



QGIS Server 3.16 User Guide

QGIS Project

24 de marzo de 2022

1	Introducción	1
2	Comenzando	3
2.1	Instalación en sistemas basados en Debian	3
2.1.1	Servidor HTTP Apache	4
2.1.2	NGINX HTTP Server	6
2.1.3	Xvfb	11
2.2	Instalación en Windows	12
2.3	Sirve un proyecto	14
2.4	Configure su proyecto	15
2.4.1	Capacidades WMS	17
2.4.2	Capacidades WFS	18
2.4.3	Capacidades WCS	18
2.4.4	Ajuste fino de OWS	18
2.5	Integración con terceras partes	18
2.5.1	Integración con QGIS Desktop	18
2.5.2	Integración con MapProxy	18
2.5.3	Integración con QWC2	19
3	Servicios	21
3.1	Web Map Service (WMS)	21
3.1.1	GetMap	22
3.1.2	GetFeatureInfo	30
3.1.3	GetPrint	32
3.1.4	GetLegendGraphics	35
3.1.5	GetProjectSettings	36
3.2	Web Feature Service (WFS)	36
3.2.1	GetFeature	37
3.3	Web Map Tile Service (WMTS)	39
3.3.1	GetCapabilities	39
3.3.2	GetTile	40
3.3.3	GetFeatureInfo	42
3.4	WFS3 (Objetos API OGC)	44
3.4.1	Representación de recurso	45
3.4.2	Puntos finales	45
3.4.3	Paginación	49
3.4.4	Filtrado de objetos espaciales	49
3.4.5	Ordenar Entidad	50
3.4.6	Selección de atributos	50
3.4.7	Personaliza las páginas HTML	51
3.5	Parámetros extra soportados por todos los tipos de solicitudes	52

3.6	REDLINING	52
3.7	Capas WMS Externas	54
3.8	Catálogo del servidor QGIS	55
4	Complementos	59
4.1	Instalación	59
4.2	Configuración del servidor HTTP	59
4.2.1	Apache	59
4.3	Como usar un complemento	60
5	Canfiguración avanzada	61
5.1	Registro	61
5.2	Variables de entorno	61
5.3	Resumen de configuración	63
5.4	Nombre corto para capas, grupos y proyecto	64
5.5	Conexión a archivo servicio	64
5.6	Agregar fuentes a su servidor linux	64
6	Servidor de desarrollo	67
7	Despliegue en contenedores	69
7.1	Imágenes de simple docker	69
7.1.1	Primero ejecute	71
7.1.2	Muestra utilizable	71
7.1.3	Limpieza	72
7.2	Pilas de Docker	72
7.2.1	Swarm/docker-compose	72
7.2.2	Kubernetes	73
7.3	Implementar nube	77
7.3.1	Caso de uso de AWS	77
8	FAQ Preguntas frecuentes	79

Introducción

QGIS Server es un WMS, WFS, OGC API de código abierto para la implementación de Características 1.0 (WFS3) y WCS que, además, implementa características cartográficas avanzadas para mapeo temático. QGIS Server es una aplicación FastCGI/CGI (Common Gateway Interface) escrita en C++ que funciona junto con un servidor web (por ejemplo, Apache, Nginx). Tiene compatibilidad con el complemento Python, lo que permite un desarrollo e implementación rápidos y eficientes de nuevas funciones.

El servidor QGIS utiliza QGIS como back-end para la lógica de los SIG y de mapa de representación. Además, la biblioteca Qt se utiliza para gráficos y para la plataforma independiente la programación en C++. En contraste con otro software de WMS, el servidor de **lqgl** utiliza reglas cartográficas como un lenguaje de configuración, tanto para la configuración del servidor y de las reglas cartográficas definidas por el usuario.

Como el escritorio QGIS y el servidor QGIS usan las mismas bibliotecas de visualización, los mapas que se publican en la web tienen el mismo aspecto que en el escritorio SIG.

En las siguientes secciones, proporcionaremos una configuración de muestra para configurar un servidor QGIS en Linux (Debian, Ubuntu y derivados) y en Windows. Para obtener más información sobre el desarrollo de complementos de servidor, lea `server_plugins`.

Se permite copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, Versión 1.3 o cualquier otra versión posterior publicada por la Free Software Foundation; sin secciones invariantes, sin textos de portada y sin textos de contraportada.

Se incluye una copia de la licencia en la sección `gnu_fdl`.

2.1 Instalación en sistemas basados en Debian

Daremos un breve y sencillo procedimiento de instalación para una configuración de trabajo mínima en sistemas basados en Debian (incluidos Ubuntu y derivados). Sin embargo, muchas otras distribuciones y sistemas operativos proporcionan paquetes para QGIS Server.

Nota: En Ubuntu puede usar su usuario habitual, anteponiendo `sudo` a los comandos que requieren permisos de administrador. En Debian puedes trabajar como administrador (`root`), sin usar `sudo`.

Requirements and steps to add official QGIS repositories to install QGIS Server on a Debian based system are provided in [QGIS installers page](#). You may want to install at least the latest Long Term Release.

Once the target version repository is configured and QGIS Server installed, you can test the installation with:

```
/usr/lib/cgi-bin/qgis_mapserv.fcgi
```

If you get the following output, the server is correctly installed.

Nota: Depending on the version of QGIS, you might see slightly different output reported when you run `qgis_mapserv.fcgi`.

```
QFSFileEngine::open: No file name specified
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.
Warning 1: Unable to find driver JP2ECW to unload from GDAL_SKIP environment_
↳variable.
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.
Warning 1: Unable to find driver JP2ECW to unload from GDAL_SKIP environment_
↳variable.
Content-Length: 206
Content-Type: text/xml; charset=utf-8

<ServiceExceptionReport version="1.3.0" xmlns="https://www.opengis.net/ogc">
```

(continúe en la próxima página)

(proviene de la página anterior)

```
<ServiceException code="Service configuration error">Service unknown or  
↳ unsupported</ServiceException>  
</ServiceExceptionReport>
```

Nota: As seen below, QGIS reports a Status 400 code, which correctly identifies the request has failed because there is no active http session. This is not a bug and indicates the server is functioning properly.

```
Application path not initialized  
Application path not initialized  
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.  
Warning 1: Unable to find driver ECW to unload from GDAL_SKIP environment variable.  
Warning 1: Unable to find driver JP2ECW to unload from GDAL_SKIP environment.  
↳ variable.  
"Loading native module /usr/lib/qgis/server/libdummy.so"  
"Loading native module /usr/lib/qgis/server/liblandingpage.so"  
"Loading native module /usr/lib/qgis/server/libwcs.so"  
"Loading native module /usr/lib/qgis/server/libwfs.so"  
"Loading native module /usr/lib/qgis/server/libwfs3.so"  
"Loading native module /usr/lib/qgis/server/libwms.so"  
"Loading native module /usr/lib/qgis/server/libwmts.so"  
QFSFileEngine::open: No file name specified  
Content-Length: 102  
Content-Type: application/json  
Server: QGIS FCGI server - QGIS version 3.16.6-Hannover  
Status: 400  
[{"code": "Bad request error", "description": "Requested URI does not match any  
↳ registered API handler"}]
```

Agreguemos un proyecto de muestra. Puede utilizar el suyo propio o uno de [Datos de demostración de entrenamiento](#):

```
mkdir /home/qgis/projects/  
cd /home/qgis/projects/  
wget https://github.com/qgis/QGIS-Training-Data/archive/release_3.16.zip  
unzip release_3.16.zip  
mv QGIS-Training-Data-release_3.16/exercise_data/qgis-server-tutorial-data/world.  
↳ qgs .  
mv QGIS-Training-Data-release_3.16/exercise_data/qgis-server-tutorial-data/  
↳ naturalearth.sqlite .
```

Por supuesto, puede utilizar su software GIS favorito para abrir este archivo y echar un vistazo a la configuración y las capas disponibles.

Para implementar correctamente el servidor QGIS, necesita un servidor HTTP. Las opciones recomendadas son **** Apache **** o **** NGINX ****.

2.1.1 Servidor HTTP Apache

Nota: In the following, please replace `qgis.demo` with the name or IP address of your server.

Instala Apache y `mod_fcgid`:

```
apt install apache2 libapache2-mod-fcgid
```

Puede ejecutar QGIS Server en su sitio web predeterminado, o configurar un host virtual específico para esto, como sigue.

En el directorio `/etc/apache2/sites-available` cree un archivo llamado `qgis.demo.conf`, con este contenido:

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName qgis.demo

  DocumentRoot /var/www/html

  # Apache logs (different than QGIS Server log)
  ErrorLog ${APACHE_LOG_DIR}/qgis.demo.error.log
  CustomLog ${APACHE_LOG_DIR}/qgis.demo.access.log combined

  # Longer timeout for WPS... default = 40
  FcgidIOTimeout 120

  FcgidInitialEnv LC_ALL "en_US.UTF-8"
  FcgidInitialEnv PYTHONIOENCODING UTF-8
  FcgidInitialEnv LANG "en_US.UTF-8"

  # QGIS log
  FcgidInitialEnv QGIS_SERVER_LOG_STDERR 1
  FcgidInitialEnv QGIS_SERVER_LOG_LEVEL 0

  # default QGIS project
  SetEnv QGIS_PROJECT_FILE /home/qgis/projects/world.qgs

  # QGIS_AUTH_DB_DIR_PATH must lead to a directory writeable by the Server's FCGI_
  ↪process user
  FcgidInitialEnv QGIS_AUTH_DB_DIR_PATH "/home/qgis/qgisserverdb/"
  FcgidInitialEnv QGIS_AUTH_PASSWORD_FILE "/home/qgis/qgisserverdb/qgis-auth.db"

  # Set pg access via pg_service file
  SetEnv PGSERVICEFILE /home/qgis/.pg_service.conf
  FcgidInitialEnv PGPASSFILE "/home/qgis/.pgpass"

  # if qgis-server is installed from packages in debian based distros this is_
  ↪usually /usr/lib/cgi-bin/
  # run "locate qgis_mapserv.fcgi" if you don't know where qgis_mapserv.fcgi is
  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin/">
    AllowOverride None
    Options +ExecCGI -MultiViews -SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
    Require all granted
  </Directory>

  <IfModule mod_fcgid.c>
    FcgidMaxRequestLen 26214400
    FcgidConnectTimeout 60
  </IfModule>

</VirtualHost>
```

Otras lecturas:

- *QGIS Server logging*
- `pg-service-file` in QGIS Server

Puede hacer lo anterior en un sistema de escritorio Linux pegando y guardando la configuración anterior después de hacer:

```
nano /etc/apache2/sites-available/qgis.demo.conf
```

Nota: Algunas de las opciones de configuración se explican en la sección del servidor *environment variables*.

Creemos ahora los directorios que almacenarán los registros del servidor QGIS y la base de datos de autenticación:

```
mkdir -p /var/log/qgis/  
chown www-data:www-data /var/log/qgis  
mkdir -p /home/qgis/qgisserverdb  
chown www-data:www-data /home/qgis/qgisserverdb
```

Nota: `www-data` es el usuario de Apache en sistemas basados en Debian y necesitamos que Apache tenga acceso a esas ubicaciones o archivos. Los comandos `chown www-data...` cambian el propietario de los respectivos directorios y archivos a `www-data`.

Ahora podemos habilitar el **host virtual**, habilita el mod `fcgid` si no está activado todavía:

```
a2enmod fcgid  
a2ensite qgis.demo
```

Ahora reinicie Apache para que se tenga en cuenta la nueva configuración:

```
systemctl restart apache2
```

Ahora que Apache sabe que debe responder las solicitudes a <http://qgis.demo>, también necesitamos configurar el sistema cliente para que sepa quién es `qgis.demo`. Lo hacemos agregando `127.0.0.1 qgis.demo` en el archivo `hosts`. Podemos hacerlo con `sh -c"echo '127.0.0.1 qgis.demo' >> /etc/hosts"`. Reemplaza `127.0.0.1` con la IP de tu servidor.

Nota: Remember that both the `qgis.demo.conf` and `/etc/hosts` files should be configured for your setup to work. You can also test the access to your QGIS Server from other clients on the network (e.g. Windows or macOS machines) by going to their `/etc/hosts` file and point the `myhost` name to whatever IP the server machine has on the network (not `127.0.0.1` as it is the local IP, only accessible from the local machine). On `*nix` machines the `hosts` file is located in `/etc`, while on Windows it's under the `C:\Windows\System32\drivers\etc` directory. Under Windows you need to start your text editor with administrator privileges before opening the `hosts` file.

QGIS Server is now available at <http://qgis.demo>. To check, type in a browser, as in the simple case:

```
http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&  
↪REQUEST=GetCapabilities
```

2.1.2 NGINX HTTP Server

Nota: In the following, please replace `qgis.demo` with the name or IP address of your server.

También puede usar QGIS Server con **NGINX**. Unlike Apache, NGINX no genera automáticamente procesos FastCGI. Los procesos FastCGI deben ser iniciados por otra cosa.

Instala NGINX:

```
apt install nginx
```

- Como primera opción, puede usar **spawn-fcgi** o **fcgiwrap** para iniciar y administrar los procesos de QGIS Server. Existen paquetes oficiales de Debian para ambos. Cuando no tiene un servidor X en ejecución y necesita, por ejemplo, imprimir, puede usar *xvfb*.
- Otra opción es confiar en **** Systemd ****, el sistema de inicio para GNU / Linux que la mayoría de las distribuciones de Linux utilizan en la actualidad. Una de las ventajas de este método es que no requiere otros componentes o procesos. Está destinado a ser simple, pero robusto y eficiente para implementaciones de producción.

NGINX Configuración

El **** incluido fastcgi_params; **** usado en la configuración anterior es importante, ya que agrega los parámetros de `/etc/nginx/fastcgi_params`:

```
fastcgi_param  QUERY_STRING       $query_string;
fastcgi_param  REQUEST_METHOD    $request_method;
fastcgi_param  CONTENT_TYPE      $content_type;
fastcgi_param  CONTENT_LENGTH    $content_length;

fastcgi_param  SCRIPT_NAME        $fastcgi_script_name;
fastcgi_param  REQUEST_URI        $request_uri;
fastcgi_param  DOCUMENT_URI       $document_uri;
fastcgi_param  DOCUMENT_ROOT      $document_root;
fastcgi_param  SERVER_PROTOCOL    $server_protocol;
fastcgi_param  REQUEST_SCHEME     $scheme;
fastcgi_param  HTTPS              $https if_not_empty;

fastcgi_param  GATEWAY_INTERFACE  CGI/1.1;
fastcgi_param  SERVER_SOFTWARE    nginx/$nginx_version;

fastcgi_param  REMOTE_ADDR        $remote_addr;
fastcgi_param  REMOTE_PORT        $remote_port;
fastcgi_param  SERVER_ADDR        $server_addr;
fastcgi_param  SERVER_PORT        $server_port;
fastcgi_param  SERVER_NAME        $server_name;

# PHP only, required if PHP was built with --enable-force-cgi-redirect
fastcgi_param  REDIRECT_STATUS    200;
```

Además, puede usar algunos *Variables de entorno* para configurar QGIS Server. En el archivo de configuración de NGINX, `/etc/nginx/nginx.conf`, debe usar la instrucción `fastcgi_param` para definir estas variables como se muestra a continuación:

```
location /qgisserver {
    gzip            off;
    include         fastcgi_params;
    fastcgi_param  QGIS_SERVER_LOG_STDERR  1;
    fastcgi_param  QGIS_SERVER_LOG_LEVEL   0;
    fastcgi_pass   unix:/var/run/qgisserver.socket;
}
```

FastCGI envoltorios

Advertencia: `fcgiwrap` es más fácil de configurar que `spawn-fcgi`, porque ya está incluido en un servicio Systemd. Pero también conduce a una solución que es mucho más lenta que usar `spawn-fcgi`. Con `fcgiwrap`, se crea un nuevo proceso de QGIS Server en cada solicitud, lo que significa que el proceso de inicialización de QGIS Server, que incluye leer y analizar el archivo del proyecto QGIS, se realiza en cada solicitud. Con `spawn-fcgi`, el proceso de QGIS Server permanece activo entre solicitudes, lo que resulta en un rendimiento mucho mejor. Por esa razón, se recomienda `spawn-fcgi` para uso en producción.

spawn-fcgi

Si quieres usar `spawn-fcgi`, el primer paso es instalar el paquete:

```
apt install spawn-fcgi
```

Luego, introduzca el siguiente bloque en la configuración de su servidor NGINX:

```
location /qgisserver {
    gzip            off;
    include         fastcgi_params;
    fastcgi_pass    unix:/var/run/qgisserver.socket;
}
```

Y reinicie NGINX para tener en cuenta la nueva configuración:

```
systemctl restart nginx
```

Finalmente, considerando que no existe un archivo de servicio predeterminado para `spawn-fcgi`, debe iniciar manualmente QGIS Server en su terminal:

```
spawn-fcgi -s /var/run/qgisserver.socket \
            -U www-data -G www-data -n \
            /usr/lib/cgi-bin/qgis_mapserv.fcgi
```

QGIS Server is now available at <http://qgis.demo/qgisserver>.

Nota: Al usar `spawn-fcgi`, puede definir directamente las variables de entorno antes de ejecutar el servidor. Por ejemplo: `export QGIS_SERVER_LOG_STDERR=1`

Por supuesto, puede agregar un script de inicio para iniciar QGIS Server en el momento del arranque o cuando lo desee. Por ejemplo, con **systemd**, edite el archivo `/etc/systemd/system/qgis-server.service` con este contenido:

```
[Unit]
Description=QGIS server
After=network.target

[Service]
;; set env var as needed
;Environment="LANG=en_EN.UTF-8"
;Environment="QGIS_SERVER_PARALLEL_RENDERING=1"
;Environment="QGIS_SERVER_MAX_THREADS=12"
;Environment="QGIS_SERVER_LOG_LEVEL=0"
;Environment="QGIS_SERVER_LOG_STDERR=1"
;; or use a file:
;EnvironmentFile=/etc/qgis-server/env
```

(continúe en la próxima página)

(proviene de la página anterior)

```
ExecStart=spawn-fcgi -s /var/run/qgisserver.socket -U www-data -G www-data -n /usr/
↳lib/cgi-bin/qgis_mapserv.fcgi
```

[Install]

```
WantedBy=multi-user.target
```

Luego habilite e inicie el servicio:

```
systemctl enable --now qgis-server
```

Advertencia: Con los comandos anteriores, spawn-fcgi genera solo un proceso de QGIS Server.

fcgiwrap

Usando `fcgiwrap` es mucho más fácil de configurar que `spawn-fcgi` pero es mucho más lento. Primero tienes que instalar el paquete correspondiente:

```
apt install fcgiwrap
```

Luego, introduzca el siguiente bloque en la configuración de su servidor NGINX:

```
1 location /qgisserver {
2     gzip            off;
3     include         fastcgi_params;
4     fastcgi_pass    unix:/var/run/fcgiwrap.socket;
5     fastcgi_param   SCRIPT_FILENAME /usr/lib/cgi-bin/qgis_mapserv.fcgi;
6 }
```

Finalmente, reinicie NGINX y `fcgiwrap` para tener en cuenta la nueva configuración:

```
systemctl restart nginx
systemctl restart fcgiwrap
```

QGIS Server is now available at <http://qgis.demo/qgisserver>.

Systemd

QGIS Server necesita un servidor X en ejecución para ser completamente utilizable, en particular para imprimir. En el caso de que ya tenga un servidor X en ejecución, puede utilizar los servicios de systemd.

Este método, para implementar QGIS Server, se basa en dos unidades Systemd:

- un `Socket unit`
- y un `Service unit`.

La **unidad QGIS Server Socket** define y crea un socket del sistema de archivos, utilizado por NGINX para iniciar y comunicarse con QGIS Server. La unidad Socket debe configurarse con `Accept = false`, lo que significa que las llamadas a la llamada del sistema «Aceptar ()» se delegan al proceso creado por la unidad de Servicio. Se encuentra en `/etc/systemd/system/qgis-server@.socket`, que en realidad es una plantilla:

[Unit]

```
Description=QGIS Server Listen Socket (instance %i)
```

[Socket]

```
Accept=false
```

```
ListenStream=/var/run/qgis-server-%i.sock
```

(continúe en la próxima página)

(proviene de la página anterior)

```
SocketUser=www-data
SocketGroup=www-data
SocketMode=0600
```

[Install]

```
WantedBy=sockets.target
```

Ahora habilite e inicie sockets:

```
for i in 1 2 3 4; do systemctl enable --now qgis-server@$i.socket; done
```

La **unidad de servicio del servidor QGIS** define e inicia el proceso del servidor QGIS. La parte importante es que la entrada estándar del proceso de servicio está conectada al enchufe definido por la unidad de enchufe. Esto debe configurarse usando `StandardInput=socket` en la configuración de la unidad de servicio ubicada en `/etc/systemd/system/qgis-server@.service`:

[Unit]

```
Description=QGIS Server Service (instance %i)
```

[Service]

```
User=www-data
Group=www-data
StandardOutput=null
StandardError=journal
StandardInput=socket
ExecStart=/usr/lib/cgi-bin/qgis_mapserv.fcgi
EnvironmentFile=/etc/qgis-server/env
```

[Install]

```
WantedBy=multi-user.target
```

Nota: El servidor QGIS *environment variables* se define en un archivo separado, `/etc/qgis-server/env`. Podría parecerse a esto:

```
QGIS_PROJECT_FILE=/etc/qgis/myproject.qgs
QGIS_SERVER_LOG_STDERR=1
QGIS_SERVER_LOG_LEVEL=3
```

Ahora inicie el servicio de socket:

```
for i in 1 2 3 4; do systemctl enable --now qgis-server@$i.service; done
```

Finalmente, para el servidor HTTP NGINX, introduzcamos la configuración para esta configuración:

```
upstream qgis-server_backend {
    server unix:/var/run/qgis-server-1.sock;
    server unix:/var/run/qgis-server-2.sock;
    server unix:/var/run/qgis-server-3.sock;
    server unix:/var/run/qgis-server-4.sock;
}

server {
    ...

    location /qgis-server {
        gzip off;
        include fastcgi_params;
        fastcgi_pass qgis-server_backend;
    }
}
```

(continué en la próxima página)

(proviene de la página anterior)

```
}
}
```

Ahora reinicie NGINX para que se tenga en cuenta la nueva configuración:

```
systemctl restart nginx
```

Gracias a Oslandia por compartir [su tutorial](#).

2.1.3 Xvfb

QGIS Server necesita un servidor X en ejecución para ser completamente utilizable, en particular para imprimir. En los servidores, generalmente se recomienda no instalarlo, por lo que puede usar `xvfb` para tener un entorno X virtual.

Si está ejecutando el servidor en un entorno gráfico/X11, no es necesario instalar `xvfb`. Mas info en <https://www.itopen.it/qgis-server-setup-notes/>.

Para instalar el paquete:

```
apt install xvfb
```

Cree el archivo de servicio, `/etc/systemd/system/xvfb.service`, con este contenido:

```
[Unit]
Description=X Virtual Frame Buffer Service
After=network.target

[Service]
ExecStart=/usr/bin/Xvfb :99 -screen 0 1024x768x24 -ac +extension GLX +render -
↳noreset

[Install]
WantedBy=multi-user.target
```

Activa, inicia y prueba el estado de `xvfb.service`:

```
systemctl enable --now xvfb.service
systemctl status xvfb.service
```

Luego, de acuerdo con su servidor HTTP, debe configurar el parámetro **DISPLAY** o usar directamente **xvfb-run**.

Con Apache

Luego puede configurar el parámetro **DISPLAY**.

Con Apache, simplemente agregue a su configuración *FastCGI* (ver arriba):

```
FcgidInitialEnv DISPLAY      ":99"
```

Ahora reinicie Apache para que se tenga en cuenta la nueva configuración:

```
systemctl restart apache2
```

Con NGINX

Luego, puede usar **xvfb-run** directamente o configurar el parámetro **DISPLAY**.

- Con spawn-fcgi usando xvfb-run:

```
xvfb-run /usr/bin/spawn-fcgi -f /usr/lib/cgi-bin/qgis_mapserv.fcgi \
-s /tmp/qgisserver.socket \
-G www-data -U www-data -n
```

- Con la variable de entorno **DISPLAY** en la configuración del servidor HTTP.

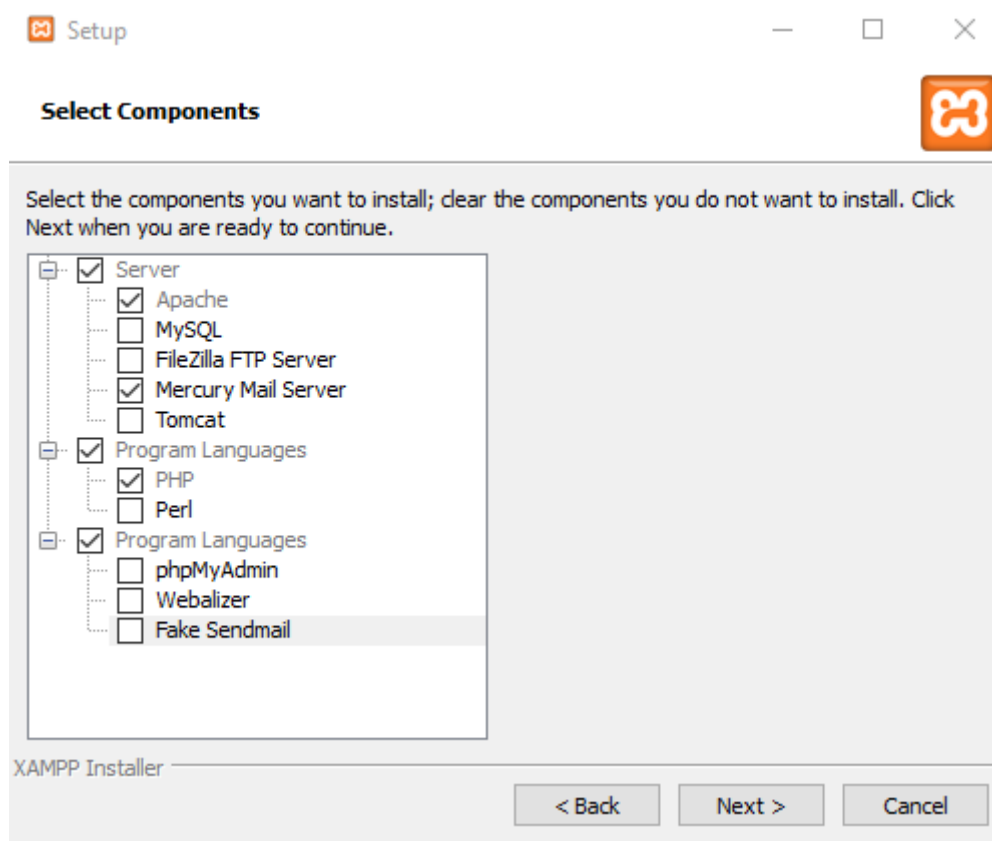
```
fastcgi_param DISPLAY ":99";
```

2.2 Instalación en Windows

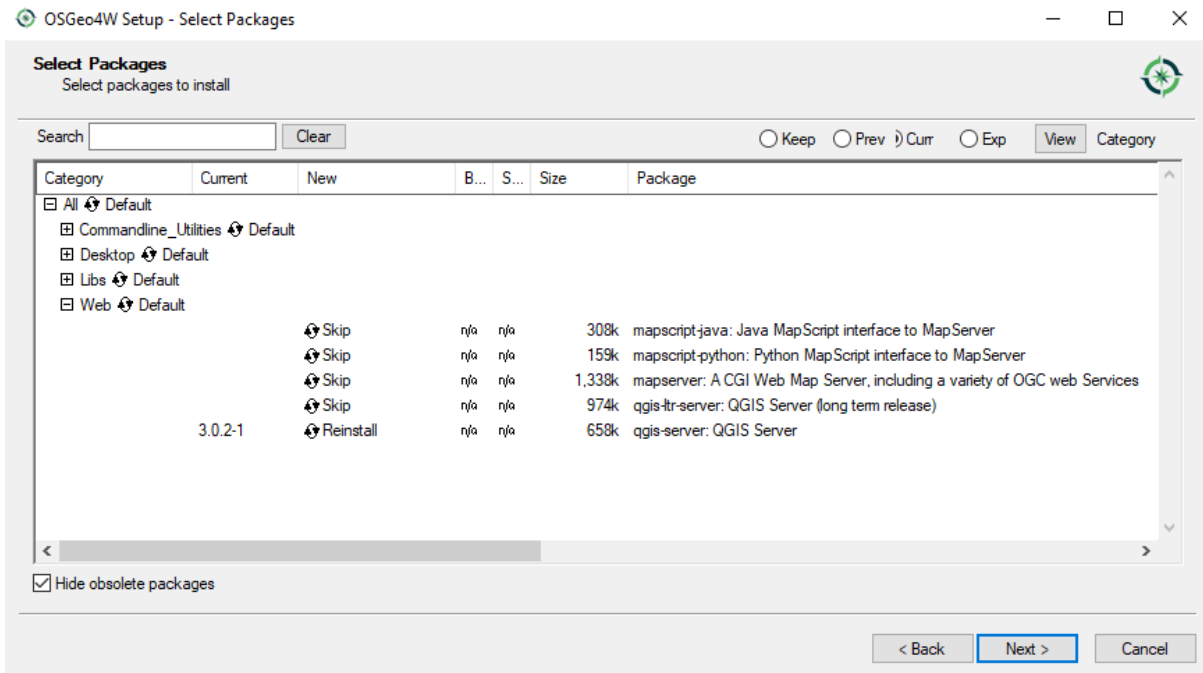
QGIS Server también se puede instalar en sistemas Windows. Si bien el paquete QGIS Server está disponible en la versión de 64 bits del instalador de red OSGeo4W (<https://qgis.org/en/site/forusers/download.html>) no hay ningún paquete Apache (u otro servidor web) disponible, por lo que debe instalarse por otros medios.

Un procedimiento sencillo es el siguiente:

1. Descarga el instalador de XAMPP (<https://www.apachefriends.org/download.html>) para Windows e instale Apache



2. Descargue el instalador de OSGeo4W, siga la «Instalación avanzada» e instale los paquetes QGIS Desktop y QGIS Server



3. Edit the httpd.conf file (C:\xampp\apache\conf\httpd.conf if the default installation paths have been used) and make the following changes:

De:

```
ScriptAlias /cgi-bin/ "C:/xampp/cgi-bin/"
```

A:

```
ScriptAlias /cgi-bin/ "C:/OSGeo4W64/apps/qgis/bin/"
```

De:

```
<Directory "C:/xampp/cgi-bin">
  AllowOverride None
  Options None
  Require all granted
</Directory>
```

A:

```
<Directory "C:/OSGeo4W64/apps/qgis/bin">
  SetHandler cgi-script
  AllowOverride None
  Options ExecCGI
  Order allow,deny
  Allow from all
  Require all granted
</Directory>
```

De:

```
AddHandler cgi-script .cgi .pl .asp
```

A:

```
AddHandler cgi-script .cgi .pl .asp .exe
```

4. Luego, en la parte inferior de httpd.conf, agregue:

```
SetEnv GDAL_DATA "C:\OSGeo4W64\share\gdal"  
SetEnv QGIS_AUTH_DB_DIR_PATH "C:\OSGeo4W64\apps\qgis\resources"  
SetEnv PYTHONHOME "C:\OSGeo4W64\apps\Python37"  
SetEnv PATH "C:\OSGeo4W64\bin;C:\OSGeo4W64\apps\qgis\bin;C:\OSGeo4W64\apps\Qt5\  
↪bin;C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem"  
SetEnv QGIS_PREFIX_PATH "C:\OSGeo4W64\apps\qgis"  
SetEnv QT_PLUGIN_PATH "C:\OSGeo4W64\apps\qgis\qtplugins;C:\OSGeo4W64\apps\Qt5\  
↪plugins"
```

5. Reinicie el servidor web Apache desde el Panel de control de XAMPP y abra la ventana del navegador para probar una solicitud GetCapabilities en QGIS Server

```
http://qgis.demo/cgi-bin/qgis_mapserv.fcgi.exe?SERVICE=WMS&VERSION=1.3.0&  
↪REQUEST=GetCapabilities
```

2.3 Sirve un proyecto

Ahora que QGIS Server está instalado y funcionando, solo tenemos que usarlo.

Obviamente, necesitamos un proyecto QGIS en el que trabajar. Por supuesto, puede personalizar completamente su proyecto definiendo la información de contacto, precisando algunas restricciones en CRS o incluso excluyendo algunas capas. Todo lo que necesita saber al respecto se describe más adelante en *Configure su proyecto*.

But for now, we are going to use a simple project already configured and previously downloaded in `/home/qgis/projects/world.qgs`, as described above.

Al abrir el proyecto y echar un vistazo rápido a las capas, sabemos que actualmente hay 4 capas disponibles:

- airports
- places
- countries
- countries_shapeburst

No es necesario que comprenda la solicitud completa por ahora, pero puede recuperar un mapa con algunas de las capas anteriores gracias a QGIS Server haciendo algo como esto en su navegador web para recuperar la capa *countries*:

```
http://qgis.demo/qgisserver?  
MAP=/home/qgis/projects/world.qgs&  
LAYERS=countries&  
SERVICE=WMS&  
VERSION=1.3.0&  
REQUEST=GetMap&  
CRS=EPSG:4326&  
WIDTH=400&  
HEIGHT=200&  
BBOX=-90,-180,90,180
```

Si obtiene la siguiente imagen, QGIS Server se está ejecutando correctamente:

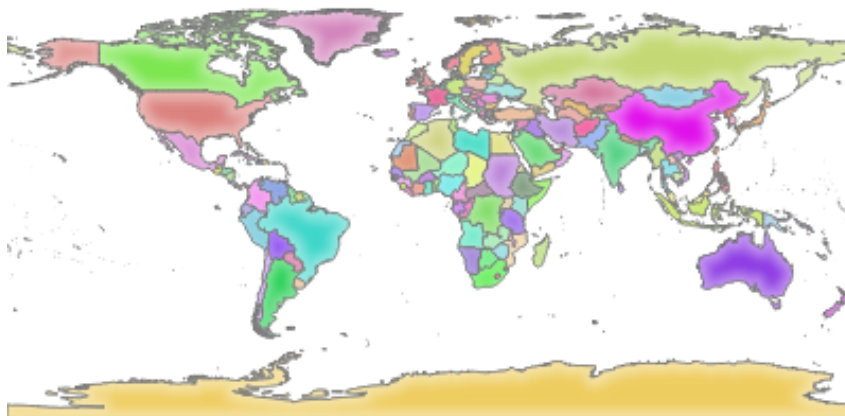


Figura 2.1: Respuesta del servidor a una solicitud básica de GetMap

Perciba que puede definir la variable de entorno **QGIS_PROJECT_FILE** para usar un proyecto por defecto en lugar de dar un parámetro **MAP** (ver *Variables de entorno*).

Por ejemplo con spawn-fcgi:

```
export QGIS_PROJECT_FILE=/home/qgis/projects/world.qgs
spawn-fcgi -f /usr/lib/bin/cgi-bin/qgis_mapserv.fcgi \  
-s /var/run/qgisserver.socket \  
-U www-data -G www-data -n
```

2.4 Configure su proyecto

Para proporcionar un nuevo servidor QGIS WMS, WFS o WCS, debe crear un archivo de proyecto QGIS con algunos datos o usar uno de su proyecto actual. Defina los colores y estilos de las capas en QGIS y el proyecto SRC, si aún no está definido.

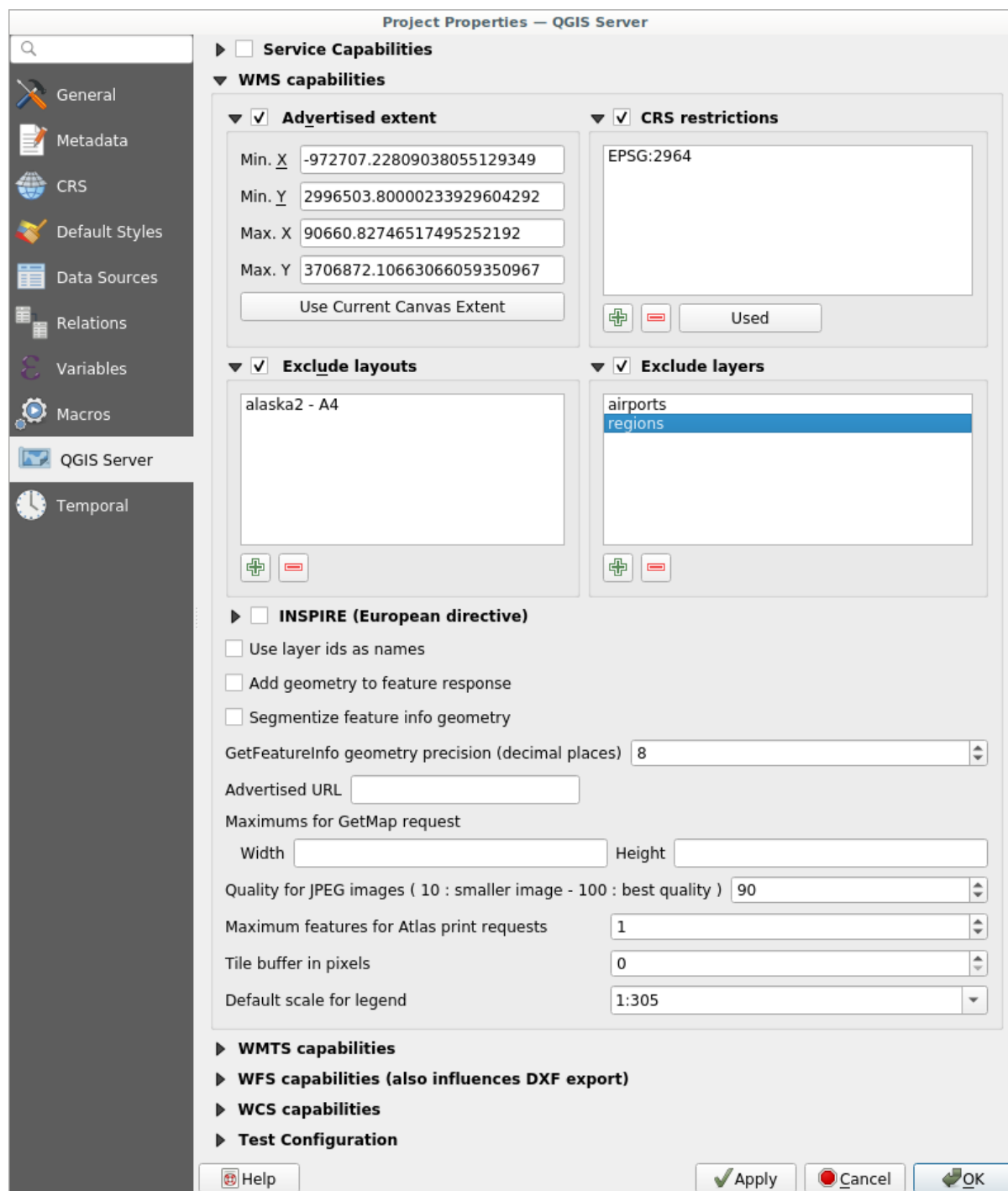





Figura 2.2: Definiciones para un proyecto WMS/WFS/WCS de QGIS Server

Luego, vaya al menú *QGIS Server* del diálogo *Proyecto* ► *Propiedades...* y proporcionar información sobre el OWS en los campos de *Service Capabilities*. Esto aparecerá en la respuesta *GetCapabilities* del WMS, WFS o WCS. Si no marca *Service capabilities*, QGIS Server utilizará la información proporcionada en el archivo `wms_metadata.xml` ubicado en la carpeta `cgi-bin`.

2.4.1 Capacidades WMS

En la sección *Capacidades de WMS*, puede definir la extensión anunciada en la respuesta *GetCapabilities* de WMS ingresando los valores mínimos y máximos de X e Y en los campos debajo de: `guiabel: `Extensión de publicidad``. Al hacer clic en *Usar extensión de lienzo actual* establece estos valores en la extensión que se muestra actualmente en el lienzo del mapa QGIS. Marcando *Restricciones de SRC*, puede restringir en qué sistemas de referencia de coordenadas (SRC) QGIS Server ofrecerá renderizar mapas. Se recomienda que restrinja el SRC ofrecido, ya que esto reduce el tamaño de la respuesta WMS *GetCapabilities*. Utilice el  botón de abajo para seleccionar esos SRC del Selector de Sistema de Referencia de Coordenadas, o haga clic en *Usada* para agregar los SRC usados en el proyecto QGIS a la lista.

Si tiene diseños de impresión definidos en su proyecto, se enumerarán en la respuesta *GetProjectSettings* y la solicitud *GetPrint* puede usarlos para crear impresiones, utilizando uno de los diseños de impresión como plantilla. Esta es una extensión específica de QGIS para la especificación WMS 1.3.0. Si desea excluir cualquier diseño de impresión de ser publicado por el WMS, marque *Excluir diseños* y haga clic en el botón  de abajo. Luego, seleccione un diseño de impresión en el cuadro de diálogo *Seleccionar diseño de impresión* para agregarlo a la lista de diseños excluidos.

Si desea excluir cualquier capa o grupo de capas de la publicación del WMS, marque *Excluir capas* y haga clic en el  botón de abajo. Esto abre el cuadro de diálogo: `guiabel: `Seleccionar capas y grupos restringidos``, que le permite elegir las capas y grupos que no desea que se publiquen. Utilice la tecla `Shift` o `:kbd:`Ctrl`` si desea seleccionar varias entradas. Se recomienda que excluya de la publicación las capas que no necesita, ya que esto reduce el tamaño de la respuesta de WMS *GetCapabilities*, lo que conduce a tiempos de carga más rápidos en el lado del cliente.

Si marca *Use los identificadores de capa como nombre*, los identificadores de capa se usarán para hacer referencia a las capas en la respuesta *GetCapabilities* o el parámetro `GetMap LAYERS`. Si no es así, se usa el nombre de la capa o el nombre corto si está definido (ver `vectorservermenu`).

Puede recibir *GetFeatureInfo* solicitado como texto sin formato, XML y GML. El predeterminado es XML.

Si lo desea, puede marcar *Añadir geometría a la respuesta de la función*. Esto incluirá el cuadro delimitador para cada entidad en la respuesta *GetFeatureInfo*. Ver también el parámetro: `ref: WITH_GEOMETRY <server_wms_getfeatureinfo>`.

Como muchos clientes web no pueden mostrar arcos circulares en geometrías, tiene la opción de segmentar la geometría antes de enviarla al cliente en una respuesta *GetFeatureInfo*. Esto permite a dichos clientes seguir mostrando la geometría de una característica (por ejemplo, para resaltar la entidad). Debe marcar la casilla de verificación *Segmentize feature info geometry* para activar la opción.

También puede usar la opción *Precisión de geometría GetFeatureInfo* para establecer la precisión de la geometría *GetFeatureInfo*. Esto le permite ahorrar ancho de banda cuando no necesita toda la precisión.

Si desea que QGIS Server publique URL de solicitud específicas en la respuesta WMS *GetCapabilities*, ingrese la URL correspondiente en el campo *URL anunciada*.

Además, puede restringir el tamaño máximo de los mapas devueltos por la solicitud *GetMap* ingresando el ancho y la altura máximos en los campos respectivos bajo *Máximos para solicitud GetMap*.

Puede cambiar el factor *Calidad para imágenes JPEG*. El factor de calidad debe estar en el rango de 0 a 100. Especifique 0 para compresión máxima, 100 para sin compresión.

Puede cambiar el límite para que las entidades de Atlas se impriman en una solicitud configurando el campo *Entidades máximas para solicitudes de impresión de Atlas*.

Cuando se utiliza QGIS Server en modo mosaico (consulte *parámetro TILED*), puede configurar *Búfer de mosaico en píxeles*. El valor recomendado es el tamaño del símbolo o ancho de línea más grande en su proyecto QGIS.

Si una de sus capas usa Map Tip display (por ejemplo para mostrar texto usando expresiones) esto se enumerará dentro de la salida *GetFeatureInfo*. Si la capa usa un mapa de valor para uno de sus atributos, esta información también se mostrará en la salida de *GetFeatureInfo*.

2.4.2 Capacidades WFS

En el área *WFS capabilities* puede seleccionar las capas que desea publicar como WFS y especificar si permitirán operaciones de actualización, inserción y eliminación. Si ingresa una URL en el campo *Advertised URL* de la sección *WFS capabilities*, QGIS Server anunciará esta URL específica en la respuesta WFS GetCapabilities.

2.4.3 Capacidades WCS

En el área *WCS capabilities*, puede seleccionar las capas que desea publicar como WCS. Si ingresa una URL en el campo *Advertised URL* de la sección *WCS capabilities*, QGIS Server anunciará esta URL específica en la respuesta WCS GetCapabilities.

2.4.4 Ajuste fino de OWS

Para capas vectoriales, el menú *Campos* del diálogo *Capa* ► *Propiedades de capa* permite definir para cada atributo si se publicará o no. De forma predeterminada, todos los atributos son publicados por su WMS y WFS. Si no desea que se publique un atributo específico, desmarque la casilla correspondiente en la columna *WMS* o *WFS*.

Puede superponer marcas de agua sobre los mapas producidos por su WMS agregando anotaciones de texto o anotaciones SVG al archivo del proyecto. Ver la sección *sec_annotations* para obtener instrucciones sobre cómo crear anotaciones. Para que las anotaciones se muestren como marcas de agua en la salida de WMS, la casilla de verificación *Fixed map position* en el cuadro de diálogo *Annotation text* debe estar desmarcada. Se puede acceder a esto haciendo doble clic en la anotación mientras una de las herramientas de anotación está activa. Para las anotaciones SVG, necesitará configurar el proyecto para guardar rutas absolutas (en el menú *General* del cuadro de diálogo :menuselection:`Proyecto -> Propiedades...`) o modificar manualmente la ruta para la imagen SVG para que represente una ruta relativa válida.

2.5 Integración con terceras partes

QGIS Server proporciona servicios web OGC como [WMS](#), [WFS](#), etc. por lo tanto, puede ser utilizado por una amplia variedad de herramientas de usuario final.

2.5.1 Integración con QGIS Desktop

QGIS Desktop es el diseñador de mapas donde QGIS Server es el servidor de mapas. Los mapas o proyectos QGIS serán servidos por el servidor QGIS para proporcionar estándares OGC. Estos proyectos de QGIS pueden ser archivos o entradas en una base de datos (usando *Proyecto -> Guardar en -> PostgreSQL* en QGIS Desktop).

Además, se debe establecer un flujo de trabajo de actualización dedicado a actualizar un proyecto utilizado por un servidor QGIS (es decir, copiar los archivos del proyecto en la ubicación del servidor y reiniciar el servidor QGIS). Por ahora, los procesos automatizados (como la recarga del servidor a través del servicio de cola de mensajes) aún no están implementados.

2.5.2 Integración con MapProxy

[MapProxy](#) es un servidor de caché de teselas y como tal puede leer y servir cualquier servidor de mapas WMS/WMTS, puede conectarse directamente a los servicios web del servidor QGIS y mejorar la experiencia del usuario final.

2.5.3 Integración con QWC2

QWC2 es una aplicación web receptiva dedicada a QGIS Server. Le ayuda a construir un visor de mapas altamente personalizado con selección de capas, información de características, etc. También hay muchos complementos disponibles como autenticación o servicio de impresión, la lista completa está disponible en este [repositorio](#).

QGIS Server puede servir datos de acuerdo con los protocolos estándar descritos por **Open Geospatial Consortium (OGC)**:

- WMS 1.1.1 y 1.3.0
- WFS 1.0.0 y 1.1.0
- OGC API - Entidades (WFS3)
- WCS 1.1.1
- WMTS 1.0.0

Se admiten parámetros y solicitudes adicionales del proveedor además del estándar original que mejora enormemente las posibilidades de personalizar su comportamiento gracias al motor de renderización QGIS.

3.1 Web Map Service (WMS)

Los estándares WMS **1.1.1** y **1.3.0** implementados en QGIS Server proporcionan una interfaz HTTP para solicitar imágenes de mapas o leyendas generadas a partir de un proyecto QGIS. Una solicitud WMS típica define el proyecto QGIS que se utilizará, las capas que se renderizarán y el formato de imagen que se generará. El soporte básico también está disponible como **Descriptor de capa con estilo (SLD)**.

Especificaciones:

- [WMS 1.1.1](#)
- [WMS 1.3.0](#)
- [perfil SLD 1.1.0 WMS](#)

Solicitudes standard proporcionadas por el servidor QGIS:

Solicitud	Descripción
<code>GetCapabilities</code>	Devuelve metadatos XML con información sobre el servidor
<code>GetMap</code>	Devuelve un mapa
<code>GetFeatureInfo</code>	Recupera datos (geometría y valores) para una ubicación de píxeles
<code>GetLegendGraphics</code>	Devuelve símbolos de leyenda

Solicitudes de proveedor proporcionadas por QGIS Server:

Solicitud	Descripción
<i>GetPrint</i>	Devuelve una composición de QGIS
<i>GetProjectSettings</i>	Devuelve información específica sobre QGIS Server

3.1.1 GetMap

Parámetros estándar para la petición **GetMap** de acuerdo con las especificaciones OGC WMS 1.1.1 y 1.3.0:

Parámetros	Necesario	Descripción
<i>SERVICE</i>	Sí	Nombre del servicio (WMS)
<i>VERSION</i>	Sí	Versión del servicio
<i>REQUEST</i>	Sí	Nombre de la petición (GetMap)
<i>LAYERS</i>	No	Capas a mostrar
<i>STYLES</i>	No	Estilo de capas
<i>SRS / CRS</i>	Sí	Sistema de referencia de coordenadas
<i>BBOX</i>	Sí	Extensión del mapa
<i>WIDTH</i>	Sí	Ancho de la imagen en píxeles.
<i>HEIGHT</i>	Sí	Altura de la imagen en píxeles
<i>FORMAT</i>	No	Formato de imagen
<i>TRANSPARENT</i>	No	fondo transparente
<i>SLD</i>	No	URL de una SLD que se utilizará para diseñar
<i>SLD_BODY</i>	No	SLD en línea (XML) que se utilizará para diseñar

En adición a los estándares, QGIS Server apoya los siguientes parámetros extra:

Parámetros	Necesario	Descripción
<i>MAP</i>	Sí	especifica el archivo de proyecto de QGIS
<i>BGCOLOR</i>	No	Especifica el color de fondo
<i>DPI</i>	No	Especifica la resolución de salida
<i>IMAGE_QUALITY</i>	No	compresión JPEG
<i>OPACITIES</i>	No	Opacidad de la capa o Grupo
<i>FILTER</i>	No	Subconjunto de entidades
<i>SELECTION</i>	No	Resalta entidades
<i>FILE_NAME</i>	No	Nombre de archivo de la capa descargada Solo para ``FORMAT=application/dxf``
<i>FORMAT_OPTIONS</i>	No	Opciones del formato de archivo especificado Solo para ``FORMAT=application/dxf``
<i>TESELADO</i>	No	Trabajando en <i>modo teselado</i>

URL ejemplo:

```
http://localhost/qgis_server?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&MAP=/home/qgis/projects/world.qgs
&LAYERS=mylayer1,mylayer2,mylayer3
&STYLES=style1,default,style3
&OPACITIES=125,200,125
&CRS=EPSG:4326
&WIDTH=400
&HEIGHT=400
&FORMAT=image/png
```

(continué en la próxima página)

(proviene de la página anterior)

```
&TRANSPARENT=TRUE
&DPI=300
&TILED=TRUE
```

SERVICIO

Este parámetro tiene que ser WMS.

Por ejemplo:

```
http://localhost/qgisserver?
SERVICE=WMS
&...
```

VERSIÓN

Este parámetro permite especificar la versión del servicio a utilizar. Los valores disponibles para el parámetro **VERSIÓN** son:

- 1.1.1
- 1.3.0

Según el número de versión, se esperan ligeras diferencias como se explica más adelante para los siguientes parámetros:

- CRS / SRS
- BBOX

SOLICITUD

Este parámetro es **GetMap** en caso de petición **GetMap**.

CAPAS

Este parámetro permite especificar las capas que se mostrarán en el mapa. Los nombres deben estar separados por una coma.

Además, QGIS Server introdujo algunas opciones para seleccionar capas mediante:

- un nombre corto
- el id de capa

El nombre corto de una capa se puede configurar mediante *Propiedades -> Metadatos* en el menú de capas. Si se define el nombre corto, entonces se usa por defecto en lugar del nombre de la capa:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&LAYERS=mynickname1,mynickname2
&...
```

Además, hay una opción de proyecto que permite seleccionar capas por su identificación en *OWS Server -> WMS Capacidades* del menú *Proyecto -> Propiedades ...*. Para activar esta opción, la casilla de verificación: *guiabel: 'Usar ID de capa como nombres`* debe estar seleccionada.

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&LAYERS=mylayerid1,mylayerid2  
&...
```

ESTILOS

Este parámetro se puede utilizar para especificar el estilo de una capa para el paso de renderizado. Los estilos deben estar separados por una coma. El nombre del estilo predeterminado es `default`.

SRS / CRS

Este parámetro permite indicar la salida del mapa Spatial Reference System en WMS **1.1.1** y tiene que ser foramdo como `EPSG:XXXX`. Note que `SRC` también es compatible si la versión actual es **1.1.1**.

Para WMS **1.3.0**, es preferible el parámetro `SRC` pero también se admite `SRS`.

Tenga en cuenta que si en la solicitud se indican los parámetros `SRC` y `SRS`, entonces es la versión actual indicada en el parámetro `VERSIÓN` la que es decisiva.

En el siguiente caso, el parámetro `SRS` se mantiene cualquiera que sea el parámetro `VERSION` porque no se indica `SRC`:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&VERSION=1.3.0  
&SRS=EPSG:2854  
&...
```

En el siguiente caso, el parámetro `SRS` se mantiene en lugar de `SRC` debido al parámetro `VERSIÓN`:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&VERSION=1.1.1  
&CRS=EPSG:4326  
&SRS=EPSG:2854  
&...
```

En el siguiente caso, el parámetro `SRC` se mantiene en lugar de `SRS` debido al parámetro `VERSION`:

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&VERSION=1.3.0  
&CRS=EPSG:4326  
&SRS=EPSG:2854  
&...
```

BBOX

Este parámetro permite especificar la extensión del mapa con unidades según el SRC actual. Las coordenadas deben estar separadas por una coma.

El parámetro BBOX se forma como `min_a, min_b, max_a, max_b` pero la definición de los ejes `a` y `b` es diferente según el parámetro `VERSION` actual:

- en **WMS 1.1.1**, el orden de los ejes siempre es este/norte
- en **WMS 1.3.0**, el orden de los ejes depende de la autoridad CRS

Por ejemplo, en el caso de `EPSG:4326` y **WMS 1.1.1**, `a` es la longitud (este) y `b` la latitud (norte), lo que lleva a una solicitud como:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.1.1
&SRS=epsg:4326
&BBOX=-180,-90,180,90
&...
```

Pero en el caso de **WMS 1.3.0**, el orden de los ejes definido en la base de datos EPSG es norte/este, por lo que `a` es la latitud y `b` la longitud:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&CRS=epsg:4326
&BBOX=-90,-180,90,180
&...
```

ANCHURA

Este parámetro permite especificar el ancho en píxeles de la imagen de salida.

ALTURA

Este parámetro permite especificar la altura en píxeles de la imagen de salida.

FORMATO

Este parámetro se puede utilizar para especificar el formato de la imagen del mapa. Los valores disponibles son:

- `jpg`
- `jpeg`
- `image/jpeg`
- `image/png`
- `image/png; mode=1bit`
- `image/png; mode=8bit`
- `image/png; mode=16bit`

- `application/dxf` Solo las capas que tienen acceso de lectura en el servicio WFS se exportan en formato DXF.

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WMS&VERSION=1.3.0
&REQUEST=GetMap
&FORMAT=application/dxf
&LAYERS=Haltungen,Normschacht,Spezialbauwerke
&STYLES=
&CRS=EPSG%3A21781&BBOX=696136.28844801,245797.12108743,696318.91114315,245939.
↪25832905
&WIDTH=1042
&HEIGHT=811
&FORMAT_OPTIONS=MODE:SYMBOLLAYERSYMBOLOLOGY;SCALE:250&FILE_NAME=plan.dxf
```

TRANSPARENTE

Este parámetro booleano se puede utilizar para especificar la transparencia del fondo. Los valores disponibles son (no distinguen entre mayúsculas y minúsculas):

- TRUE
- FALSE

Sin embargo, este parámetro se ignora si el formato de la imagen del mapa indicada con `FORMAT` es diferente de PNG.

MAPA

Este parámetro permite definir el archivo de proyecto QGIS a utilizar.

Como se mencionó en *tabla de parámetros de GetMap*, `MAP` es obligatorio porque una solicitud necesita un proyecto QGIS para funcionar. Sin embargo, la variable de entorno `QGIS_PROJECT_FILE` se puede utilizar para definir un proyecto QGIS predeterminado. En este caso específico, `MAP` ya no es un parámetro obligatorio. Para obtener más información, puede consultar *Configuración avanzada*.

BGCOLOR

Este parámetro permite indicar un color de fondo para la imagen del mapa. Sin embargo, no se puede combinar con el parámetro `TRANSPARENTE` en el caso de imágenes PNG (la transparencia tiene prioridad). El color puede ser literal o en notación hexadecimal.

URL ejemplo con la notación literal:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&BGCOLOR=green
&...
```

Ejemplo URL con la notación hexadecimal:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&VERSION=1.3.0
&BGCOLOR=0x00FF00
&...
```

DPI

Este parámetro se puede utilizar para especificar la resolución de salida solicitada.

IMAGE_QUALITY

Este parámetro solo se utiliza para imágenes JPEG. De forma predeterminada, la compresión JPEG es -1.

Puede cambiar el valor predeterminado por proyecto QGIS en el menú *OWS Server -> Capacidades WMS del Proyecto -> Propiedades...* Si desea anularlo en una solicitud GetMap puede hacerlo usando el parámetro *IMAGE_QUALITY*.

OPACITIES

Lista de valores de opacidad separados por comas. La opacidad se puede establecer a nivel de capa o de grupo. Los valores permitidos van de 0 (totalmente transparente) a 255 (totalmente opaco).

FILTER

Se puede seleccionar un subconjunto de capas con el parámetro *FILTRO*. La sintaxis es básicamente la misma que para la cadena del subconjunto QGIS. Sin embargo, existen algunas restricciones para evitar inyecciones de SQL en bases de datos a través de QGIS Server. Si se encuentra una cadena peligrosa en el parámetro, QGIS Server devolverá el siguiente error:

```
<ServiceExceptionReport>
  <ServiceException code="Security">The filter string XXXXXXXXX has been rejected.
  ↳because of security reasons.
  Note: Text strings have to be enclosed in single or double quotes. A space.
  ↳between each word / special character is mandatory.
  Allowed Keywords and special characters are IS,NOT,NULL,AND,OR,IN,=,<,<,>,>=,!=,
  ↳', ', (, ),DMETAPHONE,SOUNDEX.
  Not allowed are semicolons in the filter expression.</ServiceException>
</ServiceExceptionReport>
```

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WMS
&REQUEST=GetMap
&LAYERS=mylayer1,mylayer2,mylayer3
&FILTER=mylayer1:"col1";mylayer1,mylayer2:"col2" = 'blabla'
&...
```

En este ejemplo, el mismo filtro (el campo *col2* es igual a la cadena *blabla*) se aplica a las capas *mylayer1* y *mylayer2*, mientras que el filtro en *col1* es solo aplicado a *mylayer1*.

Nota: Es posible realizar búsquedas de atributos a través de *GetFeatureInfo* y omitir el parámetro *X / Y* si hay un *FILTRO*. QGIS Server luego devuelve información sobre las características coincidentes y genera un cuadro delimitador combinado en la salida XML.

SELECTION

El parámetro SELECCIÓN puede resaltar entidades de una o más capas. Las características vectoriales se pueden seleccionar pasando listas separadas por comas con identificadores de objetos.

```
http://localhost/qgisserver?  
SERVICE=WMS  
&REQUEST=GetMap  
&LAYERS=mylayer1,mylayer2  
&SELECTION=mylayer1:3,6,9;mylayer2:1,5,6  
&...
```

La siguiente imagen presenta la respuesta de una solicitud GetMap usando la opción SELECTION p.ej. <http://myserver.com/...&SELECTION=countries:171,65>.

Como los identificadores de esas características corresponden en el conjunto de datos de origen a **Francia y Rumanía**, están resaltados en amarillo.



Figura 3.1: Respuesta del servidor a una solicitud GetMap con el parámetro SELECTION

FORMAT-OPTIONS

Este parámetro se puede utilizar para especificar opciones para el formato seleccionado. Solo para `FORMAT=application/dxf`. Una lista de pares clave: valor separados por punto y coma:

- **ESCALA:** para ser utilizado para reglas de simbología, filtros y estilos (no escala real de los datos - los datos permanecen en la escala original).
- **MODO:** corresponde a las opciones de exportación que se ofrecen en el cuadro de diálogo de exportación QGIS Desktop DXF. Los valores posibles son `NOSYMBOLLOGY`, `FEATURESYMBOLLOGY` y `SYMBOLLAYERSYMBOLLOGY`.
- **LAYERSATTRIBUTES:** especifique un campo que contenga valores para los nombres de las capas DXF; si no se especifica, se utilizan los nombres de las capas QGIS originales.
- **USE_TITLE_AS_LAYERNAME:** si está habilitado, el título de la capa se utilizará como nombre de capa.
- **CODEC:** especificar un códec que se utilizará para la codificación. El valor predeterminado es `ISO-8859-1`. Consulte el cuadro de diálogo de exportación DXF del escritorio de QGIS para obtener valores válidos.
- **NO_MTEXT:** Utilice `TEXT` en lugar de `MTEXT` para las etiquetas.
- **FORCE_2D:** Forzar salida 2D. Esto es necesario para el ancho de la polilínea.

TILED

Por motivos de rendimiento, QGIS Server se puede utilizar en modo mosaico. En este modo, el cliente solicita varios mosaicos pequeños de tamaño fijo y los ensambla para formar el mapa completo. Al hacer esto, los símbolos en o cerca del límite entre dos teselas pueden aparecer cortados, porque solo están presentes en una de las teselas.

Establezca el parámetro `TILED` en `TRUE` para decirle al servidor QGIS que funcione en modo *teselado* y para aplicar el *búfer de teselas* configurado en el proyecto QGIS (consulte [Configure su proyecto](#)).

Cuando `TILED` es `TRUE` y cuando se configura un búfer de mosaico distinto de cero en el proyecto QGIS, las entidades fuera de la extensión del mosaico se dibujan para evitar símbolos de corte en los límites del mosaico.

`TILED` predeterminado como `FALSE`.

3.1.2 GetFeatureInfo

Parámetros estándar para la petición **GetFeatureInfo** de acuerdo con las especificaciones OGC WMS 1.1.1 y 1.3.0 :

Parámetro	Necesario	Descripción
<i>SERVICE</i>	Sí	Nombre del servicio (WMS)
<i>VERSION</i>	No	Versión del servicio
<i>REQUEST</i>	Sí	Nombre de la solicitud (GetFeatureInfo)
<i>QUERY_LAYERS</i>	Sí	Capas a consultar
<i>LAYERS</i>	Sí	Capas para mostrar (idénticas a <i>QUERY_LAYERS</i>)
<i>STYLES</i>	No	Estilo de capas
<i>SRS / CRS</i>	Sí	Sistema de referencia de coordenadas
<i>BBOX</i>	No	Extensión del mapa
<i>WIDTH</i>	Sí	Ancho de la imagen en píxeles.
<i>HEIGHT</i>	Sí	Altura de la imagen en píxeles
<i>TRANSPARENT</i>	No	fondo transparente
<i>INFO_FORMAT</i>	No	Formato de salida
<i>FEATURE_COUNT</i>	No	Número máximo de objetos espaciales a devolver
<i>I</i>	No	Columna de píxeles del punto a consultar
<i>X</i>	No	Igual que el parámetro <i>I</i> , pero en WMS 1.1.1
<i>J</i>	No	Fila de píxeles del punto a consultar
<i>Y</i>	No	Igual que el parámetro <i>J</i> , pero en WMS 1.1.1
<i>WMS_PRECISION</i>	No	La precisión (número de dígitos) que se utilizará al devolver la geometría (consulte Cómo agregar geometría a la respuesta de la función). El valor predeterminado es -1, lo que significa que se utiliza la precisión definida en el proyecto.

En adición a los estándares, QGIS Server apoya los siguientes parámetros extra:

Parámetros	Necesario	Descripción
<i>MAP</i>	Sí	especifica el archivo de proyecto de QGIS
<i>FILTER</i>	No	Subconjunto de entidades
<i>FI_POINT_TOLERANCE</i>	No	Tolerancia en píxeles para capas punto
<i>FI_LINE_TOLERANCE</i>	No	Tolerancia en píxeles para capas línea
<i>FI_POLYGON_TOLERANCE</i>	No	Tolerancia en píxeles para capas polígono
<i>FILTER_GEOM</i>	No	Filtrado de geometría
<i>WITH_MAPTIP</i>	No	Agregar consejos de mapa a la salida
<i>WITH_GEOMETRY</i>	No	Agregar geometría a la salida

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetMap
&MAP=/home/qgis/projects/world.qgs
&LAYERS=mylayer1,mylayer2,mylayer3
&CRS=EPSG:4326
&WIDTH=400
&HEIGHT=400
&INFO_FORMAT=text/xml
&TRANSPARENT=TRUE
&QUERY_LAYERS=mylayer1
&FEATURE_COUNT=3
&I=250
&J=250
```

SOLICITUD

Este parámetro es `GetFeatureInfo` en caso de la petición **GetFeatureInfo**.

INFO_FORMAT

Este parámetro se puede utilizar para especificar el formato del resultado. Los valores disponibles son:

- `text/xml`
- `text/html`
- `text/plain`
- `application/vnd.ogc.gml`
- `application/json`

QUERY_LAYERS

Este parámetro especifica las capas a mostrar en el mapa. Nombres son separados por comas.

Además, El servidor de QGIS presenta opciones para seleccionar las capas por:

- nombre corto
- id capa

Ver las `LAYERS` definidas por parámetro en *GetMap* para mas información.

FEATURE_COUNT

Este parámetro especifica el número máximo de entidades por capa que se devolverán. Por ejemplo, si `QUERY_LAYERS` se establece en `layer1, layer2` y `FEATURE_COUNT` se establece en 3, se devolverán un máximo de 3 entidades de `layer1`. Asimismo, se devolverá un máximo de 3 entidades de `layer2`.

De formar predeterminada, sólo 1 objeto espacial por capa es devuelto.

I

Este parámetro, definido en WMS 1.3.0, le permite especificar la columna de pixeles del punto de consulta.

X

Igual que el parámetro `I`, pero definido en WMS 1.1.1.

J

Este parámetro, definido en WMS 1.3.0, le permite especificar la fila de pixeles del punto de consulta.

Y

Igual que el parámetro \mathcal{J} , pero definido en WMS 1.1.1.

FI_POINT_TOLERANCE

Este parámetro especifica la tolerancia en pixeles para capas de puntos.

FI_LINE_TOLERANCE

Este parámetro especifica la tolerancia en pixeles para capas de líneas.

FI_POLYGON_TOLERANCE

Este parámetro especifica la tolerancia en pixeles para capas de polígonos.

FILTER_GEOM

Este parámetro especifica una geometría WKT con la cuál las entidades deben intersecarse.

WITH_MAPTIP

Este parámetro especifica si agregar sugerencias de mapas a la salida.

Valores disponibles son (no en caso sensitivo):

- TRUE
- FALSE

WITH_GEOMETRY

Este parámetro especifica si se deben agregar geometrías a la salida. Para usar esta función, primero debe habilitar la opción: `guiabel:Añadir geometría a la entidad de respuesta` en el proyecto QGIS. Ver: ref: *Configura tu proyecto <Creatingwmsfromproject>*.

Valores disponibles son (no en caso sensitivo):

- TRUE
- FALSE

3.1.3 GetPrint

QGIS Server tiene la capacidad de crear una salida de diseño de impresión en formato pdf o píxel. Las ventanas de diseño de impresión del proyecto publicado se utilizan como plantillas. En la solicitud **GetPrint**, el cliente tiene la posibilidad de especificar parámetros de las etiquetas y mapas de diseño contenidos.

Parámetro para la petición **GetPrint**:

Parámetros	Necesario	Descripción
<i>MAP</i>	Sí	especifica el archivo de proyecto de QGIS
<i>SERVICE</i>	Sí	Nombre del servicio (WMS)
<i>VERSION</i>	No	Versión del servicio
<i>REQUEST</i>	Sí	Nombre de la solicitud (GetPrint)
<i>LAYERS</i>	No	Capas a mostrar
<i>TEMPLATE</i>	Sí	Plantilla de diseño a usar
<i>SRS / CRS</i>	Sí	Sistema de referencia de coordenadas
<i>FORMAT</i>	No	Formato de salida
<i>ATLAS_PK</i>	No	Características del Atlas
<i>STYLES</i>	No	Estilo de capas
<i>TRANSPARENT</i>	No	fondo transparente
<i>OPACITIES</i>	No	Opacidad de la capa o Grupo
<i>SELECTION</i>	No	Resalta entidades
<i>mapX:EXTENT</i>	No	Extensión del mapa "X"
<i>mapX:LAYERS</i>	No	Capas del mapa "X"
<i>mapX:STYLES</i>	No	Estilo de capas del mapa "X"
<i>mapX:SCALE</i>	No	Escala de capas del mapa "X"
<i>mapX:ROTATION</i>	No	Rotación del mapa "X"
<i>mapX:GRID_INTERVAL_X</i>	No	Intervalo de cuadrícula en eje x del mapa "X"
<i>mapX:GRID_INTERVAL_Y</i>	No	Intervalo de cuadrícula en eje y del mapa "X"

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WMS
&VERSION=1.3.0
&REQUEST=GetPrint
&MAP=/home/qgis/projects/world.qgs
&CRS=EPSG:4326
&FORMAT=png
&TEMPLATE=Layout%201
&map0:EXTENT=-180,-90,180,90
&map0:LAYERS=mylayer1,mylayer2,mylayer3
&map0:OPACITIES=125,200,125
&map0:ROTATION=45
```

Tenga en cuenta que la plantilla de diseño puede contener más de un mapa. De esta forma, si quieres configurar un mapa específico, debes utilizar los parámetros *mapX*: donde X es un número positivo que puedes recuperar gracias a la solicitud **GetProjectSettings**.

Por ejemplo:

```
<WMS_Capabilities>
...
<ComposerTemplates xsi:type="wms:_ExtendedCapabilities">
<ComposerTemplate width="297" height="210" name="Druckzusammenstellung 1">
<ComposerMap width="171" height="133" name="map0"/>
<ComposerMap width="49" height="46" name="map1"/></ComposerTemplate>
</ComposerTemplates>
...
</WMS_Capabilities>
```

SOLICITUD

Este parámetro tiene que ser `GetPrint` para la petición **GetPrint**

TEMPLATE

Este parámetro puede ser usado para especificar el nombre de la plantilla de composición a usar para la impresión.

FORMATO

Este parámetro especifica el formato de la imagen de mapa. Los valores disponibles son:

- `png` (valor predeterminado)
- `image/png`
- `jpg`
- `jpeg`
- `image/jpeg`
- `svg`
- `image/svg`
- `image/svg+xml`
- `pdf`
- `application/pdf`

Si el parámetro `FORMAT` es diferente de uno de estos valores, entonces se devuelve una excepción.

ATLAS_PK

Este parámetro permite la activación del renderizado Atlas indicando qué características queremos imprimir. Para recuperar un atlas con todas las características, se puede usar el símbolo `*` (según el número máximo de características permitidas en la configuración del proyecto).

Cuando `FORMATO` es `pdf`, se devuelve un solo documento PDF que combina las páginas de funciones. Para todos los demás formatos, se devuelve una sola página.

mapX:EXTENT

Este parámetro especifica la extensión para un elemento del diseño de mapa como `xmin,ymin,xmax,ymax`.

mapX:ROTATION

Este parámetro especifica la rotación del mapa en grados.

mapX:GRID_INTERVAL_X

Este parámetro especifica la densidad de línea de cuadrícula en la dirección X.

mapX:GRID_INTERVAL_Y

Este parámetro especifica la densidad de línea de cuadrícula en la dirección Y.

mapX:SCALE

Este parámetro especifica la escala de mapa para un elemento de la composición de mapa. Es útil para asegurar la visibilidad basada en escala de capas y etiquetas incluso si el cliente y servidor puedan tener diferentes algoritmos para calcular el denominador de escala.

mapX:LAYERS

Este parámetro especifica las capas para un elemento de composición de mapa. Ver *GetMap Capas* para mas información en este parámetro.

mapX:STYLES

Este parámetro especifica los estilos de las capas definidos en un elemento de mapa de diseño específico. Ver *GetMap Estilos* para mas información en este parámetro.

3.1.4 GetLegendGraphics

Varios parámetros adicionales están disponibles para cambiar el tamaño de los elementos de leyenda:

- **BOXSPACE** espacio entre el marco de la leyenda y el contenido (mm)
- **FORMAT**, `image/jpeg`, `image/png` o `application/json`. Para JSON, los símbolos se codifican con Base64 y la mayoría de las demás opciones relacionadas con el diseño o las fuentes no se tienen en cuenta porque la leyenda debe construirse en el lado del cliente.
- **LAYERSPACE** espacio vertical entre capas (mm)
- **LAYERTITLESPACE** espacio vertical entre el título de la capa y los elementos siguientes (mm)
- **SYMBOLSPACE** espacio vertical entre símbolo y artículo siguiente (mm)
- **ICONLABELSPACE** espacio horizontal entre el símbolo y el texto de la etiqueta (mm)
- **SYMBOLWIDTH** ancho de la previsualización de símbolo (mm)
- **SYMBOLHEIGHT** alto de la previsualización de símbolo (mm)

Estos parámetros cambian las propiedades de fuente para títulos de capa y etiquetas de elementos:

- **LAYERFONTFAMILY / ITEMFONTFAMILY** familia de fuentes para el título de la capa / texto del elemento
- **LAYERFONTBOLD / ITEMFONTBOLD** TRUE usar una fuente en negrita
- **LAYERFONTSIZE / ITEMFONTSIZE** Tamaño de fuente en puntos
- **LAYERFONTITALIC / ITEMFONTITALIC** TRUE para usar fuente itálica
- **LAYERFONTCOLOR / ITEMFONTCOLOR** código de color Hexadecimal (p.ej. #FF0000 para rojo)
- **LAYERTITLE** FALSE para obtener solo los gráficos de leyenda sin título de capa
- **RULELABEL**:

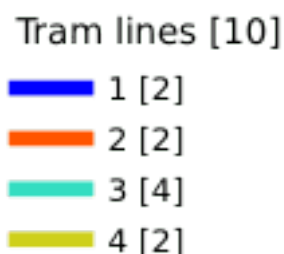
- FALSE gráficos de leyenda sin etiquetas de artículo
- AUTO ocultar la etiqueta del elemento para las capas con representación de *símbolo único*

Leyenda basada en contenido. Estos parámetros permiten que el cliente solicite una leyenda que muestre solo los símbolos de las características que caen en el área solicitada:

- **BBOX** el área geográfica para la que se debe construir la leyenda
- **CRS / SRS** el sistema de referencia de coordenadas adoptado para definir las coordenadas BBOX
- ****SRCWIDTH / SRCHEIGHT**** si se establecen, estos deben coincidir con los parámetros **WIDTH** y **HEIGHT** de la solicitud **GetMap**, para permitir que QGIS Server escale los símbolos de acuerdo con el tamaño de la imagen de la vista del mapa.

Las características de leyenda basadas en contenido se basan en la implementación de [UMN MapServer](#):

- **SHOWFEATURECOUNT** si se establece en **TRUE**, agrega en la leyenda el recuento de características de las características como en la siguiente imagen:



- **RULE** configúrelo en un nombre de regla dado para obtener solo el símbolo de regla nombrado
- **WIDTH/HEIGHT** el tamaño de la imagen de leyenda generada si el parámetro **RULE** es establecido

3.1.5 GetProjectSettings

Este tipo de solicitud funciona de manera similar a **GetCapabilities**, pero es más específico para QGIS Server y permite que un cliente lea información adicional que no está disponible en la salida de **GetCapabilities**:

- visibilidad inicial de las capas
- información acerca de atributos vectoriales y sus tipos de edición
- información acerca de orden de capa y orden de dibujado
- lista de capas publicadas en WFS
- mostrar si un grupo en el árbol de capas es mutuamente exclusivo

3.2 Web Feature Service (WFS)

Los estándares **WFS 1.0.0** y **1.1.0** implementados en QGIS Server proporcionan una interfaz HTTP para consultar características geográficas de un proyecto QGIS. Una solicitud WFS típica define el proyecto QGIS a usar y la capa a consultar.

Documento de especificaciones según el número de versión del servicio:

- [WFS 1.0.0](#)
- [WFS 1.1.0](#)

Solicitudes standard proporcionadas por el servidor QGIS:

Solicitud	Descripción
GetCapabilities	Devuelve metadatos XML con información sobre el servidor
GetFeature	Devuelve una selección de objetos espaciales
DescribeFeatureType	Devuelve una descripción de los tipos de entidades y propiedades.
Transacción	Permite insertar, actualizar o eliminar objetos

3.2.1 GetFeature

Parámetros estándar para la solicitud **GetFeature** de acuerdo con las especificaciones OGC WFS 1.0.0 y 1.1.0:

Parámetros	Necesario	Descripción
SERVICIO	Sí	Nombre del servicio
VERSIÓN	No	Versión del servicio
SOLICITUD	Sí	Nombre de la petición
TYPENAME	No	Nombre de las capas
OUTPUTFORMAT	No	Formato de Salida
RESULTTYPE	No	Tipo de resultado
PROPERTYNAME	No	Nombre de las propiedades a regresar
MAXFEATURES	No	Número máximo de objetos espaciales a devolver
SRSNAME	No	Sistema de referencia de coordenadas
FEATUREID	No	Filtrar los objetos espaciales por ids
FILTER	No	OGC Codificación de Filtro
BBOX	No	Extensión del Mapa
SORTBY	No	Ordena los resultados

En adición a los estándares, QGIS Server apoya los siguientes parámetros extra:

Parámetros	Necesario	Descripción
MAPA	Sí	especifica el archivo de proyecto de QGIS
STARTINDEX	No	Paginamiento
GEOMETRYNAME	No	Tipo de geometría para regresar
EXP_FILTER	No	Filtrado de expresión

SERVICIO

Este parámetro tiene que ser WFS en caso de la solicitud **GetFeature**.

Por ejemplo:

```
http://localhost/qgisserver?
SERVICE=WFS
&...
```

VERSIÓN

Este parámetro permite especificar la versión del servicio a utilizar. Los valores disponibles para el parámetro **VERSIÓN** son:

- 1.0.0
- 1.1.0

Si no se indica una versión en la solicitud, entonces se usa 1.1.0 por defecto.

URL ejemplo:

```
http://localhost/qgisserver?  
SERVICE=WFS  
&VERSION=1.1.0  
&...
```

SOLICITUD

Este parámetro es `GetFeature` en el caso de la solicitud **GetFeature**.

URL ejemplo:

```
http://localhost/qgisserver?  
SERVICE=WFS  
&VERSION=1.1.0  
&REQUEST=GetFeature  
&...
```

RESULTTYPE

Este parámetro se puede utilizar para especificar el tipo de resultado que se devolverá. Los valores disponibles son:

- `results`: el comportamiento predeterminado
- `hits`: devuelve solo un recuento de objetos

URL ejemplo:

```
http://localhost/qgisserver?  
SERVICE=WFS  
&VERSION=1.1.0  
&REQUEST=GetFeature  
&RESULTTYPE=hits  
&...
```

GEOMETRYNAME

Este parámetro se puede utilizar para especificar el tipo de geometría que se devolverá para las entidades. Los valores disponibles son:

- `extent`
- `centroid`
- `none`

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WFS
&VERSION=1.1.0
&REQUEST=GetFeature
&GEOMETRYNAME=centroid
&...
```

STARTINDEX

Este parámetro es estándar en WFS 2.0, pero es una extensión para WFS 1.0.0. En realidad, se puede usar para omitir algunas funciones en el conjunto de resultados y, en combinación con MAXFEATURES, brinda la capacidad de revisar los resultados.

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WFS
&VERSION=1.1.0
&REQUEST=GetFeature
&STARTINDEX=2
&...
```

3.3 Web Map Tile Service (WMTS)

El estándar WMTS 1.0.0 implementado en QGIS Server proporciona una interfaz HTTP para solicitar imágenes de mapas en mosaico generadas a partir de un proyecto QGIS. Una solicitud WMTS típica definió el proyecto QGIS a usar, algunos parámetros WMS como capas para renderizar, así como parámetros de mosaico.

Documento de especificaciones del servicio:

- [WMTS 1.0.0](#)

Solicitudes standard proporcionadas por el servidor QGIS:

Solicitud	Descripción
GetCapabilities	Devuelve metadatos XML con información sobre el servidor
GetTile	Devuelve una tesela
GetFeatureInfo	Recupera datos (geometría y valores) para una ubicación de píxeles

3.3.1 GetCapabilities

Parámetros estándar para la solicitud **GetCapabilities** de acuerdo con las especificaciones OGC WMTS 1.0.0:

Parámetros	Necesario	Descripción
SERVICIO	Sí	Nombre del servicio (WMTS)
SOLICITUD	Sí	Nombre de la solicitud (GetCapabilities)

En adición a los estándares, QGIS Server apoya los siguientes parámetros extra:

Parámetros	Necesario	Descripción
MAPA	Sí	especifica el archivo de proyecto de QGIS

URL ejemplo:

```
http://localhost/qgisserver?  
SERVICE=WMTS  
&REQUEST=GetCapabilities  
&MAP=/home/qgis/projects/world.qgs
```

SERVICIO

Este parámetro tiene que ser WMTS en caso de la solicitud **** GetCapabilities ****.

SOLICITUD

Este parámetro es `GetCapabilities` en el caso de la solicitud **GetCapabilities**.

MAPA

Este parámetro permite definir el archivo de proyecto QGIS a utilizar.

3.3.2 GetTile

Parámetros estándar para la solicitud **** GetTile **** de acuerdo con las especificaciones OGC WMTS 1.0.0:

Parámetros	Necesario	Descripción
SERVICIO	Sí	Nombre del servicio (WMTS)
SOLICITUD	Sí	Nombre de la solicitud (GetTile)
LAYER	Sí	Identificador de capa
FORMATO	Sí	Formato de salida de la tesela
TILEMATRIXSET	Sí	Nombre de la pirámide
TILEMATRIX	Sí	Enmallado
TILEROW	Sí	Coordenada de fila en la malla
TILECOL	Sí	Coordenada de columna en la malla

En adición a los estándares, QGIS Server apoya los siguientes parámetros extra:

Parámetros	Necesario	Descripción
MAPA	Sí	especifica el archivo de proyecto de QGIS

URL ejemplo:

```
http://localhost/qgisserver?  
SERVICE=WMTS  
&REQUEST=GetTile  
&MAP=/home/qgis/projects/world.qgs  
&LAYER=mylayer  
&FORMAT=image/png  
&TILEMATRIXSET=EPSG:4326  
&TILEROW=0  
&TILECOL=0
```

SERVICIO

Este parámetro tiene que ser WMTS en el caso de la solicitud **** GetTile ****.

SOLICITUD

Este parámetro es GetTile en caso de petición **GetTile**.

LAYER

Este parámetro permite especificar la capa que se mostrará en el mosaico.

Además, QGIS Server introdujo algunas opciones para seleccionar una capa mediante:

- un nombre corto
- el id de capa

El nombre corto de una capa se puede configurar mediante *Propiedades -> Metadatos* en el menú de capas. Si se define el nombre corto, entonces se usa por defecto en lugar del nombre de la capa:

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetTile
&LAYER=mynickname
&...
```

Además, hay una opción de proyecto que permite seleccionar capas por su identificación en: *menuselection:OWSServer -> WMS capabilities* menú del diálogo *Proyecto ▶ Propiedades de Proyecto*. Para activar esta opción, la casilla de verificación *Usar ID de capa como nombres* debe estar seleccionada.

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetTile
&LAYER=mylayerid1
&...
```

FORMATO

Este parámetro se puede utilizar para especificar el formato de la imagen del mosaico. Los valores disponibles son:

- jpg
- jpeg
- image/jpeg
- image/png

Si el parámetro `FORMATO` es diferente de uno de estos valores, entonces se usa el formato predeterminado PNG en su lugar.

TILEMATRIXSET

Este parámetro define el SRC que se utilizará al calcular la pirámide subyacente. Formato: EPSG: XXXX.

TILEMATRIX

Este parámetro permite definir la matriz a utilizar para el mosaico de salida.

TILEROW

Este parámetro permite seleccionar la fila del mosaico para entrar en la matriz.

TILECOL

Este parámetro permite seleccionar la columna del mosaico para entrar en la matriz.

MAPA

Este parámetro permite definir el archivo de proyecto QGIS a utilizar.

Como se mencionó en *tabla de parámetros de GetMap*, MAP es obligatorio porque una solicitud necesita un proyecto QGIS para funcionar. Sin embargo, la variable de entorno QGIS_PROJECT_FILE se puede utilizar para definir un proyecto QGIS predeterminado. En este caso específico, MAP ya no es un parámetro obligatorio. Para obtener más información, puede consultar *Configuración avanzada*.

3.3.3 GetFeatureInfo

Parámetros estándar para la solicitud **GetFeatureInfo** de acuerdo con la especificación OGC WMTS 1.0.0:

Parámetros	Necesario	Descripción
SERVICIO	Sí	Nombre del servicio (WMTS)
SOLICITUD	Sí	Nombre de la solicitud (GetFeatureInfo)
LAYER	Sí	Identificador de capa
INFOFORMAT	No	Formato de salida
I	No	coordenada X de un píxel
J	No	coordenada Y de un píxel
TILEMATRIXSET	Sí	<i>See GetTile</i>
TILEMATRIX	Sí	<i>See GetTile</i>
TILEROW	Sí	<i>See GetTile</i>
TILECOL	Sí	<i>See GetTile</i>

En adición a los estándares, QGIS Server apoya los siguientes parámetros extra:

Parámetros	Necesario	Descripción
MAPA	Sí	especifica el archivo de proyecto de QGIS

URL ejemplo:

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&MAP=/home/qgis/projects/world.qgs
&LAYER=mylayer
```

(continúe en la próxima página)

(proviene de la página anterior)

```
&INFOFORMAT=image/html
&I=10
&J=5
```

SERVICIO

Este parámetro tiene que ser WMTS en el caso de la solicitud **GetFeatureInfo**.

SOLICITUD

Este parámetro es `GetFeatureInfo` en caso de la petición **GetFeatureInfo**.

MAPA

Este parámetro permite definir el archivo de proyecto QGIS a utilizar.

Como se mencionó en *tabla de parámetros de GetMap*, MAP es obligatorio porque una solicitud necesita un proyecto QGIS para funcionar. Sin embargo, la variable de entorno `QGIS_PROJECT_FILE` se puede utilizar para definir un proyecto QGIS predeterminado. En este caso específico, MAP ya no es un parámetro obligatorio. Para obtener más información, puede consultar *Configuración avanzada*.

LAYER

Este parámetro permite especificar la capa que se mostrará en el mosaico.

Además, QGIS Server introdujo algunas opciones para seleccionar una capa mediante:

- un nombre corto
- el id de capa

El nombre corto de una capa se puede configurar mediante *Propiedades -> Metadatos* en el menú de capas. Si se define el nombre corto, entonces se usa por defecto en lugar del nombre de la capa:

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&LAYER=mynickname
&...
```

Además, hay una opción de proyecto que permite seleccionar capas por su identificación en: *menuselection:OWS Server -> WMS capabilities* menú del diálogo *Proyecto ▶ Propiedades de Proyecto*. Para activar esta opción, la casilla de verificación *Usar ID de capa como nombres* debe estar seleccionada.

```
http://localhost/qgisserver?
SERVICE=WMTS
&REQUEST=GetFeatureInfo
&LAYER=mylayerid1
&...
```

INFOFORMAT

Este parámetro permite definir el formato de salida del resultado. Los valores disponibles son:

- `text/xml`
- `text/html`
- `text/plain`
- `application/vnd.ogc.gml`

El valor predeterminado es `text/plain`.

I

Este parámetro permite definir la coordenada X del píxel para el que queremos recuperar la información subyacente.

J

Este parámetro permite definir la coordenada Y del píxel para el que queremos recuperar información subyacente.

3.4 WFS3 (Objetos API OGC)

WFS3 es la primera implementación de la nueva generación de protocolos OGC. Es descrito por el [OGC API - Features - Part 1: Core](#) documento.

Aquí hay un resumen informal rápido de las diferencias más importantes entre el conocido protocolo WFS y WFS3:

- WFS3 está basada en una API REST
- API WFS3 debe seguir las especificaciones [OPENAPI](#)
- WFS3 admite múltiples formatos de salida, pero no dicta ninguno (solo GeoJSON y HTML están disponibles actualmente en QGIS WFS3) y usa [negociación de contenido](#) para determinar qué formato se servirá al cliente
- JSON y HTML son ciudadanos de primera clase en WFS3
- WFS3 se autodocumenta (a través del punto final `/api`)
- WFS3 es totalmente navegable (a través de enlaces) y navegable

Importante: Si bien la implementación de WFS3 en QGIS puede hacer uso del parámetro `MAP` para especificar el archivo del proyecto, la especificación [OPENAPI](#) no permite parámetros de consulta adicionales. Por esta razón, se recomienda encarecidamente que `MAP` no esté expuesto en la URL y que el archivo del proyecto se especifique en el entorno por otros medios (es decir, estableciendo `QGIS_PROJECT_FILE` en el entorno a través de una regla de reescritura del servidor web).

Nota: La [API](#) punto final proporciona documentación completa de todos los parámetros y formatos de salida admitidos de su servicio. Los siguientes párrafos solo describirán los más importantes.

3.4.1 Representación de recurso

La implementación de QGIS Server WFS3 actualmente admite los siguientes formatos de representación de recursos (salida):

- HTML
- JSON

El formato que se sirve realmente dependerá de la negociación de contenido, pero se puede solicitar explícitamente un formato específico agregando un especificador de formato a los puntos finales.

Las extensiones de especificador de formato admitidas son:

- .json
- .html

Los alias de especificador de formato adicionales se pueden definir por puntos finales específicos:

- .openapi: alias para .json soportado por la **API** endpoint
- .geojson: alias para .json soportado por los endpoints **Features** y **Feature**

3.4.2 Puntos finales

La API proporciona una lista de puntos finales que los clientes pueden recuperar. El sistema está diseñado de tal manera que cada respuesta proporciona un conjunto de enlaces para navegar a través de todos los recursos proporcionados.

Los puntos finales proporcionados por la implementación de QGIS son:

Nombre	Ruta	Descripción
Página de Aterrizaje	/	Información general sobre el servicio y proporciona enlaces a todos los puntos finales disponibles.
Conformidad	/conformance	Información sobre la conformidad del servicio con los estándares.
API	/api	Descripción completa de los puntos finales proporcionados por el servicio y la estructura de los documentos devueltos
Colecciones	/collections	Lista de todas las colecciones (es decir, “capas vectoriales”) proporcionadas por el servicio
Colección	/collections/{collectionId}	Información sobre una colección (nombre, metadatos, extensión, etc.)
Prestaciones	/collections/{collectionId}/items	Lista de los objetos espaciales provistos por la colección
Objeto espacial	/collections/{collectionId}/items/{featureId}	Información acerca de un objeto espacial solo

Página de Aterrizaje

El punto final principal es la **Página de destino**. Desde esa página es posible navegar a todos los puntos finales de servicio disponibles. La **página de destino** debe proporcionar enlaces a

- la definición de API (ruta /api relaciones de enlace `service-desc` y `service-doc`),
- la declaración de conformidad (ruta /conformidad, relación de enlace `conformidad`), y
- las Colecciones (ruta /colecciones, relación de enlace `datos`).

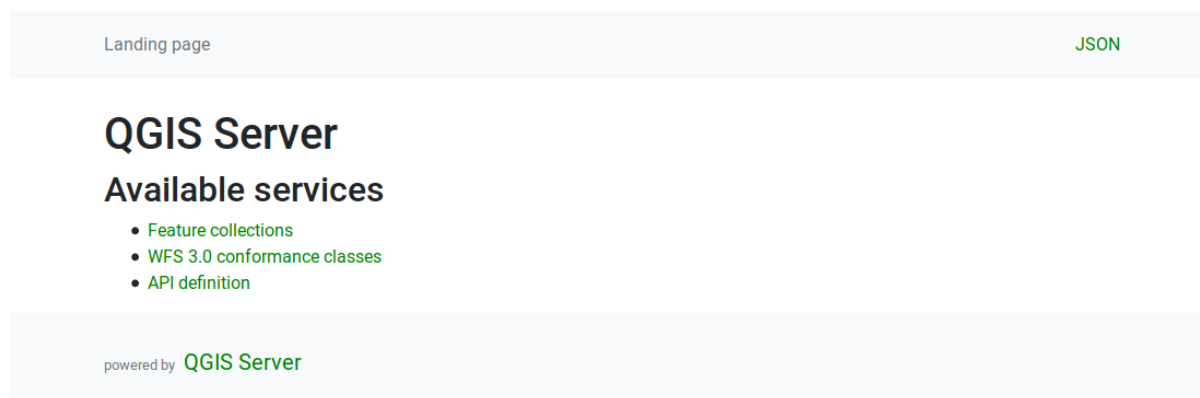


Figura 3.2: Página de aterrizaje de Servidor WFS3

Definición API

La **Definición de API** es una descripción compatible con OPENAPI de la API proporcionada por el servicio. En su representación HTML, es una página navegable donde todos los puntos finales y sus formatos de respuesta se enumeran y documentan con precisión. La ruta de este punto final es `/api`.

La definición de API proporciona una documentación completa y autorizada del servicio, incluidos todos los parámetros admitidos y los formatos devueltos.

Nota: Este punto final es análogo a `GetCapabilities` de WFS's

Lista de colecciones

El punto final de colecciones proporciona una lista de todas las colecciones disponibles en el servicio. Dado que el servicio «sirve» a un solo proyecto QGIS, las colecciones son las capas vectoriales del proyecto actual (si se publicaron como WFS en las propiedades del proyecto). La ruta de este punto final es `/collections/`.

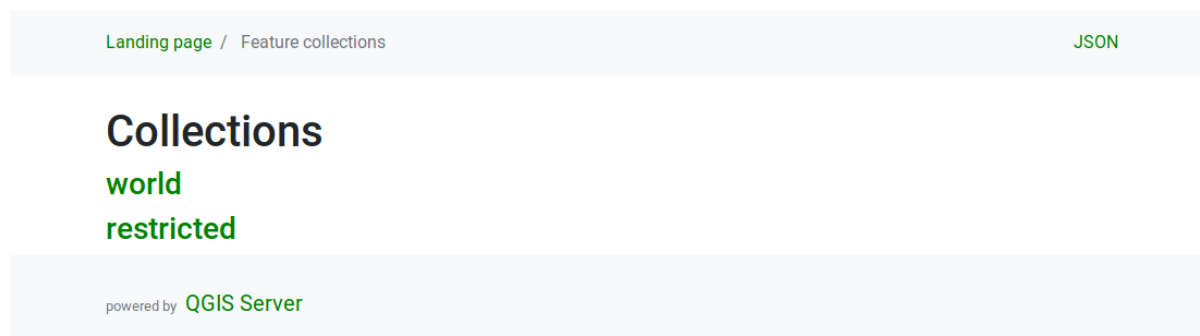


Figura 3.3: Página de lista de colecciones de servidor WFS3

Detalle de colección

Si bien el punto final de las colecciones no proporciona información detallada sobre cada colección disponible, esa información está disponible en los puntos finales `/collections/{collectionId}`. La información típica incluye la extensión, una descripción, SRC y otros metadatos.

La representación HTML también proporciona un mapa navegable con las funciones disponibles.

The screenshot shows a web page for a collection named 'world'. At the top, there is a breadcrumb trail: 'Landing page / Collections / world' and a 'JSON' link. The main heading is 'world' in green. Below it, the section 'Available CRSs' lists four CRS URLs:

- <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
- <http://www.opengis.net/def/crs/EPSSG/9.6.2/4326>
- <http://www.opengis.net/def/crs/EPSSG/9.6.2/3857>
- <http://www.opengis.net/def/crs/EPSSG/9.6.2/900913>

 The 'Extent' section lists the following values:

- West: -179.99999999999997
- South: -90.0
- East: 180.0
- North: 83.62359600000008

 To the right of the text is a map showing a world map with a blue rectangular bounding box around the central region. The map includes a zoom control in the top-left corner and a 'Leaflet | © OpenStreetMap contributors' attribution in the bottom-right corner. At the bottom of the page, it says 'powered by QGIS Server'.

Figura 3.4: Página de detalles de la colección del servidor WFS3

Lista de objetos espaciales

Este punto final proporciona una lista de todas las funciones de una colección que conocen el ID de la colección. La ruta de este punto final es `/collections/{collectionId}/items`.

La representación HTML también proporciona un mapa navegable con las funciones disponibles.

Nota: Este punto final es análogo a `GetFeature` en WFS 1 y WFS 2.

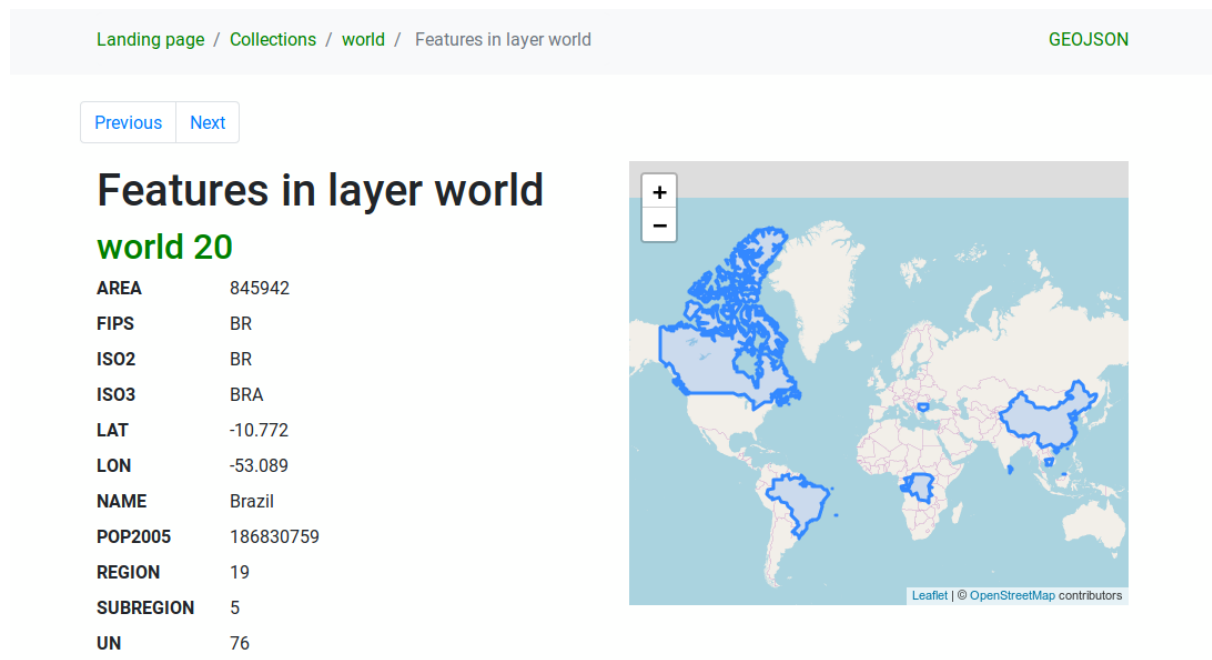


Figura 3.5: Página de lista de características del servidor WFS3

Detalle de objeto espacial

Este punto final proporciona toda la información disponible sobre una única entidad, incluidos los atributos de la entidad y su geometría. La ruta de este punto final es `/collections/{collectionId}/items/{itemId}`.

La representación HTML también proporciona un mapa navegable con la geometría de la entidad.



Figura 3.6: Página de detalles de la entidad del servidor WFS3

3.4.3 Paginación

La paginación de una larga lista de características se implementa en la API de OGC a través de enlaces `siguiente` y `anterior`, el servidor QGIS construye estos enlaces agregando `límite` y `compensación` como parámetros de cadena de consulta.

URL ejemplo:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?offset=10&limit=10
```

Nota: El valor máximo aceptable para `límite` se puede configurar con el ajuste de configuración del servidor `QGIS_SERVER_API_WFS3_MAX_LIMIT` (ver: *Variables de entorno*).

3.4.4 Filtrado de objetos espaciales

Las funciones disponibles en una colección se pueden filtrar/buscar especificando uno o más filtros.

Filtro de fecha y hora

Las colecciones con atributos de fecha y / o fecha y hora se pueden filtrar especificando un argumento `fecha` y `hora` en la cadena de consulta. De forma predeterminada, el primer campo de fecha / fecha y hora se utiliza para el filtrado. Este comportamiento se puede configurar estableciendo una dimensión de «Fecha» u «Hora» en *Servidor QGIS* -> *Dimensión* del cuadro de diálogo de propiedades de la capa.

La sintaxis de filtrado de fecha y hora se describe completamente en *Definición API* y también admite rangos (se incluyen los valores de inicio y finalización) además de valores únicos.

URL ejemplos:

Devuelve solo las funciones con coincidencia de dimensión de fecha 2019-01-01

```
http://localhost/qgisserver/wfs3/collection_one/items.json?datetime=2019-01-01
```

Devuelve solo las funciones con coincidencia de dimensión de fecha y hora 2019-01-01T01:01:01

```
http://localhost/qgisserver/wfs3/collection_one/items.json?datetime=2019-01-01T01:01:01
```

Devuelve solo las entidades con dimensión de fecha y hora en el rango 2019-01-01T01:01:01 - 2019-01-01T12:00:00

```
http://localhost/qgisserver/wfs3/collection_one/items.json?datetime=2019-01-01T01:01:01/2019-01-01T12:00:00
```

Filtro de recuadro delimitador

Se puede especificar un filtro espacial de cuadro delimitador con el parámetro `bbox`:

El orden de los elementos separados por comas es:

- Esquina inferior izquierda, longitud WGS 84
- Esquina inferior izquierda, latitud WGS 84
- Esquina superior derecha, longitud WGS 84
- Esquina superior derecha, latitud WGS 84

Nota: Las especificaciones OGC también permiten un especificador bbox de 6 elementos donde el tercer y sexto elemento son los componentes Z, esto aún no es compatible con el servidor QGIS.

URL ejemplo:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?bbox=-180,-90,180,90
```

Si el SRC del cuadro delimitador no es WGS 84, se puede especificar un SRC diferente utilizando el parámetro opcional `bbox-crs`. El identificador de formato SRC debe estar en el formato 'OGC URI <<https://www.opengis.net/def/crs/>>`_:

URL ejemplo:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?bbox=913191,5606014,  
↪913234,5606029&bbox-crs=http://www.opengis.net/def/crs/EPSG/9.6.2/3857
```

Filtros de Atributos

Los filtros de atributos se pueden combinar con el filtro de cuadro delimitador y están en la forma general: `<attribute name>=<attribute value>`. Se pueden combinar varios filtros usando el operador «Y».

URL ejemplo:

filtra todas las entidades donde nombre es igual a «mi valor»

```
http://localhost/qgisserver/wfs3/collection_one/items.json?attribute_one=my%20value
```

Coincidencias parciales también son soportadas usando el operador * («estrella»)

URL ejemplo:

filtra todas las entidades donde el atributo nombre finaliza con «valor»

```
http://localhost/qgisserver/wfs3/collection_one/items.json?attribute_one=*value
```

3.4.5 Ordenar Entidad

Es posible ordenar el conjunto de resultados por valor de campo utilizando el parámetro de consulta `sortBy`.

Los resultados se ordenan en orden ascendente de forma predeterminada. Para ordenar los resultados en orden descendente, una bandera booleana (`sortdesc`) puede establecerse:

```
http://localhost/qgisserver/wfs3/collection_one/items.json?sortBy=name&sortdesc=1
```

3.4.6 Selección de atributos

Los atributos de características devueltos por una llamada `:ref:'ogc_api_features_features_list'` se pueden limitar agregando una lista de nombres de atributos separados por comas en el argumento opcional de cadena de consulta `properties`.

URL ejemplo:

devuelve solo el atributo nombre

```
http://localhost/qgisserver/wfs3/collection_one/items.json?properties=name
```

3.4.7 Personaliza las páginas HTML

La representación HTML utiliza un conjunto de plantillas HTML para generar la respuesta. La plantilla es analizada por un motor de plantillas llamado *inja*. Las plantillas se pueden personalizar reemplazándolas (consulte: *Sobreescritura de Plantilla*). La plantilla tiene acceso a los mismos datos que están disponibles para la representación JSON y algunas funciones adicionales están disponibles para la plantilla:

Funciones personalizadas de plantilla

- `path_append(path)`: adjunta una ruta de directorio a la actual url
- `path_chomp(n)`: borra el número especificado «n» de componentes del directorio de la ruta url actual
- `json_dump()`: imprime los datos JSON pasados a la plantilla
- `static(path)`: devuelve la URL completa a la ruta estática especificada. Por ejemplo: «`static(«/style/black.css»)`» con una ruta raíz «`http://localhost/qgisserver/wfs3`» devolverá «`http://localhost/qgisserver/wfs3/static/style/black.css`».
- `links_filter(links, key, value)`: Devuelve enlaces filtrados de una lista de enlaces
- `content_type_name(content_type)`: Devuelve un nombre corto de un tipo de contenido, por ejemplo «`text/html`» devolverá «`HTML`»

Sobreescritura de Plantilla

Las plantillas y los activos estáticos se almacenan en subdirectorios del directorio de recursos de API predeterminado del servidor QGIS (`/usr/share/qgis/resources/server/api/` en un sistema Linux), el directorio base se puede personalizar cambiando la variable de entorno `QGIS_SERVER_API_RESOURCES_DIRECTORY`.

Una instalación típica de Linux tendrá el siguiente árbol de directorios:

```

/usr/share/qgis/resources/server/api/
├── ogc
│   ├── schema.json
│   ├── static
│   │   ├── jsonFormatter.min.css
│   │   ├── jsonFormatter.min.js
│   │   └── style.css
│   └── templates
│       └── wfs3
│           ├── describeCollection.html
│           ├── describeCollections.html
│           ├── footer.html
│           ├── getApiDescription.html
│           ├── getFeature.html
│           ├── getFeatures.html
│           ├── getLandingPage.html
│           ├── getRequirementClasses.html
│           ├── header.html
│           ├── leaflet_map.html
│           └── links.html

```

Para anular las plantillas, puede copiar todo el árbol a otra ubicación y señalar `QGIS_SERVER_API_RESOURCES_DIRECTORY` a la nueva ubicación.

3.5 Parámetros extra soportados por todos los tipos de solicitudes

Los siguientes parámetros adicionales son compatibles con todos los protocolos.

- **FILE_NAME**: si se establece, la respuesta del servidor se enviará al cliente como un archivo adjunto con el nombre de archivo especificado.

Nota: No disponible para WFS3.

- **MAP**: De manera similar a MapServer, el parámetro MAP se puede usar para especificar la ruta al archivo del proyecto QGIS. Puede especificar una ruta absoluta o una ruta relativa a la ubicación del ejecutable del servidor (`qgis_mapserv.fcgi`). Si no se especifica, QGIS Server busca archivos `.qgs` en el directorio donde se encuentra el ejecutable del servidor.

Ejemplo:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?\  
REQUEST=GetMap&MAP=/home/qgis/projects/world.qgs&...
```

Nota: Puede definir un **QGIS_PROJECT_FILE** como una variable de entorno para decirle al ejecutable del servidor dónde encontrar el archivo del proyecto QGIS. Esta variable será la ubicación donde QGIS buscará el archivo del proyecto. Si no está definido, usará el parámetro MAP en la solicitud y finalmente buscará en el directorio ejecutable del servidor.

3.6 REDLINING

Esta función está disponible y se puede utilizar con las solicitudes `GetMap` y `GetPrint`.

La función de marcación roja se puede utilizar para pasar geometrías y etiquetas en la solicitud que el servidor superpone sobre la imagen devuelta estándar (mapa). Esto permite al usuario poner énfasis o quizás agregar algunos comentarios (etiquetas) a algunas áreas, ubicaciones, etc. que no están en el mapa estándar.

La petición `GetMap` está en el formato:

```
http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi?map=/world.qgs&SERVICE=WMS&  
↪VERSION=1.3.0&  
REQUEST=GetMap  
...  
&HIGHLIGHT_GEOM=POLYGON((590000 5647000, 590000 6110620, 2500000 6110620, 2500000↪  
↪5647000, 590000 5647000))  
&HIGHLIGHT_SYMBOL=<StyledLayerDescriptor><UserStyle><Name>Highlight</Name>  
↪<FeatureTypeStyle><Rule><Name>Symbol</Name><LineSymbolizer><Stroke><SvgParameter↪  
↪name="stroke">%23ea1173</SvgParameter><SvgParameter name="stroke-opacity">1</  
↪SvgParameter><SvgParameter name="stroke-width">1.6</SvgParameter></Stroke></  
↪LineSymbolizer></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>  
&HIGHLIGHT_LABELSTRING=Write label here  
&HIGHLIGHT_LABELSIZE=16  
&HIGHLIGHT_LABELCOLOR=%23000000  
&HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFF  
&HIGHLIGHT_LABELBUFFERSIZE=1.5
```

El equivalente `GetPrint` está en el formato (note que el parámetro `mapX`: se agrega para decir qué mapa tiene líneas rojas):


```

http://qgisplatform.demo/cgi-bin/qgis_mapserv.fcgi?map=/world.qgs&SERVICE=WMS&
↪VERSION=1.3.0&
REQUEST=GetPrint
...
&map0:HIGHLIGHT_GEOM=POLYGON((590000 5647000, 590000 6110620, 2500000 6110620,↪
↪2500000 5647000, 590000 5647000))
&map0:HIGHLIGHT_SYMBOL=<StyledLayerDescriptor><UserStyle><Name>Highlight</Name>
↪<FeatureTypeStyle><Rule><Name>Symbol</Name><LineSymbolizer><Stroke><SvgParameter↪
↪name="stroke">%23ea1173</SvgParameter><SvgParameter name="stroke-opacity">1</
↪SvgParameter><SvgParameter name="stroke-width">1.6</SvgParameter></Stroke></
↪LineSymbolizer></Rule></FeatureTypeStyle></UserStyle></StyledLayerDescriptor>
&map0:HIGHLIGHT_LABELSTRING=Write label here
&map0:HIGHLIGHT_LABELSIZE=16
&map0:HIGHLIGHT_LABELCOLOR=%23000000
&map0:HIGHLIGHT_LABELBUFFERCOLOR=%23FFFFFFF
&map0:HIGHLIGHT_LABELBUFFERSIZE=1.5
    
```

Aquí está la imagen obtenida por la solicitud anterior en la que se dibujan un polígono y una etiqueta sobre el mapa normal:



Figura 3.7: Respuesta del servidor a una solicitud GetMap con parámetros de marcado

Puede ver que hay varios parámetros en esta solicitud:

- **HIGHLIGHT_GEOM:** Puede añadir POINT, MULTILINESTRING, POLYGON etc. Admite geometrías multiparte. Aquí hay un ejemplo: `HIGHLIGHT_GEOM=MULTILINESTRING((0 0, 0 1, 1 1))`. Las coordenadas deben estar en el SRC de la solicitud GetMap/GetPrint.

- **HIGHLIGHT_SYMBOL**: Esto controla cómo se perfila la geometría y puede cambiar el ancho, el color y la opacidad del trazo.
- **HIGHLIGHT_LABELSTRING**: Puede pasar su texto de etiquetado a este parámetro.
- **HIGHLIGHT_LABELSIZE**: Este parámetro controla el tamaño de la etiqueta.
- **HIGHLIGHT_LABELFONT**: Este parámetro controla la fuente de la etiqueta (por ejemplo, Arial)
- **HIGHLIGHT_LABELCOLOR**: Este parámetro controla el color de la etiqueta.
- **HIGHLIGHT_LABELBUFFERCOLOR**: Este parámetro controla el color del búfer de etiquetas.
- **HIGHLIGHT_LABELBUFFERSIZE**: Este parámetro controla el tamaño del búfer de etiquetas.

3.7 Capas WMS Externas

QGIS Server permite incluir capas de servidores WMS externos en las solicitudes WMS GetMap y WMS GetPrint. Esto es especialmente útil si un cliente web utiliza una capa de fondo externa en el mapa web. Por razones de rendimiento, dichas capas deben ser solicitadas directamente por el cliente web (no en cascada a través del servidor QGIS). Sin embargo, para imprimir, estas capas deben conectarse en cascada a través del servidor QGIS para que aparezcan en el mapa impreso.

Las capas externas se pueden agregar al parámetro `LAYERS` como `EXTERNAL_WMS:<layername>`. Los parámetros para las capas WMS externas (por ejemplo, url, formato, dpiMode, crs, capas, estilos) se pueden proporcionar posteriormente como parámetros de servicio. `<layername>:<parameter>`. En una solicitud GetMap, esto podría verse así:

```
http://localhost/qgisserver?  
SERVICE=WMS&REQUEST=GetMap  
...  
&LAYERS=EXTERNAL_WMS:basemap,layer1,layer2  
&STYLES=,,  
&basemap:url=http://externalserver.com/wms.fcgi  
&basemap:format=image/jpeg  
&basemap:dpiMode=7  
&basemap:crs=EPSG:2056  
&basemap:layers=orthofoto  
&basemap:styles=default
```

En una solicitud GetMap, esto podría verse así:

```
http://localhost/qgisserver?  
SERVICE=WMS  
...  
&REQUEST=GetPrint&TEMPLATE=A4  
&map0:layers=EXTERNAL_WMS:basemap,layer0:layer2  
&map0:EXTENT=<minx,miny,maxx,maxy>  
&basemap:url=http://externalserver.com/wms.fcgi  
&basemap:format=image/jpeg  
&basemap:dpiMode=7  
&basemap:crs=EPSG:2056  
&basemap:layers=orthofoto  
&basemap:styles=default
```

3.8 Catálogo del servidor QGIS

El QGIS Server Catalog es un catálogo simple que muestra la lista de proyectos QGIS atendidos por el QGIS Server. Proporciona un sitio web totalmente navegable y fácil de usar con capacidades básicas de mapeo para explorar rápidamente los conjuntos de datos expuestos a través de esos proyectos QGIS.

El QGIS Server catalog usa las variables QGIS_SERVER_LANDING_PAGE_PROJECTS_DIRECTORIES y QGIS_SERVER_LANDING_PAGE_PROJECTS_PG_CONNECTIONS (ver *Variables de entorno*)

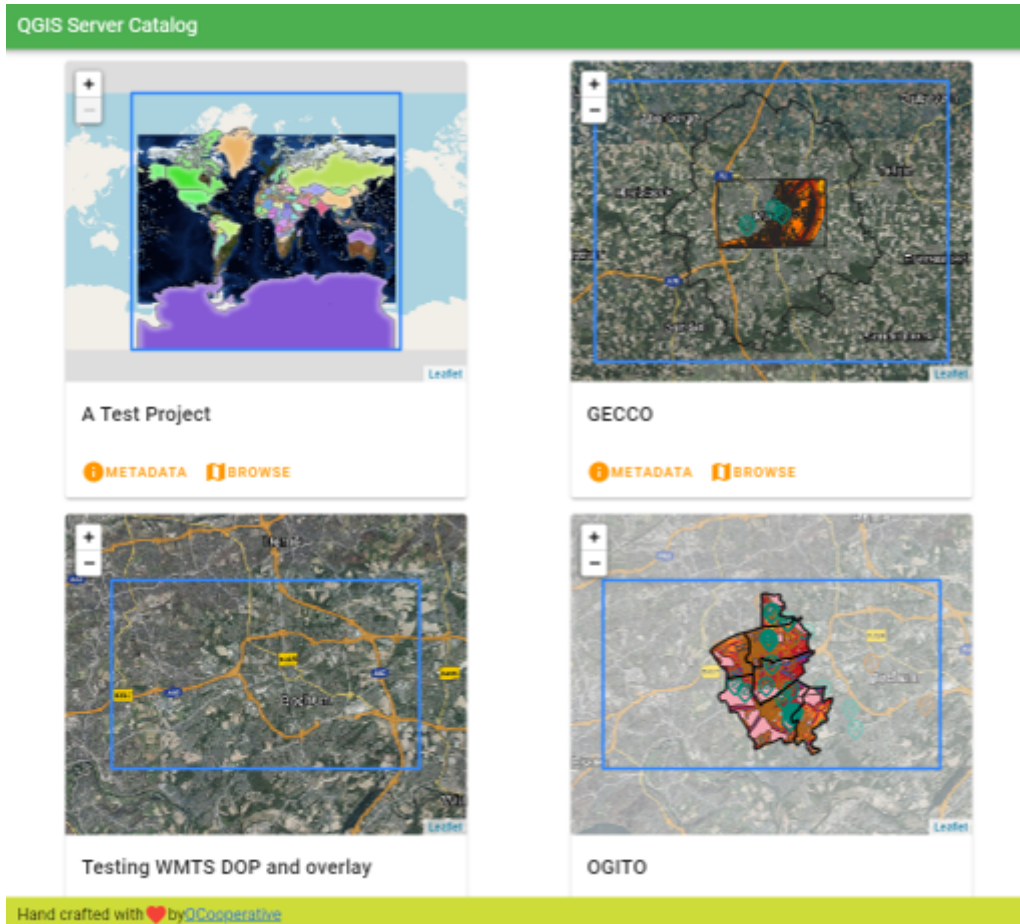


Figura 3.8: Página de lista de proyectos del catálogo de servidores

Puedes consultar los metadatos asociados a un proyecto y los servicios que presta. También se proporcionan enlaces a esos servicios.

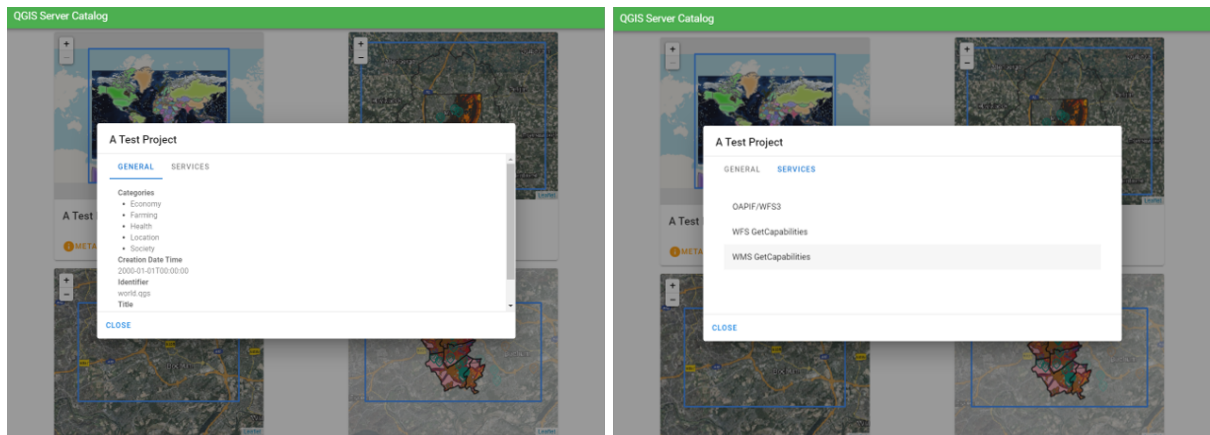


Figura 3.9: Server Catalog, metadatos asociados a un proyecto y servicios (links to) que proporciona.

Al navegar por un proyecto, se enumera el conjunto de datos que sirve.

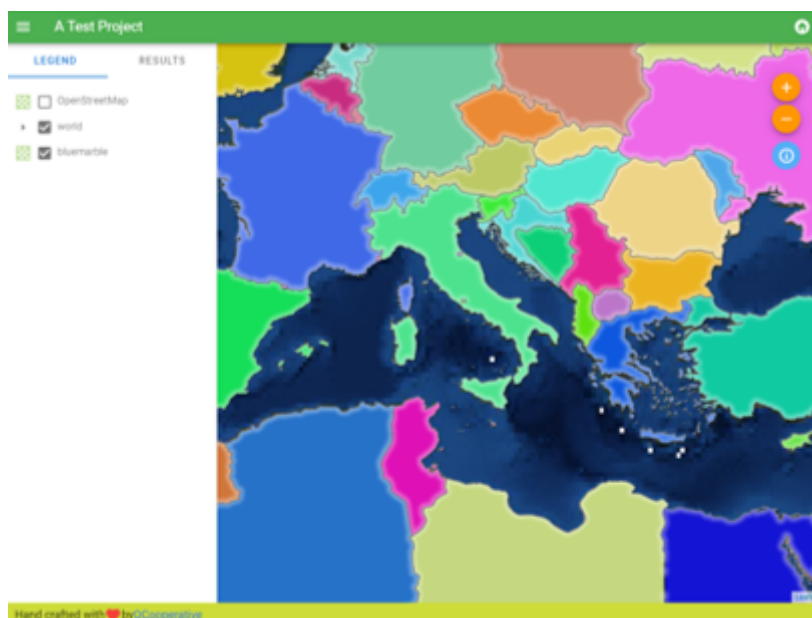


Figura 3.10: Examinar un conjunto de datos servido por un proyecto en el Server Catalog

Utilice el botón derecho del ratón en una capa para mostrar la tabla de atributos asociada a ella.

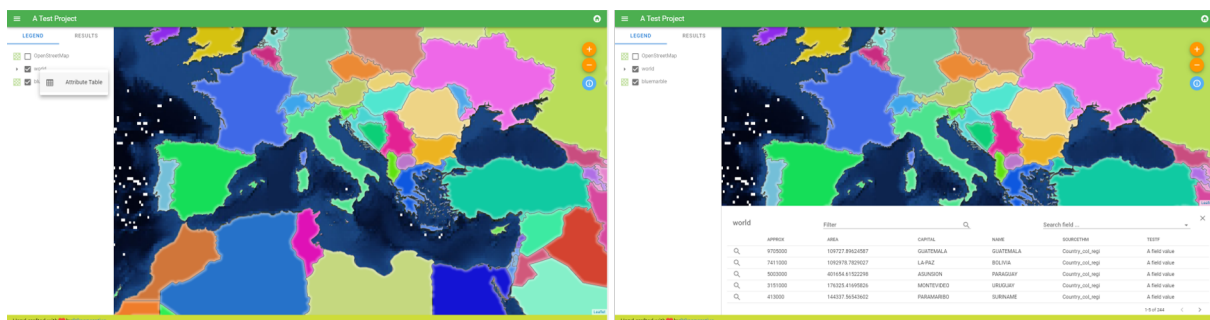


Figura 3.11: Tabla de atributos asociada a una capa

Es posible consultar información de los elementos en el mapa como se muestra en la siguiente imagen:

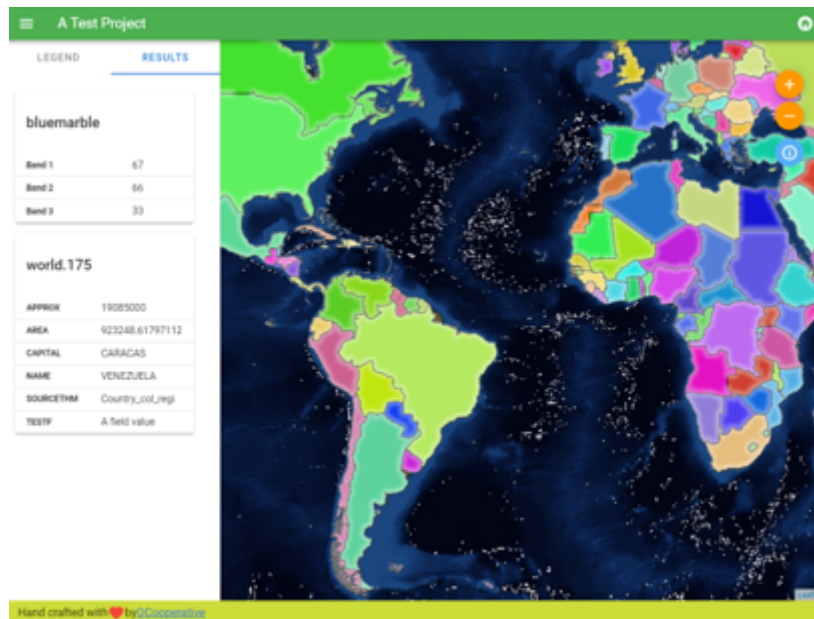


Figura 3.12: Consulta de información de un elemento del mapa

4.1 Instalación

La instalación de un complemento de ejemplo HelloWorld para probar los servidores. Se puede crear un directorio para mantener los complementos del servidor. Esto se especificará en la configuración del host virtual y transmitirlo al servidor a través de una variable de entorno.

```
mkdir -p /var/www/qgis-server/plugins
cd /var/www/qgis-server/plugins
wget https://github.com/elpaso/qgis-helloserver/archive/master.zip
unzip master.zip
mv qgis-helloserver-master HelloServer
```

4.2 Configuración del servidor HTTP

4.2.1 Apache

Para poder usar complementos de servidor, FastCGI necesita saber dónde buscar. Entonces, tenemos que modificar el archivo de configuración de Apache para indicar la variable de entorno **QGIS_PLUGINPATH** a FastCGI:

```
FcgidInitialEnv QGIS_PLUGINPATH "/var/www/qgis-server/plugins"
```

Además, es necesaria una autorización HTTP básica para jugar con el complemento HelloWorld introducido anteriormente. Entonces tenemos que actualizar el archivo de configuración de Apache por última vez:

```
# Needed for QGIS HelloServer plugin HTTP BASIC auth
<IfModule mod_fcgid.c>
    RewriteEngine on
    RewriteCond %{HTTP:Authorization} .
    RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
</IfModule>
```

A continuación, reinicie Apache:

```
systemctl restart apache2
```

4.3 Como usar un complemento

Pruebe el servidor con el complemento HelloWorld:

```
wget -q -O - "http://localhost/cgi-bin/qgis_mapserv.fcgi?SERVICE=HELLO"  
HelloServer!
```

Puede echar un vistazo a las GetCapabilities predeterminadas del servidor QGIS en:

```
http://localhost/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&  
↪REQUEST=GetCapabilities
```


5.1 Registro

Para registrar las solicitudes enviadas al servidor, debe configurar la siguiente variable de entorno:

- *QGIS_SERVER_LOG_STDERR*

Con las siguientes variables, el registro se puede personalizar aún más:

- *QGIS_SERVER_LOG_LEVEL*
- *QGIS_SERVER_LOG_PROFILE*

5.2 Variables de entorno

Puede configurar algunos aspectos de QGIS Server configurando ****** variables de entorno ******.

Según el servidor HTTP y cómo ejecute QGIS Server, hay varias formas de definir estas variables. Esto se describe completamente en *Servidor HTTP Apache*.

Nombre	Descripción	Predeterminada	Servicios
QGIS_OPTIONS_PATH	Especifica la ruta al directorio con la configuración. Funciona de la misma manera que la opción <code>--optionspath</code> de la aplicación QGIS. Está buscando un archivo de configuración en <code><QGIS_OPTIONS_PATH>/QGIS/QGIS3.ini</code> .	""	Todo
QGIS_PLUGINPATH	Útil si está utilizando complementos de Python para el servidor, esto establece la carpeta en la que se buscan complementos de Python.	""	Todo
QGIS_PROJECT_FILE	El archivo de proyecto <code>.qgs</code> o <code>.qgz</code> , normalmente pasado como un parámetro en la cadena de consulta (con <i>MAP</i>), también puede configurarlo como una variable de entorno (por ejemplo, usando el módulo Apache <code>mod_rewrite</code>). Tenga en cuenta que también puede indicar un proyecto almacenado en PostgreSQL, p. Ej. <code>postgresql://localhost:5432?sslmode=disable&dbname=mydb&schema=myschema&project=myproject</code> .	""	Todo
QGIS_SERVER_API_RESOURCES_DIRECTORY	Directorio para todos los recursos estáticos de la API OGC (como OAPIF / WFS3) (plantillas HTML, CSS, JS, ...)	dependiente del empaquetado	WFS
QGIS_SERVER_API_WFS3_MAXLIMIT	Máximo de límite en una solicitud de entidades.	10000	WFS
QGIS_SERVER_CACHE_DIRECTORY	Especifica el directorio de caché de red en el sistema de archivos.	cache en directorio de perfil	Todo
QGIS_SERVER_CACHE_SIZE	Establece el tamaño de la caché de la red en MB.	50 MB	Todo
QGIS_SERVER_DISABLE_GETPRINT	Permite una opción a nivel de proyecto para mejorar el tiempo de lectura del proyecto al deshabilitar la carga de diseños. La activación de esta opción desactiva la solicitud <code>GetPrint</code> de QGIS WMS. Configure esta bandera de proyecto QGIS para que no cargue diseños.	falso	WMS
QGIS_SERVER_IGNORE_BADLAYERS	Las capas «malas» son capas que no se pueden cargar. El comportamiento predeterminado de QGIS Server es considerar el proyecto como no disponible si contiene una capa incorrecta. El comportamiento predeterminado se puede anular estableciendo esta variable en <code>1</code> o verdadero. En este caso, las capas «malas» simplemente se ignorarán y el proyecto se considerará válido y disponible.	falso	Todo
QGIS_SERVER_LANDING_PAGE_PROJECTS_DIRECTORIES	Directorios utilizados por el servicio de página de destino para encontrar proyectos <code>.qgs</code> y <code>.qgz</code>	<code>/qgis/server/projects/directories</code>	Todo
QGIS_SERVER_LANDING_PAGE_PG_CONNECTIONS	Cadenas de conexión de PostgreSQL utilizadas por el servicio de página de destino para encontrar proyectos	<code>/qgis/server/projects/pg_connections</code>	Todo
QGIS_SERVER_LOG_FILE	Especifique la ruta y el nombre del archivo. Asegúrese de que el servidor tenga los permisos adecuados para escribir en el archivo. El archivo debe	""	Todo

5.3 Resumen de configuración

Cuando QGIS Server se está iniciando, tiene un resumen de todos los parámetros configurables gracias a las variables de entorno. Además, también se muestra el valor utilizado actualmente y el origen.

Por ejemplo con spawn-fcgi:

```
export QGIS_OPTIONS_PATH=/home/user/.local/share/QGIS/QGIS3/profiles/default/
export QGIS_SERVER_LOG_STDERR=1
export QGIS_SERVER_LOG_LEVEL=2
spawn-fcgi -f /usr/lib/cgi-bin/qgis_mapserv.fcgi -s /tmp/qgisserver.sock -U www-
↳data -G www-data -n

QGIS Server Settings:

- QGIS_OPTIONS_PATH / '' (Override the default path for user configuration): '/
↳home/user/.local/share/QGIS/QGIS3/profiles/default/' (read from ENVIRONMENT_
↳VARIABLE)

- QGIS_SERVER_PARALLEL_RENDERING / '/qgis/parallel_rendering' (Activate/
↳Deactivate parallel rendering for WMS getMap request): 'true' (read from INI_
↳FILE)

- QGIS_SERVER_MAX_THREADS / '/qgis/max_threads' (Number of threads to use when
↳parallel rendering is activated): '4' (read from INI_FILE)

- QGIS_SERVER_LOG_LEVEL / '' (Log level): '2' (read from ENVIRONMENT_VARIABLE)

- QGIS_SERVER_LOG_STDERR / '' (Activate/Deactivate logging to stderr): '1'
↳(read from ENVIRONMENT_VARIABLE)

- QGIS_PROJECT_FILE / '' (QGIS project file): '' (read from DEFAULT_VALUE)

- MAX_CACHE_LAYERS / '' (Specify the maximum number of cached layers): '100'
↳(read from DEFAULT_VALUE)

- QGIS_SERVER_CACHE_DIRECTORY / '/cache/directory' (Specify the cache
↳directory): '/root/.local/share/QGIS/QGIS3/profiles/default/cache' (read from
↳DEFAULT_VALUE)

- QGIS_SERVER_CACHE_SIZE / '/cache/size' (Specify the cache size): '52428800'
↳(read from INI_FILE)

Ini file used to initialize settings: /home/user/.local/share/QGIS/QGIS3/profiles/
↳default/QGIS/QGIS3.ini
```

En este caso particular, sabemos que **QGIS_SERVER_MAX_THREADS** y **QGIS_SERVER_PARALLEL_RENDERING** los valores se leen del archivo ini que se encuentra en **QGIS_OPTIONS_PATH** directorio (que se define a través de una variable de entorno). Las entradas correspondientes en el archivo ini son **/qgis/max_threads** y **/qgis/parallel_rendering** y sus valores son **true** y **4** hilos.

5.4 Nombre corto para capas, grupos y proyecto

Varios elementos tienen tanto un <Name> y un <Title>. El **Nombre** es una cadena de texto que se usa para la comunicación de máquina a máquina, mientras que el **Título** es para beneficio de los humanos.

Por ejemplo, un conjunto de datos podría tener el título descriptivo “Temperatura atmosférica máxima” y solicitarse utilizando el **Nombre** abreviado “ATMAX”. El usuario puede establecer el título de capas, grupos y proyectos.

El nombre de OWS se basa en el nombre utilizado en el árbol de capas. Este nombre es más una etiqueta para humanos que un nombre para la comunicación de máquina a máquina. Puede establecer un **Nombre corto** para capas, grupos o proyectos, para que QGIS Server lo utilice como nombre de identificación de capa (en parámetro por ejemplo *LAYERS*).

Puede establecer título, nombre corto y resumen para:

- **Layers:** haga clic derecho en una capa y elija *Propiedades...* -> *Servidor QGIS*-> *Descripción*.
- **Groups:** haga clic con el botón derecho en un grupo y seleccione *Establecer datos de WMS de grupo*

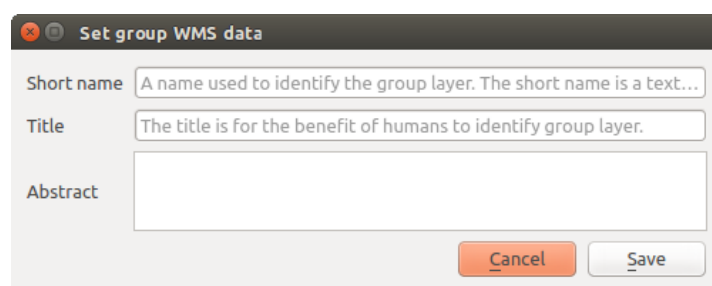


Figura 5.1: Cuadro de diálogo Establecer grupo de datos WMS

- **Project:** vaya a *Proyecto* ► *Propiedades...* ► *Servidor QGIS* ► *Capacidades de Servicio*.

5.5 Conexión a archivo servicio

Para que apache conozca el archivo de servicio PostgreSQL (consulte la sección :ref: pg-service-file), debe hacer que su archivo :file:`*.conf` tenga este aspecto:

```
SetEnv PGSERVICEFILE /home/web/.pg_service.conf

<Directory "/home/web/apps2/bin/">
  AllowOverride None
  ....
```

5.6 Agregar fuentes a su servidor linux

Tenga en cuenta que puede usar proyectos QGIS que apunten a fuentes que pueden no existir por defecto en otras máquinas. Esto significa que si comparte el proyecto, puede verse diferente en otras máquinas (si las fuentes no existen en la máquina de destino).

Para asegurarse de que esto no suceda, solo necesita instalar las fuentes que faltan en la máquina de destino. Hacer esto en sistemas de escritorio suele ser trivial (hacer doble clic en las fuentes).

Para Linux, si no tiene un entorno de escritorio instalado (o prefiere la línea de comandos), debe:

- En sistemas basados en Debian:

```
sudo su
mkdir -p /usr/local/share/fonts/truetype/myfonts && cd /usr/local/share/fonts/
↳truetype/myfonts

# copy the fonts from their location
cp /fonts_location/* .

chown root *
cd .. && fc-cache -f -v
```

- En sistemas basados en Debian:

```
sudo su
mkdir /usr/share/fonts/myfonts && cd /usr/share/fonts/myfonts

# copy the fonts from their location
cp /fonts_location/* .

chown root *
cd .. && fc-cache -f -v
```

Servidor de desarrollo

Una instalación de producción e implementación de QGIS Server generalmente implica la configuración de un componente de servidor web (por ejemplo, Apache o Nginx) que puede reenviar las solicitudes HTTP provenientes de los clientes a la aplicación binaria QGIS Server FastCGI.

Si desea probar rápidamente QGIS Server en su máquina local sin configurar e instalar una pila de servidor web completa, puede usar el servidor independiente de desarrollo de QGIS.

Esta es una aplicación independiente que proporciona un servidor web muy simple listo para servir los archivos de su proyecto.

Advertencia: El Servidor de Desarrollo Independiente no se ha desarrollado con el propósito de ser utilizado en producción, no se revisó para detectar vulnerabilidades de seguridad u otras condiciones de estrés que normalmente ocurrirán en un servidor expuesto públicamente.

Para lanzar el servidor:

```
$ qgis_mapserver
```

El puerto predeterminado que escucha el servidor de desarrollo es 8000. Salida de ejemplo:

```
QGIS Development Server listening on http://localhost:8000
CTRL+C to exit
127.0.0.1 [lun gen 20 15:16:41 2020] 5140 103ms "GET /wfs3/?MAP=/tests/testdata/
↪qgis_server/test_project.qgs HTTP/1.1" 200
127.0.0.1 [lun gen 20 15:16:41 2020] 3298 2ms "GET /wfs3/static/jsonFormatter.min.
↪js HTTP/1.1" 200
127.0.0.1 [lun gen 20 15:16:41 2020] 1678 3ms "GET /wfs3/static/jsonFormatter.min.
↪css HTTP/1.1" 200
127.0.0.1 [lun gen 20 15:16:41 2020] 1310 5ms "GET /wfs3/static/style.css HTTP/1.1
↪" 200
127.0.0.1 [lun gen 20 15:16:43 2020] 4285 13ms "GET /wfs3/collections?MAP=/tests/
↪testdata/qgis_server/test_project.qgs HTTP/1.1" 200
```

El servidor tiene algunas opciones que se pueden pasar como argumentos de línea de comando. Puede verlos todos invocando al servidor con `-h`.

```
Usage: qgis_mapserver [options] [address:port]
QGIS Development Server

Options:
-h, --help           Displays this help.
-v, --version        Displays version information.
-l <logLevel>       Sets log level (default: 0)
                    0: INFO
                    1: WARNING
                    2: CRITICAL
-p <projectPath>    Path to a QGIS project file (*.qgs or *.qgz),
                    if specified it will override the query string MAP argument
                    and the QGIS_PROJECT_FILE environment variable

Arguments:
addressAndPort       Listen to address and port (default: "localhost:8000")
                    address and port can also be specified with the environment
                    variables QGIS_SERVER_ADDRESS and QGIS_SERVER_PORT
```

Despliegue en contenedores

Hay muchas formas de utilizar aplicaciones en contenedores, desde las más simples (imágenes simples de Docker) hasta las más sofisticadas (Kubernetes, etc.).

Nota: Este tipo de implementación necesita que la *aplicación docker* <<https://www.docker.com>> _ esté instalada y en ejecución. Mira este [tutorial](#).

Consejo: Docker ejecuta una aplicación preempaquetada (también conocida como imágenes) que se pueden recuperar como fuentes (Dockerfile y recursos) para compilar o ya compiladas a partir de registros (privados o públicos).

Nota: Las descargas de paquetes QGIS Debian-Ubuntu necesitan una clave de autenticación gpg válida. Consulte las [páginas de instalación](#) para actualizar el siguiente Dockerfile con la última huella dactilar clave

7.1 Imágenes de simple docker

Como la imagen de la ventana acoplable no existe en un registro público, necesitarás construirlo. Para hacerlo, cree un directorio `qgis-server` y dentro de su directorio:

- cree un archivo `Dockerfile` con este contenido:

```
FROM debian:buster-slim

ENV LANG=en_EN.UTF-8

RUN apt-get update \
    && apt-get install --no-install-recommends --no-install-suggests --allow-
↪unauthenticated -y \
    gnupg \
    ca-certificates \
    wget \
```

(continué en la próxima página)

```

    locales \
    && localedef -i en_US -f UTF-8 en_US.UTF-8 \
    # Add the current key for package downloading - As the key changes every year
↳at least
    # Please refer to QGIS install documentation and replace it with the latest one
    && wget -O - https://qgis.org/downloads/qgis-2020.gpg.key | gpg --import \
    && gpg --export --armor F7E06F06199EF2F2 | apt-key add - \
    && echo "deb http://qgis.org/debian buster main" >> /etc/apt/sources.list.d/
↳qgis.list \
    && apt-get update \
    && apt-get install --no-install-recommends --no-install-suggests --allow-
↳unauthenticated -y \
    qgis-server \
    spawn-fcgi \
    xauth \
    xvfb \
    && apt-get remove --purge -y \
    gnupg \
    wget \
    && rm -rf /var/lib/apt/lists/*

RUN useradd -m qgis

ENV TINI_VERSION v0.17.0
ADD https://github.com/krallin/tini/releases/download/${TINI_VERSION}/tini /tini
RUN chmod +x /tini

ENV QGIS_PREFIX_PATH /usr
ENV QGIS_SERVER_LOG_STDERR 1
ENV QGIS_SERVER_LOG_LEVEL 2

COPY cmd.sh /home/qgis/cmd.sh
RUN chmod -R 777 /home/qgis/cmd.sh
RUN chown qgis:qgis /home/qgis/cmd.sh

USER qgis
WORKDIR /home/qgis

ENTRYPOINT ["/tini", "--"]

CMD ["/home/qgis/cmd.sh"]

```

- cree un archivo cmd.sh con este contenido:

```

#!/bin/bash

[[ $DEBUG == "1" ]] && env

exec /usr/bin/xvfb-run --auto-servernum --server-num=1 /usr/bin/spawn-fcgi -p 5555
↳-n -d /home/qgis -- /usr/lib/cgi-bin/qgis_mapserv.fcgi

```

- construya la imagen con:

```

docker build -f Dockerfile -t qgis-server ./

```

7.1.1 Primero ejecute

Para ejecutar el servidor, necesitará un archivo de proyecto QGIS. Puede usar uno de los suyos o elegir [esta muestra](#).

Para hacerlo, cree un directorio: archivo: *datos* dentro del directorio `qgis-server` y copie su archivo en él. Para cumplir con las siguientes explicaciones, cámbiele el nombre a `osm.qgs`.

Nota: Es posible que deba agregar las URL anunciadas en la pestaña *QGIS Server* de *Project ► Properties* si las *GetCapabilities* están rotas. Por ejemplo, si su servidor está expuesto en el puerto 8080, colocará esto para la URL anunciada `http://localhost:8080/qgis-server/`. Más información disponible en la sección *Configure su proyecto* y posteriores.

Ahora, puede ejecutar el servidor con:

```
docker network create qgis
docker run -d --rm --name qgis-server --net=qgis --hostname=qgis-server \
    -v $(pwd)/data:/data:ro -p 5555:5555 \
    -e "QGIS_PROJECT_FILE=/data/osm.qgs" \
    qgis-server
```

Opciones usadas:

- **-d:** ejecuta en segundo plano
- **--rm:** borra el contenedor cuando es detenido
- **--name:** nombre del contenedor a crear
- **--net:** (creada previamente) subred
- **--hostname:** nombre de host del contenedor, para referenciación posterior
- **-v:** directorio de datos local a montar en el contenedor
- **-p:** mapeo del puerto del host/contenedor
- **-e:** variable de entorno a usar en el contenedor

Para probar, escriba `docker ps | grep qgis-server` and you should see a line with **qgis-server**:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
↪ PORTS		NAMES		
4de8192da76e	qgis-server	"/tini -- /home/qgis..."	3 seconds ago	Up 2 seconds
↪ 0.0.0.0:5555->5555/tcp		qgis-server		

7.1.2 Muestra utilizable

Como el servidor solo acepta conexiones fastcgi, necesita un servidor HTTP que maneje este protocolo. Para hacerlo, tenemos que crear un archivo de configuración Nginx simple e iniciar una imagen Nginx.

Cree un archivo `nginx.conf` en el directorio actual con este contenido:

```
server {
    listen 80;
    server_name _;
    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
    }
    location /qgis-server {
        proxy_buffers 16 16k;
        proxy_buffer_size 16k;
    }
}
```

(continúe en la próxima página)

(proviene de la página anterior)

```
gzip off;
include fastcgi_params;
fastcgi_pass qgis-server:5555;
}
}
```

y escriba este comando:

```
docker run -d --rm --name nginx --net=qgis --hostname=nginx \
-v $(pwd)/nginx.conf:/etc/nginx/conf.d/default.conf:ro -p 8080:80 \
nginx:1.13
```

Para verificar la disponibilidad de capacidades, escriba un navegador <http://localhost:8080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

7.1.3 Limpieza

Para limpiar las imágenes cargadas, escriba:

```
docker stop qgis-server nginx
```

7.2 Pilas de Docker

El método anterior es programable, pero no es fácil de empaquetar, estandarizar o administrar.

Para trabajar con un conjunto de imágenes de la ventana acoplable, puede usar una pila de la ventana acoplable administrada por un orquestador. En una pila, las imágenes funcionan en la misma red privada y puede iniciar / detener toda la pila o implementar la pila en otros trabajadores. Hay muchos orquestadores, por ejemplo Swarm, Kubernetes y Mesos.

A continuación, presentaremos configuraciones simples con fines de prueba. No son aptos para la producción.

7.2.1 Swarm/docker-compose

Docker ahora tiene su propio orquestador: Swarm (compatible con archivos de composición de Docker). Tienes que “habilitarlo <https://docs.docker.com/get-started/orchestration/#enable-docker-swarm>” (la versión para Mac también funcionará con Linux).

Descripción de pila

Ahora que tiene Swarm funcionando, cree el archivo de servicio (consulte [Implementar en Swarm](https://docs.docker.com/get-started/swarm-deploy/)) `qgis-stack.yaml`:

```
version: '3.7'

services:
  qgis-server:
    # Should use version with utf-8 locale support:
    image: qgis-server:latest
    volumes:
      - REPLACE_WITH_FULL_PATH/data:/data:ro
    environment:
      - LANG=en_EN.UTF-8
      - QGIS_PROJECT_FILE=/data/osm.qgs
```

(continúe en la próxima página)

(proviene de la página anterior)

```

- QGIS_SERVER_LOG_LEVEL=0 # INFO (log all requests)
- DEBUG=1 # display env before spawning QGIS Server

nginx:
  image: nginx:1.13
  ports:
  - 8080:80
  volumes:
  - REPLACE_WITH_FULL_PATH/nginx.conf:/etc/nginx/conf.d/default.conf:ro
  depends_on:
  - qgis-server
    
```

Para implementar (o actualizar) la pila, escriba:

```
docker stack deploy -c qgis-stack.yaml qgis-stack
```

Verifique el estado de implementación de la pila hasta que obtenga **1/1** en la columna **réplicas**:

```
docker stack services qgis-stack
```

Algo como:

ID	NAME	MODE	REPLICAS	
↔ IMAGE	PORTS			┌
gmx7ewlvwsqt	qgis_nginx	replicated	1/1	┌
↔ nginx:1.13	*:8080->80/tcp			
l0v2e7cl43u3	qgis_qgis-server	replicated	1/1	┌
↔ qgis-server:latest				

Para verificar las capacidades de WMS, escriba un navegador web <http://localhost:8080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

Limpieza

Para limpiar, escriba:

```
docker stack rm qgis-stack
```

7.2.2 Kubernetes

Instalación

Si tiene una instalación de **Docker Desktop**, usar Kubernetes (también conocido como k8s) es bastante sencillo: [habilitar k8s](#).

Si no, siga el [tutorial de minikube](#) o [microk8s para Ubuntu](#).

Como la instalación de Kubernetes puede ser realmente compleja, solo nos centraremos en los aspectos utilizados por esta demostración. Para obtener más información / más detallada, consulte la [documentación oficial](#).

microk8s

microk8s necesita pasos adicionales: debe habilitar el registro y etiquetar la imagen del servidor qgis para que Kubernetes encuentre las imágenes creadas.

Primero, habilite el registro:

```
microk8s enable dashboard dns registry
```

Luego, etiquete y envíe la imagen a su registro recién creado:

```
docker tag qgis-server 127.0.0.1:32000/qgis-server && docker push 127.0.0.1:32000/  
↪qgis-server
```

Finalmente, agregue o complete el `/etc/docker/daemon.json` para tener su registro**127.0.0.1:32000** enumerados en el campo **registros inseguros**:

```
{  
  "insecure-registries": ["127.0.0.1:32000"]  
}
```

Creando manifiestos

Kubernetes describe los objetos para implementar en manifiestos yaml. Hay muchos tipos diferentes, pero solo usaremos implementaciones (manejar pods, es decir, imágenes de Docker) y servicios para exponer las implementaciones a propósitos internos o externos.

Manifiestos de implementación

Cree un archivo `deployments.yaml` con este contenido:

```
apiVersion: apps/v1  
kind: Deployment  
metadata:  
  name: qgis-server  
  namespace: default  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      myLabel: qgis-server  
  template:  
    metadata:  
      labels:  
        myLabel: qgis-server  
    spec:  
      containers:  
        - name: qgis-server  
          image: localhost:32000/qgis-server:latest  
          imagePullPolicy: IfNotPresent  
          env:  
            - name: LANG  
              value: en_EN.UTF-8  
            - name: QGIS_PROJECT_FILE  
              value: /data/osm.qgs  
            - name: QGIS_SERVER_LOG_LEVEL  
              value: "0"  
            - name: DEBUG  
              value: "1"
```

(continué en la próxima página)

(proviene de la página anterior)

```

    ports:
      - containerPort: 5555
    volumeMounts:
      - name: qgis-data
        mountPath: /data/
  volumes:
    - name: qgis-data
      hostPath:
        path: REPLACE_WITH_FULL_PATH/data
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: qgis-nginx
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      myLabel: qgis-nginx
  template:
    metadata:
      labels:
        myLabel: qgis-nginx
    spec:
      containers:
        - name: qgis-nginx
          image: nginx:1.13
          ports:
            - containerPort: 80
          volumeMounts:
            - name: nginx-conf
              mountPath: /etc/nginx/conf.d/default.conf
      volumes:
        - name: nginx-conf
          hostPath:
            path: REPLACE_WITH_FULL_PATH/nginx.conf

```

Manifiestos de servicio

Cree un archivo `services.yaml` con este contenido:

```

apiVersion: v1
kind: Service
metadata:
  name: qgis-server
  namespace: default
spec:
  type: ClusterIP
  selector:
    myLabel: qgis-server
  ports:
    - port: 5555
      targetPort: 5555
---
apiVersion: v1
kind: Service
metadata:

```

(continué en la próxima página)

(proviene de la página anterior)

```

name: qgis-nginx
namespace: default
spec:
  type: NodePort
  selector:
    myLabel: qgis-nginx
  ports:
    - port: 80
      targetPort: 80
      nodePort: 30080
    
```

Implementar manifiesto

Para implementar las imágenes y los servicios en Kubernetes, se puede usar el panel (haga clic en + en la parte superior derecha) o la línea de comandos.

Nota: Cuando use la línea de comando con *microk8s*, tendrá que anteponer cada comando con *microk8s*.

Para implementar o actualizar sus manifiestos:

```
kubectl apply -k ./
```

Para verificar qué está implementado actualmente:

```
kubectl get pods,services,deployment
```

Debería obtener algo como:

NAME	READY	STATUS	RESTARTS	AGE
pod/qgis-nginx-54845ff6f6-8skp9	1/1	Running	0	27m
pod/qgis-server-75df8ddd89-c7t7s	1/1	Running	0	27m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
↔ AGE				
service/Kubernetes	ClusterIP	10.152.183.1	<none>	443/TCP
↔ 5h51m				
service/qgis-exec-server	ClusterIP	10.152.183.218	<none>	5555/TCP
↔ 35m				
service/qgis-nginx	NodePort	10.152.183.234	<none>	80:30080/TCP
↔ 27m				
service/qgis-server	ClusterIP	10.152.183.132	<none>	5555/TCP
↔ 27m				

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/qgis-nginx	1/1	1	1	27m
deployment.apps/qgis-server	1/1	1	1	27m

Para leer los registros de nginx / qgis, escriba:

```
kubectl logs -f POD_NAME
```

Para verificar las capacidades de WMS, escriba un navegador web <http://localhost:30080/qgis-server/?SERVICE=WMS&VERSION=1.3.0&REQUEST=GetCapabilities>

Limpieza

Para limpiar, escriba:

```
kubectl delete -n default service/qgis-server service/qgis-nginx deployment/qgis-  
→nginx deployment/qgis-server
```

7.3 Implementar nube

Administrar su propio clúster de servidores para manejar la implementación de aplicaciones en contenedores es un trabajo complejo. Tiene que manejar varios problemas, como hardware, ancho de banda y seguridad en diferentes niveles.

Las soluciones de implementación en la nube pueden ser una buena alternativa cuando no desea centrarse en la gestión de la infraestructura.

Una implementación en la nube puede utilizar mecanismos propietarios, pero también son compatibles con las etapas explicadas anteriormente. (*docker images* y *stack management*).

7.3.1 Caso de uso de AWS

Con Amazon AWS, a través de 'ECS (Elastic Container Service)'<<https://console.aws.amazon.com/ecs/home>>`_ funcionalidades, puede usar contenedores compatibles con Docker-compose o Kubernetes para administrar su pila. Tendrá que crear un registro de imágenes <<https://console.aws.amazon.com/ecr/home>>`_ para que sus imágenes personalizadas sean accesibles.

Para usar funcionalidades similares de docker-compose, necesita instalar el cliente **ecs-cli** y tener los permisos/roles adecuados <https://docs.aws.amazon.com/AmazonECS/latest/developerguide/task_execution_IAM_role.html>`_. Luego, con la ayuda de los comandos 'ecs-cli compose` (ver el manual 'ecs-cli compose <<https://docs.aws.amazon.com/AmazonECS/latest/developerguide/cmd-ecs-cli-compose.html>>`_ y ecs-cli tutorial), puedes reutilizar el *stack description*.

Para usar Kubernetes, puede usar la consola web de AWS o la herramienta de línea de comandos eksctl y tener los permisos / roles adecuados. Luego, con un entorno kubectl bien configurado, puede reusar la *Kubernetes manifests*.

FAQ Preguntas frecuentes

- ¿Cuál es la diferencia entre QGIS Desktop y QGIS Server?

El Escritorio de QGIS tiene una interfaz gráfica de usuario y le permite crear y modificar mapas. El Servidor QGIS es una aplicación servidor que sirve sus archivos de proyectos QGIS a usuarios finales de aplicaciones por medio de servicios web OGC como [WMS](#), [WFS](#), etc..

- ¿Qué es OGC?

El [OGC \(Open Geospatial Consortium\)](#) es una organización internacional sin ánimo de lucro is an international not for profit organization comprometida en hacer estándares abiertos de calidad para la comunidad geoespacial global.

- ¿Nombre de algunos otros servidores de mapas web?

ArcGIS server, Geoserver, Mapserver, Mapnik etc.

- ¿Cómo comparar QGIS Server con otros servidores de mapas web? (2021/01/01)

Objetos espaciales	QGIS Server	GeoServer	ArcGIS Server
Desde	2006	2001	1999
Licencia	GPL	GPL	comercial
Soporte comercial	Multiples empresas	Multiples empresas	ESRI y su red de vendedores
Tecnología	C++/python	Java	C++
cache de Tesela	si	si (via GeoWebCache)	si
3D	No	No	Sí
Consultando	Filtros FES (2.0) y OGC (1.0)	Filtros CQL y OGC	filtros OGC
Generación de informe	si	si	si
Administración del Servidor	si vía terceros (LizMap, QWC2, etc.)	web + API REST	web + API REST
Capa del proyecto SIG/edición de simbología	Completa a través de la GUI dedicada	simple a través de la interfaz web	Completa a través de la GUI dedicada

- ¿Cuáles son las versiones de la especificación OGC implementadas en el servidor QGIS en comparación con otros servidores de mapas web?? (2021/01/01)

Estándares OGC	QGIS Server	GeoServer	ArcGIS Server
WMS (Web Map Service)	1.3.0 - 1.1.1	1.3.0 - 1.1.1	1.3.0 - 1.1.1
WFS (Web Feature Service)	1.1.0 - 1.0.0	2.0.0 - 1.0.0	2.0.0 - 1.0.0
OAPIF (aka WFS3)	1.0.0	no	no
WMTS (Web Map Tile Service)	1.0.0	1.0.0	1.0.0
WCS (Web Coverage Service)	1.0.0	2.0.1 - 1.0.0	2.0.1 - 1.0.0
WPS (Web Processing Service)	no	1.0.0	1.0.0
CSW (Catalogue Service for the Web)	no	2.0.2	no
SLD (Styled Layer Descriptor)	si	si	si

- *¿Que es un cache de tesela?*

Los mapas suelen ser estáticos. Como la mayoría de los clientes de mapeo procesan datos de WMS (Web Map Service) cada vez que son consultados, esto puede resultar en un procesamiento innecesario y mayores tiempos de espera.

La caché de teselas optimiza esta experiencia al guardar (almacenar en caché) imágenes de mapas, o teselas, a medida que se solicitan, actuando de hecho como un proxy entre el cliente (como OpenLayers o Google Maps) y el servidor (cualquier servidor compatible con WMS). A medida que se solicitan nuevos mapas y mosaicos, el servidor QGIS intercepta estas llamadas y devuelve mosaicos pre-renderizados si están almacenados, o llama al motor QGIS para generar nuevos mosaicos según sea necesario. Por lo tanto, una vez que se almacenan los mosaicos, la velocidad de representación del mapa aumenta muchas veces, creando una experiencia de usuario mucho mejor.

- *¿Qué es PostgreSQL?*

PostgreSQL <<https://www.postgresql.org/>> es una poderosa, base de datos relacional de objetos de código abierto para QGIS.

- *¿Qué es PostGIS?*

PostGIS es un extensor de base de datos espacial para la base de datos relacional de objetos PostgreSQL. Agrega soporte para objetos geográficos permitiendo que las consultas de ubicación se ejecuten en SQL.

- Continuará...