



Instituto Politécnico Nacional

Escuela Superior de Cómputo



Programa académico / Plan de estudios

Ingeniería en Sistemas Computacionales / 2020

Unidad de aprendizaje

Desarrollo de aplicaciones móviles nativas

Tarea 4: "Desarrollo de una Aplicación de Geolocalización con Mapas"

Objetivo: Esta tarea tiene como propósito desarrollar habilidades prácticas en la implementación de servicios de geolocalización y mapas en aplicaciones Android nativas, explorando diferentes proveedores de mapas y añadiendo funcionalidades creativas que enriquezcan la experiencia del usuario.

Ejercicio 1: Implementación Básica de Localización con OpenStreetMap

Descripción de la actividad: En esta primera parte, implementará una aplicación que muestra la ubicación actual del usuario utilizando OpenStreetMap como proveedor de mapas.

1. Configuración del Proyecto

- Cree un nuevo proyecto en Android Studio.
- Configure los permisos necesarios en el archivo AndroidManifest.xml:

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- Implemente la solicitud de permisos en tiempo de ejecución siguiendo las mejores prácticas de Android.

2. Implementación con OpenStreetMap

- Configure un WebView para mostrar mapas de OpenStreetMap.
- Integre la biblioteca Leaflet.js para la interacción con el mapa.
- Implemente FusedLocationProviderClient para obtener las coordenadas actuales del usuario.
- Genere una página HTML con Leaflet.js que muestre un mapa centrado en esas coordenadas.
- Coloque un marcador en la posición actual con la etiqueta "Mi ubicación".
- Configure un manejo adecuado de errores para cuando no se pueda obtener la ubicación.

3. Optimización de Rendimiento

- Implemente un mecanismo para actualizar la ubicación de manera eficiente.
- Configure opciones de zoom apropiadas para la visualización del mapa.
- Optimice el rendimiento del WebView para una experiencia fluida.

4. Evidencias:

- Capturas de pantalla mostrando la aplicación en funcionamiento con OpenStreetMap.

- Capturas de pantalla del flujo de solicitud de permisos.
 - Código fuente relevante comentado (clases de ubicación, implementación del WebView, etc.).
-

Ejercicio 2: Implementación Alternativa con Google Maps

Descripción de la actividad: En esta segunda parte, implementará una versión alternativa de la aplicación utilizando Google Maps como proveedor.

1. **Configuración para Google Maps**
 - Cree una nueva Activity que utilizará WebView para cargar la versión web pública de Google Maps.
 - Mantenga la misma implementación de FusedLocationProviderClient para obtener la ubicación.
 - Configure el WebView para cargar la URL de Google Maps con las coordenadas específicas.
 2. **Navegación entre Implementaciones**
 - Implemente un sistema de navegación (por ejemplo, tabs o botones) que permita al usuario cambiar entre las implementaciones de OpenStreetMap y Google Maps.
 - Asegúrese de que la transición entre las diferentes implementaciones sea fluida y mantenga el estado de la aplicación.
 3. **Comparación de Rendimiento**
 - Implemente métricas básicas de rendimiento para comparar ambas implementaciones.
 - Mida el tiempo de carga inicial, el consumo de memoria y la respuesta a la interacción del usuario.
 - Documente las diferencias observadas en términos de rendimiento y experiencia de usuario.
 4. **Evidencias:**
 - Capturas de pantalla mostrando la implementación con Google Maps.
 - Capturas de pantalla del sistema de navegación entre implementaciones.
 - Tabla comparativa de las métricas de rendimiento entre ambas implementaciones.
 - Código fuente relevante comentado.
-

Ejercicio 3: Implementación de Funcionalidades Creativas

Descripción de la actividad: En esta tercera parte, extenderá la aplicación con funcionalidades creativas que enriquezcan la experiencia del usuario.

1. **Sistema de Exploración Urbana**
 - Implemente una capa que muestre puntos de interés cercanos (monumentos, parques, restaurantes).
 - Añada un sistema de "zonas por descubrir" que se desbloquean al visitarlas físicamente.
 - Implemente una barra de progreso que muestre cuánto de una ciudad o zona ha explorado el usuario.
 - Ofrezca rutas sugeridas para descubrir áreas nuevas basadas en las zonas ya visitadas.
 2. **Gestor de Puntos de Interés Personalizados**
 - Implemente un sistema que permita al usuario guardar, categorizar y gestionar puntos de interés personalizados.
 - Utilice una base de datos NoSQL (como Room o Firebase) para el almacenamiento eficiente.
 - Permita el marcado rápido de ubicaciones con un toque largo en el mapa.
 - Añada opciones de categorización por tipos (favoritos, pendientes de visitar, recomendados, etc.).
 - Implemente funcionalidades de búsqueda y filtrado de puntos guardados.
 - Permita añadir fotografías y notas asociadas a cada punto.
-

Ejercicio 4: Implementación de una Funcionalidades Adicional

3. Implementación de una Funcionalidad Adicional (Elija UNA de las siguientes opciones)

Opción A: Generador de Rutas de Navegación

- Funcionalidad para calcular y mostrar rutas optimizadas entre dos puntos.
- Implementación de diferentes opciones de transporte (a pie, en bicicleta, en auto).
- Estimación de tiempo y distancia para cada ruta.

Opción B: Simulador de Tráfico y Vehículos

- Implementación de un sistema que simula movimiento de vehículos y condiciones de tráfico en tiempo real.
- Generación procedural de vehículos que siguen las calles del mapa.
- Visualización de patrones de tráfico con códigos de color (fluido, moderado, congestionado).
- Simulación de semáforos y otros elementos de control de tráfico.
- Efectos visuales como luces de vehículos en modo nocturno.

4. Evidencias:

- Capturas de pantalla y video demostrando el funcionamiento de las funcionalidades creativas.
- Diagrama de la arquitectura implementada para las funcionalidades adicionales.
- Explicación de los algoritmos o técnicas utilizadas.
- Código fuente relevante comentado.

Evidencias a entregar:

Suba las siguientes evidencias a Classroom:

1. Código fuente completo organizado y comentado adecuadamente.
2. Capturas de pantalla y videos demostrando la funcionalidad de cada ejercicio.
3. Incluya un archivo README.md en su repositorio, que contenga:
 - Descripción detallada del proyecto.
 - Instrucciones paso a paso para configurar y ejecutar la aplicación.
 - Diagrama o explicación de la arquitectura de la aplicación.
 - Capturas de pantalla relevantes mostrando la funcionalidad.
 - Explicación de los desafíos encontrados y cómo fueron resueltos.
 - Lista de las dependencias utilizadas y su propósito.
4. Repositorio de GitHub con el código fuente y la documentación.

Fecha de entrega:

- La fecha límite para la entrega de esta práctica es el **martes 8 de abril de 2025**. No se aceptarán entregas fuera de tiempo y forma.