



## MANUAL DE DESARROLLADOR: QGIS

EL MANUAL DE DESARROLLADOR DEL SISTEMA DE INFORMACION GEOGRAFICO EN QGIS 3 PERMITE REALIZAR NUEVOS REPORTES Y CONFIGURAR EL ENTORNO DE TRABAJO PARA SU APLICACIÓN EN EL SISTEMA DE GESTION DE AGUA POTABLE

## Tabla de contenido

1.	CONCEPTOS	3
1.1.	¿Qué es pygis?	3
1.2.	¿Qué podemos hacer con pyqgis?	3
1.3.	Variable de entorno PYQGIS_STARTUP	3
1.4.	El archivo startup.py	3
1.5.	Ejecución de código Python en el lanzamiento de QGIS	4
2.	CREAR PROYECTO PERSONALIZADO EN QGIS 3	5
2.1.	Descargar IDLE Python	5
2.2.	Sistema de gestión de paquetes (Pip)	5
2.3.	Descargar e instalar Qgis 3.10 LTR (long term release)	6
2.4.	Importar librerías	6
2.5.	Inicializar la GUI de Qgis	7
2.6.	Ejecución del .exe	7
2.7.	RUTEO	8
2.8.	EL módulo botón	9
2.9.	Menú	10
2.10.	Submenus	10
2.11.	Funcionalidad de los submenús	11
3.	ENLACE SQL SERVER CON QGIS	12
3.1.	Conexión MSSQL y QGIS	12
3.2.	Editar propiedades de la capa	14
4.	PROYECCIÓN DEL PROYECTO PERSONALIZADO	15
5.	INFORMACION DEL CENTRO	17

Financiado por:



# 1. CONCEPTOS

## 1.1. ¿QUÉ ES PYGIS?

PyQGIS es la librería de QGIS para ejecutar código Python.

QGIS tiene soporte para ejecutar scripts utilizando el lenguaje Python.

QGIS elige Python por ser uno de los lenguajes favoritos para scripting. Python es un lenguaje de programación para GIS orientado a objetos.

## 1.2. ¿QUÉ PODEMOS HACER CON PYQGIS?

- Podemos automatizar tareas GIS. Listar el número de registros de las capas, realizar geoprocetamiento, añadir y borrar elementos de una capa o tabla, crear geometrías nuevas o exportar un mapa a PDF.
- Crear complementos de QGIS (plugins) utilizando Python como lenguaje de programación. La principal ventaja de utilizar Python es la simplicidad en la distribución y un desarrollo sencillo.
- Ejecutar aplicaciones personalizadas. Con PyQGIS podemos crear aplicaciones interactivas con funcionalidades GIS. El módulo qgis.gui trae diversos componentes visuales GUI.
- Ejecutar código Python cuando QGIS inicie .

## 1.3. VARIABLE DE ENTORNO PYQGIS\_STARTUP

Puede ejecutar el código Python justo antes de que se complete la inicialización de QGIS configurando STARTUP variable de entorno en la ruta de un archivo Python existente.

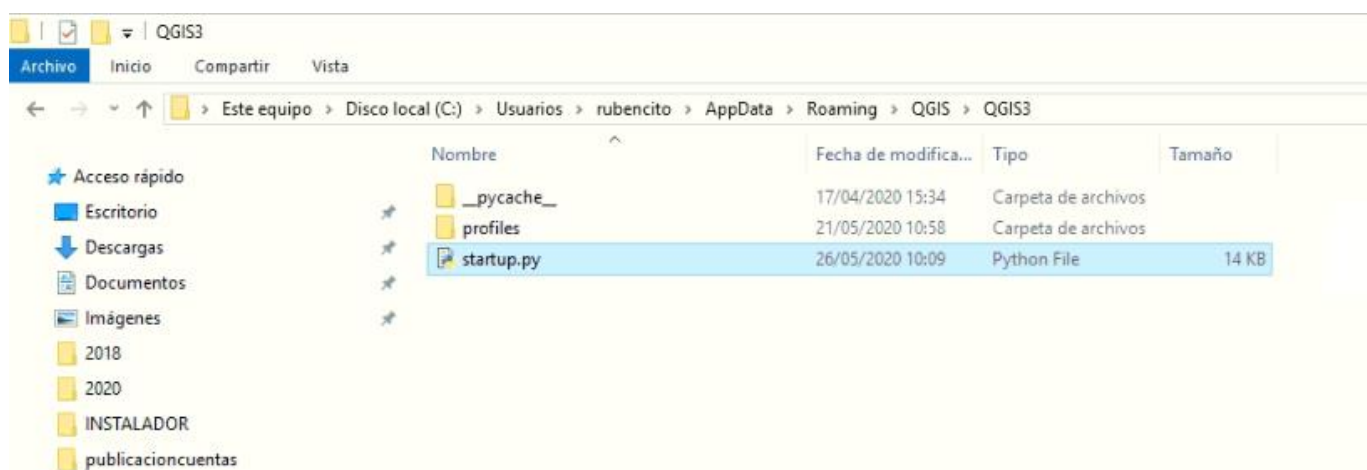
Es una de las varias formas de ejecutar código Python dentro de QGIS, porque este código se ejecutará antes de que se complete la inicialización de QGIS.

## 1.4. EL ARCHIVO STARTUP.PY

Cada vez que se inicia QGIS, .qgis2/pythonse busca en el directorio de inicio de Python del usuario, un archivo llamado startup.py, si ese archivo existe, el intérprete de Python incorporado lo ejecuta.

Este archivo se debe crear en la dirección de la carpeta de Usuarios que pertenece a QGIS.

Este archivo Python, lleva los módulos para el funcionamiento personalizado del sistema de información geográfico.



### 1.5. EJECUCIÓN DE CÓDIGO PYTHON EN EL LANZAMIENTO DE QGIS

Es posible ejecutar un código PyQGIS cada vez que se inicia QGIS. QGIS busca un archivo nombrado `startup.py` en el directorio de inicio de Python del usuario y, si lo encuentra, lo ejecuta. Este archivo es muy útil para personalizar la interfaz QGIS.

Si no existe el archivo, se deberá crear (`startup.py`), con el código que se presentará más adelante. Tener en cuenta las importaciones en la parte superior (librerías, módulos).

Cuando ejecutamos los fragmentos de código en la Consola Python, no se tuvo que importar ningún módulo, ya que se realizan automáticamente cuando se inicia la consola. Para los scripts `pyqgis` en otros lugares, como en nuestro proyecto personalizado, tenemos que importar explícitamente los módulos (clases) que queremos usar.

## 2. CREAR PROYECTO PERSONALIZADO EN QGIS 3

### 2.1. DESCARGAR IDLE PYTHON

Es un entorno gráfico de desarrollo elemental que permite editar y ejecutar programas en Python. IDLE es también un entorno interactivo en el que se pueden ejecutar instrucciones sueltas de Python.

- Ubicar la ubicación de la instalación de la IDLE Python\Scripts.



### 2.2. SISTEMA DE GESTIÓN DE PAQUETES (PIP)

Es utilizado para instalar y administrar paquetes de software escritos en Python. Muchos paquetes pueden ser encontrados en el Python Package Index (PyPI), este ya viene instalado junto a la versión 3 de Python.

- pip install PYQT5 (modulo para modificar la ui de Python)
- python -m webbrowser -t <http://www.python.org> (modulo para conectar Qgis con la web, podemos subir el proyecto personalizado a la nube)
- pip install os-sys (módulo para interactuar con aplicaciones externas desde python)
- pip install pysal

```
c:\Users\ [redacted] > cd c:\Users\ [redacted] \AppData\Local\Programs\Python\Python36\Scripts
c:\Users\ [redacted] \AppData\Local\Programs\Python\Python36\Scripts> pip install PyQT5
Collecting PyQT5
  Downloading PyQt5-5.8.1.1-5.8.0-cp35.cp36.cp37-none-win_amd64.whl (75.2MB)
    100% | [redacted] | 75.2MB 18kB/s
Requirement already satisfied: sip<4.20,>=4.19 in c:\users\ [redacted] \appdata\local\programs\python\python36\lib\site-packages (from PyQT5)
Installing collected packages: PyQT5
Successfully installed PyQT5-5.8.1
c:\Users\ [redacted] \AppData\Local\Programs\Python\Python36\Scripts>
```

## 2.3. DESCARGAR E INSTALAR QGIS 3.10 LTR (LONG TERM RELEASE)

Este es el lanzamiento a largo plazo, ya que su actualización es cada 3 años, esto nos ayudará a tener soporte constante, sin un cambio que pueda afectar el funcionamiento de nuestro módulo en Python, ya que depende mucho que coincida la versión de Python y con la de Qgis.



3.12.3  
3.10.6 LTR

DESCUBRE QGIS PARA USUARIOS HAZ PARTE DOCUMENTACIÓN

Buscar

Para obtener la Versión con Soporte a Largo Plazo (que no es el ultimo lanzamiento) seleccione **Instalación Avanzada** y seleccione **qgis-ltr-full**  
Para obtener la última versión inestable en desarrollo seleccione **Instalación Avanzada** y seleccione **qgis-dev-full**

Instaladores independientes de los paquetes de OSGeo4W

Versión mas reciente (mas rica en características):

	Instalador autónomo de QGIS Version 3.12 (64 bit)	sha256
	Instalador autónomo de QGIS Version 3.12 (32 bit)	sha256

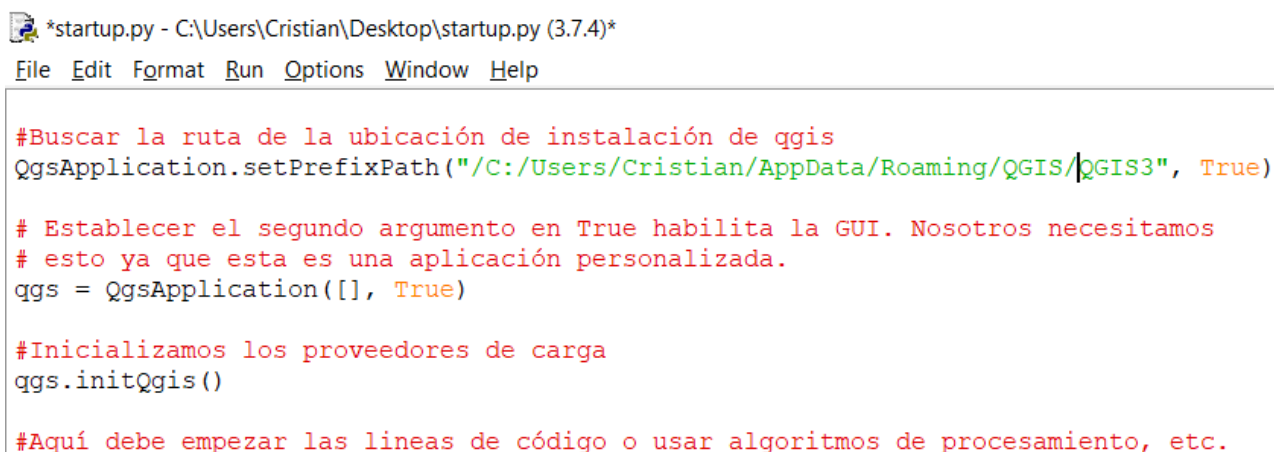
## 2.4. IMPORTAR LIBRERÍAS

Al instalar la IDLE de Python, esta trae consigo una serie de librerías listas para ser usadas y las que no están las descargaremos mediante el sistema de gestión de paquetes (pip). De esta manera le decimos al interprete que busque el módulo deseado para poder hacer uso de sus funciones y variables.

```
*startup.py - C:\Users\Cristian\Desktop\startup.py (3.7.4)*
File Edit Format Run Options Window Help
from PyQt5 import QtGui
from qgis.utils import iface
from PyQt5 import QtGui
from PyQt5.QtWidgets import QMenu
from PyQt5.QtWidgets import QAction
from PyQt5.QtGui import *
from PyQt5.QtCore import *
from qgis.core import Qgs
from qgis.core import *
from PyQt5.QtWidgets import QMessageBox
from qgis.core import QgsProject
import os
from PyQt5.QtWidgets import QMessageBox
from qgis.core import QgsProject
from datetime import datetime
import webbrowser
from qgis.PyQt.QtGui import QIcon
import os
import subprocess
import sys, string, os
```

## 2.5. INICIALIZAR LA GUI DE QGIS

Estas 3 líneas de código inicializan la GUI de Qgis, nos permitirá modificar la interfaz gráfica y poder crear nuestro proyecto personalizado.



```
*startup.py - C:\Users\Cristian\Desktop\startup.py (3.7.4)*
File Edit Format Run Options Window Help

#Buscar la ruta de la ubicación de instalación de qgis
QgsApplication.setPrefixPath("/C:/Users/Cristian/AppData/Roaming/QGIS/QGIS3", True)

# Establecer el segundo argumento en True habilita la GUI. Nosotros necesitamos
# esto ya que esta es una aplicación personalizada.
qgs = QgsApplication([], True)

#Inicializamos los proveedores de carga
qgs.initQgis()

#Aquí debe empezar las líneas de código o usar algoritmos de procesamiento, etc.
```

Primero se importa el módulo qgis. core y configuramos el prefijo de ruta. El prefijo de ruta es la ubicación donde QGIS está instalado en el sistema (en la imagen anterior podemos ver la dirección donde está instalado QGIS). Se configura en el script llamando al método setPrefixPath. El segundo argumento de setPrefixPath se establece a True, especificando las rutas predeterminadas que se deben utilizar, finalmente inicializamos todos los proveedores de carga para poder modificar la UI de QGIS.

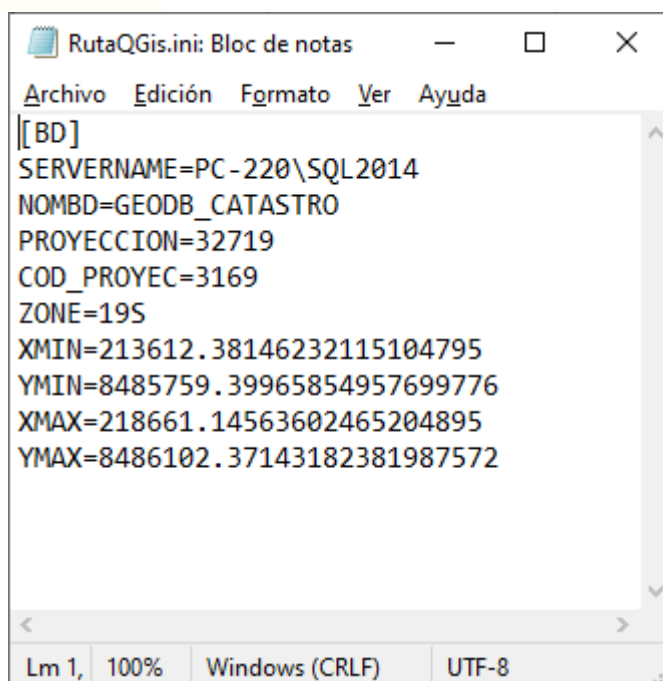
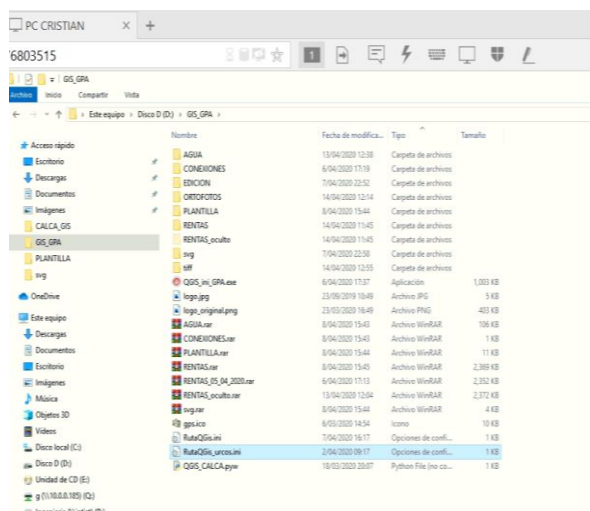
## 2.6. EJECUCIÓN DEL .EXE

Usando la librería os, ubica el ejecutable que modifica el proyecto con la data de la municipalidad, hace los cambios de SRC, EPSG, coordenadas, etc.

```
#Este mensaje se ejecuta al iniciar el QGIS, para evidenciar la ejecución del startup
#iface.messageBar().pushMessage("ADVERTENCIA", "La informacion que contiene este proyecto es confidencial", duration=0)
#QMessageBox.information(iface.mainWindow(), "AVISO", 'La informacion que contiene este proyecto es confidencial ' )
icon = QIcon(r"C:/Users/rubencito/Desktop/GIS_CALCA/file.png")
iface.mainWindow().setWindowIcon(icon)
iface.mainWindow().setWindowTitle("GUAMAN POMA DE AYALA")
#subprocess.Popen("D:\GIS_GPA\QGIS_ini_GPA.exe", shell=True)
#subprocess.Popen(['D:\GIS_GPA\QGIS_ini_GPA.exe', 'option'])
#subprocess.call(['D:\GIS_GPA\QGIS_ini_GPA.exe'], shell = True)
#subprocess.call(['D:/GIS_GPA/QGIS_ini_GPA.exe'])
#os.startfile("D:\GIS_GPA\QGIS_ini_GPA.exe")
#subprocess.Popen(['D:/GIS_GPA/QGIS_ini_GPA.exe', 'https://es.stackoverflow.com/', '-new-tab'])
#os.system(r"D:/GIS_GPA/QGIS_ini_GPA.exe")
#Ejecucion del .exe para inicializar los cambios de acuerdo a la municipalidad
os.chdir('D:\\GIS_GPA')
os.system('D:\GIS_GPA\QGIS_ini_GPA.exe')
```

## 2.7. RUTEO

En la carpeta <<D:\GIS GPA>> ubique el archivo RutaQGis.ini el cual deberá contener la ruta de acceso a la base de datos alfanumérica, que es de donde se extraen los datos.



Este archivo de texto menciona la configuración de la municipalidad seleccionada, la conexión a la base de datos de enlace, proyección donde se encuentra la localidad, código de proyección, la zona en la que esta georreferenciada y puntos de referencia en la que se ubica la localidad.



## 2.8. EL MÓDULO BOTÓN

Identifica los objetos espaciales de acuerdo con nuestro proyecto personalizado, en el manual de usuario evidencia la funcionalidad de este botón.

```

|
#Modulo para el boton de informacion en cualquier instante que el usuario haga la
#manipulacion del sistema de informacion
#se crearon las acciones para las capas de los proyectos de rentas y gestion agua
def boton():
    layer=None
    for lyr in QgsProject.instance().mapLayers().values():
        #Para RENTAS
        if lyr.name() == "Predios sin registro":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios sin Construcccion":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios por Pisos (Individual)":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios por pisos (intervalo)":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios por Estado de Conservacion":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios Por Clasificacion":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Contribuyentes exonerados":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Contribuyentes exonerados":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Contribuyentes con deudas":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        #Para Gestion_Agua
        if lyr.name() == "Tipo de Conexion":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
        if lyr.name() == "Contribuyentes por deuda":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios por el tipo de tarifa":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Contribuyentes con deuda":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Predios Conexion":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(),duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "Situacion Actual":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(), duration=10)
            iface.actionIdentify().trigger()
            break
        if lyr.name() == "INCIDENCIAS":
            layer = iface.setActiveLayer(lyr)
            iface.messageBar().pushMessage("", 'La capa activa es ' + lyr.name(), duration=10)
            iface.actionIdentify().trigger()
            break

```


## 2.9. MENÚ

Usando la librería QMenu, para darle funcionalidad a nuestro menú, este se ubicará en la barra principal.

```
#Menu rentas en la barra principal
menu = QMenu( "RENTAS", iface.mainWindow().menuBar() )
actions = iface.mainWindow().menuBar().actions()
lastAction = actions[-1]
iface.mainWindow().menuBar().insertMenu( lastAction,menu)
```

## 2.10. SUBMENUS

Dependen de la librería QgsProject previamente importada, se modulariza los lanzamientos de cada botón, más adelante estos módulos serán llamados para cumplir su ejecución.

 \*startup.py - C:\Users\Cristian\Desktop\startup.py (3.7.4)\*

File Edit Format Run Options Window Help

```
#MENU PARA ABRIR PROYECTOS DE RENTAS 1-9
def abrir1():
    project1 = QgsProject.instance()
    project1.read('D://GIS_GPA//RENTAS//PREDIOS_REGISTRADOS.qgs')

def abrir2():
    project2 = QgsProject.instance()
    project2.read('D://GIS_GPA//RENTAS//PREDIOS_SINCONSTRUCCION.qgs')

def abrir3():
    project3 = QgsProject.instance()
    project3.read('D://GIS_GPA//RENTAS//PREDIOS_POR_PISO.qgs')

def abrir4():
    project4 = QgsProject.instance()
    project4.read('D://GIS_GPA//RENTAS//PREDIOS_POR_PISO_PORINTERVALOS.qgs')

def abrir5():
    project5 = QgsProject.instance()
    project5.read('D://GIS_GPA//RENTAS//PREDIOS_POR_ESTADO_CONSERVACION.qgs')

def abrir6():
    project6 = QgsProject.instance()
    project6.read('D://GIS_GPA//RENTAS//DIFERENCIA_AREA.qgs')

def abrir7():
    project7 = QgsProject.instance()
    project7.read('D://GIS_GPA//RENTAS//PREDIO_CLASIFICACION.qgs')

def abrir8():
    project8 = QgsProject.instance()
    project8.read('D://GIS_GPA//RENTAS//PREDIOS_EXONERADOS.qgs')

def abrir9():
    project9 = QgsProject.instance()
    project9.read('D://GIS_GPA//RENTAS//CONTRIBUYENTES_CON_DEUDA.qgs')

def abrir10():
    project10 = QgsProject.instance()
    project10.read('D://GIS_GPA//RENTAS//INCIDENCIAS_RENTAS.qgs')
```

## 2.11. FUNCIONALIDAD DE LOS SUBMENÚS

Depende de la librería QAction, la ui de Qgis manda una señal a los módulos previamente mencionados.

```
#Sub menus del menu RENTAS
actionSubMenu1 = QAction('PREDIOS SIN REGISTRAR', iface.mainWindow())
menu.addAction(actionSubMenu1)
actionSubMenu1.triggered.connect(abrir1)

actionSubMenu2 = QAction('PREDIOS SIN CONSTRUCCION', iface.mainWindow())
menu.addAction(actionSubMenu2)
actionSubMenu2.triggered.connect(abrir2)

actionSubMenu3 = QAction('PREDIOS POR PISO(INDIVIDUAL)', iface.mainWindow())
menu.addAction(actionSubMenu3)
actionSubMenu3.triggered.connect(abrir3)

actionSubMenu4 = QAction('PREDIOS POR PISO(INTERVALO)', iface.mainWindow())
menu.addAction(actionSubMenu4)
actionSubMenu4.triggered.connect(abrir4)

actionSubMenu5 = QAction('PREDIOS POR ESTADO DE CONSERVACION', iface.mainWindow())
menu.addAction(actionSubMenu5)
actionSubMenu5.triggered.connect(abrir5)

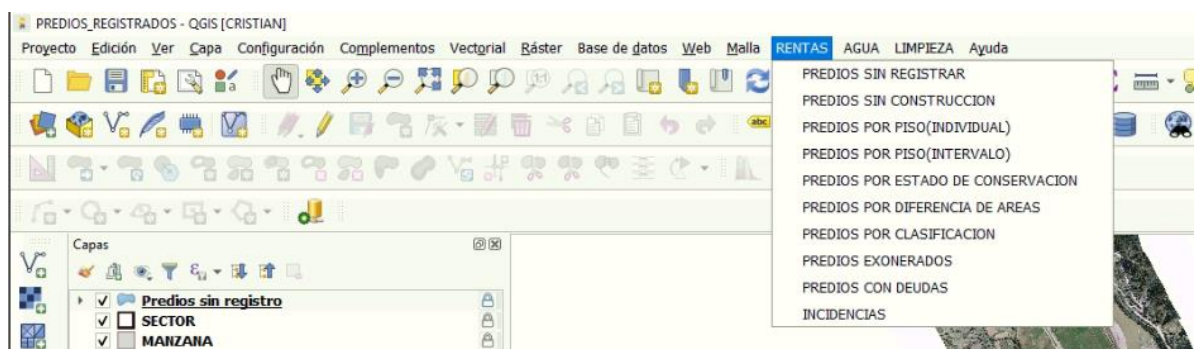
actionSubMenu6 = QAction('PREDIOS POR DIFERENCIA DE AREAS', iface.mainWindow())
menu.addAction(actionSubMenu6)
actionSubMenu6.triggered.connect(abrir6)

actionSubMenu7 = QAction('PREDIOS POR CLASIFICACION', iface.mainWindow())
menu.addAction(actionSubMenu7)
actionSubMenu7.triggered.connect(abrir7)

actionSubMenu8 = QAction('PREDIOS EXONERADOS', iface.mainWindow())
menu.addAction(actionSubMenu8)
actionSubMenu8.triggered.connect(abrir8)

actionSubMenu9 = QAction('PREDIOS CON DEUDAS', iface.mainWindow())
menu.addAction(actionSubMenu9)
actionSubMenu9.triggered.connect(abrir9)
```

Finalmente tenemos el producto, los 3 menus con sus funcionalidades cada uno:

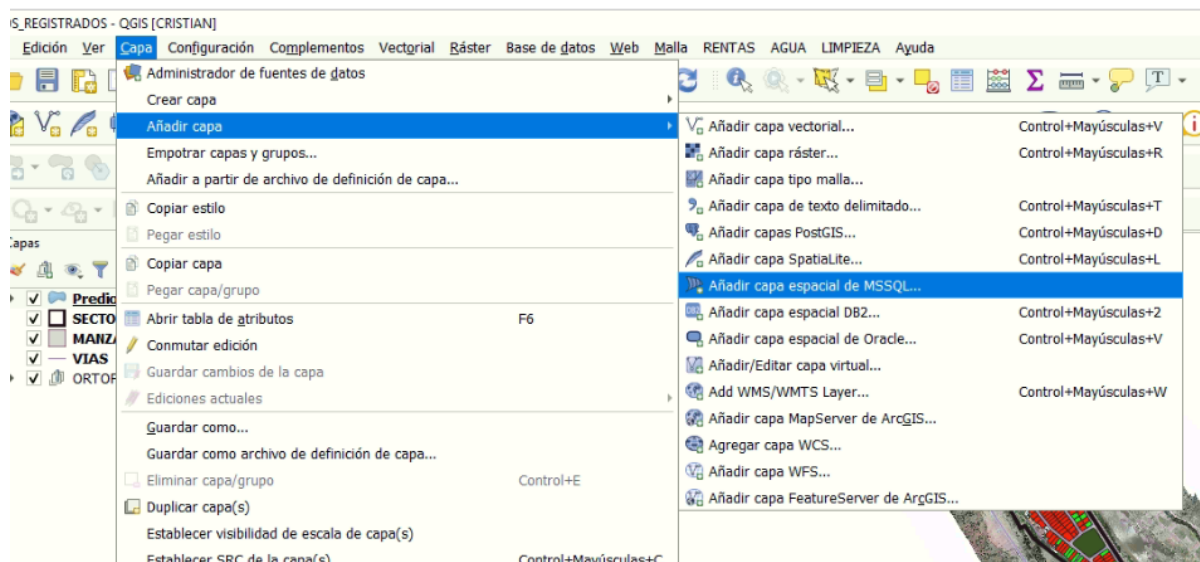


### 3. ENLACE SQL SERVER CON QGIS

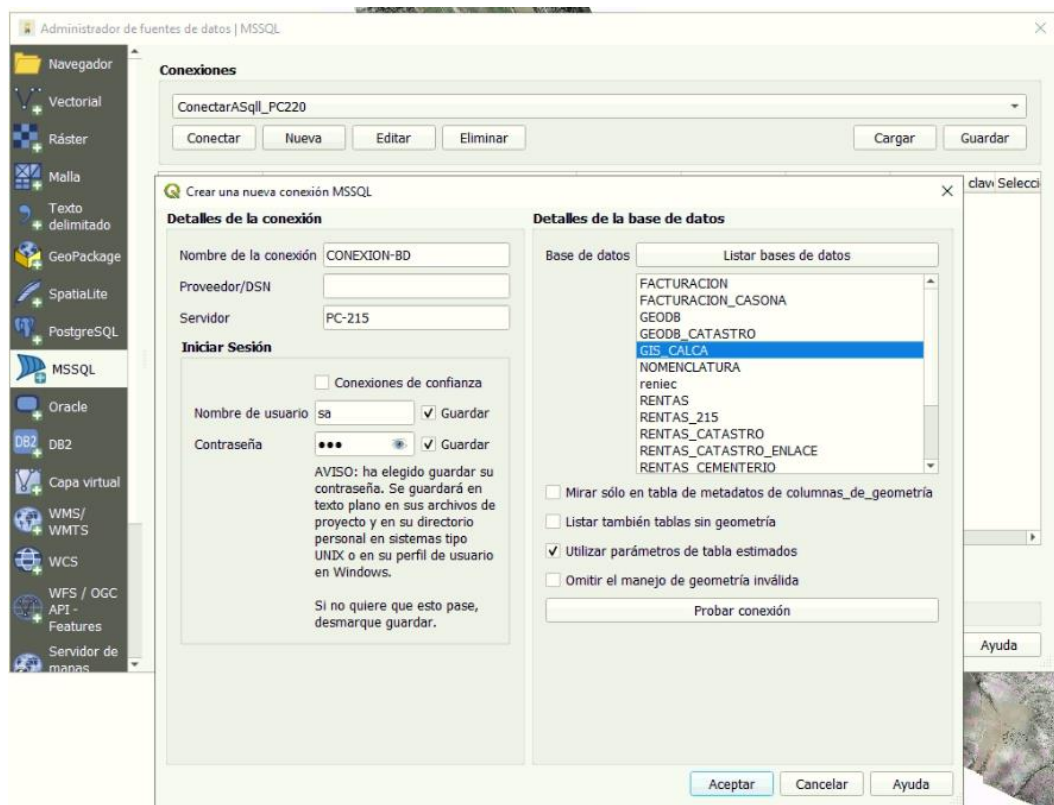
#### 3.1. CONEXIÓN MSSQL Y QGIS

Para poder cargar las capas o vistas de SQL server, debemos de realizar la conexión inicialmente.

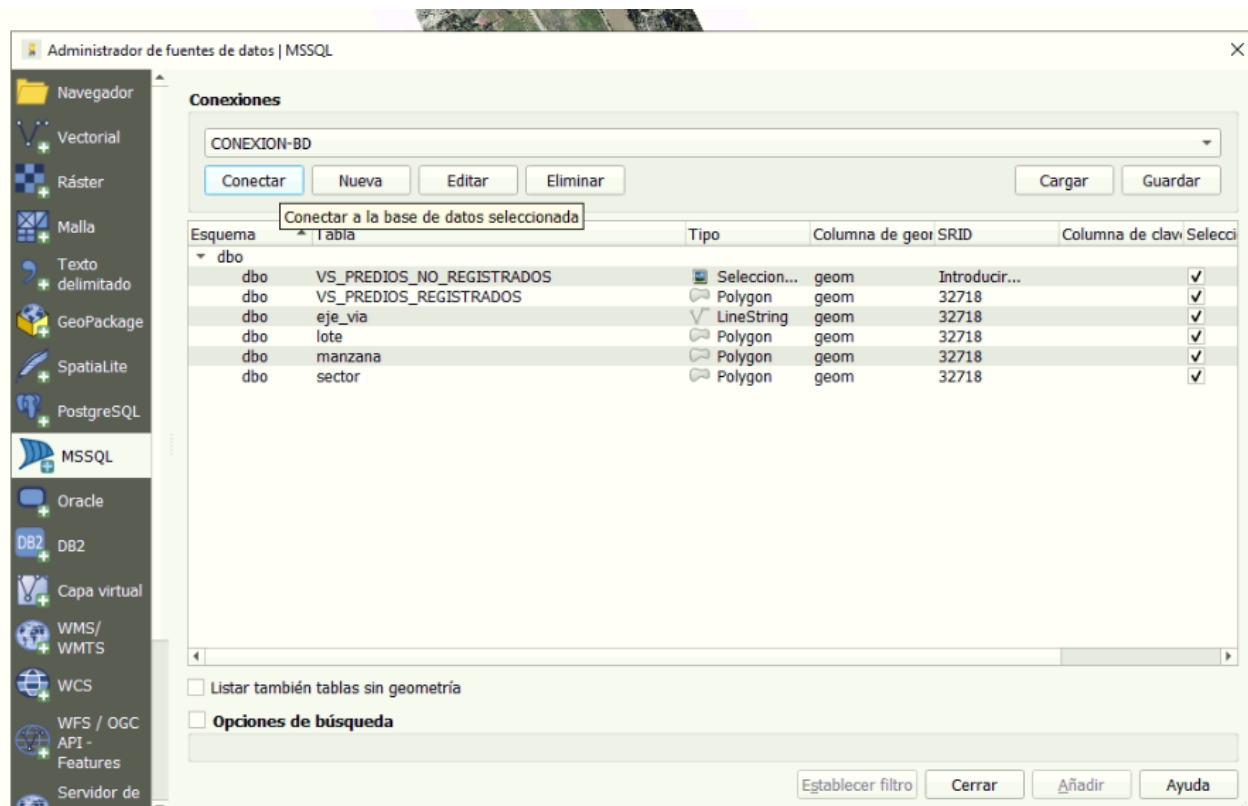
Paso 1:



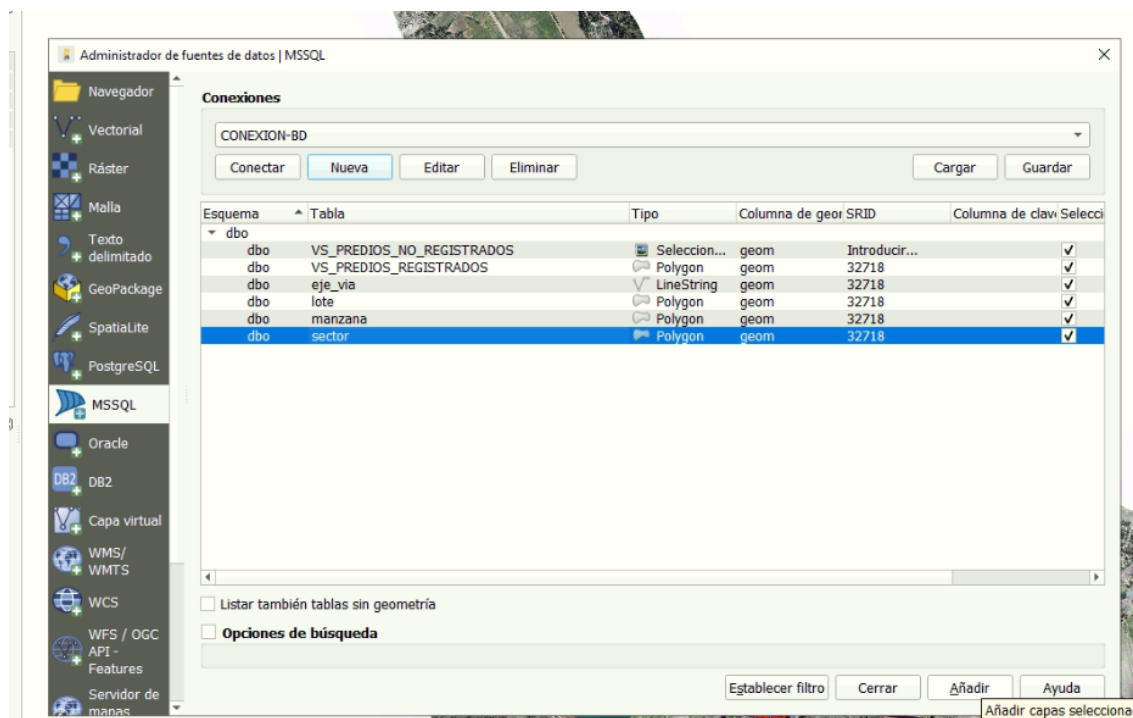
Paso 2: Seleccionamos la base de datos donde se encuentran nuestras vistas e información en general y aceptamos la conexión.



Paso 3: Finalmente hacemos clic en CONECTAR, esperamos que las vistas se carguen en la tabla, las reconoceremos por el nombre, si tiene datos GEOM, automáticamente nos presentará la proyección en la que están las vistas.



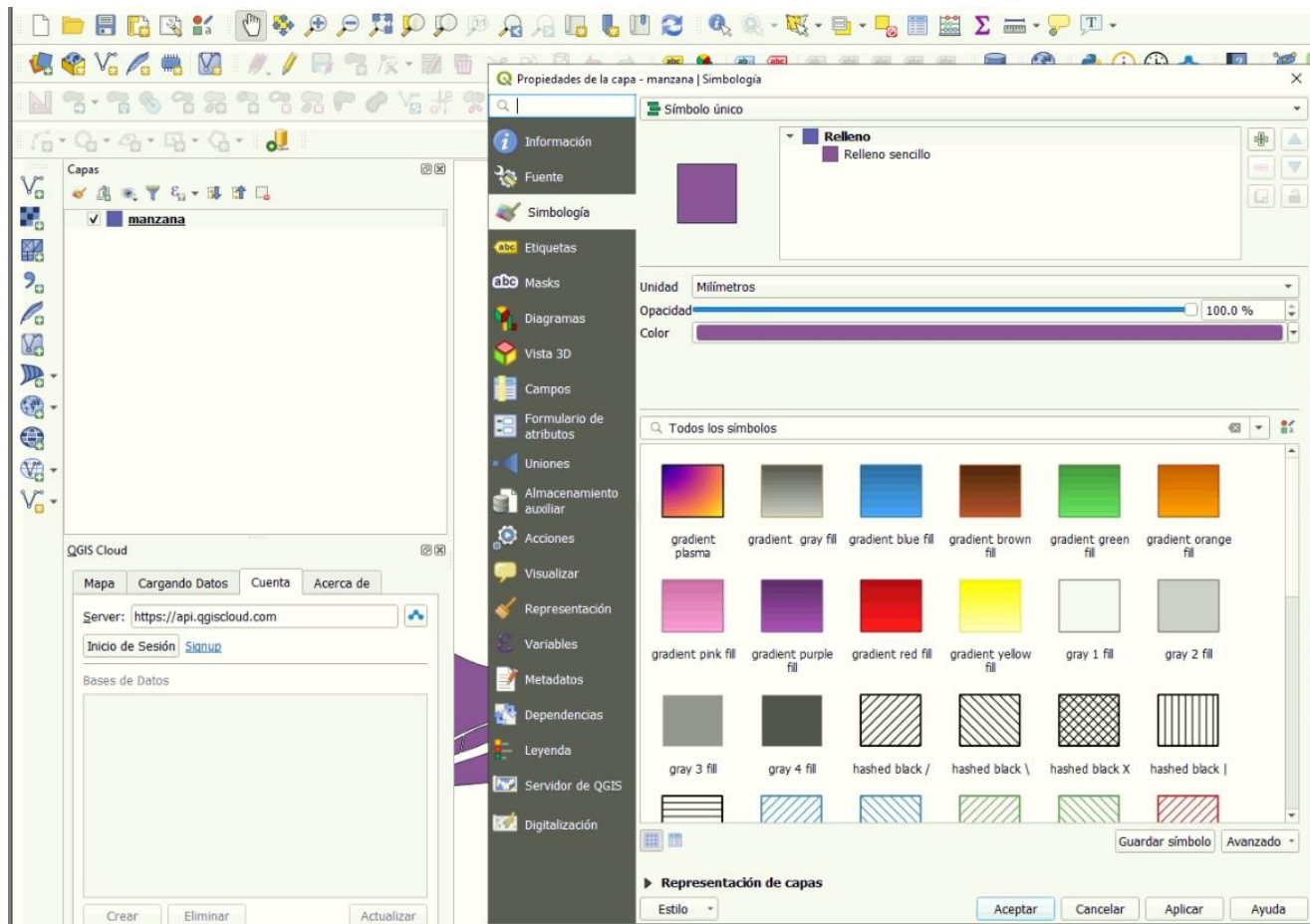
Paso 4: Ubicando la vista que queremos cargar, la seleccionamos y pulsamos el botón de AÑADIR.





### 3.2. EDITAR PROPIEDADES DE LA CAPA

Para abrir las propiedades de la capa cargada(vista), solo damos DOBLE CLIC en la capa, para este ejemplo cargamos la capa “manzana”, en este cuadro, podemos cambiar los colores, etiquetas, entre otras opciones.

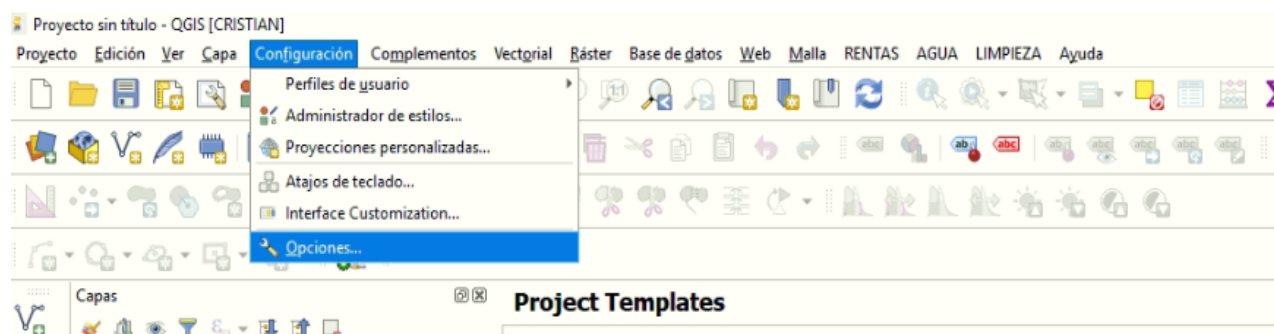


## 4. PROYECCIÓN DEL PROYECTO PERSONALIZADO

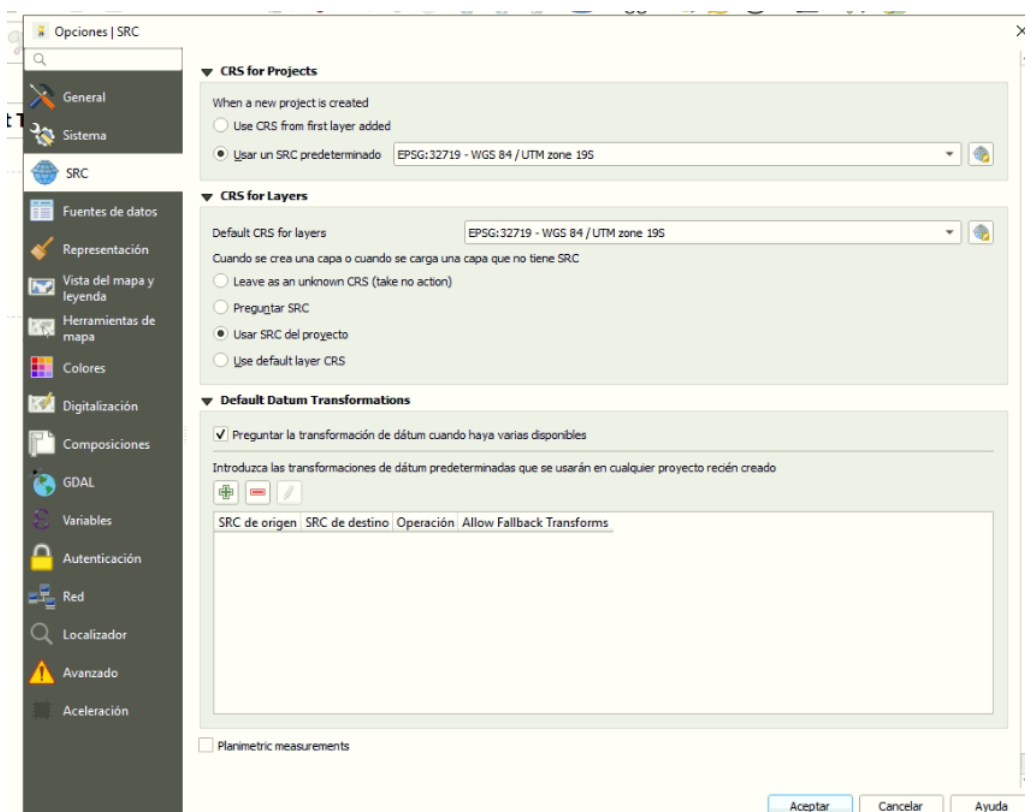
Esto se realiza solo una vez para así poder orientar la proyección de nuestro proyecto personalizado, ubicar las geometrías es importante ya que por defecto el QGIS inicializa la instalación en una proyección del medio oriente.

Usamos los siguientes pasos:

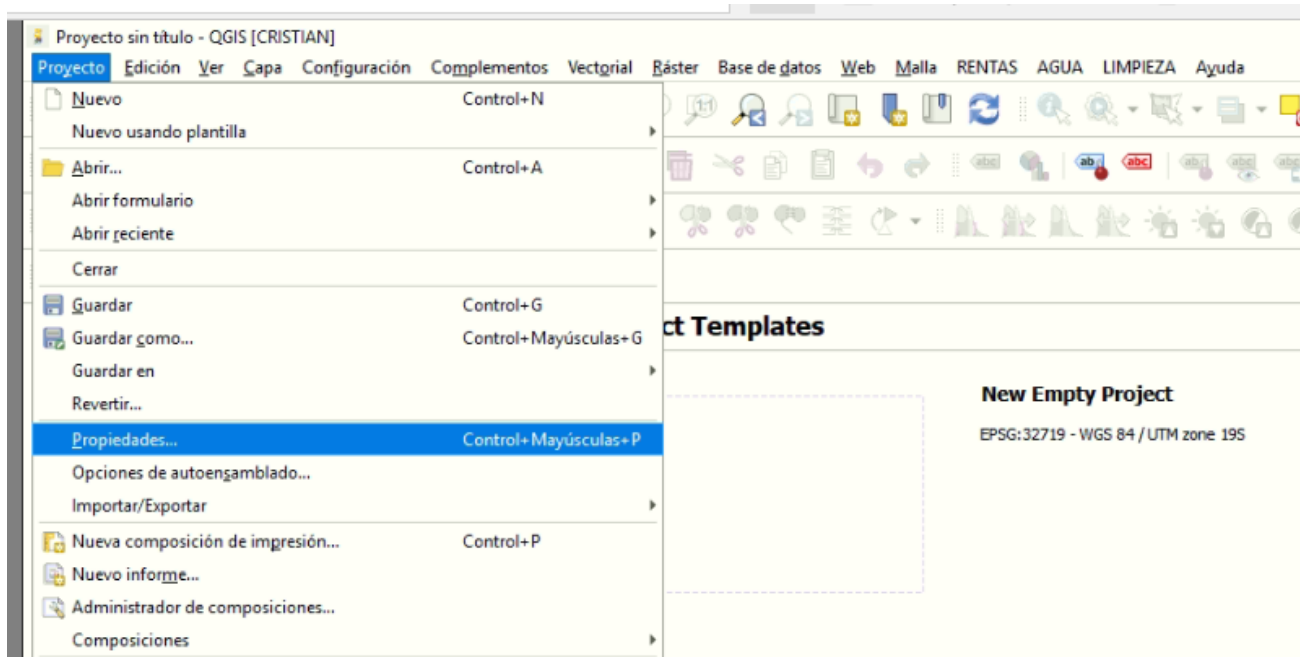
1. Vamos a Configuración>Opciones, se encuentra en la barra de menu principal.



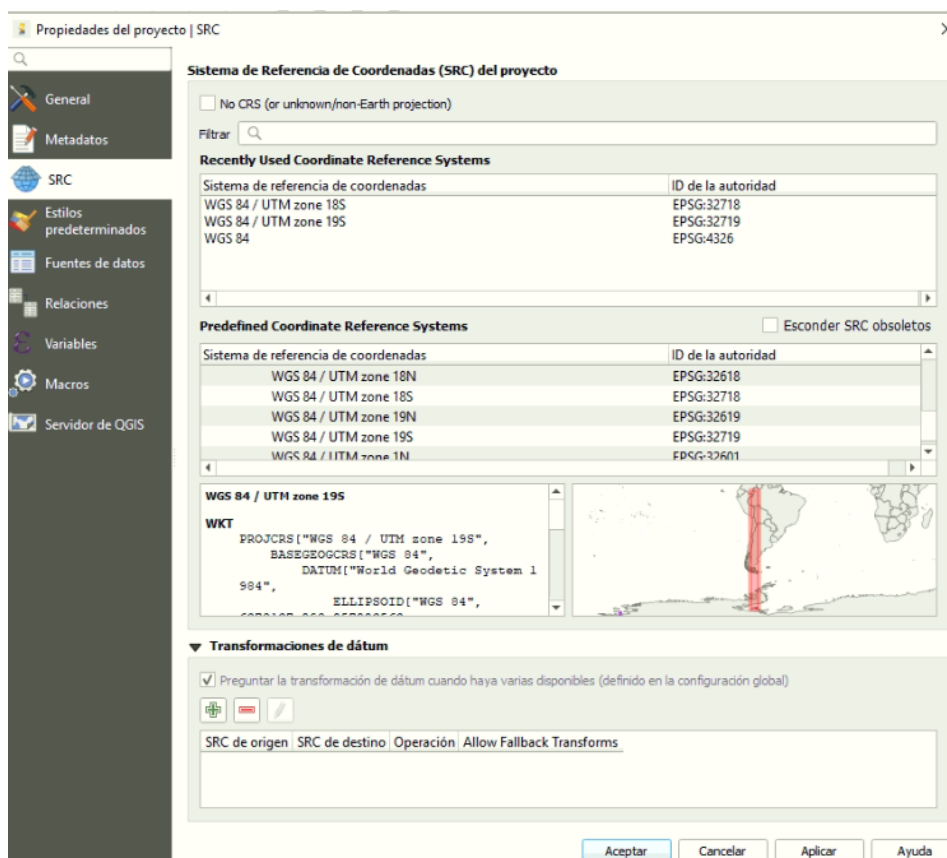
2. El canvas muestra Opciones | SRC, ahí debemos configurar el Sistema de Referencia de Coordenada (SRC) esté en la proyección de la Municipalidad elegida.



- Después hacemos lo mismo con las propiedades del proyecto por defecto. Proyecto>Propiedades



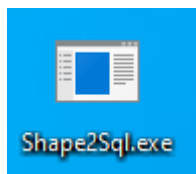
- SRC debe ser elegido de acuerdo a la proyeccion de la municipalidad. Que pertenecerá el proyecto personalizado.



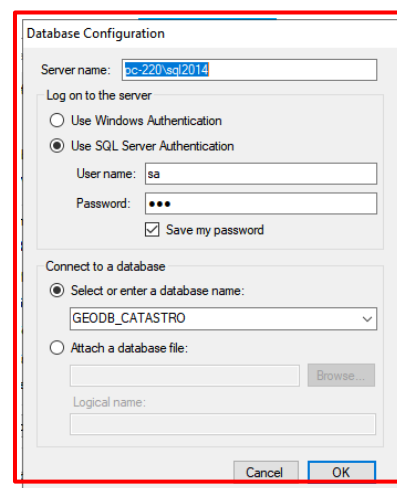
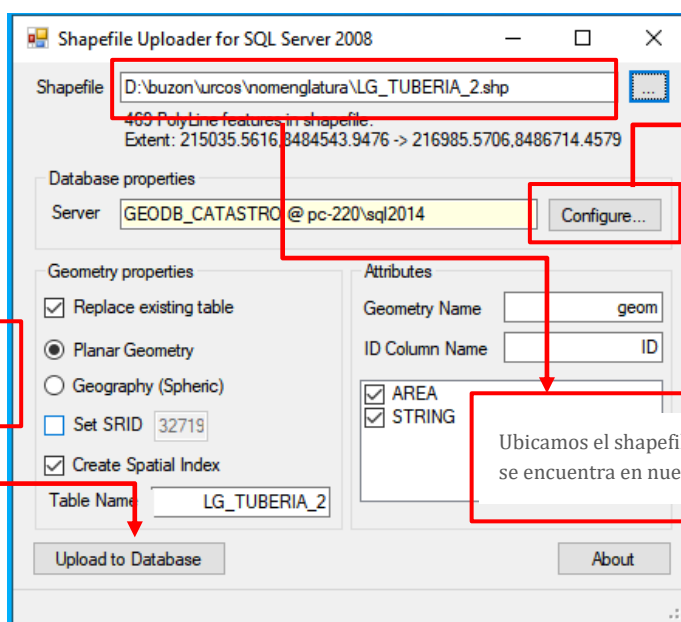


## 5. Migrar datos de shapefile a la base de datos SQL Server

1. Abrir el programa Shape2Sql.exe



2. Configuración para subir los datos de un shapefiles a una tabla de SQL Server.



Clic para subir los datos a la base de datos

Ubicamos el shapefile, donde se encuentra en nuestra PC.

Configuramos la información de la base de datos donde se subirá los datos del shapefile.

## 6. INFORMACION DEL CENTRO

Centro Guaman Poma de Ayala

Jr. Retiro N° 346 Urb. Tahuantinsuyo - Cusco

**Tel.** 084-235931

**Fax** 084-225552

[www.guamanpoma.org](http://www.guamanpoma.org)

