Universidad Nacional Autónoma de México

Facultad de Ingeniería

Ingeniería en Computación

Cómputo Móvil
Grupo 03

Profesor: Ing. Marduk Pérez de Lara Domínguez

Semestre 2025-1

Análisis técnico de la APP “City Connect”

Equipo 02

Integrantes:
Castrillo Ramírez Luis Enrique
Hernández Díaz Osvaldo
Martínez Ramírez José Ángel
Velasco Pachuca Bryan

Introducción

Objetivo de City Connect

City Connect es una aplicación que permite a los residentes de las ciudades reportar problemas urbanos como infraestructura deficiente, espacios públicos inseguros, basura acumulada o falta de transporte público en su área. Los usuarios pueden subir fotos y ubicación del problema, y la aplicación conectará estas quejas con las autoridades municipales correspondientes.

Se muestra un mapa interactivo con los reportes actuales levantados y su estado de resolución, fomentando la participación comunitaria y la transparencia en la gestión de las ciudades.

Problema y Justificación

En países cómo México, donde la Ciudad de México es la número 7 a nivel mundial con mayor densidad poblacional, se tienen 22 millones de habitantes en 1,485 kilómetros cuadrados de extensión territorial; La gestión de servicios públicos y el mantenimiento de la infraestructura se vuelven caóticas y se vuelve bastante común la carencia de servicios básicos o que necesitan reestructuraciones.

Esto indica una creciente necesidad por dar seguimiento y atención a los problemas de la ciudad, ya que se presenta el tan conocido efecto de la ventan rota, dicha teoría descrita por George Kelling en la década de 1980 describe que *“signos visibles de desinterés y deterioro pueden incitar a comportamientos delictivos. Si una ventana rota se deja sin reparar, pronto todas las ventanas estarán rotas.”*. En nuestro caso, la gestión urbana de la Ciudad de México deja que desear en muchos sentidos, por lo que consideramos que *City Connect* puede marcar la diferencia e ir revirtiendo poco a poco los estragos de la falta de atención a los problemas de gestión pública.

Requerimientos de la App

Reglas de Negocio

A continuación se describen las reglas reglas de negocio que le permitirán a *City Connect* cumplir con la operatividad deseada, cumpliendo de la mejor manera con los objetivos de la aplicación y en cumplimiento del acceso a datos e información sensible.

Categorías de reporte

Las siguientes categorías de reportes podrán ser levantadas desde el aplicativo:

- Limpieza Urbana
- Transporte y Movilidad
- Iluminación Pública
- Agua y Drenaje
- Contaminación

- Salud Pública
- Espacios Verdes y Recreación
- Acceso a Servicios Públicos
- Vivienda y Urbanización
- Animales y Fauna Urbana
- Vandalismo y Conducta Antisocial
- Desastres Naturales y Emergencias

Reporte de Problemas Urbanos

- Solo los usuarios registrados podrán enviar reportes de problemas urbanos.
- Cada reporte debe incluir al menos una descripción y una ubicación válida.
- Los reportes con imágenes deben ser de máximo 5 MB por imagen y contener hasta 3 imágenes.
- Un reporte no puede ser enviado si la categoría del problema no es seleccionada.
- No se permitirá el envío de reportes duplicados en un radio de 100 metros en un lapso de 24 horas.

Gestión de Reportes

- Cada reporte podrá tener uno de los siguientes estados: "Pendiente", "En Proceso", "Resuelto" o "Rechazado".
- Solo las autoridades correspondientes pueden cambiar el estado de un reporte.
- Los reportes rechazados deben ser enviados al usuario que lo generó con la justificación correspondiente.
- Los reportes se eliminarán del sistema 1 semana después de haber sido resueltos.

Notificaciones

- Los usuarios recibirán notificaciones cuando cambie el estado de sus reportes.
- Para el caso de reportes ajenos al usuario que lo levantó, solo se le enviarán notificaciones de reportes que se encuentren en un radio de 10 km a su ubicación.

Mapas Interactivos

- Los reportes deben mostrarse en el mapa con un marcador que indique su estado mediante colores:
    - **Rojo**: Reporte levantado.
    - **Amarillo**: Reporte en proceso de resolución.
    - **Verde**: Reporte Resuelto.
- Los usuarios pueden filtrar los reportes por categoría, estado y fecha de creación.
- Solo se mostrarán los reportes cercanos al área donde se encuentra el usuario.

Usuarios y Roles

- Existen dos tipos de usuarios principales:
    - **Ciudadanos**: Pueden levantar reportes, consultar el mapa y recibir notificaciones.

    - **Autoridades**: Pueden visualizar y actualizar el estado de los reportes en las áreas asignadas.
- Los usuarios ciudadanos deben estar autenticados mediante un correo electrónico o red social.
- Cada autoridad tendrá acceso restringido a los reportes según su jurisdicción y responsabilidad.

Seguridad y Privacidad

- La aplicación debe garantizar la confidencialidad de los datos del usuario cómo ubicación y nombre.
- No mostrar públicamente información que permita identificar a un usuario.
- Las fotos subidas a los reportes deben ser validadas para evitar contenido ofensivo o inadecuado.

Almacenamiento de Datos

- Los reportes no enviados debido a problemas de conectividad se almacenarán localmente y se sincronizará automáticamente cuando haya conexión.
- El historial de reportes estará disponible para los usuarios registrados durante un período máximo de 6 meses.

Uso de Geolocalización

- Solo se puede consultar la geolocalización directamente de los sensores del dispositivo.
- La ubicación obtenida mediante GPS debe ser precisa y asociada al reporte antes de ser enviado.
- La ubicación del usuario solo puede consultarse para el levantamiento de un reporte o la consulta de reportes cercanos.

Moderación de Contenidos

- Los reportes serán revisados automáticamente para detectar lenguaje ofensivo o spam.
- Los reportes que incumplan las normas u objetivo de la app serán automáticamente eliminados.
- Los usuarios con reportes falsos recurrentes deberán ser suspendidos permanentemente.

Conexión con Autoridades

- Cada reporte será asignado automáticamente a la autoridad correspondiente en función de su ubicación y categoría.
- Se enviarán notificaciones continuas a la autoridades con información del reporte.

Requerimientos Funcionales

La aplicación deberá cumplir con los siguientes requerimientos funcionales:

- Solo los usuarios ciudadanos con cuenta podrán levantar reportes.
- Las autoridades tendrán una cuenta creada por los administradores de *City Connect*.
- Los usuarios ciudadanos podrán realizar el levantamiento de reportes.
- Las autoridades serán las únicas que puedan cambiar el estado de un reporte.
- Cada reporte siempre deberá tener asignado un estatus, categoría y descripción.
- Los usuarios podrán realizar una carga de fotos tomadas directamente con el dispositivo.
- La aplicación tendrá acceso a la ubicación brindada por el GPS del dispositivo.
- La aplicación contará con un mapa interactivo de los reportes donde se observe en color su estado.

Requerimientos No Funcionales

La aplicación deberá cumplir con los siguientes requerimientos no funcionales:

- La aplicación deberá brindar confianza sobre la seguridad de los datos usados para levantar reportes, buscar la certificación de terceros sobre el manejo de datos.
- La consulta de reportes creados cercanos a una ubicación tiene que ser rápida, menor a 3 segundos.
- El formulario de levantamiento de reporte tiene que ser conciso y breve, no debe tomarle más de minuto y medio a un usuario levantar un reporte.
- Las notificaciones mostradas a la autoridad deben ser precisas y reflejar el grado de importancia del reporte.
- Felicitar tanto al ciudadano como a la autoridad por su labor como persona responsable.
- Una autoridad debe sentirse cómoda en utilizar la aplicación para el ejercicio de sus labores diarias.
- La aplicación deberá ser ligera y optimizada para correr en celulares de gama baja sin mostrar lentitud o fallas en la carga de componentes UI.
- La paleta de colores elegida debe reflejar confianza y serenidad.

Gestos e Interacciones Específicas

Las interacciones especiales que necesitamos para el correcto funcionamiento de la aplicación son las siguientes:

- Pellizco en pantalla para zoom in/out en el mapa.
- Doble click en el mapa para zoom in.
- Triple click en el mapa para levantar un reporte.
- Swipe de izquierda a derecha para retroceder a la pantalla anterior.
- En cualquier pantalla de la aplicación, Swipe con dos dedos de arriba hacía abajo para iniciar la búsqueda de reportes en un radio de 10 km a la ubicación del usuario.

Evaluación de Funcionalidades

Consideramos que son 5 las principales funcionalidades técnicas que el aplicativo deberá de cumplir en aras de un correcto funcionamiento que cumpla con los objetivos de *City Connect.*

1. **Levantamiento de Reportes**

   Los usuarios deberían poder enviar reportes de problemas urbanos mediante un breve formulario. Cada reporte debería incluir por lo menos una descripción, la categoría del reporte, su ubicación e imágenes como evidencia.

   Viabilidad Técnica: Consideramos que el levantamiento de reportes tiene una alta viabilidad ya que la creación de formularios en la interfaz de usuario y su integración con servicios de geolocalización (GPS) es una tarea sencilla a nivel Front-End y a nivel Back-End, la gestión y manejo de este tipo de reporte en una base de datos requeriría de pocas tablas.

   Retos tecnológicos: Restringir a un tamaño máximo de imagen para evitar sobrecarga en el almacenamiento y validar que las ubicaciones son precisas y en tiempo real.

   Conclusión: Consideramos que es una funcionalidad viable y completamente obligatoria de implementar, para afrontar los retos tecnológicos podríamos implementar una compresión automática de imágenes y una validación de la ubicación vs la IP en el Back-End.

2. **Mapa Interactivo**

   Los usuarios deberían poder visualizar los reportes en un mapa, el cual cuente con marcadores que cambien de color según el estado del problema.

   Viabilidad Técnica: Esta funcionalidad la consideramos de media a alta ya que requerirá integrarse una API de mapas como por ejemplo, Google Maps o MapKit y lograr mostrar los reportes almacenados con la visualización en tiempo real. Dicha integración requerirá de una curva de aprendizaje para la herramienta seleccionada.

   Retos tecnológicos: Tendremos que asegurar un rendimiento fluido en dispositivos de gama baja, buscar el costo mínimo asociado al uso de la API ya que la aplicación en un principio no contaría con un gran número de usuarios.

   Conclusión: Inicialmente viable utilizando una API de mapas gratuita o con costo fijo.

3. **Notificaciones de Seguimiento sobre los reportes**

   Los usuarios deberían recibir notificaciones sobre el estado de sus reportes y sobre reportes cercanos a su ubicación.

Viabilidad Técnica: Consideramos inicialmente tiene una viabilidad media a baja ya que requerimos de servicios de terceros como Firebase Cloud Messaging lo cual podría suponer un costo extra pero compensado por un mejor desempeño de *City Connect.*

Retos tecnológicos: Nuestro principal reto sería lograr sincronizar los cambios en el estado de los reportes con las notificaciones en tiempo real sin afectar el desempeño de la aplicación.

Conclusión: Inicialmente no viable, ya que presupone un costo extra y complejidad que no es indispensable en un principio pues se podría iniciar con notificaciones básicas solicitadas on-demand para reportes propios e ir moviendo la funcionalidad a notificaciones push.

4. **Conexión con Autoridades Municipales**

   Los reportes enviados por los usuarios se deberían redirigir automáticamente a las autoridades responsables, según su ubicación y categoría del reporte.

   Viabilidad Técnica: Consideramos que inicialmente y por lo menos en los primeros años de operatividad esta funcionalidad tiene una viabilidad baja ya que se requieren acuerdos con gobiernos locales para integrar la plataforma con su flujo normal de trabajo, además de confianza en la app y apertura a adoptarla como una herramienta.

   Retos tecnológicos: El reto más grande es burocrático y presupuestal ya que se requeriría establecer acuerdos con cada municipio o delegación los cuales en ocasiones no cuentan con la infraestructura tecnológica adecuada (celulares y WI-FI).

   Conclusión: Inviable en los primeros años de operación, requerirían de una plataforma estable y económicamente autosostenible para comenzar a implementarlo inicialmente a baja escala en las áreas con mejor conectividad.

5. **Modo Offline para Reportes**

   Se le debe permitir a los usuarios crear reportes en áreas sin conectividad lo que implicaría almacenar los datos localmente y posteriormente sincronizarlos cuando el dispositivo se conecte a internet.

   Viabilidad Técnica: Nosotros lo consideramos con una viabilidad media-alta ya que requerimos implementar almacenamiento local temporal en una BD ligera como SQLite y un sistema de sincronización eficiente.

   Retos tecnológicos: Una correcta sincronización entre los datos locales y los datos en el servidor así como restringir la cantidad de memoria que puedan ocupar todos los reportes en el almacenamiento local.

Conclusión: Viable ya que es una funcionalidad valiosa en zonas rurales o con mala conectividad, no obstante, requerirá de una buena sincronización de datos y un testing robusto.

Alcance y Producto Mínimo Viable (MVP)

Alcance del Proyecto

Consideramos que el proyecto tendría inicialmente 3 etapas, cada una con un alcance distinto, dichas etapas son las siguientes:

1. **Surgimiento y creación**

   Para esta etapa se estaría trabajando en construir el MPV, buscar apoyo financiero sin fines de lucro, entablar las primeras relaciones con stakeholders y buscar la constitución del aplicativo con las funcionalidades necesarias para cumplir con los objetivos preestablecidos; para esta primer etapa nuestro público objetivo serían comunidades bien integradas en ciudades con una alta conectividad en ubicaciones preestablecidas.

   En cuanto al equipo de desarrollo, se estaría trabajando únicamente con los participantes de este documento, ya que compartimos la misma idea y estamos comprometidos con su desarrollo.

   En cuanto a las funcionalidades para esta primer etapa, *City Connect* debería contar con las siguientes características:

   - Sistema de login para usuarios Ciudadanos y Administradores.
   - Sistema de levantamiento de reportes por categoría.
   - Visualización de reportes levantados mostrando su estado.
   - Métricas de reportes.
   - Filtrado de reportes por categoría y ubicación.
   - Historial de reportes levantados por el usuario.

2. **Estabilización y monetización**

   Para esta etapa se trabajaría con buscar una constitución legal del aplicativo, buscar formalmente patrocinadores, incorporar anuncios y comenzar a apuntar a un público más amplio y menos restringido.

   Para el equipo de desarrollo, se podrían buscar expandirlo según el apoyo de los patrocinadores y las percepciones económicas que comience a percibir el aplicativo, inicialmente se contratarían personas con conocimientos en ciberseguridad y cloud computing.

   En cuanto a las funcionalidades del aplicativo, para esta etapa se estaría trabajando en las siguientes características:

   - Anuncios incorporados.

- Guardado de reportes offline y sincronización.
- Visualización de los reportes en un mapa interactivo.
- Incorporación de notificaciones personalizadas.
- Movilización a proveedores de cloud computing.

3. **Incorporación al gobierno**

   Esta etapa consideramos sería la más compleja por el factor humano ya que requeriríamos hacer un esfuerzo constante de socialización entre los responsables de atender los problemas y el equipo legal y de desarrollo del aplicativo.

   Para el equipo, se requeriría contratar formalmente servicios legales que nos apoye con alinear el aplicativo con las normas mexicanas de uso de datos y seguridad, además, se podría buscar integrar al equipo de desarrollo a la entidad legal de *City Connect* con el objetivo de brindarles seguridad social y respaldo legal.

   En cuanto al aplicativo sería necesario trabajar en las siguientes características:

   - Desarrollar las funcionalidades necesarias para que las **autoridades** en el ejercicio de sus facultades puedan utilizar *City Connect* como plataforma.
   - Integración de nuevo usuario **autoridades** quien sería el encargado de atender formalmente los reportes levantados por los usuarios.
   - Brindar capacitación a las **autoridades** sobre el manejo de City Connect.
   - Integrar una nueva notificación cuando un reporte esté siendo atendido por una **autoridad.**
   - Mejorar el desempeño de los servidores, tanto en tamaño de almacenamiento como en velocidad de procesamiento.

MVP

El Producto Mínimo Viable estaría buscando atacar el primer punto **Surgimiento y creación** por lo que se estaría trabajando en las siguientes funcionalidades:

- Sistema de login para usuarios Ciudadanos y Administradores.
- Sistema de levantamiento de reportes por categoría.
- Visualización de reportes levantados mostrando su estado.
- Métricas de reportes.
- Filtrado de reportes por categoría y ubicación.
- Historial de reportes levantados por el usuario.

Todo esto buscando la menor inversión posible pues el aplicativo aún es un riesgo en términos de rentabilidad y retorno de inversión lo cual nos obliga a que el aplicativo surgiría con el esfuerzo de los autores de este documento.

Wireframes de la App

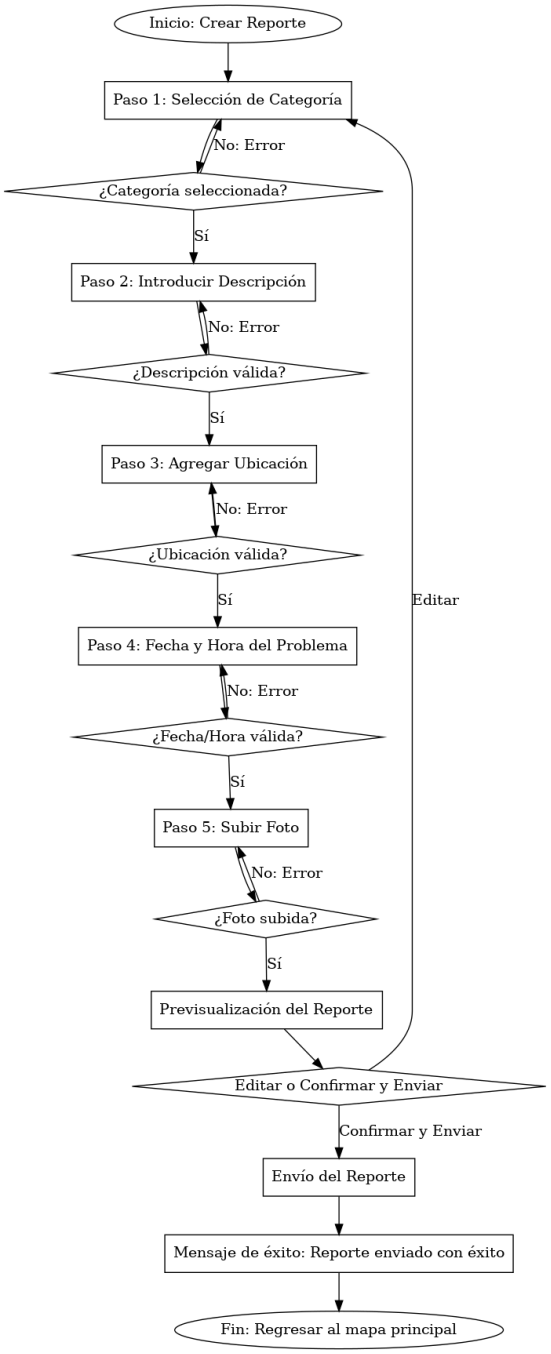
Presentación de Wireframes / Lista de pantallas

Debido que la extensión de los wireframes es bastante larga se opta por adjuntar el link a la carpeta de drive con el objetivo de tener una mejor visualización.

Frames

Explicación del Flujo de Pantallas

Mapa de Navegación



Descripción del Diagrama de Flujo: Completar y Enviar Reporte

1. **Inicio**
    - **Inicio del Proceso**:
        - Inicia cuando el usuario presiona "Crear Reporte" desde el mapa principal.
2. **Pantalla de Llenado del Reporte**
    - **Paso 1: Selección de Categoría**
        - **Acción**: Usuario selecciona la categoría (Infraestructura, Seguridad, etc.)
        - **Decisión (Validación)**:
            - ¿Se seleccionó una categoría?
                - **Sí**: Avanzar al Paso 2.
                - **No**: Mostrar mensaje de error "Debe seleccionar al menos una categoría".
    - **Paso 2: Introducir Descripción**
        - **Acción**: Usuario introduce una descripción detallada del problema.
        - **Decisión (Validación)**:
            - ¿La descripción está vacía?
                - **Sí**: Mostrar mensaje de error "La descripción no puede estar vacía".
                - **No**: Avanzar al Paso 3.
    - **Paso 3: Agregar Ubicación**
        - **Acción**: La aplicación detecta la ubicación automáticamente y la muestra.
            - **Acción**: Usuario puede ajustar la ubicación manualmente en un mapa interactivo.
        - **Decisión (Validación)**:
            - ¿La ubicación es válida (detectada o seleccionada)?
                - **Sí**: Avanzar al Paso 4.
                - **No**: Mostrar mensaje de error "Debe proporcionar una ubicación válida".
    - **Paso 4: Fecha y Hora del Problema**
        - **Acción**: Aplicación muestra la fecha y hora actuales.
            - **Acción**: Usuario puede ajustar la fecha y hora si el evento es pasado.
        - **Decisión (Validación)**:
            - ¿La fecha y hora son válidas?
                - **Sí**: Avanzar al Paso 5.
                - **No**: Mostrar mensaje de error "Formato de fecha/hora no válido".
    - **Paso 5: Subir Foto**
        - **Acción**: Usuario sube una foto del problema desde la galería o toma una nueva.
        - **Decisión (Validación)**:
            - ¿Se subió la foto?
                - **Sí**: Avanzar a la Previsualización del Reporte.

        - **No**: Mostrar mensaje de error "Debe subir una foto del problema".
3. **Previsualización del Reporte**
    - **Acción**: Se muestra un resumen de los datos: categoría, descripción, ubicación, fecha y hora, y foto.
    - **Decisión**:
        - **Botón "Editar"**: El usuario puede regresar y modificar los datos.
        - **Botón "Confirmar y Enviar"**: Se procede a enviar el reporte.
4. **Envío del Reporte**
    - **Acción**: La aplicación envía los datos a la base de datos o API correspondiente.
    - **Mensaje de éxito**: Mostrar "Reporte enviado con éxito".
5. **Fin**
    - **Acción**: El usuario regresa al mapa principal donde se refleja el reporte en el sistema.

Elementos para el Diagrama

- **Inicio / Fin**: Representados con óvalos.
- **Acciones**: Representadas con rectángulos (por ejemplo: "Seleccionar categoría", "Introducir descripción").
- **Decisiones**: Representadas con rombos (por ejemplo: "¿Se seleccionó una categoría?").
- **Flujo**: Flechas que indican la secuencia de los pasos.

Flujo General

1. Inicio
2. Selección de Categoría → [¿Categoría seleccionada?] → No (error) / Sí → Paso 2
3. Introducir Descripción → [¿Descripción válida?] → No (error) / Sí → Paso 3
4. Agregar Ubicación → [¿Ubicación válida?] → No (error) / Sí → Paso 4
5. Fecha y Hora del Problema → [¿Fecha/Hora válida?] → No (error) / Sí → Paso 5
6. Subir Foto → [¿Foto subida?] → No (error) / Sí → Previsualización del Reporte
7. Previsualización del Reporte → Botón Editar (regresar y modificar) / Botón Confirmar y Enviar → Envío del Reporte
8. Mostrar Mensaje de Éxito
9. Fin (regreso al mapa principal)

Detalle del Recorrido

Explicación paso a paso del flujo de navegación de "City Connect"

1. **Pantalla de Inicio (Splash Screen):**
    - El usuario abre la aplicación y ve el logo con el tagline "Conectando ciudadanos y ciudades".
    - Transición automática a la pantalla de inicio de sesión.
2. **Pantalla de Inicio de Sesión/Registro:**

    - Los usuarios pueden iniciar sesión con su cuenta existente o crear una nueva cuenta.
    - Si eligen "Crear cuenta", se redirige al formulario de registro.
3. **Pantalla de Mapa Interactivo (Home):**
    - Una vez autenticados, los usuarios llegan al mapa interactivo.
    - Aquí pueden:
        - Ver reportes cercanos marcados en el mapa.
        - Aplicar filtros para explorar reportes específicos.
        - Acceder al botón "Crear reporte" para iniciar el flujo de reporte.
4. **Pantalla de Crear Reporte:**
    - El usuario selecciona una categoría del problema y accede al formulario de llenado.
5. **Pantalla de Llenado Detallado del Reporte:**
    - El usuario completa:
        - Descripción del problema.
        - Foto del problema (tomada o desde la galería).
        - Ubicación (automática o manual).
        - Fecha y hora (actual o ajustada).
    - Valida la información antes de enviar.
6. **Pantalla de Confirmación de Envío del Reporte:**
    - Mensaje de éxito: "¡Reporte enviado exitosamente!".
    - Opciones:
        - Ver el estado del reporte.
        - Regresar al mapa.
7. **Pantalla de Detalle del Reporte:**
    - Los usuarios pueden ver los detalles de un reporte, incluyendo:
        - Descripción, foto, estado y comentarios de las autoridades.
    - Opciones para editar o cancelar el reporte (si está en estado pendiente).
8. **Pantalla de Notificaciones:**
    - Los usuarios reciben actualizaciones del estado de sus reportes.
    - Opciones para marcar notificaciones como leídas o eliminarlas.
9. **Pantalla de Ajustes:**
    - Permite a los usuarios personalizar su experiencia:
        - Cambiar idioma.
        - Configurar notificaciones.
        - Acceder a las políticas y términos.
10. **Pantalla de Historial de Reportes:**
    - Una lista de todos los reportes enviados por el usuario, con opciones para ordenar por estado, fecha o categoría.
11. **Pantalla de Estadísticas Comunitarias:**
    - Muestra métricas sobre la participación de la comunidad (reportes enviados, resueltos, etc.).
12. **Pantalla de Ayuda Detallada:**
    - Acceso a preguntas frecuentes, artículos de soporte y un formulario para contactar al equipo técnico.
13. **Pantalla de Configuración Avanzada:**
    - Configuraciones para accesibilidad (modo oscuro, tamaño de texto) y ajustes avanzados de notificaciones.

14. **Pantalla de Confirmación de Eliminación de Reporte:**
    - El usuario recibe una confirmación antes de eliminar un reporte.
15. **Pantalla de Cierre de Sesión:**
    - El usuario confirma si desea cerrar sesión.

Wireframes Adicionales.

La **Pantalla de Filtros para Reportes** permite a los usuarios buscar reportes específicos en el mapa. Ofrece opciones como categorías (Infraestructura, Seguridad, Salud, Transporte), estados del reporte (Pendiente, En Proceso, Resuelto) y un rango de fechas (Fecha Desde y Fecha Hasta). Después de configurar los filtros, el usuario puede aplicar los criterios seleccionados para actualizar los resultados.

La **Pantalla de Resultados de Búsqueda** muestra una lista de reportes que cumplen con los filtros aplicados. Incluye una barra de búsqueda para encontrar reportes específicos, una lista de resultados con categoría, ubicación y fecha, y opciones para ordenar por fecha o cercanía. Un botón permite regresar al mapa principal, y al seleccionar un reporte, el usuario puede ver sus detalles.

La **Pantalla de Confirmación de Eliminación de Reporte** solicita al usuario confirmar antes de eliminar un reporte. Muestra un mensaje claro ("¿Estás seguro de que deseas eliminar el reporte?") junto con botones de acción para confirmar o cancelar. Si el usuario elige eliminar, el sistema elimina el reporte y muestra un mensaje de éxito.

La **Pantalla de Centro de Ayuda** ofrece soporte a los usuarios. Incluye categorías (como "Reportar problemas", "Seguridad de cuenta" y "Configuración") para navegar entre temas, una barra de búsqueda para encontrar artículos específicos y una lista de artículos recientes con títulos y descripciones cortas. Al seleccionar un artículo, el usuario puede leer las instrucciones completas.

La **Pantalla de Historial de Reportes** muestra un resumen de todos los reportes en la comunidad, filtrados por categorías como Infraestructura o Seguridad. También permite buscar reportes específicos, ordenar los resultados por cercanía o fecha, y explorar los detalles de cada reporte al seleccionarlos.

La **Pantalla de Configuración Avanzada** permite personalizar notificaciones (frecuencia y categorías) y configurar opciones de accesibilidad como modo oscuro y tamaño de texto ajustable. También incluye preferencias generales y la opción de restablecer configuraciones a valores predeterminados. Los cambios se guardan al confirmar, mostrando un mensaje de éxito.

Análisis de Datos y Servicios por Pantalla

Descripción de los Datos

Lógica de conexión con servicios necesarios

1. Almacenamiento de datos

- Base de datos relacional o no relacional (Firebase o PostgreSQL):

  Se utiliza para almacenar datos estructurados como texto, coordenadas y estados de los informes.

  - Estructura: Cada informe contiene un ID único, descripción, categoría, coordenadas, estado y referencia a las imágenes almacenadas.
  - Conexión: La aplicación realiza solicitudes al backend para guardar o recuperar datos mediante una API REST.

2. Gestión de imágenes
   - Servicio en la nube (Firebase Storage o AWS S3):

     Las imágenes se cargan al servidor, y su URL se guarda en la base de datos asociados al reporte correspondiente.

     - Conexión: Las imágenes se suben desde el dispositivo a través de la API del servicio de almacenamiento y se vinculan a los datos del informe.
3. Geolocalización
   - API de mapas (Google Maps o MapKit):

     Las coordenadas GPS ingresadas se utilizan para marcar los informes en un mapa interactivo.

     - Conexión: El backend envía las coordenadas al servicio de mapas, que devuelve un marcador en el mapa visible en la aplicación.
4. Notificaciones
   - Firebase Cloud Messaging:

     Envía actualizaciones automáticas a los usuarios cuando cambian los estados de los informes.

     - Conexión: El backend interactúa con Firebase para programar y enviar notificaciones push.
5. Sincronización en tiempo real
   - Firebase Realtime Database o WebSockets:

     Para mostrar actualizaciones instantáneas en el mapa interactivo o en el estado de los informes.

     - Conexión: El frontend escucha cambios en los datos directamente desde el servidor.

Integración con dispositivos móviles

Dispositivos Soportados

Pensando en una audiencia de comunidades de recursos escasos en ciudades en desarrollo, ***City Connect*** está diseñado para ser accesible y funcional en una amplia variedad de dispositivos:

1. Plataformas y sistemas operativos soportados

- **Android:**

  Soporte para dispositivos con Android 6.0 (Marshmallow) o versiones superiores, ya que esta versión aún es común en dispositivos de gama baja en regiones en desarrollo.

- **iOS:**

  Compatibilidad con iOS 12 o versiones posteriores, para incluir modelos antiguos de iPhone que todavía están en uso.

Tamaños de pantalla

- Teléfonos móviles:
  - Pantallas pequeñas (4 pulgadas): Para dispositivos básicos y económicos.
  - Pantallas medianas (5-6,5 pulgadas): La mayoría de los teléfonos en mercados emergentes.
  - Pantallas grandes (7 pulgadas o más): Compatibilidad con dispositivos tipo phablet.
- Tablets (opcional):

  Soporte básico para tablets pequeñas, aunque el diseño está optimizado para teléfonos.

Orientación

- Retrato (portrait):

  La aplicación está diseñada principalmente para uso en orientación vertical, ya que es la más común para la interacción diaria.

- Paisaje (landscape):

  Compatible en funciones como el mapa interactivo, que se beneficia de una vista más amplia.

Optimización

- Se prioriza la eficiencia en dispositivos de gama baja con procesadores básicos y almacenamiento limitado.

- Se utilizan imágenes comprimidas y cargas ligeras para minimizar el uso de datos móviles, considerando que la conectividad puede ser limitada.

Este enfoque garantiza que ***City Connect*** sea accesible y útil para la mayor cantidad posible de usuarios en contextos económicos y tecnológicos desafiantes.

Uso de Almacenamiento Local

El almacenamiento local estaría principalmente destinado para guardar los reportes que se realicen offline de esta manera se podrían leer y actualizar con el servidor una vez que la conectividad Wi-Fi regrese con el dispositivo.

Para este objetivo se buscaría implementar SQLite con el propósito de contar con una versión ligera de SQL y aprovechar la interactividad que ya existe con sistemas Android y IOs. En caso de detectar mayor complejidad, al ser información sencilla podríamos hacer uso de un archivo XML mientras se implementa la conexión con la Base de Datos.

Sensores Utilizados

Para el correcto funcionamiento de todas las características de ***City Connect*** se requerirá utilizar los siguientes 2 sensores.

1. **GPS:**

   Utilizado para obtener la ubicación precisa de los problemas reportados. Permite detectar automáticamente las coordenadas del dispositivo o ajustar la ubicación manualmente en un mapa interactivo. Es esencial asociar los informes a ubicaciones específicas y mostrarlos en un mapa.

2. **Cámara:**

   Usada para capturar imágenes como evidencia visual de los problemas urbanos. Los usuarios pueden tomar fotos directamente o subirlas desde la galería. Estas imágenes se almacenan en la nube y se vinculan al informe.

Lenguajes de Programación y Herramientas de Desarrollo

Lenguajes de Programación

- **Android**: Se utilizará **Kotlin**, el lenguaje moderno y oficial para el desarrollo de apps Android. Ofrece una sintaxis concisa, y tiene plena integración con el ecosistema de Android.
- **iOS**: Para el desarrollo en dispositivos iOS, se utilizará **Swift**, el lenguaje de programación oficial de Apple, que es rápido, seguro y fácil de mantener.

Herramientas y Frameworks

En cuanto a herramientas y frameworks decidimos dividirlas en las siguiente 2 secciones:

- **Servidor:**

  Base de datos: Del lado del servidor será necesario contar con una base de datos robusta que nos permita guardar tanto datos de usuarios como de los reportes, para esto en un principio estaríamos utilizando PostgreSQL, dado que es un motor de Bases de Datos gratuito y posteriormente estaremos migrando a Firebase de Google.

  Notificaciones Push: Para las notificaciones Push utilizaremos los servicios de terceros como Firebase Cloud Messaging que permite tener una buena gestión en el envío de notificaciones.

  Servidor: Cómo framework estaríamos buscando la utilización de Django o Node.js ya que ambos son de uso gratuito.

- **Dispositivo móvil:**

  Base de datos: Para la base de datos datos del lado del dispositivo móvil es necesario una base de datos ligera, por lo tanto, se estaría utilizando SQLite que tiene compatibilidad tanto con Android como con IOs.

Permisos y Políticas de Publicación

Para garantizar que la aplicación cumpla con los requisitos de las principales tiendas de aplicaciones, como App Store y Google Play, necesitamos cubrir temas relacionados con los permisos, la privacidad de los datos y adecuaciones técnicas necesarias para su publicación.

En primer lugar, se deben especificar los permisos que utilizará la aplicación de manera clara y precisa. Primeramente contamos con el acceso a la ubicación del usuario mediante GPS, lo cual es esencial para vincular los reportes con coordenadas geográficas y mostrar problemas cercanos en el mapa interactivo. También será necesario solicitar acceso a la cámara y a la galería para que los usuarios puedan tomar fotografías de los problemas urbanos o adjuntar imágenes desde su dispositivo. Otro permiso importante es el de conexión a internet, indispensable para sincronizar los reportes con el servidor y cargar mapas interactivos. Por último, se requiere acceso al almacenamiento local para guardar temporalmente los reportes en caso de que el usuario esté sin conexión.

En cuanto a las políticas de privacidad, es importante garantizar que todos los datos sean utilizados exclusivamente para los fines específicos de la aplicación, asegurando que no se comercializarán ni compartirán con terceros sin el consentimiento del usuario. Además, ofrecemos a los usuarios la posibilidad de acceder a sus datos almacenados y, si lo desean, eliminarlos por completo del sistema.

La implementación técnica de estas medidas también incluye el cifrado de los datos tanto en tránsito como en reposo, para proteger la información contra accesos no autorizados. Asimismo, es recomendable cumplir con normativas internacionales como el GDPR, en caso de que la app se utilice en regiones donde estas regulaciones se aplican, asegurando un manejo adecuado de los datos sensibles.

Por último, para garantizar la aprobación de las tiendas de aplicaciones, es necesario seguir sus lineamientos específicos. En el caso de la App Store, Apple solicita una descripción clara de los permisos que se utilizarán y cómo beneficiarán al usuario, además de un enlace directo a la política de privacidad. Por otro lado, Google Play requiere completar un formulario de seguridad de datos que detalle el uso y la protección de la información personal. También es importante que la aplicación sea funcional en dispositivos de distintas capacidades técnicas, desde gamas altas hasta equipos más básicos, para ampliar su accesibilidad.

Esto con el fin de generar confianza entre los usuarios al demostrar un compromiso con la privacidad y la seguridad de sus datos.

Equipo de Trabajo y Roles

1. **Ángel – Desarrollador Backend:**
- Diseñar, implementar y mantener la lógica del servidor, asegurando que la base de datos y los servicios funcionen correctamente.
- Configurar servidores en la nube y garantizar la seguridad de los datos de usuarios y reportes.
- Integrar API externas, como Google Maps, para funcionalidades como la geolocalización y notificaciones.
- Implementar el sistema de autenticación segura y manejo de usuarios.
- Optimizar el rendimiento del backend para manejar grandes volúmenes de datos y usuarios.

2. **Enrique – Desarrollador Frontend:**
- Implementar la interfaz móvil para Android e iOS, asegurando que sea funcional y responsiva.
- Traducir los diseños UI/UX en componentes de código funcional.
- Integrar funcionalidades como el mapa interactivo, la creación de informes y notificaciones push.
- Garantizar la compatibilidad de la aplicación con dispositivos de diferentes gamas y resoluciones.
- Realizar pruebas de rendimiento y experiencia de usuario en diferentes dispositivos.

3. **Osvaldo – Control de Calidad (QA) y Gestión de Producto:**
- Diseñar y ejecutar pruebas funcionales, de rendimiento y de seguridad para verificar el correcto funcionamiento de la aplicación.
- Identificar errores y problemas de usabilidad, documentarlos y supervisar su resolución.
- Realizar pruebas de compatibilidad para asegurar que la aplicación funcione en diferentes dispositivos y sistemas operativos.

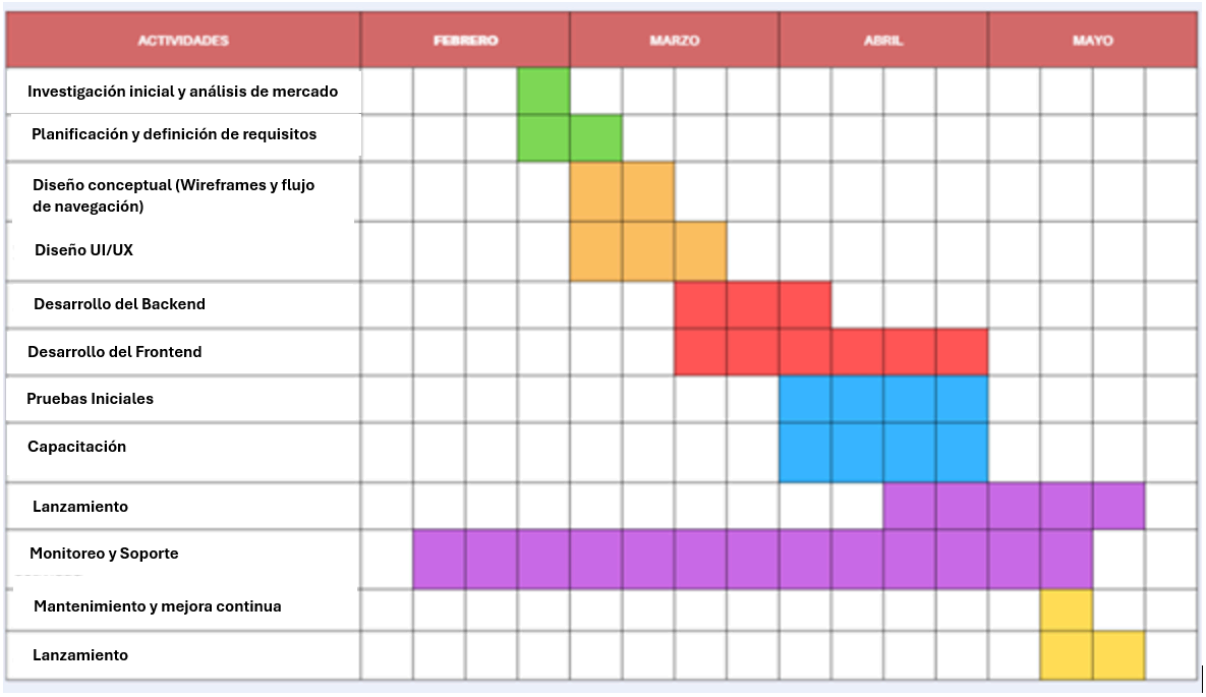
- Establecer estándares de calidad y monitorear su cumplimiento durante el desarrollo.
- Coordinar las tareas del equipo y mantener la organización del proyecto.
- Definir el alcance y las prioridades del proyecto según los objetivos establecidos.
- Comunicar avances y resultados a las partes interesadas.
- Recopilar y analizar la retroalimentación de los usuarios para proponer mejoras y funcionalidades futuras.
- Asegurar que el proyecto cumpla con los tiempos y los recursos asignados.

4. **Bryan – Diseñador UI/UX:**

- Cree wireframes y prototipos de alta fidelidad que reflejen el flujo de navegación y la estructura visual de la aplicación.
- Diseñar una interfaz de usuario intuitiva, accesible y visualmente atractiva.
- Colaborar con el equipo frontend para garantizar que el diseño sea implementado correctamente.
- Realizar investigaciones y pruebas de usabilidad para mejorar la experiencia del usuario.
- Asegurar la coherencia visual y funcional en todas las pantallas de la aplicación.

Estimación de Tiempo y Costos

Cronograma de Desarrollo



Costos de Mantenimiento

1. **Infraestructura básica**

- **Almacenamiento y servidores en la nube:**

Se utilizarán servicios básicos como Firebase o AWS para almacenar informes, imágenes y datos de los usuarios.

Costo estimado: $2,000 USD/año

- **Notificaciones push:**

  Uso de un servicio gratuito o económico como Firebase Cloud Messaging para enviar actualizaciones de los informes.

  Costo estimado: $500 USD/año

2. **Actualizaciones mínimas**

- **Corrección de errores y actualizaciones de seguridad:**

  Incluye ajustes necesarios en el backend y el frontend para mantener la funcionalidad básica y seguridad.

  Costo estimado: $2,000 USD/año

- **Compatibilidad con sistemas operativos:**

  Adaptación a nuevas versiones de sistemas operativos.

  Costo estimado: $1,000 USD/año

3. **Soporte técnico**

- **Atención al usuario:**

  Resolución de dudas y problemas básicos reportados por los usuarios, utilizando herramientas gratuitas o económicas para gestionar solicitudes.

  Costo estimado: $1,000 USD/año

4. **Licencias**

- **Renovación en tiendas de aplicaciones:**

  Pago anual para mantener la aplicación disponible en App Store y Google Play.

  Costo estimado: $100 USD/año

5. **Promoción básica**

- **Publicidad orgánica y mínima inversión en redes sociales:**

  Campañas en redes sociales enfocadas en mostrar los beneficios de la app y fomentar su uso.

  Costo estimado: $500 USD/año

Resumen de costos anuales

1. Infraestructura básica: $2,500 USD
2. Actualizaciones mínimas: $3,000 USD
3. Soporte técnico: $1,000 USD
4. Licencias: $100 USD
5. Promoción básica: $500 USD

Costo total anual estimado: **$7,100 USD**

Referencias

- Asun Luján. (2024, April 8). Estas son las 20 ciudades más pobladas del mundo en 2024. Viajes.nationalgeographic.com.es; Viajes National Geographic. [Consultado el 7 de noviembre de 2024]. https://viajes.nationalgeographic.com.es/a/estas-son-diez-ciudades-mas-pobladas-mundo-2022_18248
- ACCIONA. (2023, December 18). En qué consiste la teoría de las ventanas rotas | People ACCIONA. People ACCIONA. [Consultado el 8 de noviembre de 2024]. https://people.acciona.com/es/tendencias-e-inspiracion/teoria-ventanas-rotas/
- Escobar, S. (2023, April 21). En México, 8 de cada 10 viviendas carecen de servicios básicos o necesitan reestructuraciones: Sedatu. El Economista. [Consultado el 9 de noviembre de 2024]. https://www.eleconomista.com.mx/econohabitat/En-Mexico-8-de-cada-10-viviendas-carecen-de-servicios-basicos-o-necesitan-reestructuraciones-Sedatu-20230420-0135.html
- Nations, U. (2022). Conectar los gobiernos con los ciudadanos | Naciones Unidas. United Nations. [Consultado el 10 de noviembre de 2024]. https://www.un.org/es/desa/connecting-governments-to-citizens-2
- Firebase Cloud Messaging. (2023). Firebase. [Consultado el 11 de noviembre de 2024]. https://firebase.google.com/docs/cloud-messaging?hl=es-419
- websa100. (2019, March 19). Notificaciones push. Qué son y por qué debes usarlas - SEOptimer. Seoptimer.com. [Consultado el 12 de noviembre de 2024]. https://www.seoptimer.com/es/blog/notificaciones-push-que-son-por-que-usarlas/
- Cómo guardar datos con SQLite. (2024). Android Developers. [Consultado el 13 de noviembre de 2024]. https://developer.android.com/training/data-storage/sqlite?hl=es-419
- (2022, August 4). Bases de datos para Android [Top 3 de 2024] | KeepCoding. KeepCoding Bootcamps. [Consultado el 15 de noviembre de 2024]. https://keepcoding.io/blog/3-bases-de-datos-para-android/