PRUEBA DESARROLLO SOFWARE QT

Anteriormente había trabajado Qt (versión 4.6), no me había enfrentado a realizar una solución que utilizara mapas y peticiones http a un servidor.

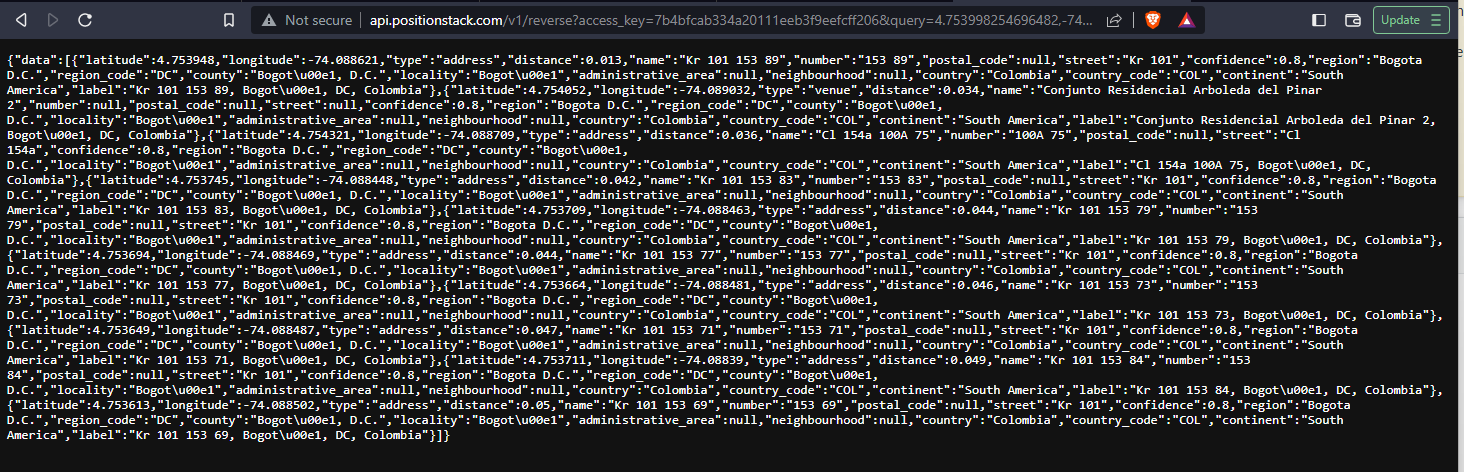
Desarrollé el proyecto con Qt creator versión 9.0, con Qt framework versión 5.15.2 sobre Windows 11.

Para empezar, utilicé como base el ejemplo que viene instalado con el framework en el cual utiliza webenginewidgets para cargar una interfaz con un mapa proporcionado por los servicios de Google, en el cual utiliza las clases QWebEngineView y QWebEnginePage.

Para buscar la información de ubicación geográfica a partir de coordenadas GPS encontré en internet positionstack, un api de tipo rest que, a partir de las coordenadas geográficas, devuelve la siguiente información:

**latitude, longitude, type, distance, name, number, postal\_code, street, confidence, region, region\_code, county, locality, administrative\_area, neighbourhood, country, country\_code, continent, label**

Esta información llega en un JSON array, el tamaño varía dependiendo que tanta información de interés tiene disponible, en una ciudad grande el array tiene más elementos en comparación a un punto alejado de la civilización.



Respuesta a un punto ubicado en una ciudad

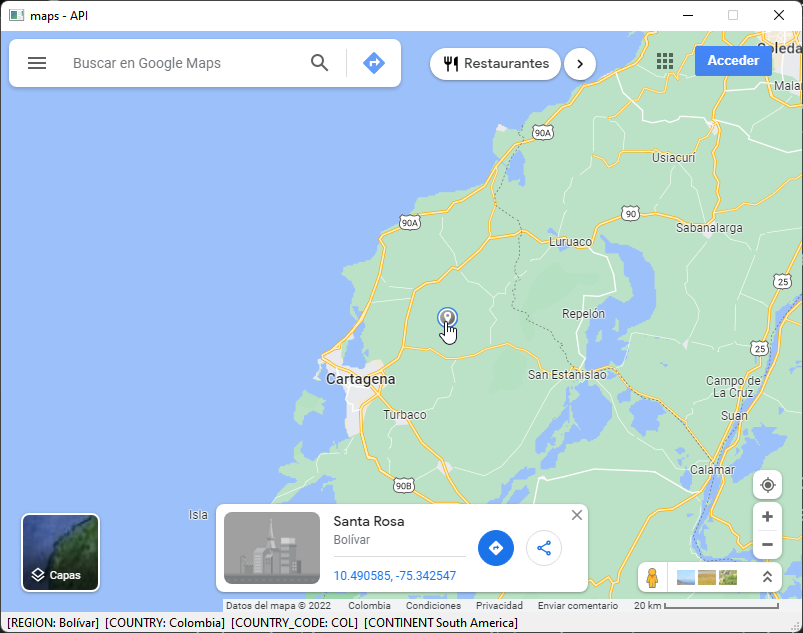


Respuesta a un punto ubicado en el llano colombiano.

Para más información del api consultar la siguiente URL:

https://positionstack.com/documentation

De los datos entregados por el api, seleccioné 4 para mostrar: region, country, country\_code y continent. Esta información se muestra en la barra de estado de la aplicación.



La prueba requería mostrar otros datos del país como tiempo actual, moneda, lenguaje, pero no encontré un API gratuito que me entregara estos datos, pero el concepto para obtener esos datos es similar a como se desarrolló con el api anterior.

Para realizar este rest api utilicé la clase QNetworkAccessManager para transferir datos de la red, para realizar la solicitud empleé la clase QNetworkRequest y para capturar la respuesta usé QNetworkReply.

Qt cuenta con las clases QJsonObject, QJsonValue y QJsonArray para manipular datos que vienen en formato JSON, la cuales se trabajaron en este proyecto.

Desafortunadamente no pude realizar la prueba técnica tal cual la solicitaron, y esto fue porque no encontré como capturar los eventos del mouse cuando se encuentra sobre el objeto m\_view de la clase QWebEngineView.

Como parte de la solución a este reto, la aplicación solicita al usuario que copie la latitud y longitud y haga clic sobre la barra de tareas, en ese momento se lee el portapapeles, se verifica que sea una coordenada válida, se realiza la petición de los datos a positionstack, cuando retornan, se valida el Json, se extraen los 4 campos de interés y se muestran en la barra de tareas



No poseo una pantalla táctil y no he desarrollado aplicaciones de Qt para Android, por lo que no tuve como probar el funcionamiento de esta aplicación en una pantalla táctil, pero por desarrollos anteriores, sé que el framework de Qt tiene soporte para pantallas táctiles.

Anexo se encuentra un video mostrando el funcionamiento de la aplicación.