**Universidad Nacional Autónoma de México**



Lic. En Informática

Profesor:

Martínez Fernández Juan Manuel

Materia:

Programación de Dispositivos Móviles

# De Cuenta - Alumno:

415139405 - Christian Daniel Gallegos Sánchez

Semestre - Grupo:

Semestre 6 – Grupo 8691

Tema - Actividad:

**Examen Final Practico 2018-1 Grupo 8691**

Fecha: 02/12/17

***Objetivo: Sube al repositorio de la última actividad las imágenes mostrando la aplicación corriendo en el emulador y como aplicación de escritorio. Manda la dirección web de tu repositorio a tu asesor.***

El repositorio que contiene el commit de la llamada al API de OpenWeather, con la configuración necesaria para pasar los parámetros de latitud y longitud del GPS de Android, es el siguiente: <https://github.com/TemplarWithin/Clima-2>

Mientras que el repositorio que contiene el commit de la aplicación de OpenWeather que se usó para ensamblar la apk, incluyendo tanto las modificaciones funcionales para que esta cuente con sus iconos integrados, como para que opere con el apoyo del GPS del dispositivo, es el siguiente: <https://github.com/TemplarWithin/OpenWeatherApp>

Dicho esto, y en lo que respecta a la aplicación que corre en el escritorio desde el servidor local, podemos señalar que esta es capaz de recibir dos parámetros con un tipo de dato flotante, los cuales representan la latitud y longitud que a su vez referencian la ubicación geoespacial de una ciudad o país, para que, por medio del API de la página de Open Weather, se regrese al usuario un JSON sobre con información de la temperatura y el clima; tal y como se muestra a continuación con la corrida de la aplicación, para los valores de Longitud y Latitud de Blue Springs, en Estados Unidos:

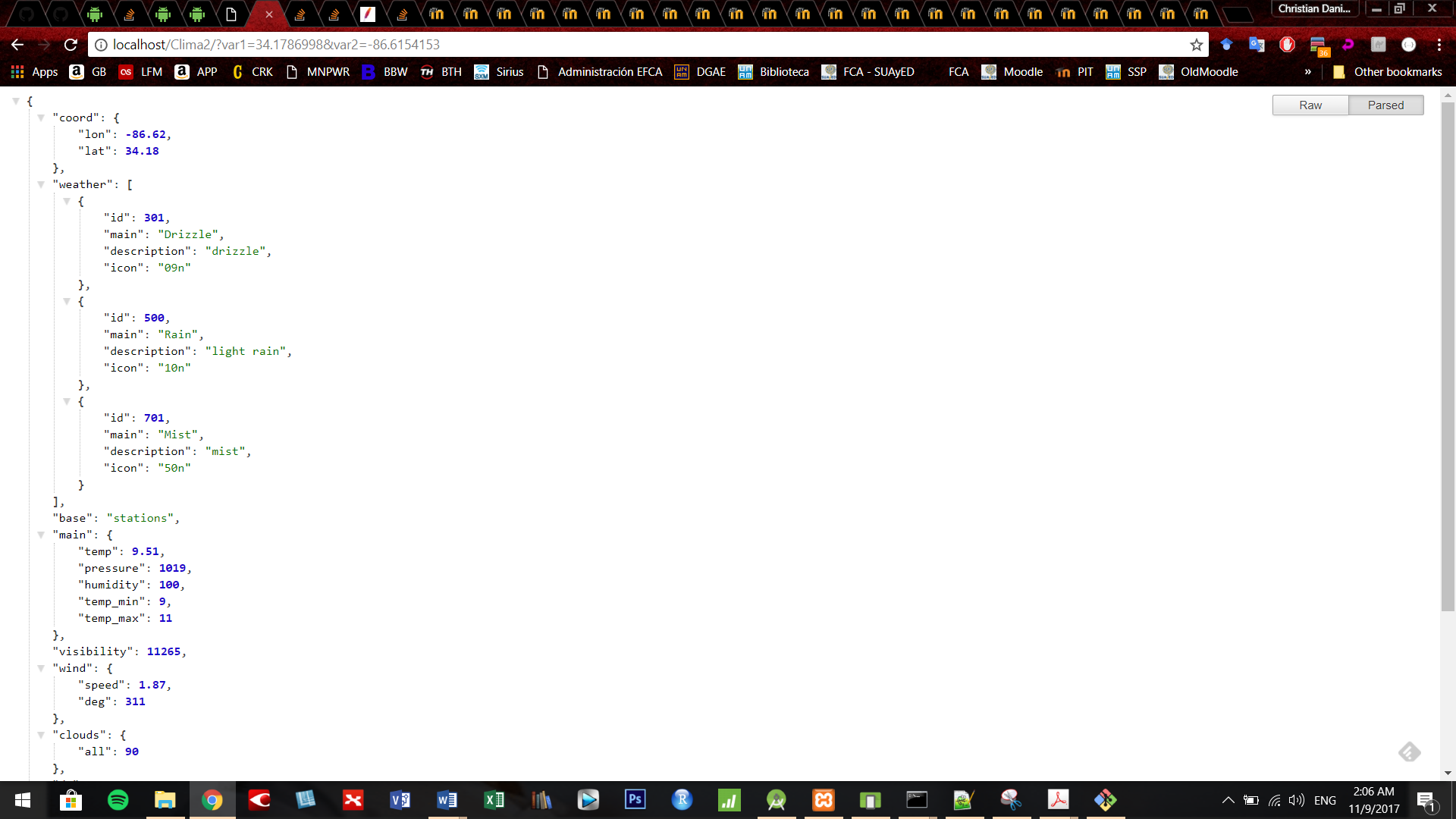


Ilustración 1 – Consulta de Clima de Blue Springs en Servidor Local (Latitud = 34.1786998 / Longitud = -86.6154153)

Por su parte, la aplicación de Android que se conecta con la aplicación de escritorio para generar un Linear Layout en el cual mostrar la descripción del clima, así como imágenes representativas del mismo, para una ubicación geoespacial proporcionada por el GPS del emulador (o el teléfono Android del cual se corre), así como la temperatura de dicha ubicación, se apoya de:

* Los métodos onSaveInstanceState() y onCreate(), para que la actividad se muestre del mismo modo en cualquier orientación de la pantalla.
* Los métodos de Volley para ejecutar operaciones de lectura y escritura en la red para escribir en un cache, respuestas de la forma de un objeto tipo Response, con las cuales dar lectura al API de Open Weather.
* Los métodos del Service Location API Client del conjunto de Google Play Services para obtener las coordenadas de longitud y latitud en las que se encuentra el dispositivo o el emulador.
* Y en la implementación de iconografía con el apoyo del Image Asset Studio que se encuentra integrada dentro de Android Studio

De modo que, al instalarse esta en el dispositivo Android, podrá apreciarse la misma, con la resolución adecuada para cada densidad de pantalla que la admita, tal y como se muestra a continuación:

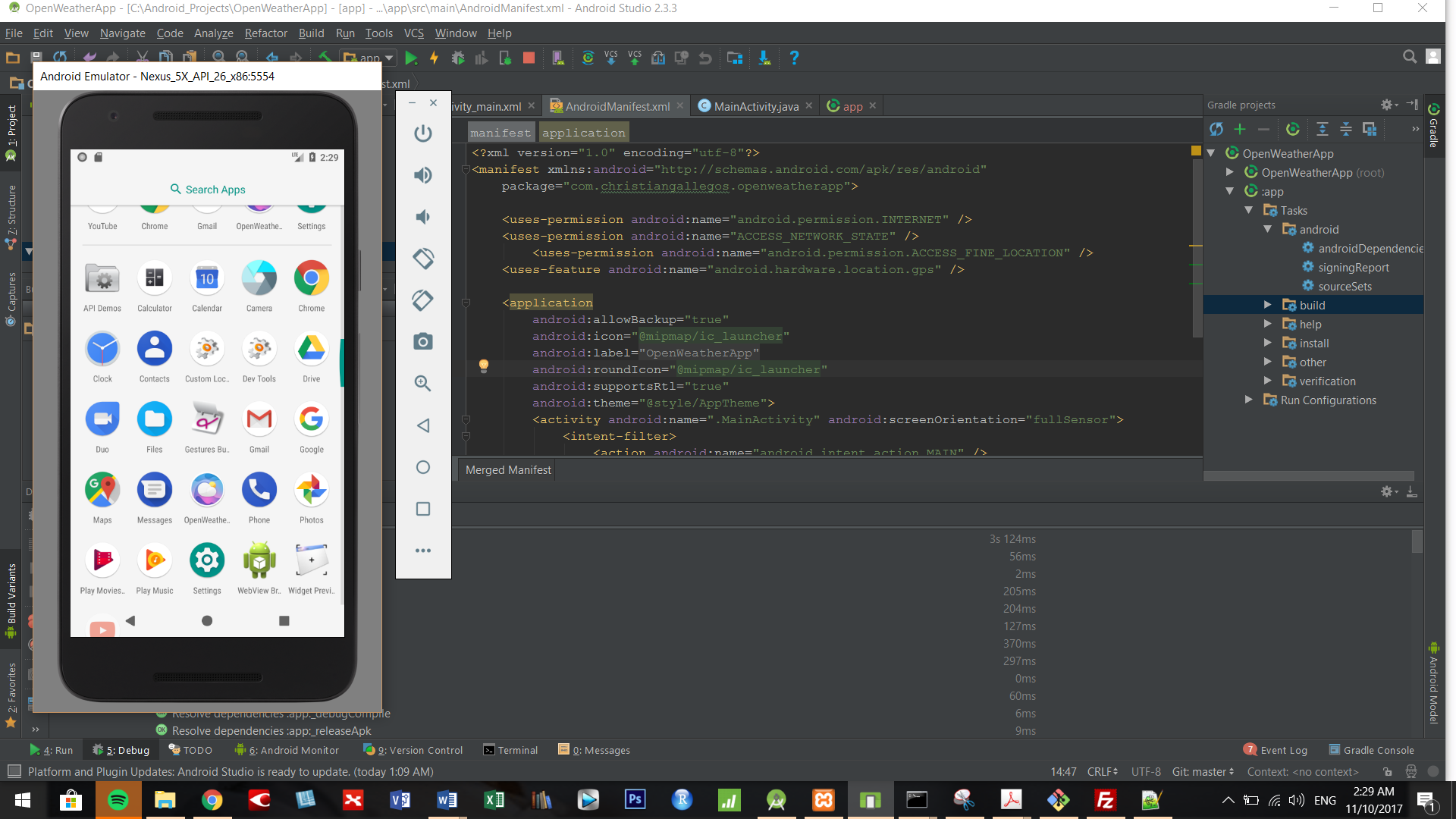


Ilustración 2 - Aplicación de OpenWeather Con Icono de App

Tal que, una vez ejecutada, esta será capaz de recibir los parámetros flotantes enviados por el GPS del dispositivo Android, para con estos realizar una consulta indirecta al API de OpenWeather, tal que este pueda a su vez devolverle el JSON requerido, para con este extraer la información de la temperatura y el clima que se mostrará al usuario final, tal y como se muestra a continuación para la simulación de la corrida de la aplicación en un emulador ubicado en Blue Springs, en Estados Unidos:

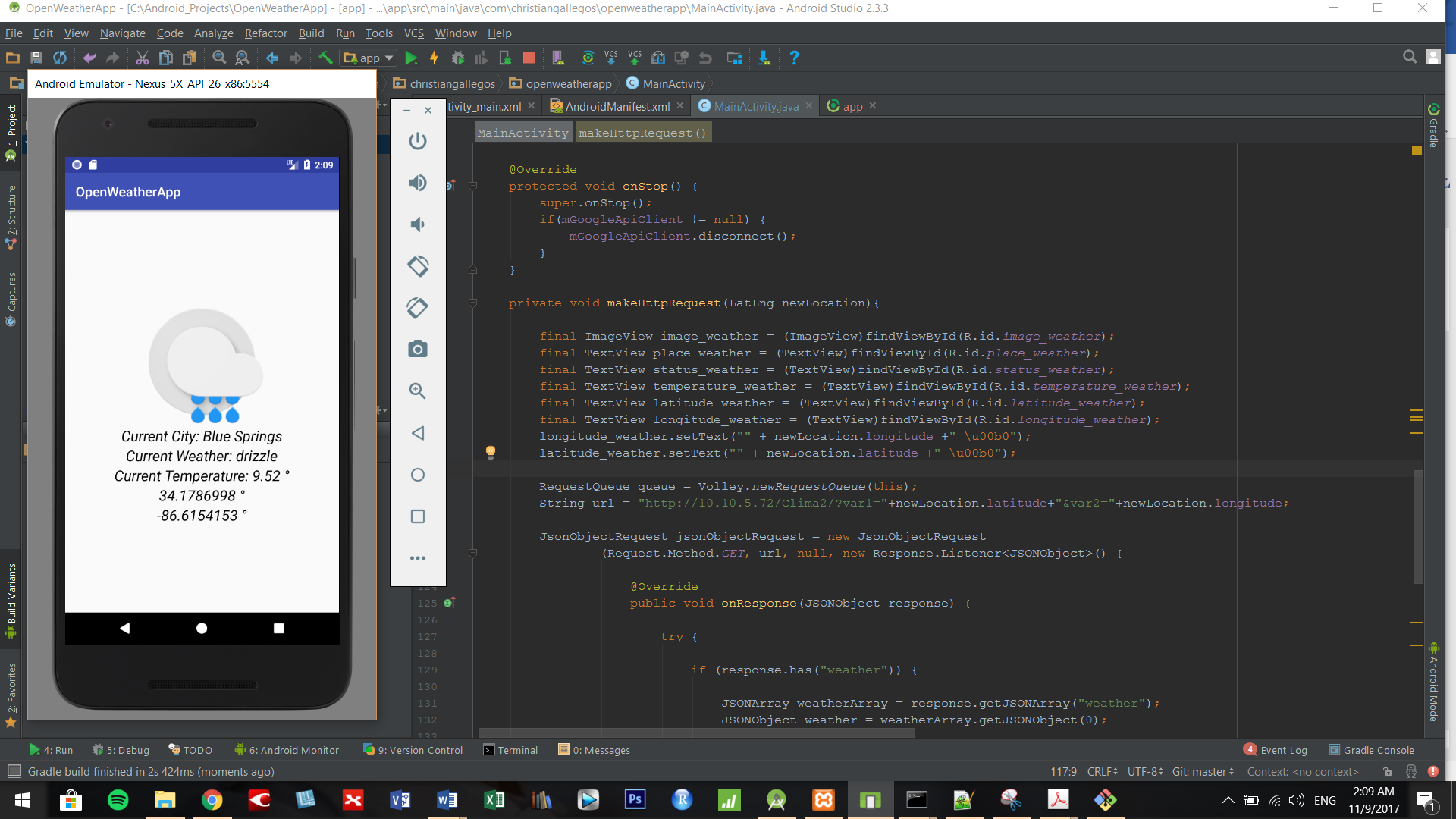


Ilustración 3 – Consulta de Clima de Blue Springs A Través del GPS de la Aplicación de Android

# Bibliografía

Fernández, J. M. (02 de Septiembre de 2017). *Carga de imágenes dinámicas en Android para OpenWeather.* Obtenido de https://www.youtube.com/watch?v=UWMs2iufAfE

Google. (10 de Octubre de 2017). *Connecting to the Network [Conectandose a la Red].* Obtenido de https://developer.android.com/training/basics/network-ops/connecting.html

Google. (11 de Noviembre de 2017). *Create App Icons with Image Asset Studio [Crear Iconos de Aplicacion con Android Studio].* Obtenido de https://developer.android.com/studio/write/image-asset-studio.html

Google. (08 de Octubre de 2017). *Getting The User Location [Obteniendo la ubicacion del Usuario].* Obtenido de https://developer.android.com/training/location/retrieve-current.html

Open Weather. (27 de Septiembre de 2017). *Open Weather City List.* Obtenido de http://openweathermap.org/help/city\_list.txt

StackOverflow. (11 de Noviembre de 2017). *Build unsigned APK file with Android Studio [Construir Aplicacion Sin Firma en Android Studio].* Obtenido de https://stackoverflow.com/questions/16709848/build-unsigned-apk-file-with-android-studio

Udacity. (27 de Septiembre de 2017). *Android Basics: User Interface.* Obtenido de https://www.udacity.com/course/android-basics-user-interface--ud834

UNAM FCA-SuayED. (2012). Apunte Electronico - Programacion de Dispositivos Moviles. Mexico, DF.