

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

Facultad de Ingeniería en Ciencias de la Computación y Telecomunicaciones.

Sistema de Información Geográfica



“DESARROLLO DE UNA APLICACIÓN MÓVIL PARA EL CORTE PLANIFICADO DE SERVICIOS PÚBLICOS”

Grupo: K

Docente:

Ing. Ubaldo Perez Ferreira

Integrantes:

- Cambara Dorbigny Sergio David
- Copa Condo Aldair
- Iporo Chulque Jose Daniel
- Ponce Yepez Elizabet Luzia
- Terrazas Cossio Ana Belia
- Velásquez Sánchez René

Santa Cruz - Bolivia 2024

Tabla de contenido

Perfil del Proyecto	5
1. Introducción	5
2. Antecedentes	6
3. Descripción del problema	6
4. Situación problemática	6
5. Situación deseada	7
6. Objetivos	7
6.1. Objetivos Generales:	7
6.2. Objetivos Específicos:	7
7. Alcance	7
7.1. Requerimientos funcionales:	7
7.2. Requerimientos no funcionales:	8
8. Metodología (Proceso Unificado de Desarrollo de Software - PUDS)	9
8.1. Planificación	10
8.2. Análisis	10
8.3. Diseño	11
8.4. Implementación	11
8.5. Pruebas	11
9. Tecnología (Lenguajes y herramientas)	11
9.1. Herramientas	11
Capítulo I	12
Marco Teórico:	12
Flutter	12
Google Maps Api	13
Integración en Aplicaciones Móviles:	13
Rutas Óptimas:	13
Geolocalización	13
Importancia en Aplicaciones Móviles:	14
Funcionalidades Basadas en Ubicación	14
Precisión y exactitud	14
Rutas Óptimas	14

Algoritmos de Rutas Óptimas	14
Aplicación en el Proyecto	14
Capítulo II	14
2.1. Requisitos del Proyecto	14
2.2. Funcionalidad	15
2.2.1. Gestión de Cortes de Servicios	15
2.2.2. Cálculo de Rutas Óptimas	16
2.2.3. Gestión Notificaciones	17
2.2.4. Formulario de Reporte de Corte	18
2.2.5. Formulario de Feedback	18
2.2.6. Reporte de Actividad	18
2.2.7. Reporte de Uso	18
2.3. Descripción de Actores	18
2.4. Caso de usos	19
2.5. Priorizar Caso de Uso	19
2.6. Detallar Casos de Uso	19
Iteración N. 1	19
CU01 Iniciar sesión	19
CU02 Buscar Direcciones	20
CU03 Visualización de Puntos de Corte	20
Iteración N.2	21
CU4 Mapeo de Puntos de Corte	21
CU5 Opciones de Filtrado	21
CU6 Selección de Rutas	22
2.7. Modelo de Análisis	22
Capítulo III	23
Flujo de trabajo: Análisis	23
3.1. Análisis de Arquitectura	23
3.2. Identificar Paquetes y Encapsulamiento del caso de uso dentro del paquete	23
3.3. Diagrama de Actividades	24
3.3.1. Diagrama de Actividades: Inicio de Sesión	24

3.3.2. Diagrama de Actividades: Búsqueda y filtrado	24
3.3.3. Diagrama de Actividades: Registro de corte	24
Capítulo IV	25
Flujo de Trabajo: Diseño	25
4.1. Diseño de Arquitectura	25
4.1.1. Diagrama de Componentes	25
4.2. Diagrama UML	25
4.2.1. Diagrama UML parte 1: Configuración de estilos y Home Screen	25
4.2.2. Diagrama UML parte 2: Formularios y Componentes de la Home Screen	25
4.2.3. Diagrama UML parte 3: Geolocalización y Otros Componentes	25
4.3. Resultados del Proyecto	25
4.3.1. Funcionalidad	25
4.3.2. Formularios de Interacción	26
4.3.3. Reportes	27
Conclusiones	27
Recomendación	28
Bibliografía	28
Anexo	28
Manual de Usuario	28

Perfil del Proyecto

1. Introducción

Con el paso de los años, los dispositivos móviles han avanzado significativamente en eficiencia. Un ejemplo destacado es la aplicación Maps de Google, que es una herramienta de búsqueda de ubicaciones que permite geolocalizar un punto concreto, calcular rutas, encontrar los lugares de interés más cercanos o ver la apariencia de un lugar a pie de calle a través de Google Street View. Fue desarrollada en 2005, inicialmente para Internet Explorer y Mozilla Firefox en PC. Sin embargo, su uso se ha generalizado de la mano de los smartphones.

El GPS facilita la planificación de rutas desde nuestro lugar de inicio hasta nuestro destino. Google Maps realiza esta tarea, ofreciendo rutas para caminar, conducir, andar en bicicleta y usar el transporte público.

La aplicación que presentaremos permite planificar la mejor ruta entre varios puntos, facilitando a la persona encargada de realizar el corte de servicio de Agua haga el uso eficiente de esta herramienta.

En esta aplicación, podremos ver la lista de usuarios programados para el corte de servicio, así como la ubicación exacta de los medidores, incluyendo su latitud y longitud. Además, la app nos mostrará la mejor ruta para visitar varios puntos de manera eficiente.

Esta herramienta cuenta con una base de datos modesta, acomodada y exclusiva en la Localidad de San Ignacio de Velasco, una colección tremenda de puntos que marcan los diferentes domicilios de los usuarios. Con una opción de búsqueda por zona o por nombre.

Una aplicación cuidadosamente desarrollada, diseñada para quienes necesitan asistencia en la planificación de recorridos entre diversos puntos asignados. Muestra de manera ordenada los diferentes destinos y calcula la mejor ruta para evitar desviaciones, ahorrando tiempo y combustible.

2. Antecedentes

La Cooperativa de Servicios de San Ignacio de Velasco (COOSIV RL) enfrenta desafíos significativos en la gestión de cortes de servicios públicos. Tradicionalmente, los procesos de corte se realizaban de manera manual, lo que generaba ineficiencias en:

- Planificación de rutas
- Registro de información
- Control de tiempo y recursos
- Seguimiento de actividades

Actualmente, la cooperativa requiere una solución tecnológica que optimice el proceso de cortes de servicios, reduciendo los tiempos de desplazamiento y mejorando la precisión en el registro de información.

3. Descripción del problema

A la hora de trabajar sobre mapas no hay mejores herramientas que Google Maps, Street View y ArcGis entre otras. Pero cuando se trata sobre manejo de mapas para aplicaciones móviles estas potentes herramientas nos ofrecen sus librerías para la programación en Android.

El proceso comienza con la obtención de una aplicación que muestre y organice los diferentes puntos, teniendo en cuenta que se trata de muchos usuarios. Esta herramienta optimiza las rutas para minimizar el esfuerzo del encargado del corte y reducir el desperdicio de tiempo.

La aplicación es extremadamente intuitiva y fácil de usar, de modo que cualquier persona, sin excepción, podrá comprender y manejar su funcionamiento sin dificultad.

4. Situación problemática

La gestión actual de cortes de servicios públicos en San Ignacio de Velasco presenta ineficiencias críticas: los técnicos de COOSIV RL realizan desplazamientos sin optimización, perdiendo aproximadamente 2 horas diarias en rutas ineficientes, consumiendo combustible

innecesariamente y registrando manualmente información propensa a errores. La falta de una herramienta tecnológica sistematizada genera sobrecostos operativos, imprecisión en los registros y dificultad para realizar un seguimiento efectivo de las actividades de corte de servicios.

5. Situación deseada

La solución tecnológica propuesta busca transformar radicalmente este escenario mediante una aplicación móvil que optimiza automáticamente rutas, permite el registro digital preciso de cortes, calcula tiempos de desplazamiento, genera trazabilidad en tiempo real y reduzca hasta un 40% los costos operativos, convirtiendo un proceso manual y dispendioso en una operación eficiente, tecnológicamente avanzada y estratégicamente planificada.

6. Objetivos

6.1. Objetivos Generales:

Desarrollar una aplicación móvil que permita a los usuarios reportar y recibir información sobre cortes de servicios públicos, utilizando la tecnología de consumo de web services y la optimización de rutas de Google Maps para mejorar la comunicación y la atención al cliente en este sector.

6.2. Objetivos Específicos:

- Conectar la aplicación con los sistemas de información de las empresas de servicios públicos para obtener datos actualizados sobre el estado de los cortes.
- Calcular la ruta más rápida y eficiente para que los técnicos de las empresas de servicios públicos lleguen a los lugares donde se han reportado cortes.

7. Alcance

7.1. Requerimientos funcionales:

Búsqueda de lugares:

- Los usuarios pueden buscar lugares que pertenezcan a la UAGRM.
- Los usuarios pueden ingresar palabras clave o sugerencias para buscar lugares específicos.
- Los usuarios pueden ver detalles adicionales de los lugares encontrados, como la descripción, las fotos y la ubicación en el mapa.

Visualización en el mapa:

- Los usuarios pueden ver los lugares encontrados en un mapa interactivo.
- Los lugares se muestran con marcadores o iconos en el mapa.
- Los usuarios pueden hacer zoom, desplazarse y explorar el mapa para encontrar lugares específicos.

7.2. Requerimientos no funcionales:

Usabilidad:

- La interfaz de usuario debe ser intuitiva y fácil de usar para los usuarios.
- La aplicación debe ser compatible con diferentes tamaños de pantalla y dispositivos móviles.
- El tiempo de respuesta de la aplicación debe ser rápido para una experiencia de usuario fluida.

Rendimiento:

- La aplicación debe cargar y responder rápidamente, minimizando los tiempos de espera.
- La aplicación debe manejar eficientemente grandes volúmenes de datos y mantener un rendimiento óptimo incluso en condiciones de carga alta.

Disponibilidad:

- La aplicación debe estar disponible y accesible para los usuarios en todo momento, con un tiempo de inactividad mínimo para mantenimiento o actualizaciones planificadas.

8. Metodología (Proceso Unificado de Desarrollo de Software - PUDS)

La metodología utilizada en la creación de esta app es una de las más conocidas que el Proceso Unificado de Desarrollo de Software nos ofrece y es nada más y nada menos que el Modelo en Cascada.

Es un enfoque tradicional en el desarrollo de software que sigue un proceso secuencial y lineal. Es uno de los primeros métodos utilizados en ingeniería de software y se caracteriza por su estructura rígida y etapas claramente definidas. En este modelo, el desarrollo del software se divide en fases que se completan una tras otra. La primera descripción formal del modelo de cascada se cita a menudo a un artículo publicado por Winston Royce W. en 1970, aunque Royce no utiliza el término "cascada" de este artículo.

El modelo de cascada define las siguientes etapas que deben cumplirse de forma sucesiva:

- Requisitos: Se recopilan y documentan todos los requisitos del sistema.
- Diseño: Se elabora un diseño detallado basado en los requisitos recopilados.
- Implementación: Se desarrolla el software siguiendo el diseño especificado.
- Verificación: Se prueba el software para asegurarse de que cumple con los requisitos y funciona correctamente.
- Mantenimiento: Se realizan ajustes y correcciones después de que el software ha sido entregado y está en uso.

Cada fase debe completarse antes de pasar a la siguiente, y generalmente no se permite volver a fases anteriores. Este enfoque es adecuado para proyectos con requisitos

bien definidos y cambios mínimos durante el desarrollo, pero puede ser menos flexible frente a cambios y adaptaciones en proyectos más complejos o en evolución.

8.1. Planificación

Objetivo: Definir el alcance y viabilidad del proyecto

Actividades:

- Identificar necesidades del cliente
- Definir objetivos del sistema
- Estimar recursos
- Calcular presupuesto
- Establecer cronograma

8.2. Análisis

Objetivo: Recopilar y documentar requisitos detallados

Actividades:

- Entrevistas con stakeholders
- Definir requisitos funcionales
- Definir requisitos no funcionales
- Crear casos de uso
- Modelar procesos de negocio

Entregables:

- Especificación de Requisitos de Software (ERS)
- Diagramas de casos de uso
- Matriz de requisitos

8.3. Diseño

Objetivo: Crear la arquitectura del sistema

Actividades:

- Diseño arquitectónico
- Diseño de base de datos
- Diseño de interfaces
- Definir tecnologías
- Crear diagramas estructurales

Entregables:

- Diagrama de arquitectura
- Modelo de datos
- Diseño de interfaces de usuario
- Especificaciones técnicas

8.4. Implementación

Objetivo: Desarrollar el software según diseño

Actividades:

- Configurar entorno de desarrollo
- Codificar módulos
- Implementar base de datos
- Desarrollar funcionalidades
- Seguir estándares de codificación

Entregables:

- Código fuente

- Base de datos implementada
- Módulos funcionales

8.5. Pruebas

Objetivo: Verificar funcionamiento y calidad

Tipos de Pruebas:

- Pruebas unitarias
- Pruebas de integración
- Pruebas de sistema
- Pruebas de aceptación
- Pruebas de rendimiento

Actividades:

- Diseñar casos de prueba
- Ejecutar pruebas
- Reportar y corregir defectos
- Validar requisitos

Entregables:

- Informe de pruebas
- Registro de defectos

9. Tecnología (Lenguajes y herramientas)

9.1. Herramientas

9.1.1. Entornos de desarrollo integrados (IDE)

Visual Studio Code: Editor de código fuente desarrollado por Microsoft que admite el desarrollo multiplataforma, incluyendo aplicaciones móviles para Android e iOS.

9.1.2. Lenguaje de Programación

Dart: Lenguaje de programación utilizado por Flutter. Dart es un lenguaje moderno y orientada a objetos con características como tipado estático opcional, recolección de basura y un enfoque en la creación de interfaces de usuarios reactivas.

9.1.3. Frameworks Y SDK (Software Development Kit)

Flutter: Framework de Desarrollo de aplicaciones móviles multiplataforma utilizando Dart.

Android SDK: Conjunto de herramientas y bibliotecas proporcionadas por Google, que permiten el desarrollo nativo de aplicaciones móviles.

Librerías utilizadas en Flutter:

- geolocator: ^10.1.0
- google_maps_flutter: ^2.5.0
- google_polyline_algorithm: ^3.1.0
- image: ^4.1.3
- image_gallery_saver: ^2.0.3
- intl: ^0.18.1
- path_provider: ^2.1.1
- permission_handler: ^11.0.0
- photo_view: ^0.14.0
- record: ^4.4.4
- shared_preferences: ^2.2.2
- timezone: ^0.9.2
- url_launcher: ^6.2.1
- web_socket_channel: ^2.1.0
- rxdart: ^0.27.0

- xml: ^6.0.1
- xml2json: ^5.3.0

9.1.4. Sistema de Gestor de Base de Datos

FireBase: Plataforma de desarrollo de aplicaciones móviles de Google que proporciona servicios como autenticación, base de datos en tiempo real, notificación push.

9.1.5. Herramientas de control de versiones

GitHub: Plataforma basada en Git que facilita la colaboración y el trabajo en equipo en proyectos de desarrollo de software.

9.1.6. Herramientas de gestión de proyectos

Trello: Herramienta de gestión de proyectos basada en tableros que permite organizar y dar seguimiento al trabajo en equipo.

9.1.7. Herramienta de modelado y diseño de software

Enterprise Architect: Es una herramienta de modelado visual que permite a los equipos de desarrollo y a los arquitectos de software crear, documentar y comunicar modelos y diseños de sistemas de manera eficiente.

Capítulo I

Marco Teórico

Flutter

Flutter es un framework de desarrollo de aplicaciones de código abierto creado por Google que permite construir aplicaciones nativas para múltiples plataformas desde una única base de código. Se basa en el lenguaje de programación Dart.

Google Maps Api

La API de Google Maps se integra en Flutter a través del paquete `google_maps_flutter`, que permite incluir mapas interactivos en la aplicación y utilizar funcionalidades de navegación.

Integración en Aplicaciones Móviles:

Visualización de Mapas: El paquete `google_maps_flutter` permite mostrar mapas en la aplicación, añadiendo marcadores, polígonos y líneas para representar ubicaciones y rutas.

Rutas Óptimas:

Matrix API es una herramienta que te permite obtener información sobre la distancia, el tiempo de viaje y las rutas entre varios puntos de origen y destino.

Web Services

Los web services son componentes de software que se comunican a través de protocolos estándar (como HTTP, SOAP, REST) para permitir la interoperabilidad entre sistemas diferentes. En el contexto de tu aplicación:

Servicios RESTful: Usan HTTP para realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar) y suelen ofrecer datos en formato JSON o XML. Son ampliamente utilizados debido a su simplicidad y escalabilidad.

APIs: Las APIs (Interfaces de Programación de Aplicaciones) permiten que diferentes aplicaciones se comuniquen y compartan datos.

Geolocalización

La geolocalización representa un conjunto de tecnologías y metodologías que permiten determinar la ubicación geográfica de un dispositivo mediante múltiples sistemas de posicionamiento como GPS, redes móviles, señales Wi-Fi y sensores integrados. Su complejidad radica en la combinación de múltiples técnicas para lograr una precisión óptima, considerando factores como la densidad de infraestructura, condiciones atmosféricas y características del entorno urbano o rural.

Importancia en Aplicaciones Móviles:

Funcionalidades Basadas en Ubicación

Transforma aplicaciones estáticas en experiencias interactivas, también permite personalizar según la ubicación física y mejoran la toma de decisiones en tiempo real.

Precisión y exactitud

Las Tecnologías de Geolocalización pueden ser:

- GPS (Sistema de posicionamiento global)
- Triangulación de torres de telefonía
- Redes WiFi
- Servicios de mapeo

Métodos de precisión

1. Geolocalización Múltiple

- a. Combinación de diferentes tecnologías
- b. Reducción de márgenes de error
- c. Mejora la precisión en entornos urbanos y rurales

2. Algoritmos de corrección

- a. Filtros de Kalman
- b. Compensación de señales débiles
- c. Eliminación de interferencias

Los Desafíos de precisión son las zonas urbanas densas, las interferencias de edificios, las condiciones climáticas y las limitaciones de hardware.

Rutas Óptimas

Los algoritmos de rutas óptimas son métodos computacionales diseñados para encontrar el camino más eficiente entre dos puntos en un gráfico o red, minimizando un criterio específico como distancia, tiempo, costo o una combinación de estos.

Componentes de optimización

1. Variables de evaluación

- Distancia física
- Tiempo de desplazamiento
- Consumo de combustible
- Restricciones de tráfico
- Condiciones del terreno

2. Técnicas de cálculo

- Algoritmo de Dijkstra
- A* (Estrella A)
- Floyd-Warshall
- Algoritmos genéticos

Algoritmos de Rutas Óptimas

1. Algoritmo de Dijkstra

```
def dijkstra(grafo, inicio, fin):  
    distancias = {nodo: float('inf') for nodo in grafo}  
    distancias[inicio] = 0  
    pendientes = [(0, inicio)]  
  
    while pendientes:  
        distancia_actual, nodo_actual = heapq.heappop(pendientes)  
  
        if nodo_actual == fin:
```

```

        return distancia_actual

    for vecino, peso in grafo[nodo_actual].items():

        distancia = distancia_actual + peso

        if distancia < distancias[vecino]:

            distancias[vecino] = distancia

            heapq.heappush(pendientes, (distancia, vecino))

    return float('inf')

```

2. Algoritmo A*

```

def a_star(grafo, inicio, fin):

    frontera = PriorityQueue()

    frontera.put((0, inicio))

    came_from = {}

    costo_actual = {inicio: 0}

    while not frontera.empty():

        actual = frontera.get()[1]

        if actual == fin:

            break

        for siguiente in grafo[actual]:

            nuevo_costo = costo_actual[actual] + grafo[actual][siguiente]

            if siguiente not in costo_actual or nuevo_costo <
costo_actual[siguiente]:

                costo_actual[siguiente] = nuevo_costo

                prioridad = nuevo_costo + heuristica(fin, siguiente)

```

```
frontera.put((prioridad, siguiente))

came_from[siguiente] = actual

return reconstruir_ruta(came_from, inicio, fin)
```

Aplicación en el Proyecto

En el contexto específico de una aplicación para cortes de servicios públicos, estas tecnologías se integran para crear una solución que optimice los recorridos de técnicos, registre geográficamente las actividades, minimice recursos operativos y proporciona trazabilidad detallada de cada intervención, transformando un proceso según manual en una operación tecnológicamente eficiente y estratégicamente planificada.

Capítulo II

2.1.Requisitos del Proyecto

A partir del documento, resumiré los requisitos funcionales clave para la aplicación móvil para cortes de servicios públicos planificados:

Autenticación y configuración inicial

1. Funcionalidad de inicio de sesión donde el usuario se autentica en la aplicación móvil
2. Importar el listado de usuarios/puntos de servicio a cortar desde el sistema ERP de la empresa

Planificación de rutas y navegación

1. Generar una ruta óptima en el mapa de la ciudad para cortes de servicios
2. El punto de partida siempre es la oficina de la empresa.
3. Optimización de ruta basada en:
 - Tiempo o distancia mínima
 - Líneas azules que representan la ruta más corta
 - Considerando 10 minutos por punto de corte

- Velocidad de desplazamiento de la motocicleta de 15 km/h.
- 4. Mostrar el tiempo total y la distancia de la ruta
- 5. Utilice la API de Google Maps o una herramienta similar para calcular la ruta

Proceso de corte de servicio

1. Cada punto de corte marcado con un marcador rojo en el mapa
2. Al tocar un marcador aparece el formulario de registro de corte.
3. Después de cortar, los puntos cambian de color:
 - Verde: corte exitoso
 - Naranja: Visitado pero no cortado
4. Capacidad de tomar fotografías de medidores.
5. Registre la lectura del medidor o la observación si la lectura no es posible

Gestión de datos

1. Importar listas de cortes desde el servidor
2. Exportar cortes completados al servidor mediante servicios web específicos:
 - W2Corte_ReporteCortesParaSubirSIG(1,0,0)
3. Ver lista de cortes completados
4. Almacenamiento de base de datos local en dispositivo móvil

2.2.Funcionalidad

Los requisitos funcionales clave centrándose en la gestión de cortes de servicio, el cálculo de rutas óptimas y la gestión de notificaciones.

2.2.1. Gestión de Cortes de Servicios

Autenticación y configuración de usuarios

- Autenticación de usuarios en la aplicación móvil
- Importar lista de usuarios/puntos de servicio desde el sistema ERP de la empresa

- Definir zona de corte asignada para cada usuario

Proceso de registro de cortes

- Seleccionar los puntos de servicio que se van a cortar
- Formulario de inscripción de corte detallado con:
 - Entrada de lectura del medidor
 - Opción para agregar observaciones
 - Capacidad de tomar fotografías del medidor.
- Marcar puntos de corte en el mapa:
 - Marcador rojo: corte pendiente
 - Marcador verde: corte realizado con éxito
 - Marcador naranja: Visitado pero no cortado

Gestión de datos

- Almacenamiento de base de datos de dispositivos móviles locales
- Capacidades de sincronización del servidor
 - Importar listas de cortes desde el servidor
 - Exportar cortes completados mediante servicios web
- Ver y administrar la lista de cortes completados

2.2.2. Cálculo de Rutas Óptimas

Características de la planificación de rutas

- Punto de partida: Oficina de la empresa
- Criterios de optimización:
 - Tiempo mínimo
 - Distancia mínima
- Funciones de visualización de ruta:
 - Líneas azules que indican el camino más corto

- Puntos de corte numerados (1 a n)
- Flechas direccionales que muestran la ruta

Parámetros de cálculo de ruta

- Modo de viaje: Motocicleta
- Velocidad media: 15 km/h
- Tiempo por punto de corte: 10 minutos
- Herramientas recomendadas:
 - API de Google Maps
 - Servicios de mapeo alternativos

Visualización de información de ruta

- Tiempo total de ruta
- Distancia total de la ruta
- Número de puntos de corte
- Número de puntos ya recortados

2.2.3. Gestión Notificaciones

Notificaciones del sistema

- Alertas para:
 - Recortes de servicios pendientes
 - Rutas incompletas
 - Estado de sincronización
- Actualizaciones en tiempo real sobre el proceso de corte

Notificaciones de interacción del usuario

- Mensajes de confirmación para:
 - Registro de corte exitoso
 - Sincronización del servidor

- Finalización de ruta
- Mensajes de advertencia para:
 - No se puede leer el medidor
 - Problemas de GPS/ubicación
 - Problemas de conectividad de red

Informes y seguimiento

- Generar informes de corte
- Seguimiento del rendimiento del usuario
- Monitorizar la eficiencia de la ruta
- Proporcionar un resumen de los cortes diarios/semanales

2.2.4. Formulario de Reporte de Corte

Objetivo: Registre información detallada y precisa de cada corte de servicio realizado.

En el registro tendrá

Código de Lector

Fecha y Hora

Identificador Único de Corte

La Ubicación que puede ver sera:

Dirección Completa

Coordenadas GPS

Referencia de Ubicación

Los datos del medidor:

Número de Medidor

Lectura Actual

Estado del Medidor

Foto del Medidor

Podrá ver detalladamente el corte

Tipo de Corte (Programado/Emergencia)

Motivo del Corte

Observaciones

Tiempo de Ejecución

El estado de los cortes:

Corte Realizado (Sí/No)

Razones de No Realización

Acciones Adicionales

2.2.5. Formulario de Feedback

Objetivo: Recopilar retroalimentación del proceso de corte y experiencia del usuario.

Tendremos información del lector, datos personales y laborales e identificación de la jornada.

La evaluación del proceso de este es la dificultad del recorrido, el tiempo de trabajo y las condiciones de acceso.

Los problemas encontrados, las sugerencias de mejora y los obstáculos en la ruta.

La satisfacciones serían la evaluación de herramientas y los comentarios cualitativos

2.2.6. Reporte de Actividad

Objetivo: Generar un resumen detallado de las actividades realizadas, como ser la fecha del reporte, el periodo de análisis y el total de cortes realizados

Las estadísticas de cortes pueden ser cortes programados, cortes ejecutados, cortes pendientes. La distribución geográfica será por cortes por zona, las rutas cubiertas y el tiempo promedio por ruta.

2.2.7. Reporte de Uso

Objetivo: Analizar la utilización de la aplicación móvil.

El reporte tendrá el número total de usuarios, la frecuencia de Uso y analizar el tiempo promedio de sesión.

el uso de funcionalidades abarcara la función más utilizada, la función menos utilizada y el tiempo de uso por función.

El rendimiento de la aplicación tendrá el tiempo de carga, errores reportados y las actualizaciones realizadas.

2.3. Descripción de Actores

Usuario Registrado

- Puede iniciar sesión
- Accede a funcionalidades principales
- Puede crear y gestionar proyectos

Administrador

- Gestiona usuarios
- Configura parámetros del sistema
- Acceso total a funcionalidades

2.4. Caso de usos

CU01 Iniciar Sesión

CU02 Buscar Direcciones

CU03 Visualización Puntos Corte

CU04 Mapeo Puntos de Corte

CU05 Opciones Filtrado

CU06 Selección Rutas

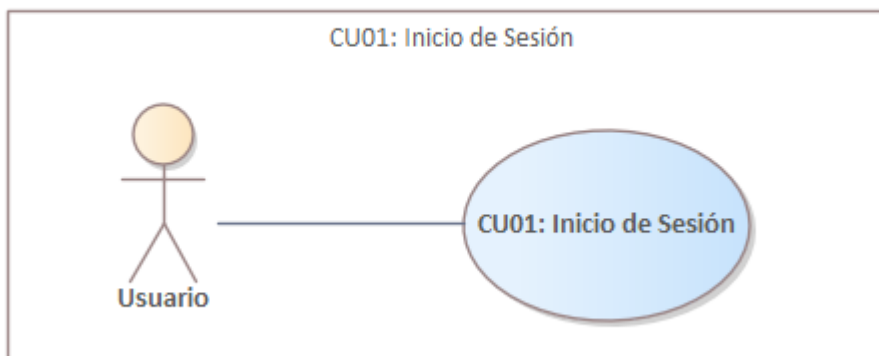
2.5. Priorizar Caso de Uso

Caso de Uso	Prioridad	Complejidad	Importancia
CU01 Iniciar Sesión	Alta	Baja	Crítico
CU02 Buscar Direcciones	Media	Media	Importante
CU03 Visualización Puntos Corte	Alta	Alta	Crítico
CU04 Mapeo Puntos de Corte	Media	Alta	Importante
CU05 Opciones Filtrado	Baja	Baja	Complementario
CU06 Selección Rutas	Media	Media	Importante

2.6. Detallar Casos de Uso

Iteración N. 1

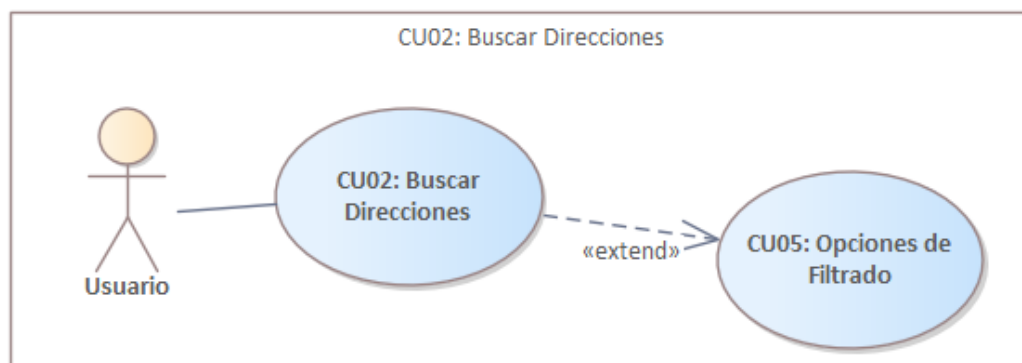
CU01 Iniciar sesión



CASO DE USO	CU01: Iniciar Sesión
--------------------	-----------------------------

PROPÓSITO	Autenticar usuarios en el sistema
RESUMEN	Inicia sesión, y puede Acceder a funcionalidades principales.
ACTORES	Usuario, Administrador
PRE CONDICIÓN	<ul style="list-style-type: none"> • Usuario registrado • Credenciales válidas
FLUJO DE SUCESO	<ol style="list-style-type: none"> 1. Usuario ingresa correo 2. Usuario ingresa contraseña 3. Sistema valida credenciales 4. Sistema redirige a dashboard
POST CONDICIÓN	
EXCEPCIÓN	Flujos Alternativos: <ul style="list-style-type: none"> • Credenciales incorrectas • Recuperar contraseña

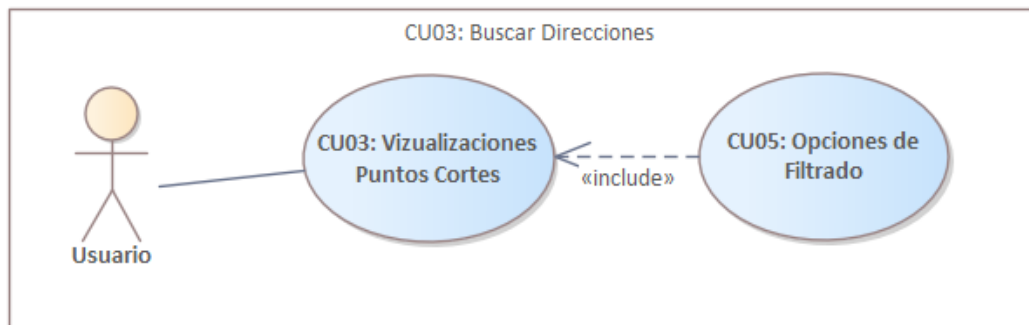
CU02 Buscar Direcciones



CASO DE USO	CU02: Buscar Direcciones
PROPÓSITO	Localizar direcciones específicas
RESUMEN	Puede buscar la dirección específica en el sistema.
ACTORES	Usuario
PRE CONDICIÓN	<ul style="list-style-type: none"> • Usuario autenticado • Acceso a sistema de búsqueda
FLUJO DE SUCESO	Flujo Principal: <ol style="list-style-type: none"> 1. Usuario accede a módulo de búsqueda 2. Ingresa términos de búsqueda 3. Sistema muestra resultados 4. Usuario selecciona dirección
POST CONDICIÓN	<ul style="list-style-type: none"> • Acceder al Sistema

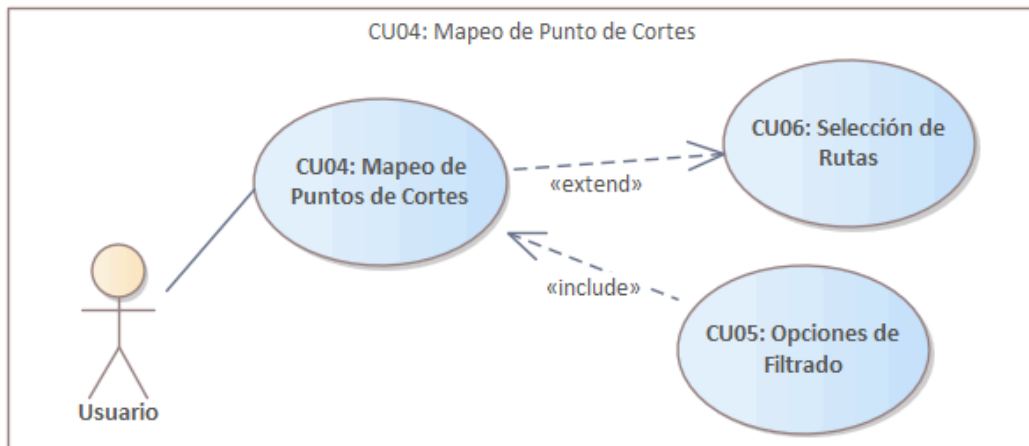
	<ul style="list-style-type: none"> • Ser Administrador
EXCEPCIÓN	Flujos Alternativos: <ul style="list-style-type: none"> • Sin resultados • Filtrar resultados

CU03 Visualización de Puntos de Corte



CASO DE USO	CU03: Visualización de Puntos de Corte
PROPÓSITO	Mostrar de manera clara y detallada los puntos de corte existentes en el sistema
RESUMEN	Puede lograr ver los puntos de corte detallados.
ACTORES	Administrador
PRE CONDICIÓN	<ul style="list-style-type: none"> • Usuario autenticado • Permisos de acceso a visualización • Existencia de puntos de corte registrados
FLUJO DE SUCESO	Flujo Principal: <ol style="list-style-type: none"> 1. Usuario accede a la sección de puntos de corte 2. Sistema recupera puntos de corte disponibles 3. Muestra lista/mapa de puntos de corte 4. Permite interacción con cada punto (detalles, información)
POST CONDICIÓN	<ul style="list-style-type: none"> • Puntos de corte visualizados correctamente • Información detallada accesible
EXCEPCIÓN	Flujos Alternativos: <ul style="list-style-type: none"> • Sin puntos de corte registrados • Filtrar puntos de corte • Visualización en diferentes formatos (lista, mapa, gráfico)

CU4 Mapeo de Puntos de Corte



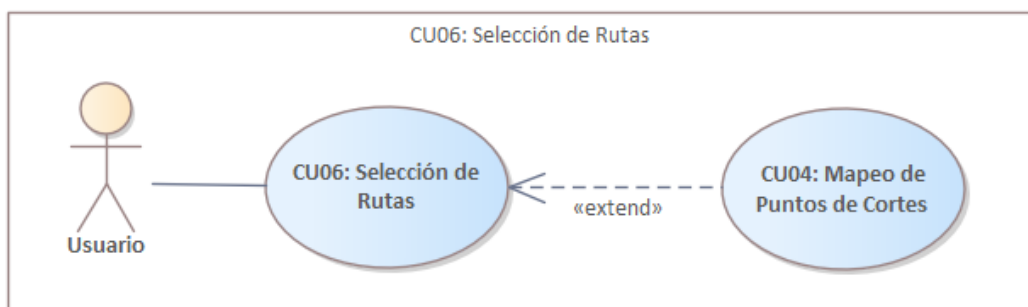
CASO DE USO	CU4: Mapeo de Puntos de Corte
PROPÓSITO	Generar un mapeo detallado y preciso de los puntos de corte.
RESUMEN	Tendrá las funcionalidades como: <ul style="list-style-type: none">● Zoom in/out● Capas de información● Detalle de cada punto● Mediciones y distancias● Exportación de mapeo
ACTORES	Usuario Registrado
PRE CONDICIÓN	<ul style="list-style-type: none">● Usuario autenticado● Acceso a herramientas de mapeo● Datos de puntos de corte disponibles
FLUJO DE SUCESO	Flujo Principal: <ol style="list-style-type: none">1. Usuario selecciona opción de mapeo2. Sistema carga herramienta de mapeo3. Presenta puntos de corte en mapa interactivo4. Permite manipulación y análisis de puntos
POST CONDICIÓN	<ul style="list-style-type: none">● Mapeo generado correctamente● Información detallada disponible
EXCEPCIÓN	Flujos Alternativos: <ul style="list-style-type: none">● Cambiar tipo de mapa● Ajustar visualización● Generar reportes

CU5 Opciones de Filtrado

CASO DE USO	CU05: Opciones de Filtrado
PROPÓSITO	Proporcionar opciones avanzadas de filtrado para puntos de corte

RESUMEN	Combinación de múltiples filtros, guardar configuraciones de filtro, exportar resultados filtrados. Podremos ver opciones de Filtrado: <ul style="list-style-type: none"> a. Por ubicación geográfica b. Por tipo de punto de corte c. Por estado/condición d. Por fecha de registro e. 5. Por características específicas
ACTORES	Usuario Registrado
PRE CONDICIÓN	<ul style="list-style-type: none"> ● Usuario autenticado ● Puntos de corte existentes ● Acceso a herramientas de filtrado
FLUJO DE SUCESO	Flujo Principal: <ol style="list-style-type: none"> 1. Usuario accede a panel de filtros 2. Selecciona criterios de filtrado 3. Sistema aplica filtros 4. Muestra resultados filtrados
POST CONDICIÓN	<ul style="list-style-type: none"> ● Filtros aplicados correctamente ● Resultados precisos
EXCEPCIÓN	Flujos Alternativos: <ul style="list-style-type: none"> ● Resetear filtros ● Filtrado avanzado ● Personalización de vista

CU6 Selección de Rutas



CASO DE USO	CU06: Selección de Rutas
PROPÓSITO	Permitir la selección y análisis de rutas basadas en puntos de corte
RESUMEN	Sus funcionalidades que tendrá: <ol style="list-style-type: none"> a. Selección de ruta origen/destino b. Análisis de puntos de corte en ruta c. Cálculo de distancias d. Estimación de tiempos

	e. Alternativas de ruta
ACTORES	Usuario Registrado
PRE CONDICIÓN	<ul style="list-style-type: none"> • Usuario autenticado • Puntos de corte disponibles • Rutas registradas en el sistema
FLUJO DE SUCESO	Flujo Principal: <ol style="list-style-type: none"> 1. Usuario accede a módulo de rutas 2. Sistema muestra rutas disponibles 3. Permite selección y análisis de rutas 4. Visualización detallada de ruta
POST CONDICIÓN	<ul style="list-style-type: none"> • Ruta seleccionada correctamente • Información detallada disponible
EXCEPCIÓN	Flujos Alternativos: <ul style="list-style-type: none"> • Generar nueva ruta • Comparar múltiples rutas • Exportar información de ruta

2.7. Modelo de Análisis

Capítulo III

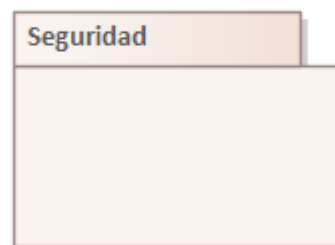
Flujo de trabajo: Análisis

3.1. Análisis de Arquitectura

3.2. Identificar Paquetes

Paquete Gestionar Seguridad

Este paquete gestiona todo lo relacionado con los usuarios del sistema, como ser la Autenticación Usuario, el Gestor Usuarios, Control de acceso y Perfil Usuario



Paquete de Gestión de Cortes

Este paquete se encarga de gestionar las Cortes de Corte, Puntos de Corte, la Optimización de Rutas, Calcular la distancia de Mapa Rutas

Gestion de Cortes

Paquete de Gestión de Informes

Este paquete gestiona todo lo relacionado con los informes como ser generando estos, analizador estadístico, visualizador de datos y exportador Reportes.

Interfaz Usuario

3.3.Encapsulamiento del caso de uso dentro del paquete

pkg Paquete de Seguridad

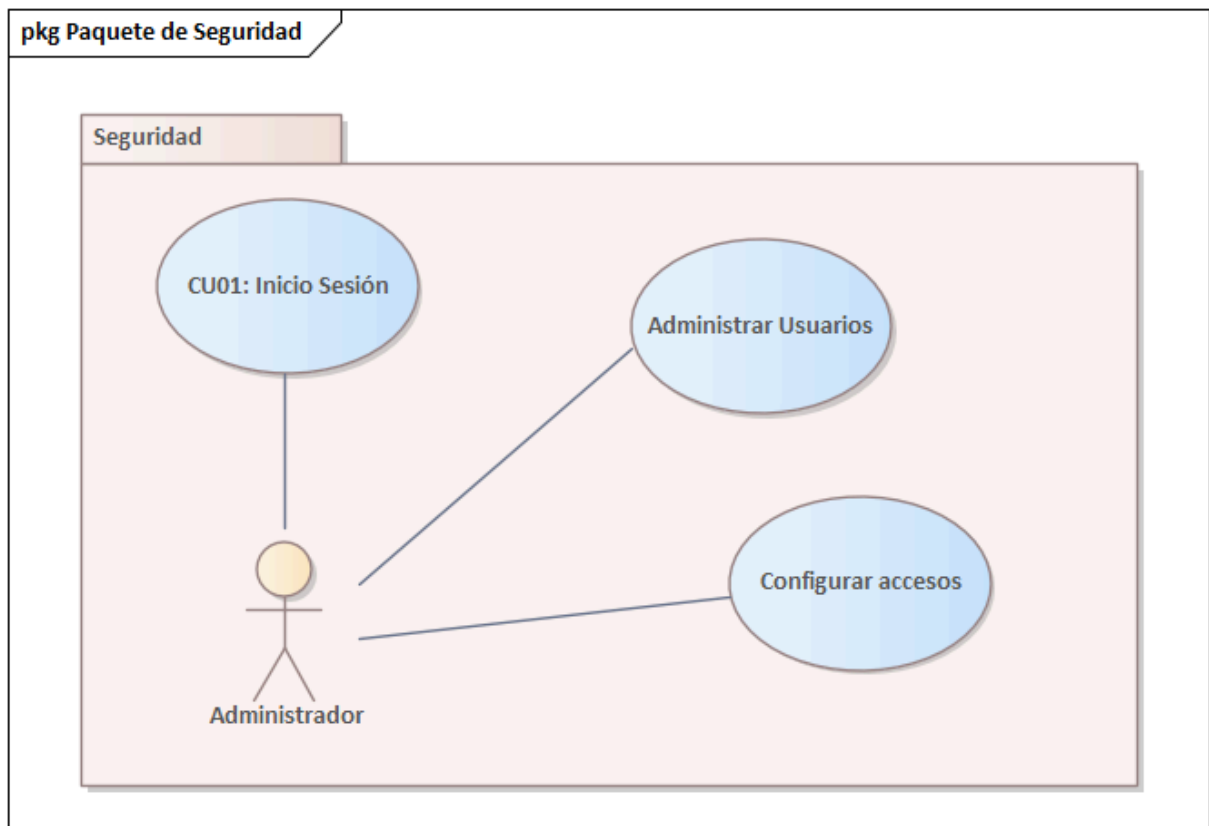
Seguridad

CU01: Inicio Sesión

Administrar Usuarios

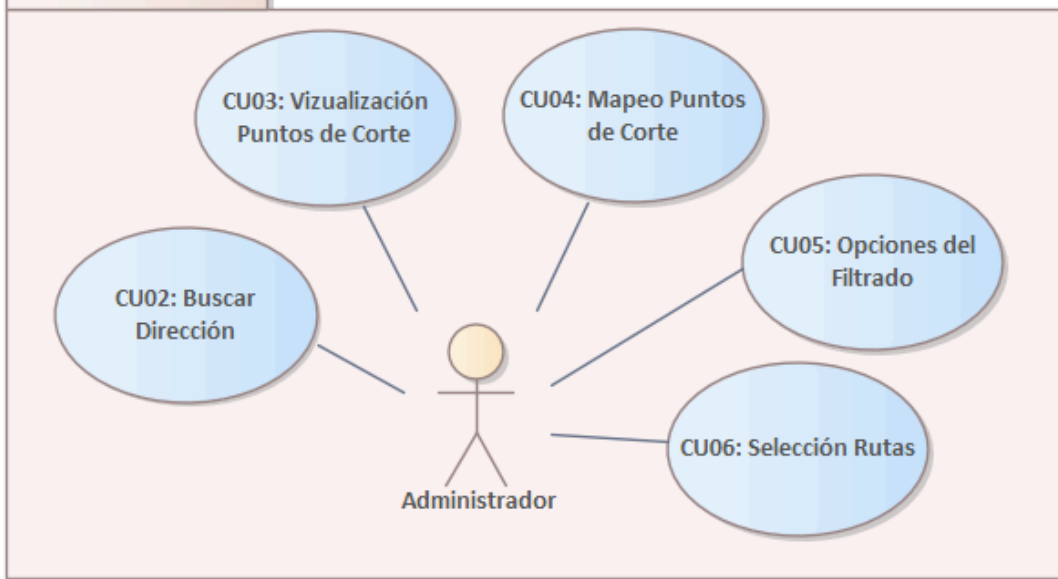
Configurar accesos

Administrador



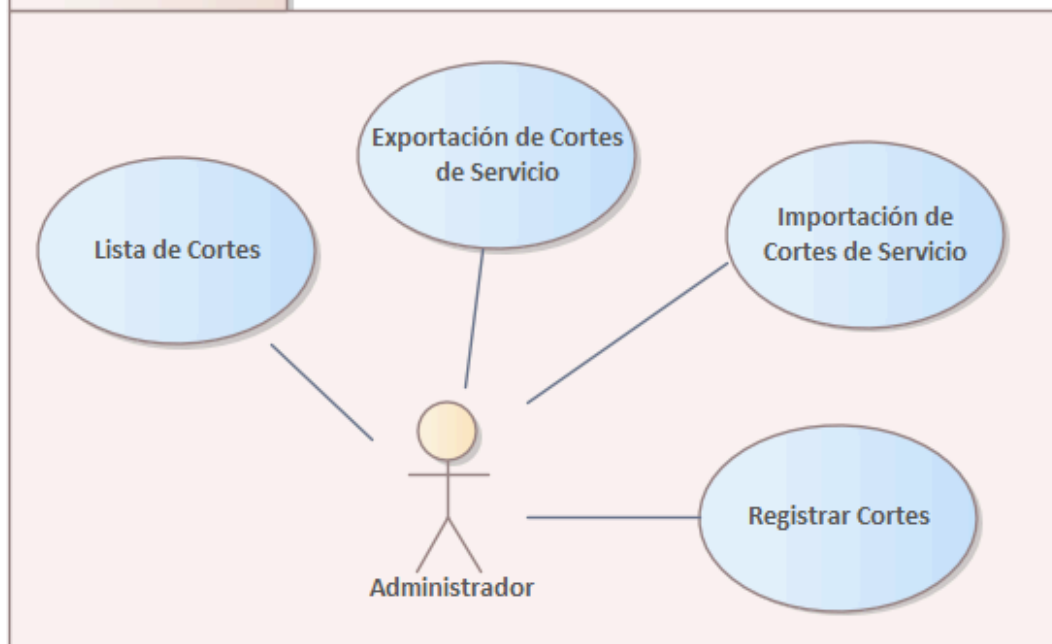
pkg Paquete de Cortes

Gestion de Cortes



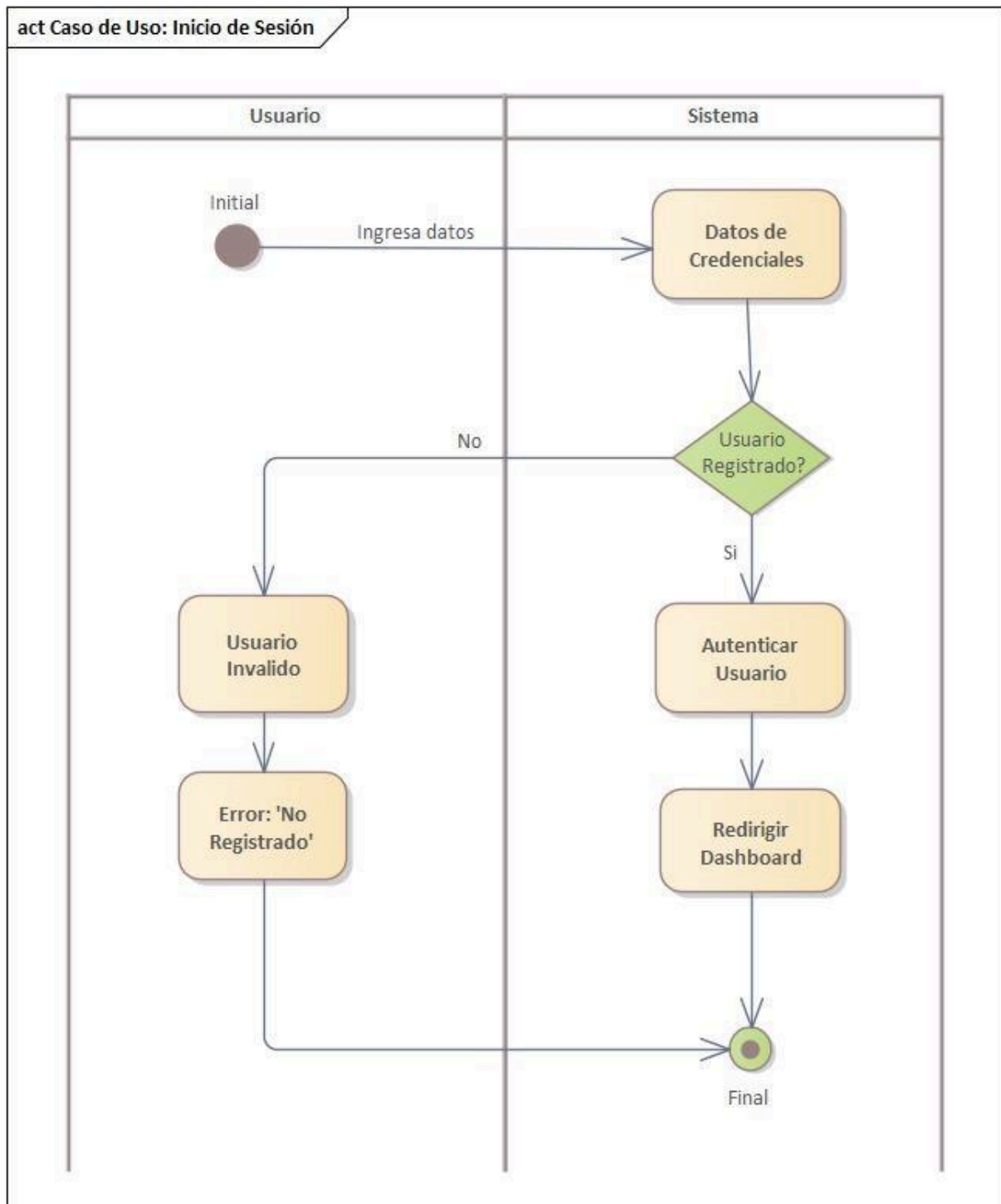
pkg Paquete de Informe

Gestión Informe



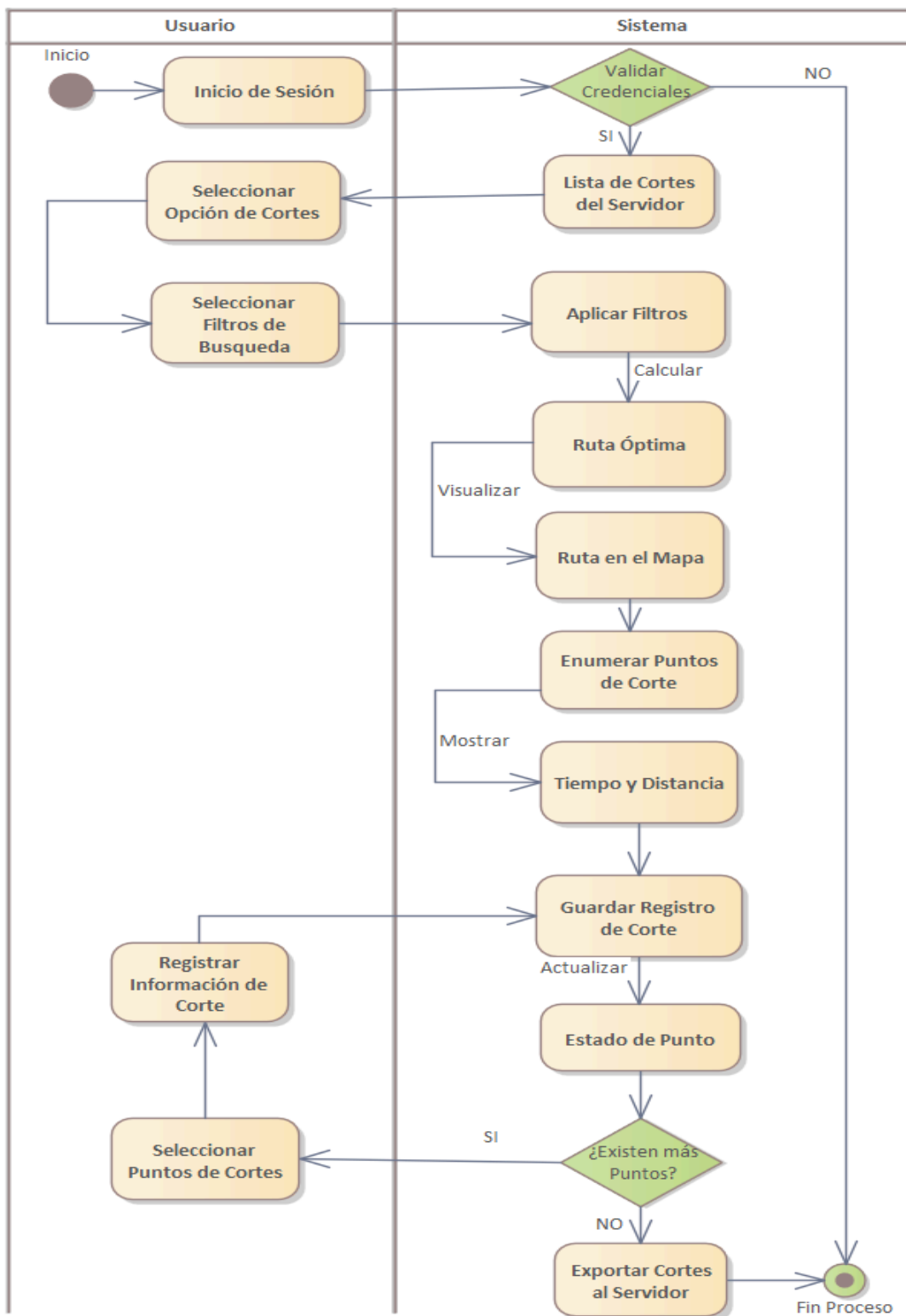
3.4. Diagrama de Actividades

3.3.1. Diagrama de Actividades: Inicio de Sesión

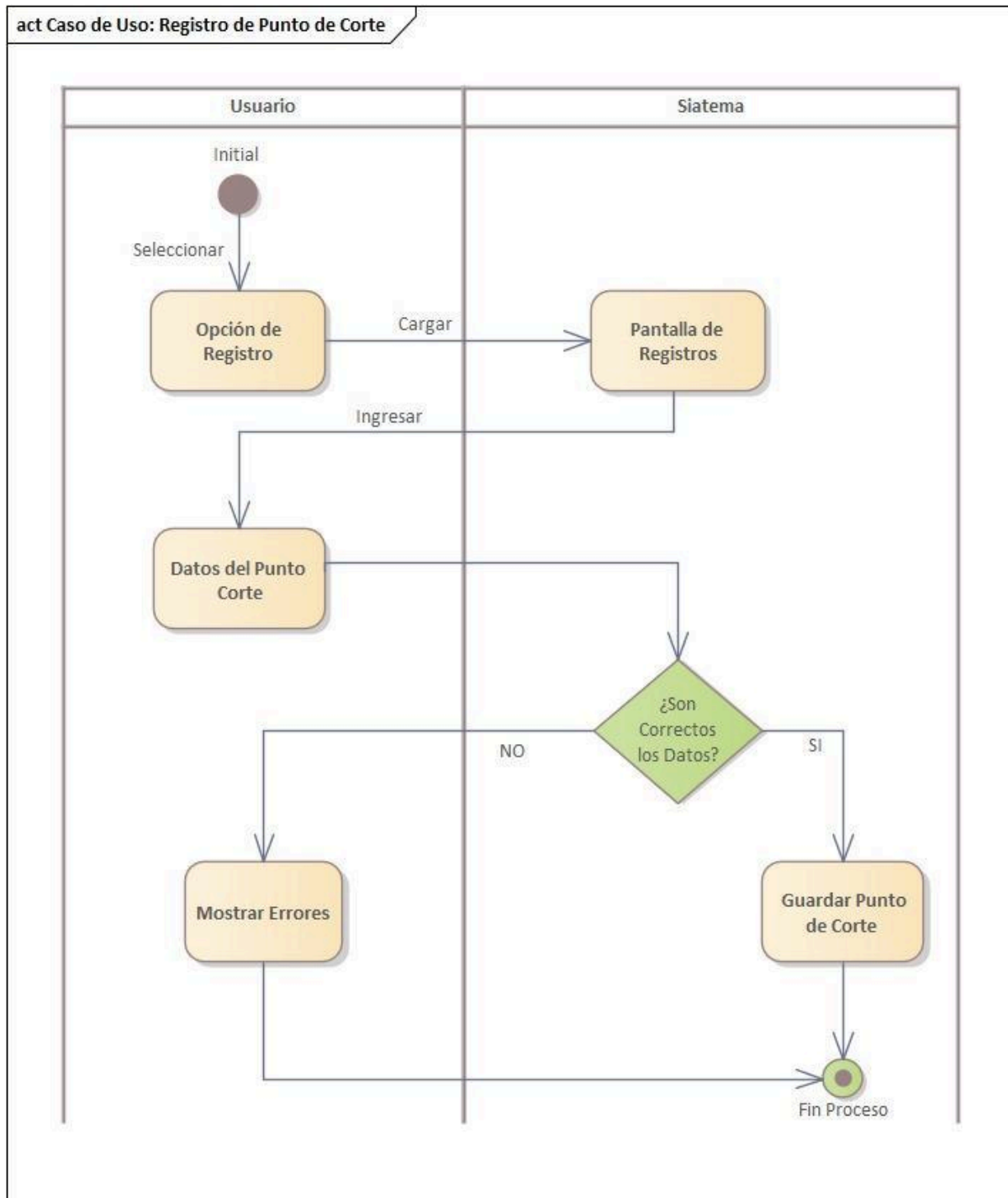


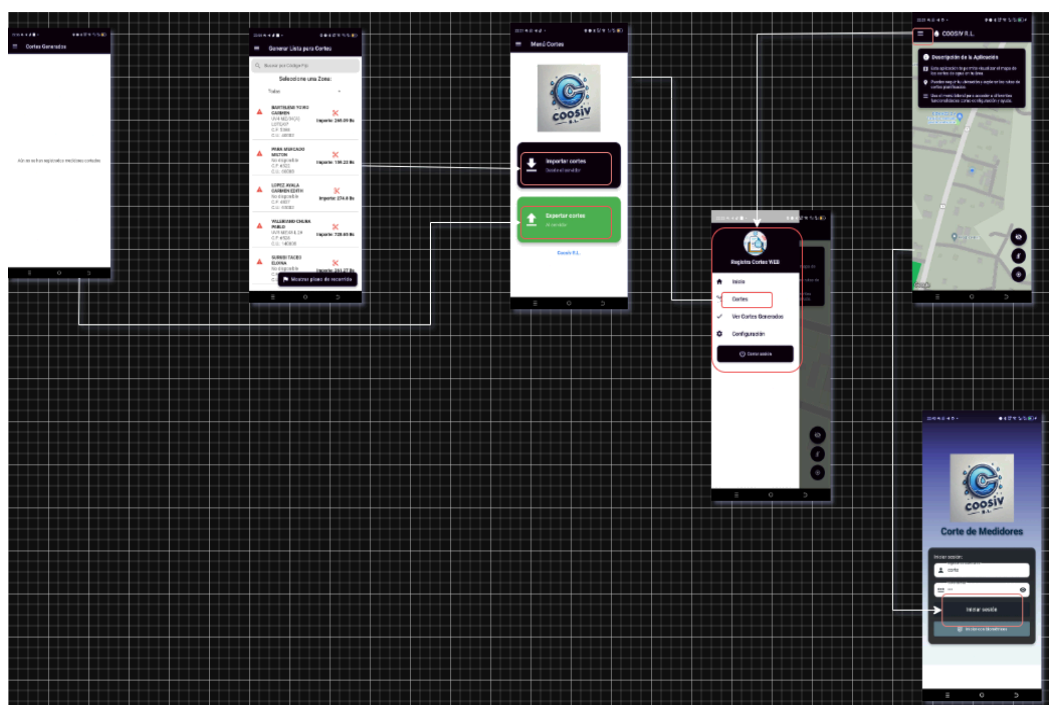
3.3.2. Diagrama de Actividades: Búsqueda y filtrado

act Diagrama Actividad: Búsqueda y Filtrado

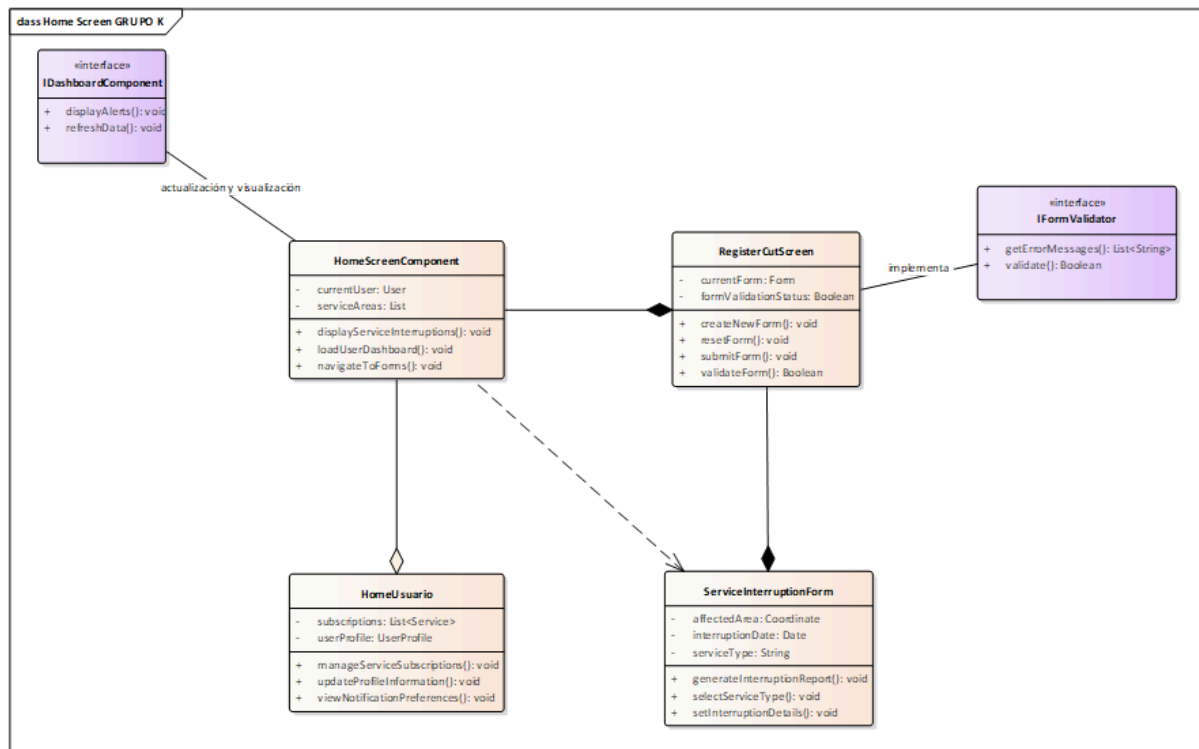


3.3.3. Diagrama de Actividades: Registro de corte

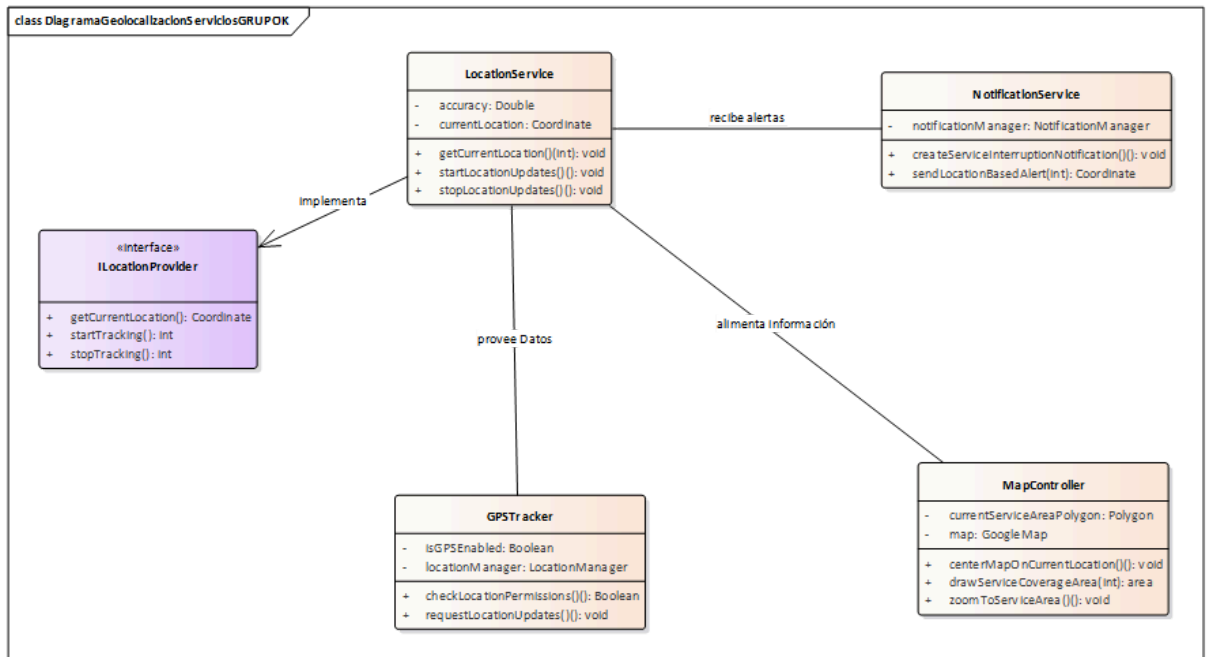




4.2.2. Diagrama UML parte 2: Formularios y Componentes de la Home Screen



4.2.3. Diagrama UML parte 3: Geolocalización y Otros Componentes



4.3.Resultados del Proyecto

4.3.1. Funcionalidad

- **Inicio de Sesión:** Los usuarios registrados pueden iniciar sesión utilizando su usuario y contraseña.
- **Lista de cortes:** Se consumirá un servicio web en la cual traerá las rutas de corte de los medidores
- **Registro de cortes:** Se elegirá el medidor a cortar a través del mapa más cercano a nuestra ubicación actual.

4.3.2. Formularios de Interacción



4.3.3. Reportes



Conclusiones

El trabajo realizado para la elaboración de esta app ha sido una explosión de ingenio para nuestro grupo a la hora de aprender a usar nuevas herramientas como ser las librerías que nos permiten trabajar con mapas y datos de los mismos.

Todo el esfuerzo invertido se refleja en la app que presentamos a continuación como proyecto de la materia, cumpliendo los requerimientos propuestos por el docente.

El trabajo y esfuerzo generados para la elaboración de esta app ha sido de experiencias nuevas, básicas pero gratificantes para enriquecer nuestros conocimientos y nuestra sed de curiosidad.

Vale decir un trabajo que nos obligó a ir más allá de lo que conocemos y un reto casi superado hasta la siguiente fase del proyecto.

Recomendación

Una recomendación sería enfocar el desarrollo inicial en las funcionalidades que generan mayor valor para los usuarios, como la autenticación y la gestión óptima de rutas con APIs confiables (ej. Google Maps). Esto garantizará una base sólida. Además, prioriza el diseño de una interfaz intuitiva, que permita a los lectores operar fácilmente en campo, y asegura un sistema robusto de sincronización de datos que funcione tanto en línea como fuera de línea. Complementa esto con pruebas continuas con usuarios reales para refinar la experiencia y detectar mejoras antes del despliegue definitivo.

Bibliografía

Desarrollo de Aplicaciones Móviles con Flutter

Flutter Documentation. (n.d.). Retrieved from <https://docs.flutter.dev/>

Dart Language. (n.d.). Retrieved from <https://dart.dev/>

Integración de Web Services

Allamaraju, S. (2010). RESTful Web Services Cookbook: Solutions for Improving Scalability and Simplicity. O'Reilly Media.

Integración de Google Maps

Google Maps Platform Documentation. (n.d.). Retrieved from

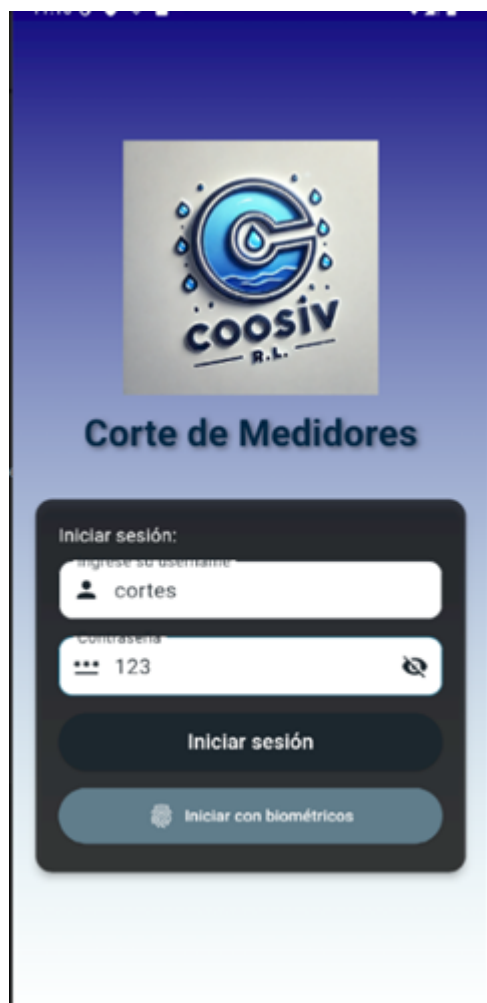
Google Developers. (n.d.). Making HTTP Requests. Retrieved from

Anexo

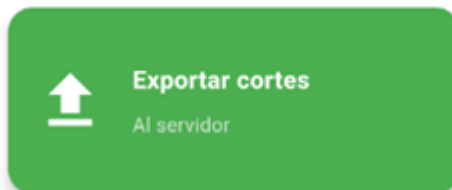
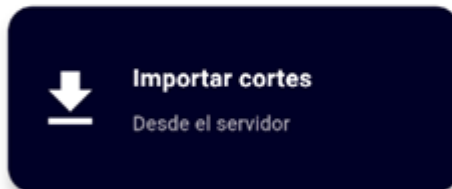
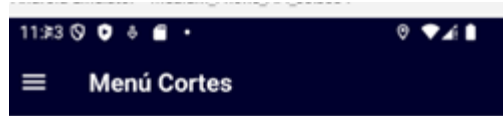
Manual de Usuario

(Aplicación móvil para cortes de Servicios públicos)

1. Ingresa a la Aplicación: **Inicia Sesión**

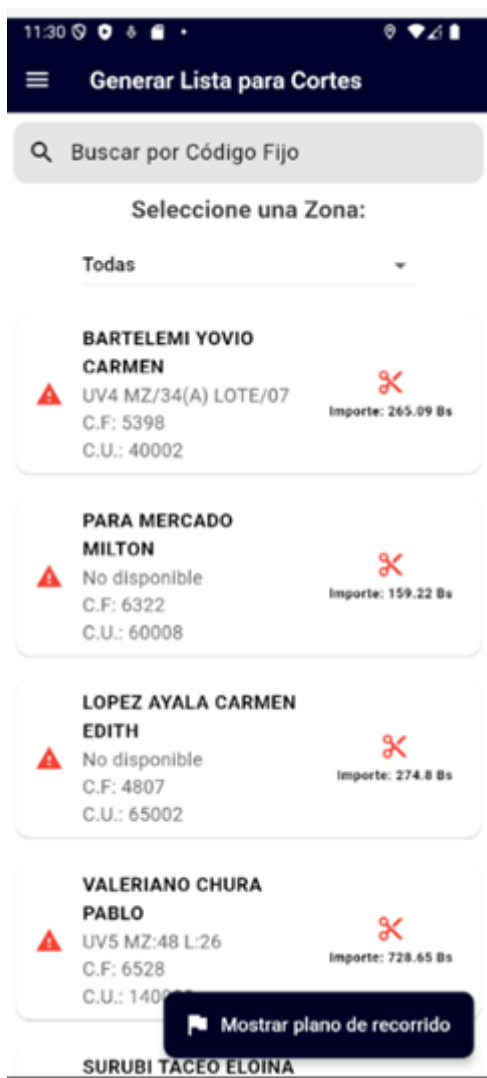


2. Verificará el usuario y luego aparecerá el siguiente menú principal al seleccionar **'Cortes'**, se mostrará El siguiente menú con las opciones de **'Importar Cortes'** y **'Exportar Cortes'**.

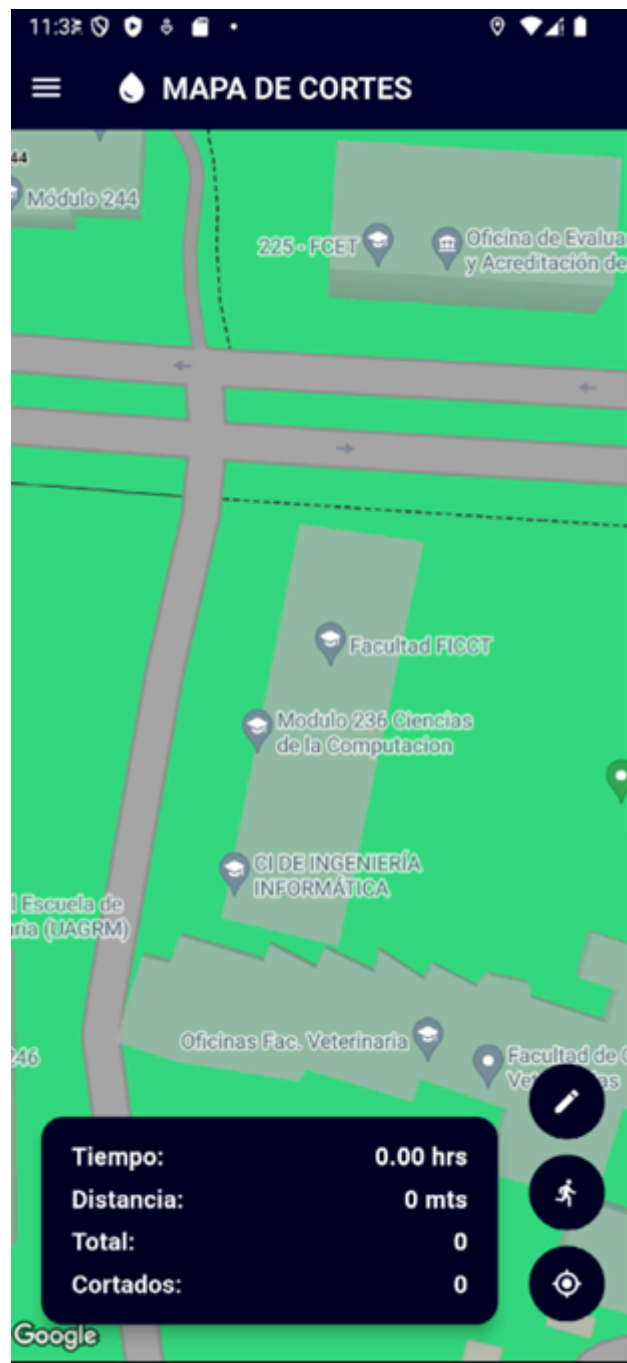


Coosiv R.L.

3. Elegimos la opción **Importar Cortes** y generará la lista de usuarios a los que se les realizará el corte del servicio. La opción '**Buscar por Código**' permite buscar un usuario por su código y mostrar sus datos, mientras que la opción '**Seleccionar por Zona**' muestra a todos los usuarios de la zona seleccionada.



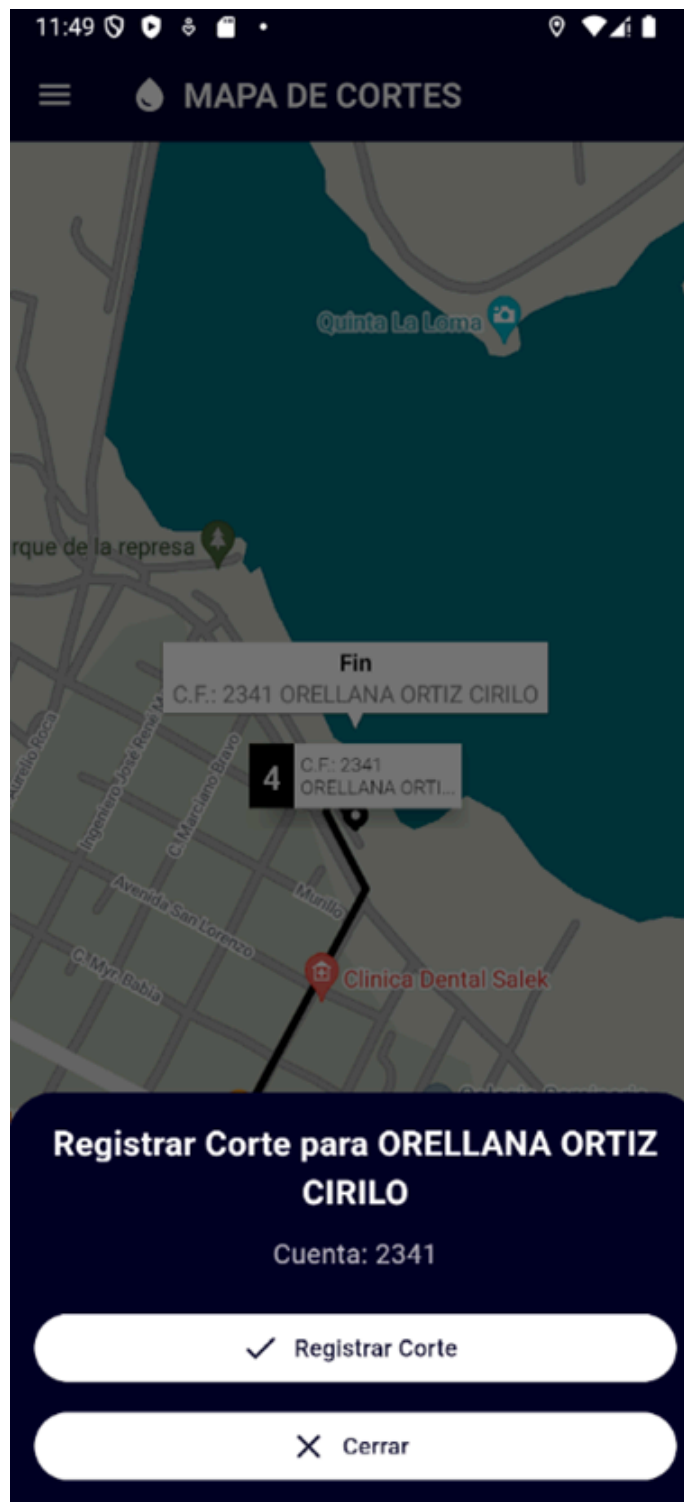
4. Una vez ahí, si elegimos **Mostrar plano de Recorrido** nos posicionará en nuestra ubicación y si elegimos el botón con el icono de lápiz la aplicación procederá a trazar de manera secuencial el recorrido de los distintos puntos, destacando la ruta óptima desde nuestra ubicación actual para visitar cada uno de ellos.



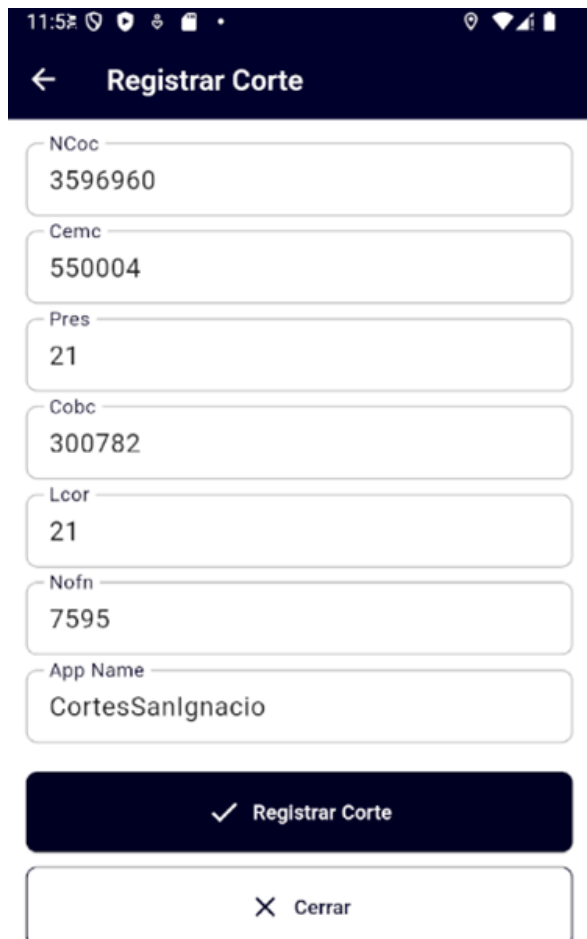
Marcando la mejor ruta, enumerándolos e indicando al técnico por cuál medidor debería comenzar, cuáles son los siguientes y con qué medidor debería finalizar, como se muestra en la siguiente imagen.



5. Una vez determinado el medidor a cortar, aparecerán las siguientes opciones:



6. Al seleccionar '**Registrar corte**', el corte será registrado y, una vez guardado, no se podrá registrar nuevamente. En su lugar, solo aparecerá un mensaje asegurando haber registrado con éxito el corte.



11:52

← Registrar Corte

NCoc
3596960

Cemc
550004

Pres
21

Cobc
300782

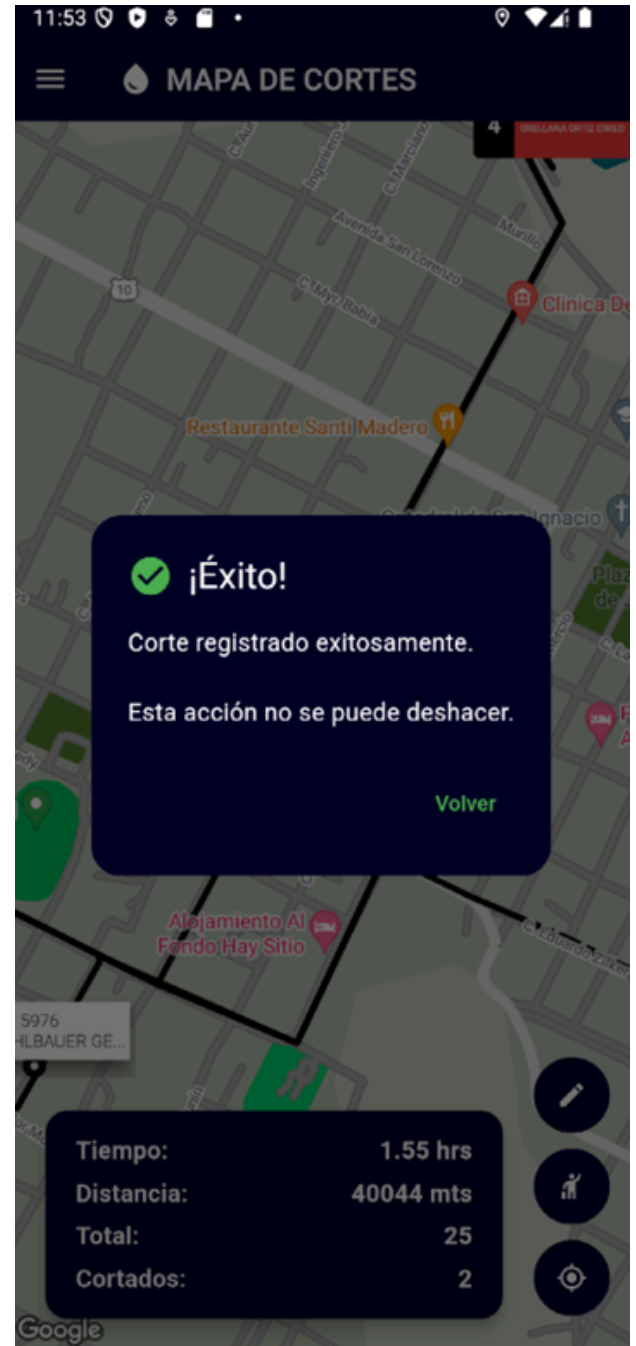
Lcor
21

Nofn
7595

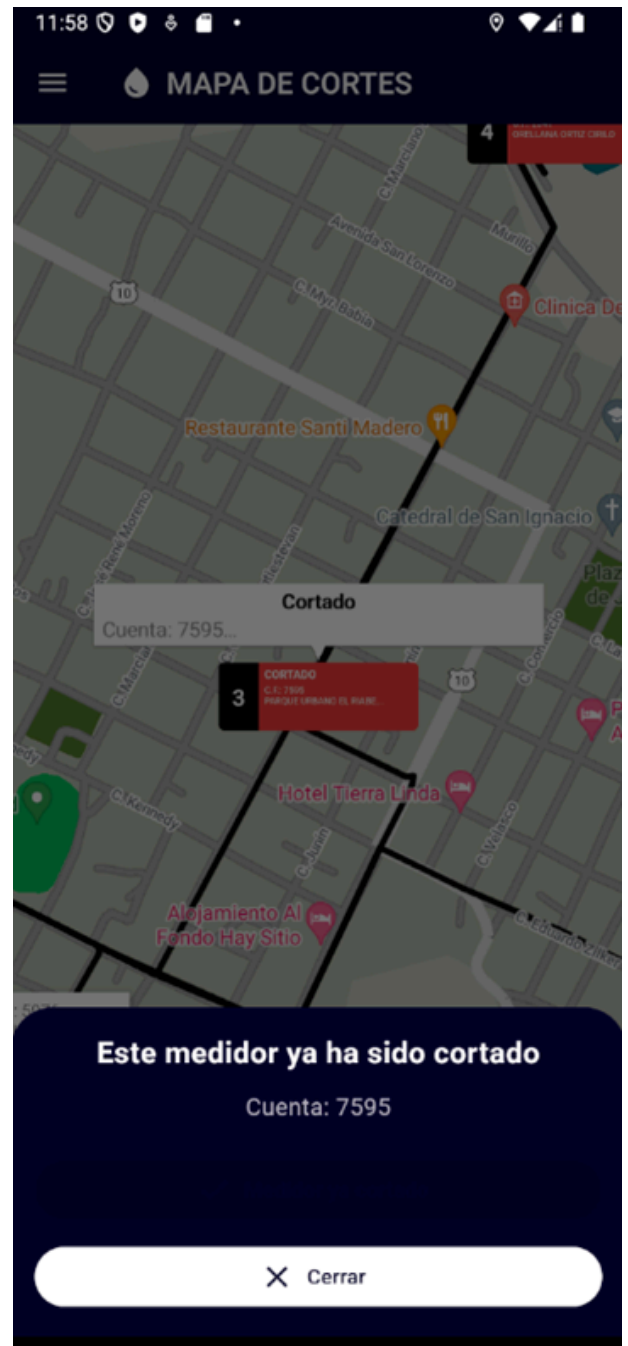
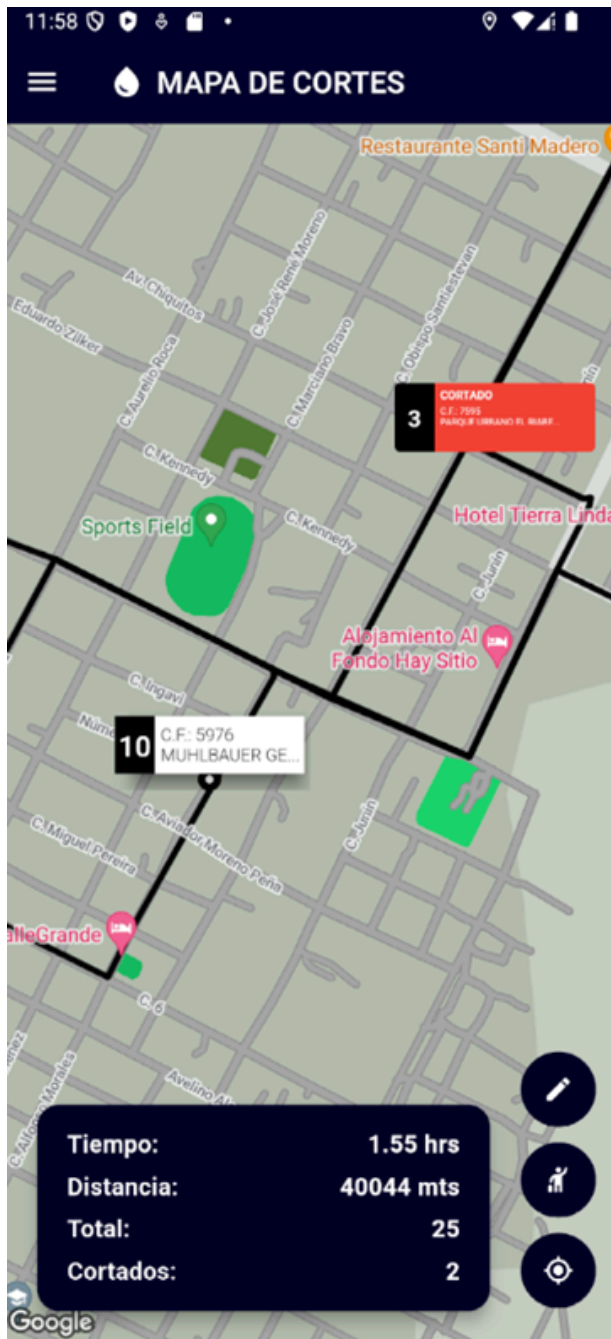
App Name
CortesSanIgnacio

✓ Registrar Corte

✕ Cerrar



7. Al dar click en **'Volver'** regresa al **MAPA DE CORTES** y si observamos la siguiente imagen podemos ver que el medidor cortado aparece en rojo y con el mensaje CORTADO, y si le damos un click encima nos indica que el medidor ya ha sido cortado.



- 8 En Cortes Generados, registra los cortes de fueron registrados con éxito.

