



Primera Fase Proyecto

MyPlanner: calendario con notas.

Integrantes:

Heysel Guadalupe Argueta Hernández AH230907

Ashley Nicole Aguilar Ramírez AR232346

Contenido

Introducción	3
PERFIL DE PROYECTO	4
Descripción General:.....	4
Objetivos del Proyecto:	4
Características Principales:	5
Tecnologías y Herramientas:	6
Plan de Desarrollo:	6
Cronograma Estimado:	8
Herramientas a utilizar	8
Diagrama grafico del diseño de la arquitectura de software para la aplicación móvil	9
Cronograma del trabajo.....	9
Mockups:	13
Diagrama Uml	15
Relaciones entre las Clases:	15
Descripción del Diagrama:	16

Introducción

La gestión eficiente del tiempo y las actividades es esencial para mantener un equilibrio en la vida moderna. Desde profesionales ocupados hasta estudiantes y personas con múltiples responsabilidades, la capacidad de organizar y recordar tareas puede marcar una gran diferencia en la productividad y el bienestar. En este contexto, las aplicaciones de calendario han emergido como herramientas clave para facilitar la planificación y el seguimiento de actividades. Con esto en mente, presentamos MyPlanner, una aplicación diseñada para ofrecer una solución integral y accesible para la gestión de tu tiempo. MyPlanner permite a los usuarios visualizar su calendario, agregar, eliminar y modificar actividades, y recibir notificaciones oportunas sobre eventos próximos. Y al agregar actividades, estas se verán marcadas en el calendario, facilitando aún más la organización de tareas diarias. Además, cuenta con una función de inicio de sesión seguro, garantizando la protección de tu información personal.

PERFIL DE PROYECTO

Nombre del Proyecto: MyPlanner

Descripción General:

El proyecto consiste en desarrollar una aplicación móvil de agenda personal para dispositivos Android. Esta aplicación permitirá a los usuarios gestionar sus actividades diarias mediante la creación de notas y recordatorios con alertas configurables. La aplicación utilizará Firebase como backend para garantizar la seguridad de los datos, sincronización en tiempo real, y permitirá a los usuarios acceder a su información desde cualquier dispositivo Android.

Objetivos del Proyecto:

1. Organización Eficiente: Facilitar a los usuarios la organización de sus tareas diarias y recordatorios importantes en un solo lugar.
2. Accesibilidad: Proveer acceso a las notas y recordatorios desde cualquier dispositivo Android, garantizando que los datos estén siempre actualizados gracias a la sincronización en la nube.
3. Seguridad y Privacidad: Asegurar la privacidad de los datos de los usuarios mediante la autenticación segura con Firebase Authentication y almacenamiento cifrado de datos en Firebase.
4. Experiencia de Usuario: Ofrecer una experiencia de usuario intuitiva, con un diseño visualmente atractivo y funcionalidades fáciles de usar.

Características Principales:

1. Gestión de Notas:

- Crear y Editar Notas: Los usuarios pueden crear nuevas notas con títulos y descripciones detalladas. Además, pueden editar notas existentes.
- Etiquetas y Categorías: Los usuarios pueden organizar sus notas mediante etiquetas o categorías personalizables.

2. Recordatorios y Alertas:

- Configurar Recordatorios: Los usuarios pueden establecer recordatorios para tareas importantes.
- Notificaciones Push: Las alertas se enviarán mediante notificaciones push, recordando a los usuarios sobre eventos o tareas pendientes.

3. Sincronización en la Nube:

- Firebase Realtime Database/Firestore: Utilización de Firebase para almacenar las notas y recordatorios de manera que se sincronicen en tiempo real entre dispositivos.
- Autenticación Firebase: Autenticación mediante correo electrónico y contraseña, Google Sign-In, o redes sociales para asegurar que cada usuario acceda solo a sus propios datos.
- Almacenamiento Seguro: Implementación de reglas de seguridad de Firebase para asegurar que los datos de cada usuario estén protegidos y cifrados.

4. Interfaz de Usuario (UI/UX):

- Diseño Basado en Material Design: Interfaz moderna que sigue las guías de Material Design, garantizando una experiencia de usuario coherente y atractiva.
- Tema Oscuro/Claro: Opción para cambiar entre tema claro y oscuro, adaptable a las preferencias del usuario o configuración del dispositivo.

5. Funciones Adicionales:

- Integración con Calendario: Sincronización de los recordatorios con el calendario del dispositivo Android para una mejor planificación.

Tecnologías y Herramientas:

- Lenguaje de Programación: Kotlin o Java
- IDE: Android Studio
- Backend: Firebase (Authentication, Firestore/Realtime Database, Cloud Messaging)
- Interfaz de Usuario: Material Design, XML Layouts
- Notificaciones: Firebase Cloud Messaging (FCM), AlarmManager
- Pruebas: JUnit, Espresso para pruebas unitarias y de interfaz
- Control de Versiones: Git/GitHub para control de versiones y colaboración

Arquitectura del Proyecto:

1. Arquitectura MVVM (Model-View-ViewModel):

- Modelo (Model): Representa los datos de la aplicación, incluyendo las notas y recordatorios, almacenados en Firebase.
- Vista (View): La interfaz de usuario (XML layouts) que interactúa con el ViewModel.
- ViewModel: Maneja la lógica de presentación y la comunicación entre la vista y el modelo, asegurando una separación clara de responsabilidades.

2. Servicios y Repositorios:

- Repositorio de Datos: Manejo de las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en Firebase.
- Servicio de Notificaciones: Implementación de servicios en segundo plano para gestionar las alertas y notificaciones.

Plan de Desarrollo:

1. Fase de Diseño (1 semana):

- Análisis de Requisitos: Definir en detalle los requerimientos funcionales y no funcionales del proyecto.
- Diseño de la Interfaz: Creación de wireframes y mockups de las pantallas principales.

- Esquema de Base de Datos: Diseñar la estructura de la base de datos en Firebase, incluyendo las colecciones y documentos necesarios.

2. Fase de Desarrollo (7 semanas):

- Configuración del Proyecto: Configuración inicial de Android Studio y Firebase, incluyendo la integración de las dependencias necesarias.

- Implementación de Funcionalidades Básicas:

- Creación y edición de notas.

- Configuración de recordatorios y alertas.

- Integración con Firebase:

- Configuración de Firebase Authentication y Firestore/Realtime Database.

- Sincronización de datos entre dispositivos.

- Desarrollo de Interfaz de Usuario: Implementación de las pantallas de la aplicación siguiendo los principios de Material Design.

3. Fase de Pruebas (1 semana):

- Pruebas Unitarias: Implementación de pruebas unitarias para asegurar la correcta funcionalidad de los componentes individuales.

- Pruebas de Integración: Verificación de la integración entre los diferentes módulos y la correcta sincronización con Firebase.

- Pruebas de Usabilidad: Realización de pruebas con usuarios para garantizar una experiencia de usuario óptima.

- Pruebas de Rendimiento: Evaluación del rendimiento de la aplicación en diferentes dispositivos Android.

4. Fase de Lanzamiento (1 semana):

- Preparación para el Lanzamiento: Revisión final del código, optimización y corrección de errores.

Cronograma Estimado:

- Fase de Diseño: 1 semanas
- Fase de Desarrollo: 7semanas
- Fase de Pruebas: 1 semanas
- Lanzamiento: 1 semana

Herramientas a utilizar

Android Studio: Como IDE (Integrated Development Environment), Android Studio es altamente funcional y versátil que ofrece herramientas útiles para el desarrollo de aplicaciones Android, así como también para realizar las pruebas emuladas de la aplicación.

Firebase: Como base de datos se utilizará Firebase Realtime Database para almacenar la información de los usuarios de la aplicación.

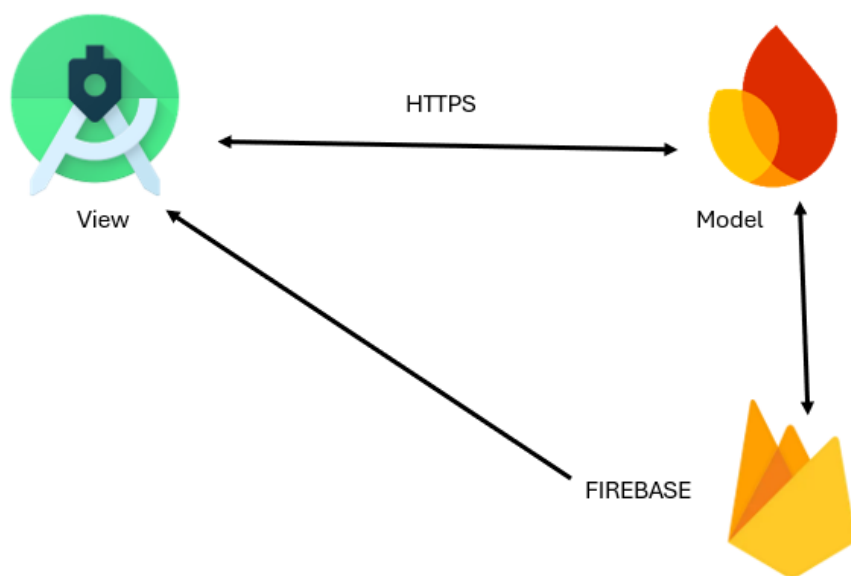
GitHub: Utilizaremos GitHub como plataforma de control de versiones para almacenar y gestionar el código fuente del proyecto de forma colaborativa. GitHub nos permitirá trabajar en equipo, realizar seguimiento de cambios, gestionar problemas y solicitudes de extracción, y garantizar la integridad y disponibilidad del código.

Notion: Utilizaremos Notion como una herramienta de gestión de proyectos para organizar y planificar el desarrollo de la aplicación. Notion nos permitirá hacer seguimiento de tareas, documentar el proyecto y colaborar de forma efectiva con el equipo de desarrollo.

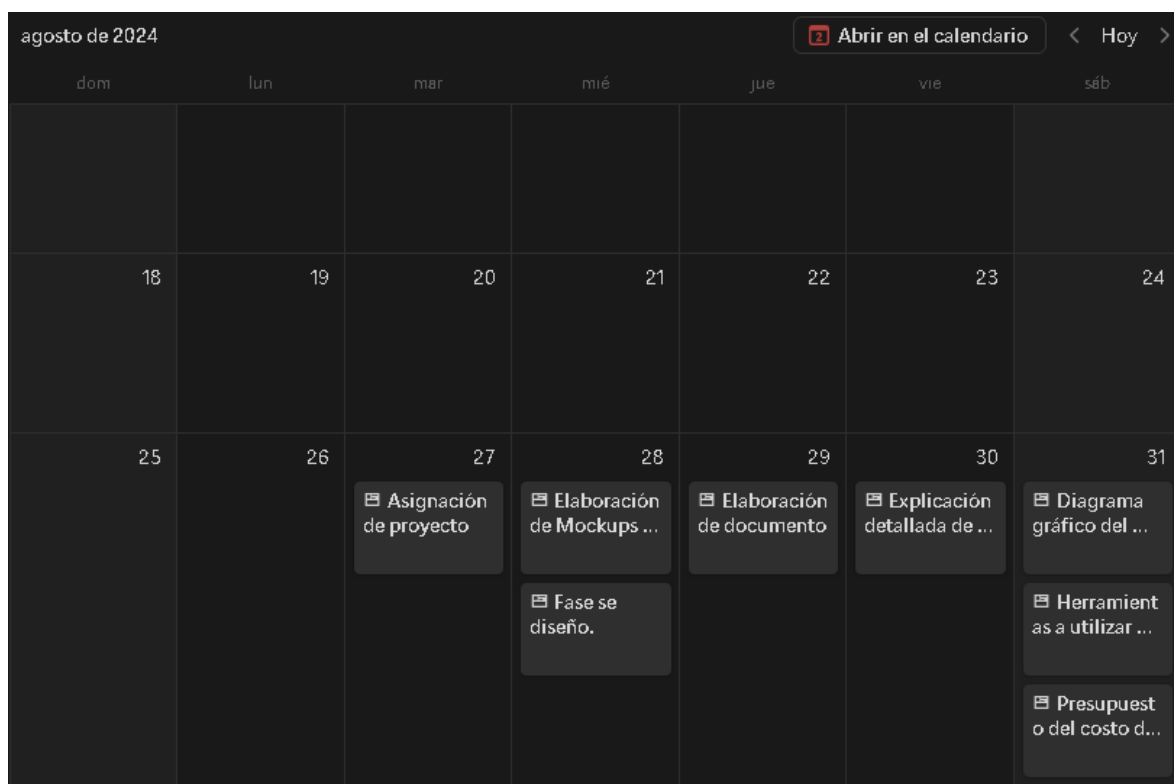
Flaticon: Como librería de iconos para ser utilizados en nuestra aplicación.

Figma: Para trabajar el diseño UX/UI con la realización de mock ups.

Diagrama grafico del diseño de la arquitectura de software para la aplicación móvil



Cronograma del trabajo



septiembre de 2024							Abrir en el calendario		<	Hoy	>
dom	lun	mar	mié	jue	vie	sáb					
	Fase de desarrollo.										
15	16	17	18	19	20	21					
	Trabajar en la base de datos tanto en la gestión y modelo de datos.										
22	23	24	25	26	27	28					
	Implementación de código en Android Studio.										
29	30	1 oct	2	3	4	5					
	Ejecutar y ver errores posibles.		Solucionar errores posibles.								

septiembre de 2024							Abrir en el calendario		<	Hoy	>
dom	lun	mar	mié	jue	vie	sáb					
1 sept	2	3	4	5	6	7					
Finalización de documento	Entrega de primera fase										
8	9	10	11	12	13	14					
	Implementar lógica de negocio - Servidores y Frameworks Backend.										
	Fase de desarrollo.										
15	16	17	18	19	20	21					
	Trabajar en la base de datos tanto en la gestión y modelo de datos.										

<div> <div>☰ All Contents</div> <div> <div>Drafts</div> <div>Planned</div> <div>Published</div> </div> </div>		
Aa Name	Channels	<div>Planned</div> <div>Published</div>
 <u>Asignación de proyecto</u>		27 de agosto de 2024
 <u>Elaboración de Mockups y paleta de colores.</u>		28 de agosto de 2024
 <u>Fase se diseño.</u>		28 de agosto de 2024
 <u>Elaboración de documento</u>		29 de agosto de 2024
 <u>Explicación detallada de la lógica a utilizar</u>		30 de agosto de 2024
 <u>Diagrama gráfico del diseño de la arquitectura de software.</u>		31 de agosto de 2024
 <u>Herramientas a utilizar durante el desarrollo</u>		31 de agosto de 2024
 <u>Presupuesto del costo de la aplicación.</u>		31 de agosto de 2024
 <u>Finalización de documento</u>	<div>ABRIR</div>	1 de septiembre de 2024
 <u>Finalización de documento</u>		1 de septiembre de 2024
 <u>Entrega de primera fase</u>		2 de septiembre de 2024
 <u>Fase de desarrollo.</u>		9 de septiembre de 2024
 <u>Implementar lógica de negocio - Servidores y Frameworks Backend.</u>		9 de septiembre de 2024 → 13 de septiembre de 2024
 <u>Trabajar en la base de datos tanto en la gestión y modelo de datos.</u>		16 de septiembre de 2024 → 20 de septiembre de 2024
 <u>Implementación de código en Android Studio.</u>		23 de septiembre de 2024 → 27 de septiembre de 2024
 <u>Ejecutar y ver errores posibles.</u>	<div>ABRIR</div>	30 de septiembre de 2024 → 1 de octubre de 2024

2024

Implementación de código en Android Studio.

23 de septiembre de 2024 → 27 de septiembre de 2024

Ejecutar y ver errores posibles.

30 de septiembre de 2024 → 1 de octubre de 2024

Solucionar errores posibles.

2 de octubre de 2024 → 4 de octubre de 2024

Fase de pruebas finales.

7 de octubre de 2024

Últimos arreglos al proyecto

10 de octubre de 2024 → 13 de octubre de 2024

Entrega de proyecto final.

15 de octubre de 2024

+ Nuevo

Cerrar Clic o Ctrl+X
Redimensionar Arrastrar

areas

Por proyecto Tablero Todas las tareas +

Primeros pasos en Proyectos y tareas 4

Aa Proyecto de cátedra - DSM

Estado

Responsable

Con retraso

Prioridad

Resumen

IA

David	Listo	david.jaimesantos	2 de septiembre de 20	Alta	Elaboración de perfil de proyecto
Heysel	Listo		8 de septiembre de 20	Alta	Elaborar las herramientas de desarrollo y Mockups
Nicole	Listo			Alta	Elaboración de Introducción de presupuesto de pri
Amilcar	Listo	Carlos Mejia		Alta	Elaborar diagramas UML

Mockups:



Iniciar sesión

Logo

Nombre

Usuario

Contraseña

Ingresar

¿No tienes una cuenta?
Registrarse

Registrarse

Logo


Nombre

Correo

Contraseña

Registrarse

Usuario



Nombre de usuario

Cerrar sesión





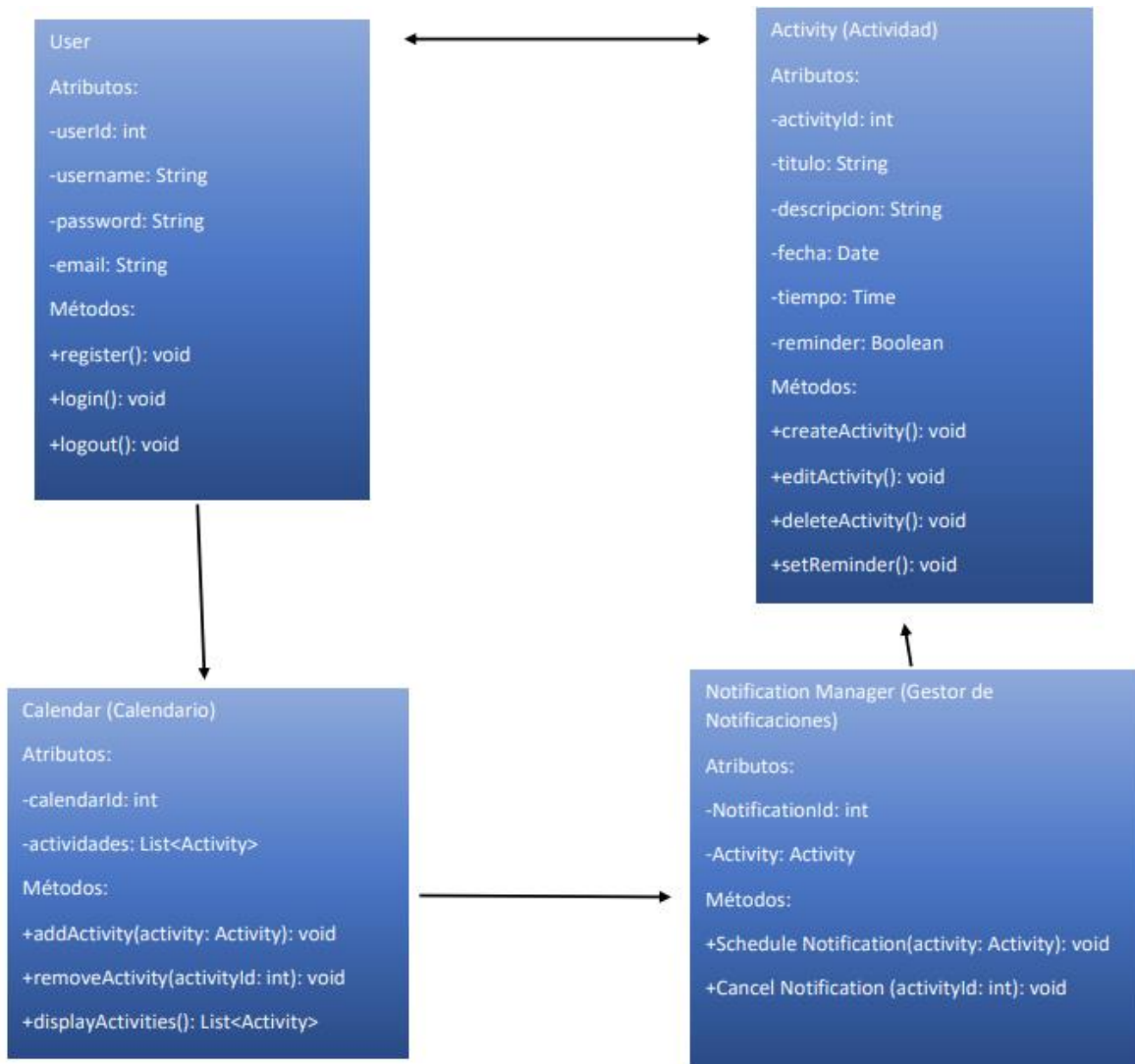
    

Diagrama Uml



Relaciones entre las Clases:

User puede tener muchas Activity.

Activity pertenece a un Calendar.

Calendar contiene muchas Activity.

NotificationManager está relacionado con Activity para manejar los recordatorios.

Descripción del Diagrama:

User es la clase que gestiona el registro e inicio de sesión de los usuarios.

Activity representa una tarea o evento que el usuario puede crear, editar, o eliminar.

Calendar es la clase que agrupa todas las actividades y las muestra en un formato de calendario.

NotificationManager maneja la programación de alertas y recordatorios para las actividades registradas.