Práctica 1 Localización y GPS

Alumno: Jalife Guillén Daniel Gibran

Materia: Programación Móvil

Secuencia: 6NM61

Objetivo: Utilizar el GPS del dispositivo para obtener la localización del usuario.

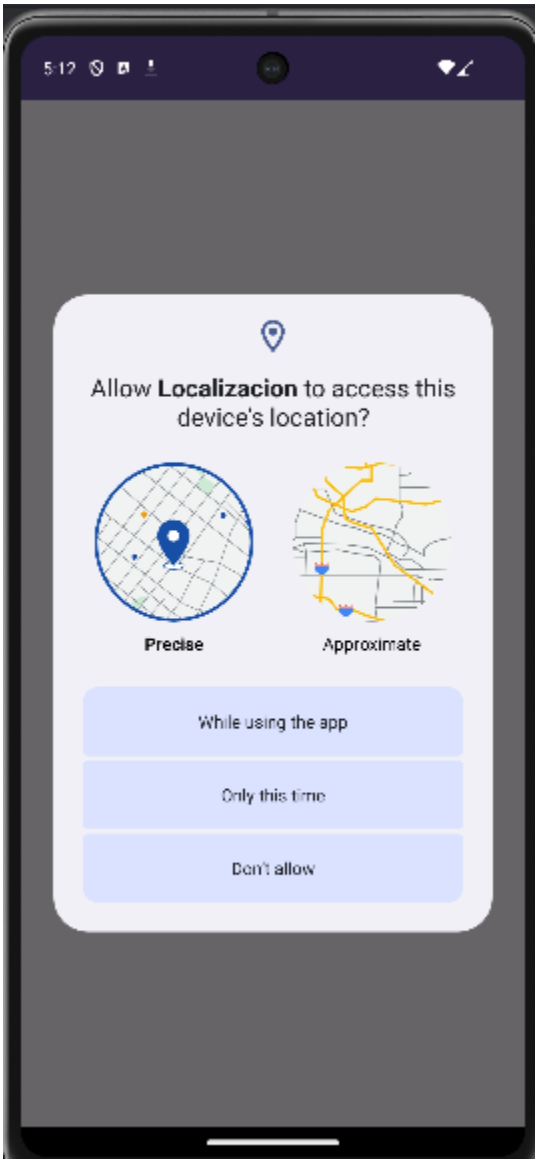
Código utilizado en la aplicación:

```
package com.example.localizacion

import android.annotation.SuppressLint
import android.content.Context
import androidx.appcompat.app.AppCompatActivity
import android.widget.*
import android.Manifest
import android.location.LocationManager
import android.os.Bundle
import com.google.android.gms.location.LocationServices
import android.content.pm.PackageManager
import android.location.Location
import androidx.core.app.ActivityCompat
import com.google.android.gms.location.FusedLocationProviderClient
import com.google.android.gms.location.LocationRequest
import com.google.android.gms.tasks.CancellationToken
import com.google.android.gms.tasks.CancellationTokenSource
import com.google.android.gms.tasks.OnTokenCanceledListener

class MainActivity : AppCompatActivity() {

    private lateinit var mFusedLocationClient: FusedLocationProviderClient

    private lateinit var tvLatitude: TextView
    private lateinit var tvLongitude: TextView
    private lateinit var btnLocate: Button

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        tvLatitude = findViewById(R.id.tvLatitude)
        tvLongitude = findViewById(R.id.tvLongitude)
        btnLocate = findViewById(R.id.btnLocate)

        mFusedLocationClient =
LocationServices.getFusedLocationProviderClient(this)

        btnLocate.setOnClickListener {
            getLocation()
        }
```

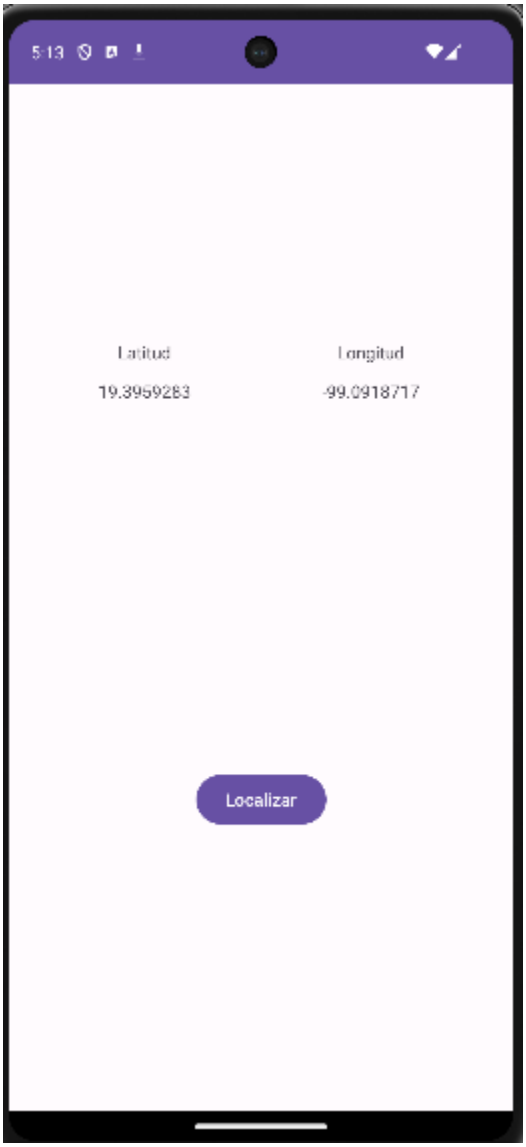
```
    }

    companion object{
        const val PERMISSION_ID = 33
    }

    private fun checkGranted(permission: String): Boolean{
        return ActivityCompat.checkSelfPermission(this, permission) ==
                PackageManager.PERMISSION_GRANTED
    }

    private fun checkPermissions() =
        checkGranted(Manifest.permission.ACCESS_COARSE_LOCATION) &&
                checkGranted(Manifest.permission.ACCESS_FINE_LOCATION)

    private fun requestPermissions() {
        ActivityCompat.requestPermissions(
            this,
            arrayOf(Manifest.permission.ACCESS_COARSE_LOCATION,
                Manifest.permission.ACCESS_FINE_LOCATION),PERMISSION_ID)
    }

    private fun isLocationEnabled(): Boolean {
        var locationManager: LocationManager =
            getSystemService(Context.LOCATION_SERVICE) as LocationManager
        return
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER) ||
                locationManager.isProviderEnabled(
                    LocationManager.NETWORK_PROVIDER
                )
    }

    @SuppressLint("MissingPermission")
    private fun getLocation() {
        if (checkPermissions()) {
            if (isLocationEnabled()) {

mFusedLocationClient.getCurrentLocation(LocationRequest.PRIORITY_HIGH_ACCURAC
Y, object : CancellationToken() {
                    override fun onCanceledRequested(p0:
OnTokenCanceledListener) = CancellationTokenSource().token

                    override fun isCancellationRequested() = false
                }).addOnSuccessListener(this) { location: Location? ->
                    tvLatitude.text = location?.latitude.toString()
                    tvLongitude.text = location?.longitude.toString()
            }
            }
        }else{requestPermissions()
        }
}
}
```

Aplicación funcionando:

La APP pide el permiso:

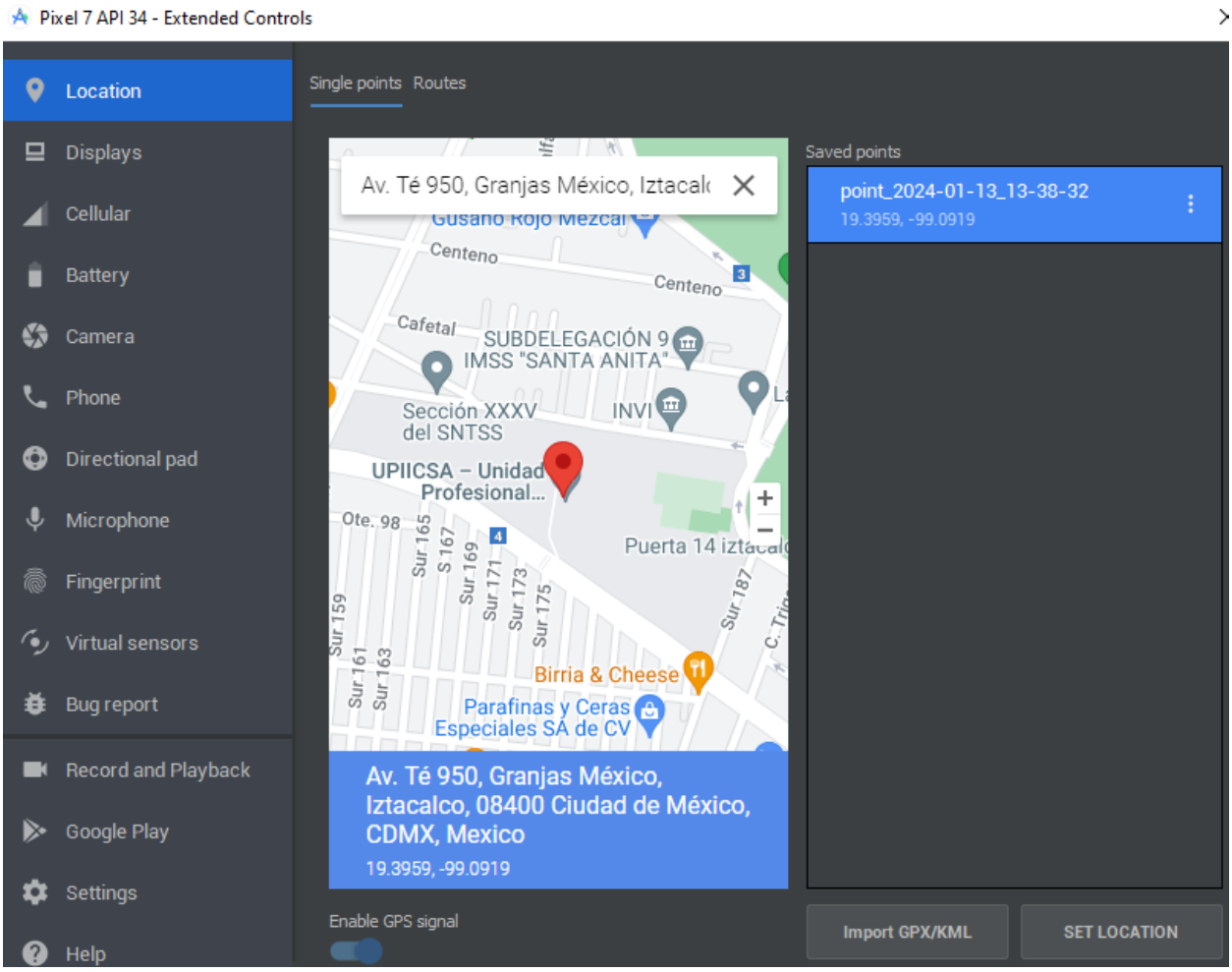


La APP regresa la localización:



Notas:

Localización establecida para el emuladors:



Utilizar la función “lastLocation” únicamente retorna la última aplicación conocida, por lo que regresa “null”. Para que regrese algo diferente a “null”, se tendría que utilizar una app que obtenga la localización actual, como Google Maps. Es por esa razón que se realizaron los cambios en esa parte.