**

Now you can try to run QGIS:

```
$HOME/apps/bin/qgis
```

If all has worked properly the QGIS application should start up and appear on your screen.

## **F. Creation of MSYS environment for compilation of Quantum GIS**

### **F.1. Initial setup**

#### **F.1.1. MSYS**

This is the environment that supplies many utilities from UNIX world in Windows and is needed by many dependencies to be able to compile.

Download from here:

<http://puzzle.dl.sourceforge.net/sourceforge/mingw/MSYS-1.0.11-2004.04.30-1.exe>

Install to `c:\msys`

All stuff we're going to compile is going to get to this directory (resp. its subdirs).

#### **F.1.2. MinGW**

Download from here:

<http://puzzle.dl.sourceforge.net/sourceforge/mingw/MinGW-5.1.3.exe>

Install to `c:\msys\mingw`

It suffices to download and install only `g++` and `mingw-make` components.

### F.1.3. Flex and Bison

Flex and Bison are tools for generation of parsers, they're needed for GRASS and also QGIS compilation.

Download the following packages:

<http://gnuwin32.sourceforge.net/downlinks/flex-bin-zip.php>

<http://gnuwin32.sourceforge.net/downlinks/bison-bin-zip.php>

<http://gnuwin32.sourceforge.net/downlinks/bison-dep-zip.php>

Unpack them all to `c:\msys\local`

## F.2. Installing dependencies

### F.2.1. Getting ready

Paul Kelly did a great job and prepared a package of precompiled libraries for GRASS. The package currently includes:

- zlib-1.2.3
- libpng-1.2.16-noconfig
- xdr-4.0-mingw2
- freetype-2.3.4
- fftw-2.1.5
- PDCurses-3.1
- proj-4.5.0
- gdal-1.4.1

It's available for download here:

<http://www.stjohnspoint.co.uk/grass/wingrass-extralibs.tar.gz>

Moreover he also left the notes how to compile it (for those interested):

<http://www.stjohnspoint.co.uk/grass/README.extralibs>

Unpack the whole package to `c:\msys\local`



### F.2.2. GDAL level one

Since Quantum GIS needs GDAL with GRASS support, we need to compile GDAL from source - Paul Kelly's package doesn't include GRASS support in GDAL. The idea is following:

1. compile GDAL without GRASS
2. compile GRASS
3. compile GDAL with GRASS

So, start with downloading GDAL sources:

<http://download.osgeo.org/gdal/gdal141.zip>

Unpack it to some directory, preferably `c:\msys\local\src`.

Start MSYS console, go to `gdal-1.4.1` directory and run the commands below. You can put them all to a script, e.g. `build-gdal.sh` and run them at once. The recipe is taken from Paul Kelly's instructions - basically they just make sure that the library will be created as DLL and the utility programs will be dynamically linked to it...

```
CFLAGS="-O2 -s" CXXFLAGS="-O2 -s" LDFLAGS=-s ./configure --without-libtool
--prefix=/usr/local --enable-shared --disable-static --with-libz=/usr/local
--with-png=/usr/local
make
make install
rm /usr/local/lib/libgdal.a
g++ -s -shared -o ./libgdal.dll -L/usr/local/lib -lz -lpng ./frmts/o/*.o
./gcore/*.o ./port/*.o ./alg/*.o ./ogr/ogrsf_frmts/o/*.o
./ogr/ogrgeometryfactory.o ./ogr/ogrpoint.o ./ogr/ogrcurve.o
./ogr/ogrlinestring.o ./ogr/ogrlinearring.o ./ogr/ogrpolygon.o
./ogr/ogrutils.o ./ogr/ogrgeometry.o ./ogr/ogrgeometrycollection.o
./ogr/ogrmultipolygon.o ./ogr/ogrsurface.o ./ogr/ogrmultipoint.o
./ogr/ogrmultilinestring.o ./ogr/ogr_api.o ./ogr/ogrfeature.o
./ogr/ogrfeaturedefn.o ./ogr/ogrfeaturequery.o ./ogr/ogrfeaturestyle.o
./ogr/ogrfielddefn.o ./ogr/ogrspatialreference.o ./ogr/ogr_srsnode.o
./ogr/ogr_srs_proj4.o ./ogr/ogr_fromepsg.o ./ogr/ogrct.o ./ogr/ogr_opt.o
./ogr/ogr_srs_esri.o ./ogr/ogr_srs_pci.o ./ogr/ogr_srs_usgs.o
./ogr/ogr_srs_dict.o ./ogr/ogr_srs_panorama.o ./ogr/swq.o
./ogr/ogr_srs_validate.o ./ogr/ogr_srs_xml.o ./ogr/ograssemblepolygon.o
./ogr/ogr2gmlgeometry.o ./ogr/gml2ogrgeometry.o
install libgdal.dll /usr/local/lib
cd ogr
```

```
g++ -s ogrinfo.o -o ogrinfo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s ogr2ogr.o -o ogr2ogr.exe -lgdal -L/usr/local/lib -lpng -lz -lgdal
g++ -s ogrtindex.o -o ogrtindex.exe -lgdal -L/usr/local/lib -lpng -lz -lgdal
install ogrinfo.exe ogr2ogr.exe ogrtindex.exe /usr/local/bin
cd ../apps
g++ -s gdalinfo.o -o gdalinfo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_translate.o -o gdal_translate.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdaladdo.o -o gdaladdo.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdalwarp.o -o gdalwarp.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_contour.o -o gdal_contour.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdaltindex.o -o gdaltindex.exe -L/usr/local/lib -lpng -lz -lgdal
g++ -s gdal_rasterize.o -o gdal_rasterize.exe -L/usr/local/lib -lpng -lz -lgdal
install gdalinfo.exe gdal_translate.exe gdaladdo.exe gdalwarp.exe
gdal_contour.exe gdaltindex.exe gdal_rasterize.exe /usr/local/bin
```

Finally, manually edit `gdal-config` in `c:\msys\local\bin` to replace the static library reference with `-lgdal`:

```
CONFIG_LIBS="-L/usr/local/lib -lpng -lz -lgdal"
```

GDAL build procedure can be greatly simplified to use `libtool` with a `libtool` line patch: configure `gdal` as below: `./configure --with-ngpython --with-xerces=/local/ --with-jasper=/local/ --with-grass=/local/grass-6.3.cvs/ --with-pg=/local/pgsql/bin/pg_config.exe`

Then fix `libtool` with: `mv libtool libtool.orig cat libtool.orig | sed 's/max_cmd_len=8192/max_cmd_len=32768/g'> libtool`

`Libtool` on windows assumes a line length limit of 8192 for some reason and tries to page the linking and fails miserably. This is a work around.

Make and make install should be hassle free after this.

### F.2.3. GRASS

Grab sources from CVS or use a weekly snapshot, see:

<http://grass.itc.it/devel/cvs.php>

In `MSYS` console go to the directory where you've unpacked or checked out sources (e.g. `c:\msys\local\src\grass-6.3.cvs`)

Run these commands:

```
export PATH="/usr/local/bin:/usr/local/lib:$PATH"
./configure --prefix=/usr/local --bindir=/usr/local
--with-includes=/usr/local/include --with-libs=/usr/local/lib --with-cxx
--without-jpeg --without-tiff --without-postgres --with-opengl=windows
--with-fftw --with-freetype --with-freetype-includes=/usr/local/include/freetype2
--without-x --without-tcltk --enable-x11=no --enable-shared=yes
--with-proj-share=/usr/local/share/proj
make
make install
```

It should get installed to `c:\msys\local\grass-6.3.cvs`

By the way, these pages might be useful:

- [http://grass.gdf-hannover.de/wiki/WinGRASS\\_Current\\_Status](http://grass.gdf-hannover.de/wiki/WinGRASS_Current_Status)
- <http://geni.ath.cx/grass.html>

#### F.2.4. GDAL level two

At this stage, we'll use GDAL sources we've used before, only the compilation will be a bit different.

But first in order to be able to compile GDAL sources with current GRASS CVS, you need to patch them, here's what you need to change:

[http://trac.osgeo.org/gdal/attachment/ticket/1587/plugin\\_patch\\_grass63.diff](http://trac.osgeo.org/gdal/attachment/ticket/1587/plugin_patch_grass63.diff)

(you can patch it by hand or use `patch.exe` in `c:\msys\bin`)

Now in MSYS console go to the GDAL sources directory and run the same commands as in level one, only with these differences:

1. when running `./configure` add this argument: `-with-grass=/usr/local/grass-6.3.cvs`
2. when calling `g++` on line 5 (which creates `libgdal.dll`), add these arguments:  
`-L/usr/local/grass-6.3.cvs/lib -lgrass_vect -lgrass_dig2 -lgrass_dgl -lgrass_rtree -lgrass_linkm -lgrass_dbmiclient -lgrass_dbmibase -lgrass_I -lgrass_gproj -lgrass_vask -lgrass_gmath -lgrass_gis -lgrass_datetime`

Then again, edit `gdal-config` and change line with `CONFIG_LIBS`

```
CONFIG_LIBS="-L/usr/local/lib -lpng -L/usr/local/grass-6.3.cvs/lib
-lgrass_vect -lgrass_dig2 -lgrass_dgl -lgrass_rtree -lgrass_linkm
-lgrass_dbmiclient -lgrass_dbmibase -lgrass_I -lgrass_gproj -lgrass_vask
-lgrass_gmath -lgrass_gis -lgrass_datetime -lz -L/usr/local/lib -lgdal"
```

Now, GDAL should be able to work also with GRASS raster layers.

### F.2.5. GEOS

Download the sources:

<http://geos.refractive.net/geos-2.2.3.tar.bz2>

Unpack to e.g. `c:\msys\local\src`

To compile, I had to patch the sources: in file `source/headers/timeval.h` line 13. Change it from:

```
#ifdef _WIN32
```

to:

```
#if defined(_WIN32) && defined(_MSC_VER)
```

Now, in MSYS console, go to the source directory and run:

```
./configure --prefix=/usr/local  
make  
make install
```

### F.2.6. SQLITE

You can use precompiled DLL, no need to compile from source:

Download this archive:

[http://www.sqlite.org/sqlitedll-3\\_3\\_17.zip](http://www.sqlite.org/sqlitedll-3_3_17.zip)

and copy `sqlite3.dll` from it to `c:\msys\local\lib`

Then download this archive:

[http://www.sqlite.org/sqlite-source-3\\_3\\_17.zip](http://www.sqlite.org/sqlite-source-3_3_17.zip)

and copy `sqlite3.h` to `c:\msys\local\include`

### F.2.7. GSL

Download sources:

<ftp://ftp.gnu.org/gnu/gsl/gsl-1.9.tar.gz>

Unpack to `c:\msys\local\src`

Run from MSYS console in the source directory:

```
./configure  
make  
make install
```

### F.2.8. EXPAT

Download sources:

<http://dfn.dl.sourceforge.net/sourceforge/expat/expat-2.0.0.tar.gz>

Unpack to `c:\msys\local\src`

Run from MSYS console in the source directory:

```
./configure  
make  
make install
```

### F.2.9. POSTGRES

We're going to use precompiled binaries. Use the link below for download:

<http://wwwmaster.postgresql.org/download/mirrors-ftp?file=%2Fbinary%2Fv8.2.4%2Fwin32%2Fpostgresql-8.2.4-1-binaries-no-installer.zip>

copy contents of `pgsql` directory from the archive to `c:\msys\local`

## F.3. Cleanup

We're done with preparation of MSYS environment. Now you can delete all stuff in `c:\msys\local\src` - it takes quite a lot of space and it's not necessary at all.

---

## G. Building with MS Visual Studio

!!\ This section describes a process where you build all dependencies yourself. See the section after this for a simpler procedure where we have all the dependencies you need pre-packaged and we focus just on getting Visual Studio Express set up and building QGIS.

Note that this does not currently include GRASS or Python plugins.

### G.1. Setup Visual Studio

This section describes the setup required to allow Visual Studio to be used to build QGIS.

#### G.1.1. Express Edition

The free Express Edition lacks the platform SDK which contains headers and so on that are needed when building QGIS. The platform SDK can be installed as described here:

<http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/>

Once this is done, you will need to edit the `<vsinstalldir>\Common7\Tools\vsvars` file as follows:

```
Add%PlatformSDKDir%\Include\atl and%PlatformSDKDir%\Include\mfc to the
@set INCLUDE entry.
```

This will add more headers to the system INCLUDE path. Note that this will only work when you use the Visual Studio command prompt when building. Most of the dependencies will be built with this. You will also need to perform the edits described here to remove the need for a library that Visual Studio Express lacks:

<http://www.codeproject.com/wtl/WTLExpress.asp>

#### G.1.2. All Editions

You will need `stdint.h` and `unistd.h`. `unistd.h` comes with GnuWin32 version of flex & bison binaries (see later). `stdint.h` can be found here:

<http://www.azillionmonkeys.com/qed/pstdint.h>.

Copy both of these to `<vsinstalldir>\VC\include`.

## G.2. Download/Install Dependencies

This section describes the downloading and installation of the various QGIS dependencies.

### G.2.1. Flex and Bison

Flex and Bison are tools for generation of parsers, they're needed for GRASS and also QGIS compilation.

Download the following packages and run the installers:

<http://gnuwin32.sourceforge.net/downloads/flex.php>

<http://gnuwin32.sourceforge.net/downloads/bison.php>

### G.2.2. To include PostgreSQL support in Qt

If you want to build Qt with PostgreSQL support you need to download PostgreSQL, install it and create a library you can later link with Qt.

Download from [.../binary/v8.2.5/win32/postgresql-8.2.5-1.zip](#) from an PostgreSQL.org Mirror and install.

PostgreSQL is currently build with MinGW and comes with headers and libraries for MinGW. The headers can be used with Visual C++ out of the box, but the library is only shipped in DLL and archive (.a) form and therefore cannot be used with Visual C++ directly.

To create a library copy following sed script to the file mkdef.sed in PostgreSQL lib directory:

```
/Dump of file / {
s/Dump of file \([^\ ]*\)$ /LIBRARY \1/p
a\
EXPORTS
}
/[^\ ]*ordinal hint/,/^ [^\ ]*Summary/ {
/^ [^\ ]*[0-9]\+ / {
s/^ [^\ ]*[0-9]\+[^\ ]*[0-9A-Fa-f]\+[^\ ]*[0-9A-Fa-f]
\+ [^\ ]*\([^\ ]*=[^\ ]*\).*$/ \1/p
}
}
```

and process execute in the Visual Studio C++ command line (from Programs menu):

```
cd c:\Program Files\PostgreSQL\8.2\bin
dumpbin /exports ..\bin\libpq.dll | sed -nf ../lib/mkdef.sed >..\lib\libpq.def
cd ..\lib
lib /def:libpq.def /machine:x86
```

You'll need an sed for that to work in your path (e.g. from cygwin or msys).

That's almost it. You only need to the include and lib path to INCLUDE and LIB in vcvars.bat respectively.

### G.2.3. Qt

Build Qt following the instructions here:

[http://wiki.qgis.org/qgiswiki/Building\\_QT\\_4\\_with\\_Visual\\_C%2B%2B\\_2005](http://wiki.qgis.org/qgiswiki/Building_QT_4_with_Visual_C%2B%2B_2005)

### G.2.4. Proj.4

Get proj.4 source from here:

<http://proj.maptools.org/>

Using the Visual Studio command prompt (ensures the environment is setup properly), run the following in the src directory:

```
nmake -f makefile.vc
```

Install by running the following in the top level directory setting PROJ\_DIR as appropriate:

```
set PROJ_DIR=c:\lib\proj

mkdir %PROJ_DIR%\bin
mkdir %PROJ_DIR%\include
mkdir %PROJ_DIR%\lib

copy src\*.dll %PROJ_DIR%\bin
copy src\*.exe %PROJ_DIR%\bin
copy src\*.h %PROJ_DIR%\include
copy src\*.lib %PROJ_DIR%\lib
```

This can also be added to a batch file.



### G.2.5. GSL

Get gsl source from here:

<http://david.geldreich.free.fr/downloads/gsl-1.9-windows-sources.zip>

Build using the gsl.sln file

### G.2.6. GEOS

Get geos from svn (svn checkout <http://svn.refrations.net/geos/trunk> geos). Edit geos\source\makefile.vc as follows:

Uncomment lines 333 and 334 to allow the copying of version.h.vc to version.h.

Uncomment lines 338 and 339.

Rename geos\_c.h.vc to geos\_c.h.in on lines 338 and 339 to allow the copying of geos\_c.h.in to geos\_c.h.

Using the Visual Studio command prompt (ensures the environment is setup properly), run the following in the top level directory:

```
nmake -f makefile.vc
```

Run the following in top level directory, setting GEOS\_DIR as appropriate:

```
set GEOS_DIR="c:\lib\geos"
```

```
mkdir %GEOS_DIR%\include
```

```
mkdir %GEOS_DIR%\lib
```

```
mkdir %GEOS_DIR%\bin
```

```
xcopy /S/Y source\headers\*.h %GEOS_DIR%\include
```

```
copy /Y capi\*.h %GEOS_DIR%\include
```

```
copy /Y source\*.lib %GEOS_DIR%\lib
```

```
copy /Y source\*.dll %GEOS_DIR%\bin
```

This can also be added to a batch file.

### G.2.7. GDAL

Get gdal from svn (svn checkout <https://svn.osgeo.org/gdal/branches/1.4/gdal> gdal).

Edit nmake.opt to suit, it's pretty well commented.

Using the Visual Studio command prompt (ensures the environment is setup properly), run the following in the top level directory:

```
nmake -f makefile.vc
```

and

```
nmake -f makefile.vc devinstall
```

### G.2.8. PostGIS

Get PostGIS and the Windows version of PostgreSQL from here:

<http://postgis.refrations.net/download/>

Note the warning about not installing the version of PostGIS that comes with the PostgreSQL installer. Simply run the installers.

### G.2.9. Expat

Get expat from here:

[http://sourceforge.net/project/showfiles.php?group\\_id=10127](http://sourceforge.net/project/showfiles.php?group_id=10127)

You'll need expat-win32bin-2.0.1.exe.

Simply run the executable to install expat.

### G.2.10. CMake

Get CMake from here:

<http://www.cmake.org/HTML/Download.html>

You'll need cmake-<version>-win32-x86.exe. Simply run this to install CMake.

### G.3. Building QGIS with CMAKE

Get QGIS source from svn (svn co <https://svn.qgis.org/repos/qgis/trunk/qgis> qgis).

Create a 'Build' directory in the top level QGIS directory. This will be where all the build output will be generated.

Run Start->All Programs->CMake->CMake.

In the 'Where is the source code:' box, browse to the top level QGIS directory.

In the 'Where to build the binaries:' box, browse to the 'Build' directory you created in the top level QGIS directory.

Fill in the various \*\_INCLUDE\_DIR and \*\_LIBRARY entries in the 'Cache Values' list.

Click the Configure button. You will be prompted for the type of makefile that will be generated. Select Visual Studio 8 2005 and click OK.

All being well, configuration should complete without errors. If there are errors, it is usually due to an incorrect path to a header or library directory. Failed items will be shown in red in the list.

Once configuration completes without error, click OK to generate the solution and project files.

With Visual Studio 2005, open the qgis.sln file that will have been created in the Build directory you created earlier.

Build the ALL\_BUILD project. This will build all the QGIS binaries along with all the plugins.

Install QGIS by building the INSTALL project. By default this will install to c:\Program Files\qgis<version> (this can be changed by changing the CMAKE\_INSTALL\_PREFIX variable in CMake).

You will also either need to add all the dependency dlls to the QGIS install directory or add their respective directories to your PATH.

## H. Building under Windows using MSVC Express

!!\ Note: Building under MSVC is still a work in progress. In particular the following don't work yet: python, grass, postgis connections.

!!\ This section of the document is in draft form and is not ready to be used yet.

Tim Sutton, 2007

## H.1. System preparation

I started with a clean XP install with Service Pack 2 and all patches applied. I have already compiled all the dependencies you need for gdal, expat etc, so this tutorial wont cover compiling those from source too. Since compiling these dependencies was a somewhat painful task I hope my pre-compiled libs will be adequate. If not I suggest you consult the individual projects for specific build documentation and support. Lets go over the process in a nutshell before we begin:

- \* Install XP (I used a Parallels virtual machine)
- \* Install the premade libraries archive I have made for you
- \* Install Visual Studio Express 2005 sp1
- \* Install the Microsoft Platform SDK
- \* Install command line subversion client
- \* Install library dependencies bundle
- \* Install Qt 4.3.2
- \* Check out QGIS sources
- \* Compile QGIS
- \* Create setup.exe installer for QGIS

## H.2. Install the libraries archive

Half of the point of this section of the MSVC setup procedure is to make things as simple as possible for you. To that end I have prepared an archive that includes all dependencies needed to build QGIS except Qt (which we will build further down). Fetch the archive from:

[http://qgis.org/uploadfiles/msvc/qgis\\_msvc\\_deps\\_except\\_qt4.zip](http://qgis.org/uploadfiles/msvc/qgis_msvc_deps_except_qt4.zip)

Create the following directory structure:

```
c:\dev\cpp\
```

And then extract the libraries archive into a subdirectory of the above directory so that you end up with:

```
c:\dev\cpp\qgislibs-release
```

/!\ Note that you are not obliged to use this directory layout, but you should adjust any instructions that follow if you plan to do things differently.

## H.3. Install Visual Studio Express 2005

First thing we need to get is MSVC Express from here:

<http://msdn2.microsoft.com/en-us/express/aa975050.aspx>

The page is really confusing so dont feel bad if you cant actually find the download at first! There are six coloured blocks on the page for the various studio family members (vb / c# / j# etc). Simply choose your language under the 'select your language' combo under the yellow C++ block, and your download will begin. Under internet explorer I had to disable popup blocking for the download to be able to commence.

Once the setup commences you will be prompted with various options. Here is what I chose :

\* Send usage information to Microsoft (No) \* Install options: \* Graphical IDE (Yes) \* Microsoft MSDN Express Edition (No) \* Microsoft SQL Server Express Edition (No) \* Install to folder: C:\Program Files\Microsoft Visual Studio 8\ (default)

It will need to download around 90mb of installation files and reports that the install will consume 554mb of disk space.

## H.4. Install Microsoft Platform SDK2

Go to this page:

<http://msdn2.microsoft.com/en-us/express/aa700755.aspx>

Start by using the link provided on the above page to download and install the platform SDK2.

The actual SDK download page is once again a bit confusing since the links for downloading are hidden amongst a bunch of other links. Basically look for these three links with their associated 'Download' buttons and choose the correct link for your platform:

PSDK-amd64.exe	1.2 MB	Download
PSDK-ia64.exe	1.3 MB	Download
PSDK-x86.exe	1.2 MB	Download

When you install make sure to choose 'custom install'. These instructions assume you are installing into the default path of:

C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\

We will go for the minimal install that will give us a working environment, so on the custom installation screen I made the following choices:

Configuration Options  
+ Register Environmental Variables (Yes)  
Microsoft Windows Core SDK

---

+ Tools	(Yes)
+ Tools (AMD 64 Bit)	(No unless this applies)
+ Tools (Intel 64 Bit)	(No unless this applies)
+ Build Environment	
+ Build Environment (AMD 64 Bit)	(No unless this applies)
+ Build Environment (Intel 64 Bit)	(No unless this applies)
+ Build Environment (x86 32 Bit)	(Yes)
+ Documentation	(No)
+ Redistributable Components	(Yes)
+ Sample Code	(No)
+ Source Code	(No)
+ AMD 64 Source	(No)
+ Intel 64 Source	(No)
Microsoft Web Workshop	(Yes) (needed for shlwapi.h)
+ Build Environment	(Yes)
+ Documentation	(No)
+ Sample Code	(No)
+ Tools	(No)
Microsoft Internet Information Server (IIS) SDK	(No)
Microsoft Data Access Services (MDAC) SDK	(Yes) (needed by GDAL for odbc)
+ Tools	
+ Tools (AMD 64 Bit)	(No)
+ Tools (AMD 64 Bit)	(No)
+ Tools (x86 32 Bit)	(Yes)
+ Build Environment	
+ Tools (AMD 64 Bit)	(No)
+ Tools (AMD 64 Bit)	(No)
+ Tools (x86 32 Bit)	(Yes)
+ Documentation	(No)
+ Sample Code	(No)
Microsoft Installer SDK	(No)
Microsoft Table PC SDK	(No)
Microsoft Windows Management Instrumentation	(No)
Microsoft DirectShow SDK	(No)
Microsoft Media Services SDK	(No)
Debuggin Tools for Windows	(Yes)

/!\ Note that you can always come back later to add extra bits if you like.

/!\ Note that installing the SDK requires validation with the Microsoft Genuine Advantage application. Some people have a philosophical objection to installing this software on their computers. If you are one of them you should probably consider using the MINGW build instructions described elsewhere

in this document.

The SDK installs a directory called

`C:\Office10`

Which you can safely remove.

After the SDK is installed, follow the remaining notes on the page link above to get your MSVC Express environment configured correctly. For your convenience, these are summarised again below, and I have added a couple more paths that I discovered were needed:

- 1) open Visual Studio Express IDE
- 2) Tools -> Options -> Projects and Solutions -> VC++ Directories
- 3) Add:

Executable files:

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin`

Include files:

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include`

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\atl`

`C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\mfc`

Library files: `C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib`

- 4) Close MSVC Express IDE
- 5) Open the following file with notepad:

`C:\Program Files\Microsoft Visual Studio 8\VC\VCProjectDefaults\corewin_express.vsprops`

and change the property:

`AdditionalDependencies="kernel32.lib"`

To read:

`AdditionalDependencies="kernel32.lib user32.lib gdi32.lib winspool.lib comdlg32.lib  
advapi32.lib shell32.lib ole32.lib oleaut32.lib uuid.lib"`

The notes go on to show how to build a mswin32 application which you can try if you like - I'm not going to recover that here.

## H.5. Edit your vsvars

Backup your vsvars32.bat file in

C:\Program Files\Microsoft Visual Studio 8\Common7\Tools

and replace it with this one:

```
@SET VSINSTALLDIR=C:\Program Files\Microsoft Visual Studio 8
@SET VCINSTALLDIR=C:\Program Files\Microsoft Visual Studio 8\VC
@SET FrameworkDir=C:\WINDOWS\Microsoft.NET\Framework
@SET FrameworkVersion=v2.0.50727
@SET FrameworkSDKDir=C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0
@if "%VSINSTALLDIR%"==" " goto error_no_VSINSTALLDIR
@if "%VCINSTALLDIR%"==" " goto error_no_VCINSTALLDIR

@echo Setting environment for using Microsoft Visual Studio 2005 x86 tools.

@rem
@rem Root of Visual Studio IDE installed files.
@rem
@set DevEnvDir=C:\Program Files\Microsoft Visual Studio 8\Common7\IDE

@set PATH=C:\Program Files\Microsoft Visual Studio 8\Common7\IDE;
C:\Program Files\Microsoft Visual Studio 8\VC\BIN;
C:\Program Files\Microsoft Visual Studio 8\Common7\Tools;
C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\bin;
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727;
C:\Program Files\Microsoft Visual Studio 8\VC\VCPackages;%PATH%
@rem added by Tim
@set PATH=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Bin;%PATH%
@set INCLUDE=C:\Program Files\Microsoft Visual Studio 8\VC\INCLUDE;%INCLUDE%
@rem added by Tim
@set INCLUDE=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Include;%INCLUDE%
@set INCLUDE=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
R2\Include\mfc;%INCLUDE%
@set INCLUDE=%INCLUDE%;C:\dev\cpp\qgislibs-release\include\postgresql
@set LIB=C:\Program Files\Microsoft Visual Studio 8\VC\LIB;
C:\Program Files\Microsoft Visual Studio 8\SDK\v2.0\lib;%LIB%
```



```
@rem added by Tim
@set LIB=C:\Program Files\Microsoft Platform SDK for Windows Server 2003
    R2\Lib;%LIB%
@set LIB=%LIB%;C:\dev\cpp\qgislibs-release\lib
@set LIBPATH=C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727

@goto end

:error_no_VSINSTALLDIR
@echo ERROR: VSINSTALLDIR variable is not set.
@goto end

:error_no_VCINSTALLDIR
@echo ERROR: VCINSTALLDIR variable is not set.
@goto end

:end
```

## H.6. Environment Variables

Right click on 'My computer' then select the 'Advanced' tab. Click environment variables and create or augment the following "System" variables (if they don't already exist):

Variable Name:	Value:
EDITOR	vim
INCLUDE	C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Include\.
LIB	C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Lib\.
LIB_DIR	C:\dev\cpp\qgislibs-release
PATH	C:\Program Files\CMake 2.4\bin; %SystemRoot%\system32; %SystemRoot%; %SystemRoot%\System32\Wbem; C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin\.; C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\Bin\WinNT\; C:\Program Files\svn\bin;C:\Program Files\Microsoft Visual

```
        Studio 8\VC\bin;  
        C:\Program Files\Microsoft Visual Studio 8\Common7\IDE;  
        "c:\Program Files\Microsoft Visual Studio 8\Common7\Tools";  
        c:\Qt\4.3.2\bin;  
        "C:\Program Files\PuTTY"  
QTDIR      c:\Qt\4.3.2  
SVN_SSH     "C:\\Program Files\\PuTTY\\plink.exe"
```

== Building Qt4.3.2 ==

You need a minimum of Qt 4.3.2 here since this is the first version to officially support building the open source version of Qt for windows under MSVC.

Download Qt 4.x.x source for windows from

<http://www.trolltech.com>

Unpack the source to

c:\Qt\4.x.x\

=== Compile Qt ===

Open the Visual Studio C++ command line and cd to c:\Qt\4.x.x where you extracted the source and enter:

```
configure -platform win32-msvc2005  
nmake  
nmake install
```

Add -qt-sql-odbc -qt-sql-psql to the configure line if your want odbc and PostgreSQL support build into Qt.

/!\ Note: For me in some cases I got a build error on qsscreenshot.pro. If you are only interested in having the libraries needed for building Qt apps, you can probably ignore that. Just check in c:\Qt\4.3.2\bin to check all dlls and helper apps (assistant etc) have been made.

=== Configure Visual C++ to use Qt ===

After building configure the Visual Studio Express IDE to use Qt:

- 1) open Visual Studio Express IDE
- 2) Tools -> Options -> Projects and Solutions -> VC++ Directories
- 3) Add:

Executable files: \$(QTDIR)\bin

Include files: \$(QTDIR)\include \$(QTDIR)\include\Qt \$(QTDIR)\include\QtCore  
\$(QTDIR)\include\QtGui \$(QTDIR)\include\QtNetwork \$(QTDIR)\include\QtSvg  
\$(QTDIR)\include\QtXml \$(QTDIR)\include\Qt3Support \$(LIB\_DIR)\include (needed during  
qgis compile to find stdint.h and unistd.h)

Library files: \$(QTDIR)\lib

Source Files: \$(QTDIR)\src

Hint: You can also add

QString = t=<d->data, su>, size=<d->size, i>

to AutoExp.DAT in C:\Program Files\Microsoft Visual Studio 8  
\Common7\Packages\Debugger before

[Visualizer]

That way the Debugger will show the contents of QString when you point at or watch a variable in the debugger. There are probably much more additions - feel free to add some - I just needed QString and took the first hit in google I could find.

== Install Python ==

Download <http://python.org/ftp/python/2.5.1/python-2.5.1.msi> and install it.

== Install SIP ==

Download <http://www.riverbankcomputing.com/Downloads/sip4/sip-4.7.1.zip>

and extract it into your c:\dev\cpp directory.

From a Visual C++ command line cd to the directory where you extract SIP and run:

```
c:\python25\python configure.py -p win32-msvc2005 nmake nmake install
```

== Install PyQt4 ==

Download <http://www.riverbankcomputing.com/Downloads/PyQt4/GPL/PyQt-win-gpl-4.3.1.zip>

and extract it into your c:\dev\cpp directory.

From a Visual C++ command line cd to the directory where you extracted PyQt4 and run:

```
c:\python25\python configure.py -p win32-msvc2005 nmake nmake install ""
```

## H.7. Install CMake

Download and install cmake 2.4.7 or better, making sure to enable the option

Update path for all users

## H.8. Install Subversion

You "must" install the command line version if you want the CMake svn scripts to work. Its a bit tricky to find the correct version on the subversion download site as they have som misleadingly named similar downloads. Easiest is to just get this file:

<http://subversion.tigris.org/downloads/1.4.5-win32/apache-2.2/svn-win32-1.4.5.zip>

Extract the zip file to

C:\Program Files\svn

And then add

C:\Program Files\svn\bin

To your path.

## H.9. Initial SVN Check out

Open a cmd.exe window and do:

```
cd \  
cd dev  
cd cpp  
svn co https://svn.qgis.org/repos/qgis/trunk/qgis
```

At this point you will probably get a message like this:

```
C:\dev\cpp>svn co https://svn.qgis.org/repos/qgis/trunk/qgis  
Error validating server certificate for 'https://svn.qgis.org:443':  
- The certificate is not issued by a trusted authority. Use the  
  fingerprint to validate the certificate manually!  
Certificate information:  
- Hostname: svn.qgis.org  
- Valid: from Sat, 01 Apr 2006 03:30:47 GMT until Fri, 21 Mar 2008 03:30:47 GMT  
- Issuer: Developer Team, Quantum GIS, Anchorage, Alaska, US  
- Fingerprint: 2f:cd:f1:5a:c7:64:da:2b:d1:34:a5:20:c6:15:67:28:33:ea:7a:9b  
(R)eject, accept (t)emporarily or accept (p)ermanently?
```

Press 'p'to accept and the svn checkout will commence.

## H.10. Create Makefiles using cmake.exe

I won't be giving a detailed description of the build process, because the process is explained in the first section (where you manually build all dependencies) of the windows build notes in this document. Just skip past the parts where you need to build GDAL etc, since this simplified install process does all the dependency provisioning for you.

```
cd qgis
mkdir build
cd build
cmake ..
```

Cmake should find all dependencies for you automatically (it uses the LIB\_DIR environment to find them all in c:\dev\cpp\qgislibs-release). Press configure again after the cmake gui appears and when all the red fields are gone, and you have made any personalisations to the setup, press ok to close the cmake gui.

Now open Visual Studio Express and do:

File -> Open -> Project / Solution

Now open the cmake generated QGIS solution which should be in :

```
c:\dev\cpp\qgis\build\qgisX.X.X.sln
```

Where X.X.X represents the current version number of QGIS. Currently I have only made release built dependencies for QGIS (debug versions will follow in future), so you need to be sure to select 'Release' from the solution configurations toolbar.

Next right click on ALL\_BUILD in the solution browser, and then choose build.

Once the build completes right click on INSTALL in the solution browser and choose build. This will by default install qgis into c:\program files\qgisX.X.X.

## H.11. Running and packaging

To run QGIS you need to at the minimum copy the dlls from c:\dev\cpp\qgislibs-release\bin into the c:\program files\qgisX.X.X directory.

---

## I. GNU General Public License

### GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors'reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

---

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as *you*.

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under



---

the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipient's exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from

---

distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPE-

---

CIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## **I.1. Quantum GIS Qt exception for GPL**

In addition, as a special exception, the QGIS Development Team gives permission to link the code of this program with the Qt library, including but not limited to the following versions (both free and commercial): Qt/Non-commercial Windows, Qt/Windows, Qt/X11, Qt/Mac, and Qt/Embedded (or with modified versions of Qt that use the same license as Qt), and distribute linked combinations including the two. You must obey the GNU General Public License in all respects for all of the code used other than Qt. If you modify this file, you may extend this exception to your version of the file, but you are not obligated to do so. If you do not wish to do so, delete this exception statement from your version.

**Literatura**

**Referencias web**