**Instituto Politécnico Nacional**

**Centro de Estudios Científicos Y Tecnológicos**

**No. 12 “José María Morelos”**

Anleu Esparza Angel Jafed 2023120047

Chavez Martinez Jorge 2023120236

Cruz Velasco Alyson Ivanna 2022120232

Garcia Solorzano Jimena 2023120478

López Roldan Santiago 2023120663

5IV11

Tecnico en Informatica

Programación de dispositivos móviles

Luis Antonio Hernandez Cervantes

Aplicación Móvil: Calendario Medicinal

CDMX a 27 de Septiembre del 2024

Índice

Descripción del proyecto………………………………………………………2

Motivación……………………………………………………………………..2

Objetivos del Proyecto………………………………………………………...2

Alcance del Proyecto………………………………………………………...3,4

Investigación y Justificación………………………………………………5,6,7

**Descripción del proyecto**

**✓ Resumen de la aplicación móvil propuesta, incluyendo la idea**

**principal y objetivo***s.*

Esta aplicación tiene como base ayudar al usuario a recordar el horario de sus tratamientos, es decir, le llegará una notificación a las determinadas horas para recordarle tomar el medicamento según lo que el doctor le recete.

**Motivación**

**✓ Razones para elegir esta aplicación y su relevancia o problema que**

**busca resolver.**

Esta aplicación tiene como motivación apoyar y recordar en qué momentos debemos de suministrar el medicamento, ya que, como humanos solemos olvidar tomar a las horas sugeridas o continuar con el tratamiento, por lo que, esta aplicación nos facilita el proceso indicado por el médico.

**Objetivos del Proyecto**

***✓ Objetivo General:***

*i. Descripción del propósito general de la aplicación:*

Hacerle llegar al usuario una notificación recordando que debe de tomar su medicamento, así como, guardar y notificar cuando tenga que acudir a cita médica , también poner una alarma y apagarla solo hasta que el usuario indique que ya se haya tomado el medicamento y un acceso directo al número de emergencias, en caso de un imprevisto

***✓ Objetivos Específicos:***

*i. Metas concretas que se buscan alcan*z*ar con el proyecto:*

Recordar, notificar y administrar al usuario sus citas y tratamientos médicos.

**Alcance del Proyecto**

**✓ Funcionalidades Principales:**

i. Lista de las características y funciones clave que debe tener la aplicación.

* + **Registro de Medicamentos**:
    1. Permitir a los usuarios añadir medicamentos con información como nombre, dosis, frecuencia (por ejemplo, "una vez al día" o "cada 8 horas") y duración del tratamiento.
  + **Calendario Simple e Interactivo**:
    1. Visualización de calendario o lista mensual/semanal con las dosis de los medicamentos programados por día.
    2. Opción de agregar citas médicas y eventos relacionados.
  + **Notificaciones de Recordatorio**:
    1. Recordatorios simples y personalizados para avisar al usuario cuando debe tomar su medicamento.
    2. Posibilidad de posponer la notificación o marcar como iniciado o tomado dicho tratamiento.
    3. Opción para establecer recordatorios para renovar recetas o acudir a citas médicas.
  + **Historial de Medicación**:
    1. Registro básico de la medicación tomada con fechas y horas.
  + **Perfil de Usuario**:
    1. Datos personales como nombre, edad y fecha de nacimiento, estatura, peso o talla, tipo de sangre, alergias, registro de enfermedades crónicas, transmisibles, orgánicas o infecciosas que limiten físicamente al usuario.

### **Funciones Adicionales**

1. **Configuración de Notificaciones**:
   * Opción para activar o desactivar notificaciones.
2. **Edición y Eliminación de Medicamentos**:
   * Posibilidad de modificar o eliminar medicamentos ya registrados.
3. **Notas Breves**:
   * Opción para agregar notas/recomendaciones simples a cada medicamento (por ejemplo, “tomar con agua” o “evitar consumir lácteos”).

### **Seguridad Básica**

4. **Acceso directo con el 911**

* + Botón que al ser presionado llame directamente al número de emergencia 911**.**

5. **Protección por PIN**:

* + Opción de agregar un PIN simple para proteger el acceso a la aplicación.

### **Limitaciones Técnicas**

1. **Sincronización en la Nube**:
   * La versión básica de la aplicación no contará con sincronización en la nube o respaldo de datos, lo que implica que si el usuario cambia de dispositivo o lo pierde, también perderá la información guardada.
2. **Compatibilidad Limitada**:
   * La aplicación estará diseñada solo para dispositivos Android, lo que limita su uso para usuarios de iOS o aquellos que deseen utilizar la aplicación en diferentes plataformas.
3. **Notificaciones Dependientes del Sistema Operativo**:
   * Las notificaciones dependerán de la gestión de energía y políticas de optimización de batería de Android, lo que puede hacer que algunas notificaciones no se entreguen en tiempo real si el dispositivo está en modo de ahorro de energía.

**Investigación y Justificación**

***✓ Investigación Preliminar:***

*i. Análisis de Aplicaciones similares*

Dentro las aplicaciones en el mercado, se identificaron varias herramientas populares para la gestión de medicamentos, como Medisafe, MyTherapy, Pill Reminder y Dosecast.

Estas aplicaciones tienen características que para el usuario son esenciales, como el registro de medicamentos, notificaciones recordatorios, y en algunas aplicaciones, seguimiento de síntomas o la posibilidad de informarle al médico. Sin embargo, la mayoría de las aplicaciones tiene funcionalidades que para el usuario no son necesarias, lo cual le puede complicar la experiencia de usuarios.

*ii. Tendencia del Mercado*

Las aplicaciones relacionadas con la salud han estado creciendo en popularidad, especialmente con el aumento de la tecnología de salud móvil (myHealth). Algunas tendencias incluyen:

* Interfaz amigable para adultos mayores y personas con enfermedades crónicas.
* Notificaciones personalidades, tanto en frecuencia como en tipo (sonido, vibración, etc).
* Acceso sin conexión para usuarios en áreas de baja conectividad o que prefieren evitar el uso de datos.
* Simplicidad en el diseño, enfocada en resolver problemas específicos sin sobrecargar al usuario con funciones innecesarias.

Otra tendencia es la privacidad y seguridad de los datos de la salud, lo que ha impulsado el uso de PINs, autenticación biométrica y encriptada de datos en las aplicaciones avanzadas.

*iii. Tecnologías Relevantes*

* Lenguaje de programación: Kotlin y java, los dos lenguajes más comunes para desarrollar aplicaciones nativas en android.
* Notificaciones: Uso del servicio AlarmManager o WorkManager de android para gestionar notificaciones recurrentes, incluso cuando la aplicación no está activa.
* Base de datos: SQlite como base de datos local para almacenar información de los medicamentos, debido a su capacidad para funcionar sin conexión y su simplicidad de implementación.
* Diseño de Interfaz: El enfoque en Material Design, que ofrece una interfaz moderna, intuitiva y consistente en dispositivos Android.

***✓ Justificación:***

*i. Simplicidad y Usabilidad*

Basado en la investigación, una de las principales decisiones fue optar por una interfaz simple y amigable, eliminando funciones complejas que podrían ser innecesarias para la mayoría de usuarios. Muchas aplicaciones actuales ofrecen una amplia variedad de opciones (seguimiento de síntomas, informes detallados, etc), lo que puede ser confuso o innecesario para aquellos que buscan solo un recordatorio básico, por ello, el diseño se enfocará en funciones claves como:

* Registro simple de medicamentos.
* Notificaciones claras y personalizables.
* Historial básico.

Este enfoque responde a la necesidad de atender a los usuarios que no requieren funcionalidades avanzadas, como las que sí están presenten en aplicaciones más complejas, como MyTherapy o CareClinic.

*ii.Accesibilidad*

La aplicación se enfoca en la accesibilidad para los adultos mayores, ya que muchas aplicaciones actuales tienen interfaces complicadas que no siempre son amigables para este grupo de usuarios. La aplicación contará con una interfaz sencilla, con fuentes más grandes, y botones claros para facilitar la navegación.

*iii. funcionalidades sin conexión*

El permitir el uso sin conexión se debe a los usuarios que se encuentran en áreas con conectividad limitada o prefieren evitar el uso de datos móviles. Esto se va a lograr mediante SQlite (A través del código en java) como una base de datos local, permitiendo que los medicamentos, recordatorios y el historial estén siempre accesibles, sin necesidad de sincronización constante con la nube.

*iv. Enfoque en notificaciones*

Las notificaciones son el núcleo de la aplicación, por lo que se decide por la implementación de alarmManager o WorkManager (envia a traves de uso de java, una notificación en una hora determinada), a través de la clase *NotificationCompat.Build*e.

Se crea un canal de notificación.

Se construye la notificación usando *NotificationCompat.Builder*.

Para enviar la notificación, se usa *NotificationManagerCompat* para que llegue la notificación.

*AlarmManager* programa tareas en horarios específicos, como enviar una notificación en una hora determinada.

*BroadcastReceiver* será el encargado de disparar la notificación cuando el AlarmManager lo active.

*WorkManager* es más adecuado para tareas en segundo plano o notificaciones repetitivas, ya que maneja eficientemente la gestión de batería y condiciones de red.

**Planificación del Proyecto**

***✓ Cronograma***

*i. Calendario de actividades e hitos importantes, incluyendo fechas de entregas intermedias y finales.*

| **Fecha:** | **Parte:** | **Contenido** |
| --- | --- | --- |
| Viernes 4 de octubre | 1 | 🖭 Documento donde se  establezcan las bases del  proyecto. |
| Miércoles 20 de  noviembre | 2 | 🖭 Documentación  🖭 Avances de la aplicación. |
| 6 al 10 de enero 2025 | 3 | 🖭 Presentación final del  proyecto.  🖭 Documentación completa.  🖭 Pruebas en tiempo real. |

***✓ Recursos Necesarios:***

*i. Herramientas, tecnologías y otros recursos necesarios para el desarrollo.*

### **1. Herramientas de Desarrollo**

#### *a) Android Studio*

* **Descripción**: Es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones Android. Proporciona todo lo necesario para desarrollar, probar y depurar aplicaciones.
* **Funcionalidades clave**:
  + Editor de código avanzado con soporte para Kotlin y Java.
  + Simuladores para probar la app en diferentes dispositivos y versiones de Android.
  + Integración con Gradle para automatizar la compilación y administración de dependencias.
  + Herramientas de diseño visual para construir interfaces gráficas mediante **XML** o **Jetpack Compose**.

#### **b) Java**

* **Descripción**: Java es el lenguaje más tradicional para el desarrollo Android y sigue siendo ampliamente utilizado. Aunque Kotlin es recomendado, Java es una alternativa válida.
* **Ventajas**: Es ampliamente conocido, bien documentado y compatible con la mayoría de las bibliotecas de Android.

#### **c) Git y GitHub**

* **Descripción**: Git es un sistema de control de versiones que permite realizar un seguimiento de los cambios en el código fuente, mientras que GitHub es una plataforma para alojar proyectos.
* **Usos**:
  + Control de versiones para colaborar de forma eficiente y mantener un historial de cambios.
  + Almacenamiento de repositorios en línea para compartir el código con otros colaboradores.
  + Creación de ramas para trabajar en nuevas funcionalidades sin afectar la versión principal de la aplicación.

### 

### **2. Tecnologías para Funcionalidades Específicas**

#### **a) SQLite**

* **Descripción**: Es una base de datos relacional ligera y local que viene integrada en Android.
* **Usos**:
  + Almacenamiento local de datos de medicamentos, horarios y registros de toma de medicinas.
  + Permite consultar, insertar, actualizar y eliminar datos de manera eficiente, sin necesidad de conexión a Internet.

#### **b) Firebase (opcional)**

* **Descripción**: Plataforma de Google que ofrece una variedad de servicios como base de datos en tiempo real, autenticación de usuarios y análisis.
* **Usos**:
  + **Firebase Realtime Database**: Si deseas sincronizar los datos de medicamentos entre diferentes dispositivos o en la nube.
  + **Firebase Cloud Messaging (FCM)**: Para enviar notificaciones push a los usuarios de forma remota.

#### **c) AlarmManager y WorkManager**

* **Descripción**: APIs de Android para gestionar tareas programadas y notificaciones.
* **Usos**:
  + **AlarmManager**: Programar notificaciones locales en momentos específicos para recordarle al usuario tomar sus medicamentos.
  + **WorkManager**: Manejar tareas en segundo plano, como recordatorios de medicamentos, incluso cuando la aplicación está cerrada o el dispositivo está en modo de ahorro de batería.

#### **d) NotificationCompat**

* **Descripción**: Clase utilizada para crear y mostrar notificaciones en dispositivos Android, compatible con versiones anteriores.
* **Usos**: Enviar notificaciones personalizadas que alerten al usuario sobre la toma de medicamentos.