Componentes básicos para la creación de plugins QGIS con python en ambientes windows y linux

Essential components for development QGIS plugins with python in windows and linux environments

DOI: 10.46932/sfjdv3n1-117

Received in: Jan 30st, 2021
Accepted in: Feb 1th, 2022

Emmanuel Juárez Carbajal
Maestro en Ingeniería
Institución: Universidad Autónoma de Guerrero
Dirección: Av. Lázaro Cárdenas, S/N, Facultad de Ingeniería, Ciudad Universitaria, Chilpancingo, Guerrero, C.P. 39087, México
E-mail: emmanueljuarez@uagro.mx

Félix Molina Ángel
Maestro en Ciencias, en Ciencias Computacionales
Institución: Universidad Autónoma de Guerrero
Dirección: Av. Lázaro Cárdenas, S/N, Facultad de Ingeniería, Ciudad Universitaria, Chilpancingo, Guerrero, C.P. 39087, México
E-mail: molina@uagro.mx

René Vázquez Jiménez
Doctor en Geomática
Institución: Universidad Autónoma de Guerrero
Dirección: Av. Lázaro Cárdenas, S/N, Facultad de Ingeniería, Ciudad Universitaria, Chilpancingo, Guerrero, C.P. 39087, México
E-mail: rvazquez@uagro.mx

Iván Gallardo Bernal
Doctor en Sistemas Computacionales
Institución: Universidad Autónoma de Guerrero
Dirección: Av. Lázaro Cárdenas, S/N, Facultad de Ingeniería, Ciudad Universitaria, Chilpancingo, Guerrero, C.P. 39087, México
E-mail: drivangallardo@gmail.com

RESUMEN
El desarrollo de plugins enriquecen y aumentan la funcionalidad de QGIS, convirtiéndolo en un programa de Sistemas de Información Geográfica (SIG) de fuente abierta muy potente, sin embargo, resulta complejo el desarrollo de éstos para usuarios con poco conocimiento en estas aplicaciones o para personas que no están relacionadas con trabajos de este tipo. El objetivo de este artículo es instruir en la construcción de un plugin QGIS mediante un procedimiento paso a paso y la integración de las herramientas necesarias. Las herramientas que se integran para la construcción de un plugin son: QGIS 3.8, Python 3.7 y la biblioteca PyQGIS. Un plugin es un pequeño programa complementario de otro programa dándole un mejor funcionamiento al programa receptor. Como resultado se obtiene un plugin con su interfaz gráfica y con esto, el usuario final podría manipularlo dependiendo de la tarea programada en QGIS.

Palabras clave: pyqgis, plugin, qgis, sig, python, linux, windows.

ABSTRACT
Development of plugins enriches and increases the functionality of QGIS, turning it into a very powerful Open Source Geographic Information System (GIS) program, but it is a bit complex to develop these for users with little knowledge of this application or for people who are not related to jobs of this type. The aim of this work is to instruct in the construction of a QGIS plugin through a step-by-step procedure and the integration of the necessary tools. The tools that are integrated for the construction of a plugin are: QGIS 3.8 Python 3.7 and the PyQGIS library. A plugin is a small complementary program for another program, giving to the receiving program better performance. As a result, a plugin with its graphical interface is obtained and with this, the end user could manipulate it depending on the task programmed in GIS.

Keywords: pyqgis, plugin, qgis, gis, python, linux, windows.

1 INTRODUCCIÓN

El auge de las tecnologías ha sido de gran ayuda en la actualidad y la integración de distintas multidisciplinas para obtener mejores resultados ha tenido gran aceptación para dar soporte a diferentes tipos de investigaciones. Tal es el caso del desarrollo de plugins a través de lenguajes de programación (como C++ y Python) que son ejecutados en ambiente SIG (QGIS), lo cual ha significado un avance que ha permitido realizar mejoras tanto en el procesamiento de datos como en los resultados obtenidos. La problemática general que se ha identificado es que muchas veces se recurre a la creación de plugins para realizar tareas específicas que contribuyen a la solución a ciertos problemas planteados en los trabajos de investigación, pero no muestran el proceso de desarrollo del plugin.

Vázquez (2017), da a conocer las principales características de algunos SIG basados en software libre, mostrando las posibilidades que poseen para el cálculo de estadísticos simples en secuencias temporales de mapas ráster. Como caso de estudio; con datos climáticos de Cataluña, España, se desarrolló un módulo para Sextante que en conjunto con gvSIG facilitan el cálculo de estadísticos simples sobre secuencias de mapas ráster. Lapuente (2018), describe el funcionamiento de QGIS 3.0 (Sistema de Información Geográfica de código libre) y de un plugin creado especialmente para poder automatizar la comprobación de geometrías vectoriales a partir de datos abiertos de OpenStreerMap (OSM) y los datos de otra base de datos topográfica del Institut Cartogràfic i Geològic de Catalunya (ICGC).

Bortagaray (2018) plantea un desarrollo e implementación con rutinas computacionales útiles para el procesamiento y análisis de series temporales de índice de Vegetación Normalizado (Normalized Difference Vegetation Index, NDVI) extraídas de imágenes obtenidas por el sensor MODIS (Moderate Resolution Imaging Spectroradiometer), los datos obtenidos de los programas propios pueden utilizarse, por ejemplo, para analizar y comparar la vegetación de distintos puntos geográficos a través del tiempo.

Se propuso que se ejecutasen en forma de plugins, codificados en Python, como aporte para el sistema de información geográfica de código libre QGIS.

Así como los avances tecnológicos ayudan al mejoramiento de tareas, existe la problemática de que algunos usuarios interesados en el mejoramiento de sus investigaciones no cuentan con el conocimiento en el desarrollo de plugins en QGIS y optan en seguir con la misma rutina de trabajo, siendo un proceso de construcción no tan complicado pero dos cosas son necesarias: en principio, tener experiencia en programación y segundo, el manejo de los SIG. Este documento está orientado a personas con un mínimo de conocimiento de lo antes mencionado.

El objetivo de este trabajo es desarrollar una metodología que muestre el proceso de construcción de un plugin para QGIS con Python en los ambientes operativos más comunes: Windows y Linux, para que los usuarios que deseen reducir los tiempos de ejecución puedan aplicar esta guía para la creación de plugins en QGIS.

1.1 DESCRIPCIÓN DE PYQGIS

PyQGIS es un acrónimo formado por las palabras Python y QGIS. Python es un lenguaje de programación que, por su facilidad de aprendizaje y versatilidad, es ampliamente usado en el mundo de los Sistemas de Información Geográfica y otros ámbitos. PyQGIS es una biblioteca de QGIS para ejecutar código Python y tiene soporte para ejecutar scripts utilizando el lenguaje.

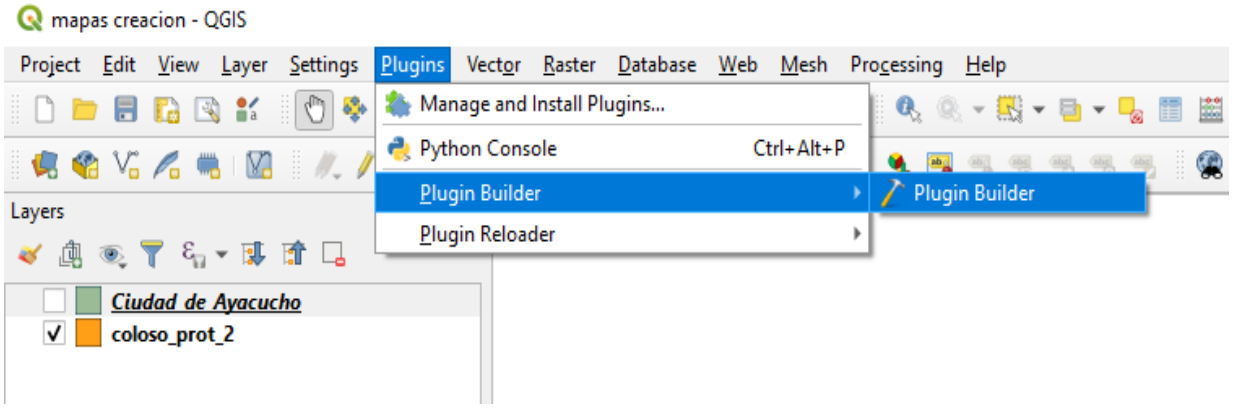
2 METODOLOGÍA PARA EL DESARROLLO DE PLUGINS EN QGIS

En este apartado se describe el procedimiento para la construcción de plugins en QGIS, explicando cada etapa para lograr el objetivo y así obtener el plugin con su interfaz gráfica para el usuario final. Esta metodología es aplicable para entornos Microsoft Windows y Linux, con QGIS versión 3.8 y Python 3.7 para Windows y, QGIS versión 3.10 y Python 3.8 para Linux, tomando en cuenta también las bibliotecas de QGIS para el desarrollo gráfico del plugin llamada PyQt5.

2.1 PRIMER PASO PARA EL DESARROLLO DEL PLUGIN

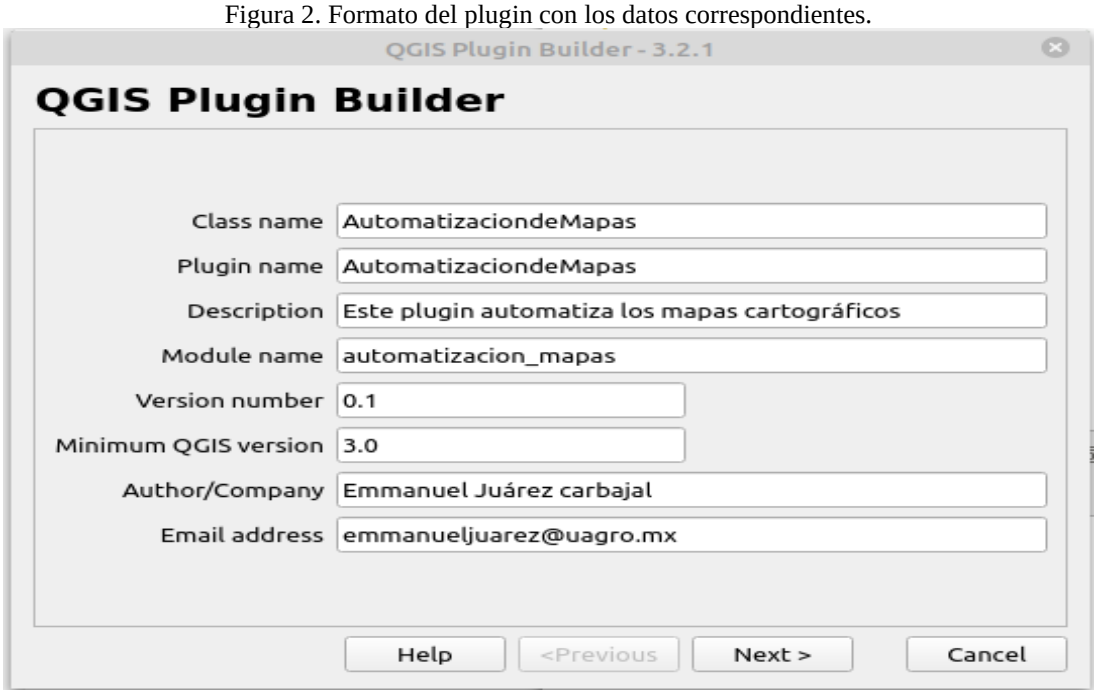
Como inicio se presenta la interfaz de QGIS y la herramienta que tiene integrada para la construcción del plugin, para eso se selecciona: **Plugins -> Manage and Install Plugins** (Si es la primera vez, desde la ventana de Plugins, se selecciona Plugin Builder, para su instalación)**,** de esta manera se instala **Plugin Builder**. Ya instalado aparecerá en el menú Plugins, y al seleccionar **Plugin Builder** despliega la interfaz en la ventana que se ilustra en la figura 1.

Figura 1. Interfaz QGIS de Plugin Builder.



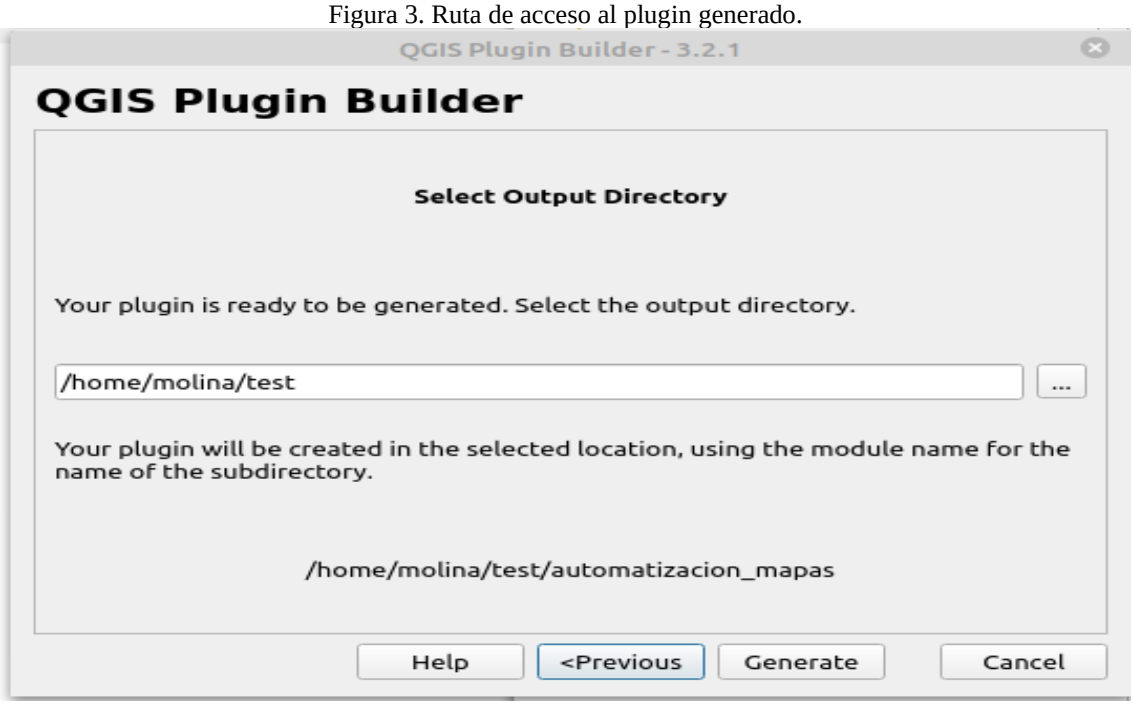
Al seleccionar **Plugin Builder** se presenta una nueva opción que despliega la ventana donde se ingresan los datos que identifican al plugin tales como el nombre de la clase, nombre del plugin, descripción, versión, autor y un correo electrónico de contacto del desarrollador. En la figura 2 se ilustra la ventana donde se ingresan los datos de ejemplo.

Figura 2. Formato del plugin con los datos correspondientes.



Al llenar los campos solo se da clic en **<Next>** en ésta y en las próximas ventanas hasta llegar a la ventana donde se especifica la ruta para guardar los archivos relacionados con el plugin generado. En la figura 3 se muestra la ventana donde se selecciona la ruta de acceso, y se termina dando clic en el botón **Generate**.

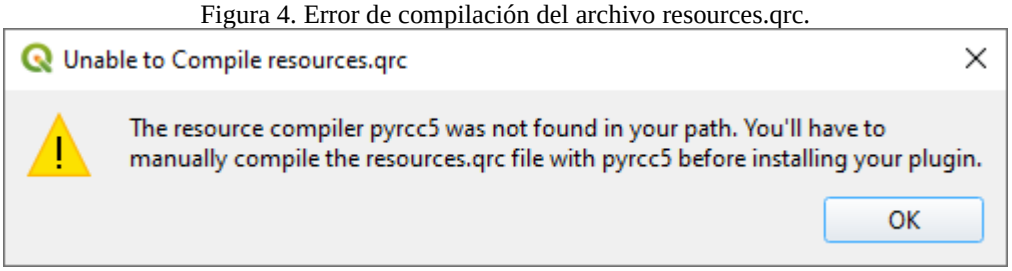
Figura 3. Ruta de acceso al plugin generado.



2.2 SEGUNDO PASO PARA EL DESARROLLO DEL PLUGIN

Si es la primera vez que se intenta crear un plugin, al hacer clic en **Generate**, podría desplegar un cuadro de diálogo mostrando un problema de compilación del archivo denominado **resources.qrc**, este archivo es una parte fundamental para el funcionamiento del plugin ya que sin él no podría funcionar dentro de QGIS. En la figura 4 se muestra el mensaje de error de compilación.

Figura 4. Error de compilación del archivo resources.qrc.



Para evitar este problema, a continuación, se describen dos procedimientos: el primero debe seguirse en el caso de sistemas con Microsoft Windows, y el segundo es aplicable para sistemas con LinuxMint específicamente.

Caso 1. En sistemas con Windows es necesario ejecutar un script .bat que corrige este problema. Este script (compile.bat) que se crea en la carpeta donde será creado el plugin, incluye tres llamadas de ambiente y la última línea que sirve para generar el módulo Python **resources.py**, como se ilustra en la figura 5.

Figura 5. Script para la generación del archivo resources.py.

compile.bat: Bloc de notas

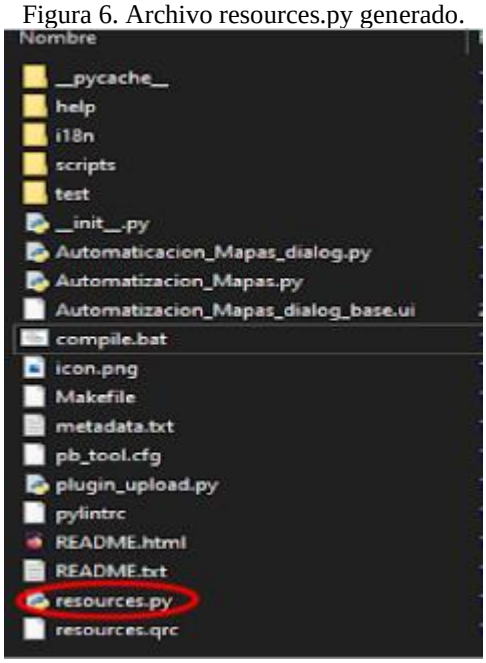
Archivo Edición Formato Ver Ayuda

```
@echo off
call "C:\OSGeo4W64\bin\o4w_env.bat"
call "C:\OSGeo4W64\bin\qt5_env.bat"
call "C:\OSGeo4W64\bin\py3_env.bat"

@echo on
pyrcc5 -o resources.py resources.qrc
```

Al ejecutar el script **compile.bat**, pyrcc5 toma como entrada el archivo **resources.qrc** y lo convierte en un módulo Python **resources.py**, mismo que puede observase como resultado de la ejecución del script, en la figura 6.

Figura 6. Archivo resources.py generado.



Caso 2. Instalar la biblioteca de soporte para la creación de interfaces gráficas en Python (Debian). Este paquete contiene tres herramientas de ayuda para desarrolladores PyQt: un compilador de interfaces de usuario (pyuic5), un generador de archivo de recursos (pyrcc5) y un buscador de cadenas de traducción (pylupdate5). Para su instalación, desde una ventana Terminal de Linux, se lleva a cabo la actualización de paquetes e instalación de pyqt5-dev-tools.

$ sudo apt update

$ sudo apt install pyqt5-dev-tools

Se recomienda instalar este paquete antes de iniciar con la creación de plugins para evitar el error de compilación al tratar de generar un plugin.

Resuelto el problema de conversión, al hacer clic en el botón **Generate**, mostrará la última ventana del proceso, con el título "**Plugin Builder Results**", esto quiere decir que el plugin se creó satisfactoriamente en el directorio o carpeta de salida especificado. En las figuras 7a y 7b se muestran los resultados del plugin creado, de acuerdo con el sistema operativo usado.

Figura 7a. Resultados del plugin creado en Windows.

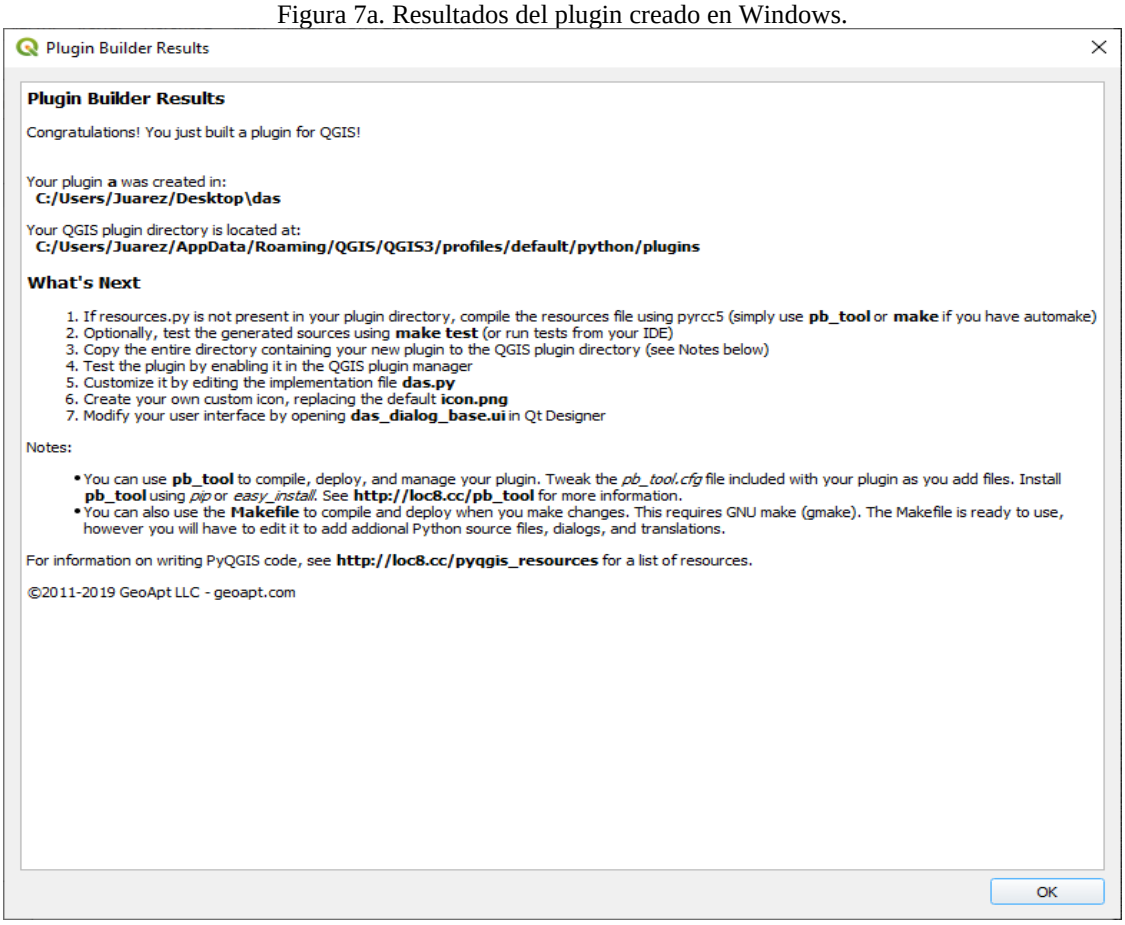
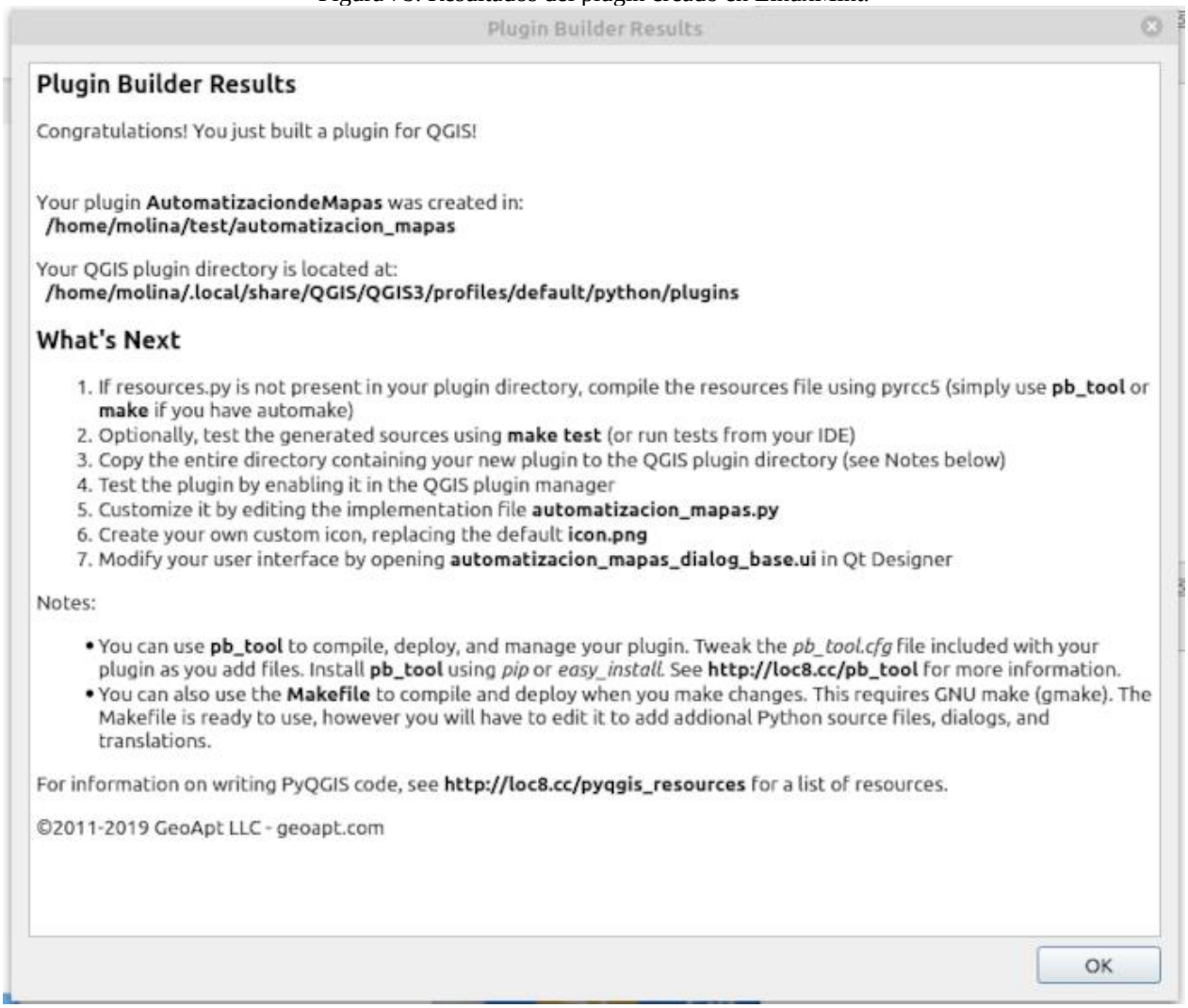


Figura 7b. Resultados del plugin creado en LinuxMint.



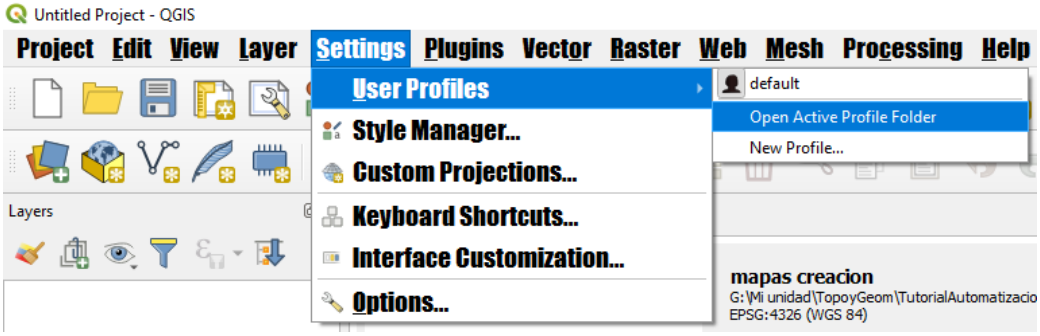
3 RESULTADOS

Como resultado se obtiene el plugin con todos los archivos necesarios que se utilizan para programar las tareas y diseñar la interfaz de usuario. Para que el plugin funcione se codifica la solución en Python y el diseño de la interfaz gráfica con la biblioteca Qt Designer que es parte de QGIS. Para el caso del ejemplo se utilizaron los archivos **Automatizacion_Mapas.py** y **Automatizacion_Mapas_dialog_base.ui** como base del plugin. En el módulo **.py** se codifica la lógica de las tareas a realizar y en él **.ui** se diseña la interfaz gráfica de usuario.

Es importante señalar que es necesario reubicar la carpeta del plugin creado a la carpeta de plugins de QGIS para que pueda visualizarse en la interfaz de SIG. Al seleccionar Open **Active Profile Folder**, desde el menú **Settings->User Profiles**, podrá observase la ruta base donde debe guardarse la carpeta de cada plugin creado. En la figura 8, se muestra la ruta completa de acuerdo con el perfil del usuario **Juarez**.

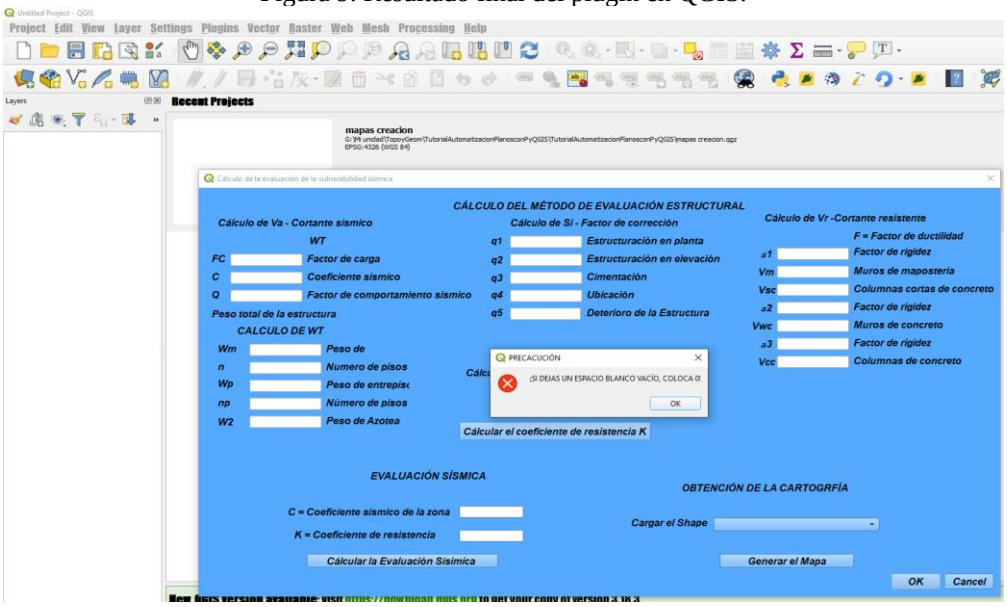
C:\Users\Juarez\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins

Figura 8. Ruta base de los plugins según el perfil de usuario.



El resultado final es la presentación del plugin funcionando dentro del entorno de QGIS, con su interfaz gráfica y su codificación. Se puede observar que solo se colocó lo necesario para agilizar algunos cálculos que son programados para previas tareas. La figura 9 ilustra el resultado del plugin en funcionamiento dentro de QGIS.

Figura 9. Resultado final del plugin en QGIS.



4 COMENTARIOS FINALES

4.1 RESUMEN DE RESULTADOS

Los resultados de este trabajo incluyen paso a paso la manera de realizar un plugin teniendo en cuenta los detalles que pudieran presentarse, siendo el caso de la compilación del archivo resources.qrc, el cual es un archivo muy importante para que el plugin funcione. También se describe la manera de utilizar el plugin dentro de QGIS ya que sin la colocación de la carpeta del plugin a la base de datos de QGIS no se podría visualizar y mucho menos utilizarse. Si se siguen estos pasos al pie de la letra puede crearse cualquier plugin sin ningún problema.

5 CONCLUSIONES

Los resultados demuestran el funcionamiento de un plugin, teniendo en cuenta los problemas que se llegaron a encontrar para realizarlo, tomando como ambientes de operación Microsoft Windows y LinuxMint.

RECOMENDACIONES

Se recomienda trabajar en el Sistema Operativo con el cual, el usuario se desempeñe mejor con las herramientas y el desarrollo será el mismo en cualquier S.O solo el error de compilación es el que cambia, así que queda a criterio del usuario para realizar el desarrollo de un plugin. Para que solucionen el error de compilación se deja un enlace https://www.qgistutorials.com/en/docs/3/building_a_python_plugin.html para que el interesado pueda consultar los errores en otros sistemas operativos y así solucionarlo.

REFERENCIAS

Baquero, L., S., Lozano, D., A., (2019). Plugin para la zonificación de amenaza por movimientos en masa implementando el uso de imágenes Sentinel 2, en el municipio de Chíquiza (Boyacá), (Tesis de Grado), especialización en sistemas de información geográfica, centro de investigación y desarrollo en información geográfica- CIAF, Bogotá, D.C.

Bortagaray, N. (2018). Desarrollo e implantación de algoritmos para GIS en análisis de series de tiempo, (Tesis de licenciatura), Universidad Nacional de Córdoba, Facultad de Matemáticas, Astronomía, Física y Computación, Córdoba, Argentina.

Francisco José de Caldas, Instituto Geográfico Agustín Codazzi, centro de investigación y desarrollo en información geográfica, Bogotá, Colombia.

Lapuente, C. (2018). Plugin en QGIS para automatizar la completitud semántica vectorial a partir de datos OpenStreerMap, (tesis de grado), Universidad Oberta de Catalunya, España.

Manrique, D., A. (2017). Desarrollo de un complemento de QGIS, Para la estimación de curvas de intensidad, duración y frecuencia a partir de datos del IDEAM, monografía proyecto de trabajo de grado, Universidad distrital

Vázquez, R. (2017). Uso de sistemas de información geográfica libres para la protección del medio ambiente. Caso de estudio: manipulación de mapas ráster con datos climáticos. Universidad y Sociedad, 10(2), 158-164. Consultado por Internet, el 23 de abril de 2021, en: http://rus.ucf.edu.cu/index.php/rus

Debian. Development tools for PyQt5. Consultado por Internet, el 25 de enero de 2022, en: https://packages.debian.org/stretch/pyqt5-dev-tools