

**Desarrollo de Aplicaciones Multiplataforma Online**

****

**Proyecto Final de Ciclo de**

**Desarrollo de Aplicaciones Multiplataforma Online**

**Título:**

**Apellidos y nombre del autor/a: Sergio Bello Gonzalo**

**Curso Académico: 2º DAM**

**Fecha Entrega: 18/05/2024**

Contenido

[1 Resumen del proyecto 3](#_Toc166950570)

[1.1 ¿Qué se propone? 3](#_Toc166950571)

[1.2 ¿De qué está formado? 4](#_Toc166950572)

[2 Justificación y objetivos del proyecto 4](#_Toc166950573)

[2.1 Qué se propone 4](#_Toc166950574)

[2.2 A quién se dirige 5](#_Toc166950575)

[2.3 Qué se pretende alcanzar 5](#_Toc166950576)

[3 Desarrollo del proyecto 5](#_Toc166950577)

[3.1 Metodologías utilizadas 5](#_Toc166950578)

[3.2 Descripción de los componentes de la aplicación 6](#_Toc166950579)

[3.2.1 Arquitectura 6](#_Toc166950580)

[3.2.2 Frontend 8](#_Toc166950581)

[3.2.3 Backend 11](#_Toc166950582)

[3.2.4 Tecnologías utilizadas 15](#_Toc166950583)

[3.3 Problemas/Dificultades y soluciones 16](#_Toc166950584)

[3.4 Resultados obtenidos 19](#_Toc166950585)

[4 Conclusiones 20](#_Toc166950586)

[5 Bibliografía 20](#_Toc166950587)

# **Resumen del proyecto**

La aplicación propuesta es una herramienta de toma de notas diseñada para ofrecer una experiencia simple y funcional a los usuarios de dispositivos Android y iOS. En un mercado saturado de aplicaciones con características excesivas y modelos de monetización complicados, nuestra aplicación se destaca por su enfoque en la simplicidad y la utilidad.

Esta aplicación busca llenar un vacío en el mercado al ofrecer una alternativa sin complicaciones para aquellos que buscan una forma eficiente de tomar notas en sus dispositivos móviles. Al eliminar características superfluas y evitar modelos de monetización intrusivos, nuestra aplicación se centra en las necesidades básicas de los usuarios, proporcionando una experiencia de usuario intuitiva y satisfactoria.

## ¿Qué se propone?

* **Simplicidad en el diseño y la funcionalidad**: La aplicación se centrará en una interfaz de usuario limpia y fácil de entender, minimizando la cantidad de características superfluas y enfocándose en las funcionalidades básicas necesarias para tomar y organizar notas de manera eficiente.
* **Usabilidad intuitiva**: Se implementará una experiencia de usuario intuitiva que permita a los usuarios crear, editar y organizar sus notas de manera rápida y sin complicaciones. Esto incluirá funciones como la capacidad de crear nuevas notas con un solo toque y realizar búsquedas rápidas dentro de la aplicación.
* **Compatibilidad y escalabilidad**: La aplicación se desarrollará utilizando tecnologías modernas y estándares de la industria para garantizar la compatibilidad con una amplia gama de dispositivos Android y iOS. Además, se diseñará con la escalabilidad en mente, lo que permitirá agregar nuevas funcionalidades y mejoras en futuras actualizaciones.

## ¿De qué está formado?

* **Interfaz de Usuario Intuitiva**: Se desarrollará una interfaz de usuario (UI) intuitiva y fácil de usar que permita a los usuarios navegar por la aplicación de manera eficiente y acceder a todas las funcionalidades principales de manera clara y concisa.
* **Funcionalidades Básicas de Toma de Notas**: La aplicación contará con funcionalidades básicas para la creación, edición y eliminación de notas. Esto incluirá la capacidad de escribir texto, agregar. Además, también se podrá utilizar el formato Markdown para una mejor organización y visualización del contenido.
* **Sincronización y Respaldo de Datos**: Se explorarán opciones para permitir la sincronización y el respaldo de datos entre dispositivos, lo que garantizará que los usuarios puedan acceder a sus notas desde cualquier lugar y dispositivo compatible.
* **Compatibilidad con Dispositivos Android y iOS**: La aplicación se desarrollará teniendo en cuenta la compatibilidad con una amplia gama de dispositivos Android, lo que garantizará que pueda ejecutarse de manera eficiente en diferentes tamaños de pantalla y versiones del sistema operativo.

# **Justificación y objetivos del proyecto**

## Qué se propone

El proyecto se propone desarrollar una aplicación de toma de notas que destaque por su simplicidad y utilidad. Sin micropagos y ofreciendo una alternativa libre de complicaciones en el mercado actual.

## A quién se dirige

La aplicación se dirige a usuarios de dispositivos Android que buscan una herramienta de toma de notas que sea fácil de usar, sin características innecesarias y sin modelos de monetización intrusivos. Está diseñada para personas que valoran la simplicidad y la eficiencia en sus aplicaciones móviles.

## Qué se pretende alcanzar

* Simplicidad y Funcionalidad: Se pretende crear una aplicación que sea fácil de entender y de usar, con una interfaz intuitiva y funcionalidades básicas de toma de notas.
* Experiencia sin pagos abusivos: Se busca ofrecer una experiencia de usuario libre de distracciones y obstáculos, eliminando características superfluas y modelos de monetización intrusivos.

# **Desarrollo del proyecto**

## Metodologías utilizadas

Metodología Utilizada: **Scrum**

- Scrum es una metodología ágil que se enfoca en la entrega iterativa e incremental de software, lo que permite una adaptación flexible a los cambios en los requisitos del proyecto. En el contexto de nuestro proyecto de desarrollo de una aplicación de toma de notas, se seguirían los siguientes pasos:

* Sprints: Se dividiría el proyecto en sprints de duración fija (dos semanas). Al comienzo de cada sprint.
* Revisiones del Sprint: Al final de cada sprint, se llevaría a cabo una revisión del sprint donde se mostraría el trabajo realizado al Product Owner. Se recibirían retroalimentación y se ajustarían las prioridades del producto según sea necesario.
* Retrospectivas del Sprint: También al final de cada sprint, el equipo realizaría una retrospectiva para reflexionar sobre el proceso y identificar áreas de mejora. Se propondrían acciones concretas para optimizar el trabajo en futuros sprints.
* Iteración y Mejora Continua: Se repetirían los ciclos de sprint, revisión y retrospectiva hasta que se hayan implementado todas las funcionalidades planificadas para la aplicación. Se fomentaría la colaboración y la mejora.

## Descripción de los componentes de la aplicación

### Arquitectura

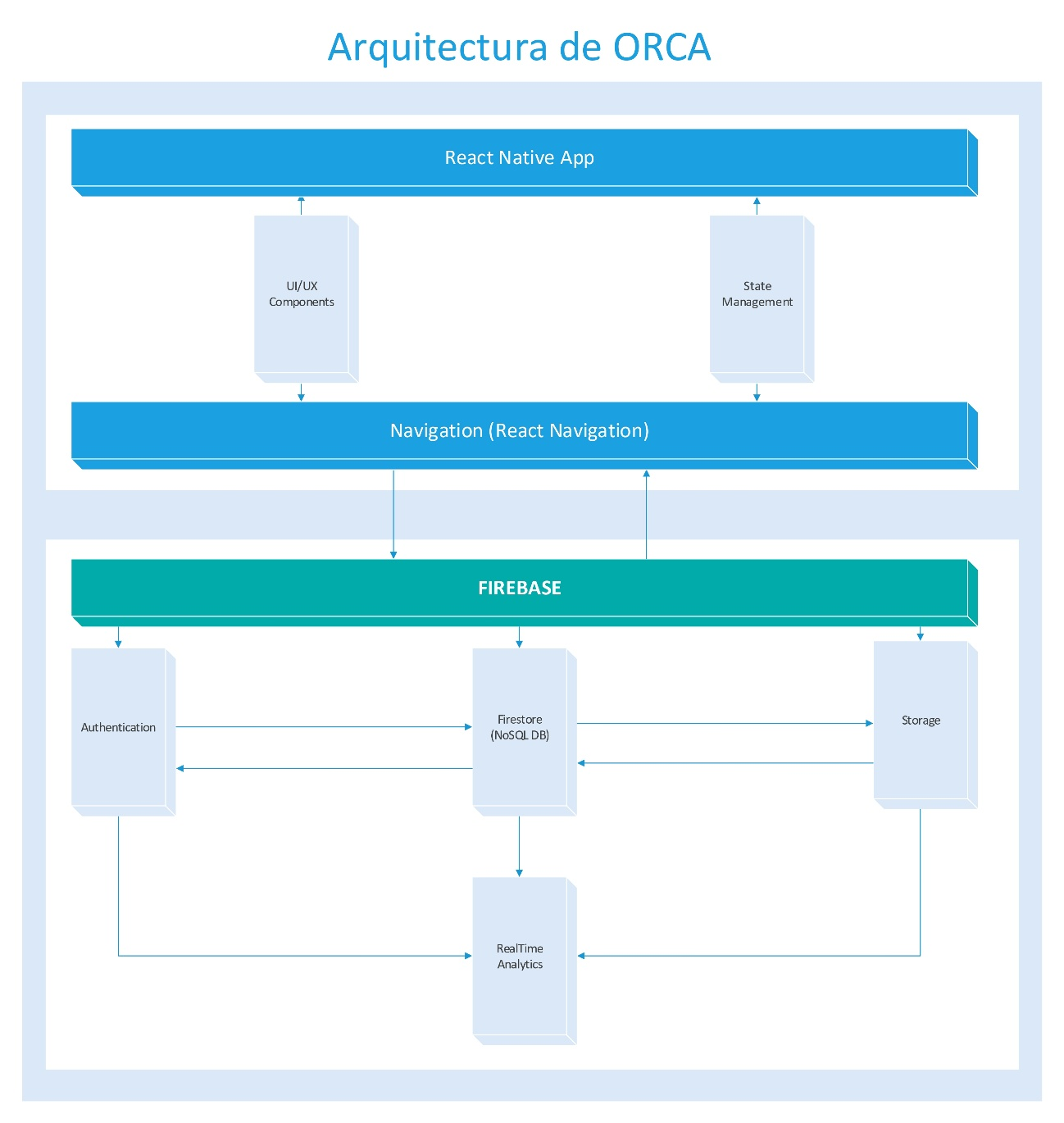


Figura 1:Arquitectura de ORCA

La arquitectura de nuestra aplicación móvil está diseñada para ofrecer una experiencia fluida y segura a los usuarios, aprovechando las tecnologías modernas disponibles en el ecosistema de desarrollo de aplicaciones móviles. Exploraremos los componentes principales de nuestra arquitectura, que incluyen React Native como framework principal para la interfaz de usuario y Firebase como plataforma integral para la gestión de datos y la autenticación.

I. React Native:

React Native es un marco de desarrollo de aplicaciones móviles que nos permite construir interfaces de usuario nativas utilizando JavaScript y React. En nuestra arquitectura, React Native desempeña un papel fundamental en la creación de componentes de UI/UX y en la gestión del estado de la aplicación.

UI/UX Components: Utilizamos React Native para desarrollar componentes de interfaz de usuario altamente personalizables y reutilizables, lo que nos permite mantener una apariencia coherente en toda la aplicación y mejorar la experiencia del usuario.

State Management: Implementamos técnicas de gestión de estado para administrar el estado de la aplicación de manera eficiente, asegurando una coherencia en los datos en toda la aplicación.

II. Navegación (React Navigation):

Para la navegación entre pantallas en nuestra aplicación, utilizamos React Navigation, una biblioteca popular en el ecosistema de React Native. React Navigation nos permite gestionar la navegación de forma intuitiva y flexible, ofreciendo una experiencia de usuario fluida al moverse dentro de la aplicación.

III. Firebase:

Firebase es una plataforma de desarrollo de aplicaciones móviles desarrollada por Google que ofrece una variedad de servicios integrales para satisfacer las necesidades de desarrollo de aplicaciones modernas.

Authentication: Utilizamos Firebase Authentication para gestionar la autenticación de usuarios de manera segura y sin problemas. Con Firebase Authentication, podemos ofrecer opciones de inicio de sesión seguras, como correo electrónico/contraseña, verificación de los mismos, autenticación social y más.

Firestore (NoSQL DB): Firestore es una base de datos NoSQL en la nube que utilizamos para almacenar y sincronizar datos en tiempo real entre los clientes de la aplicación. Con Firestore, podemos estructurar y consultar datos de manera eficiente, lo que permite una experiencia de usuario ágil y receptiva.

Storage: Firebase Storage nos proporciona un almacenamiento seguro y escalable para archivos como imágenes, videos y otros recursos multimedia. Utilizamos Firebase Storage para almacenar y recuperar archivos de manera eficiente, asegurando una experiencia de usuario sin problemas al cargar y compartir archivos en la aplicación.

Realtime Analytics: Firebase Analytics nos ofrece información valiosa sobre el rendimiento y la interacción de los usuarios dentro de la aplicación. Utilizamos Firebase Analytics para comprender el comportamiento del usuario, realizar análisis de la aplicación y tomar decisiones informadas para mejorar continuamente la experiencia del usuario.

### Frontend

React Native es un framework de desarrollo de aplicaciones móviles que permite a los desarrolladores crear aplicaciones para iOS y Android utilizando JavaScript y React. Utiliza el mismo diseño que React, lo que facilita a los desarrolladores crear interfaces de usuario declarativas y basadas en componentes. React Native permite compartir una gran cantidad de código entre plataformas, lo que agiliza el proceso de desarrollo y mantenimiento de aplicaciones móviles multiplataforma.

Esquema de pantallas:

* RegisterScreen (Registro):
  + Descripción: Esta pantalla permite a los usuarios registrarse en la aplicación ingresando su información, como correo electrónico y contraseña.
  + Cambio a esta pantalla: El usuario será dirigido a esta pantalla cuando seleccione la opción de registro desde la pantalla de inicio de sesión (LoginScreen).
* LoginScreen (Acceso):
  + Descripción: En esta pantalla, los usuarios pueden iniciar sesión en la aplicación ingresando su correo electrónico y contraseña.
  + Cambio a esta pantalla: El usuario será dirigido aquí cuando abra la aplicación por primera vez o cuando seleccione la opción de iniciar sesión desde la pantalla de registro (RegisterScreen).
* CreateNotes (Crear nota):
  + Descripción: Esta pantalla permite a los usuarios crear nuevas notas ingresando un título y un contenido para la nota.
  + Cambio a esta pantalla: Después de iniciar sesión correctamente, si el usuario no tiene ninguna nota previa, se le dirigirá automáticamente a esta pantalla para crear su primera nota. Además, el usuario puede acceder a esta pantalla desde la lista de notas (ListNotes) seleccionando la opción de crear nueva nota.
* ListNotes (Dashboard):
  + Descripción: En esta pantalla, se muestra una lista de todas las notas creadas por el usuario. Desde aquí, el usuario puede ver el contenido de cada nota y editarlas si es necesario.
  + Cambio a esta pantalla: Después de iniciar sesión correctamente y si el usuario tiene notas previamente creadas, se le dirigirá automáticamente a esta pantalla. Además, el usuario puede acceder a esta pantalla desde otras partes de la aplicación, como la pantalla de inicio (CreateNotes o ShowEditNotes).
* ShowEditNotes (Editar nota):
  + Descripción: Esta pantalla muestra el contenido completo de una nota seleccionada, permitiendo al usuario editar su contenido si lo desea.
  + Cambio a esta pantalla: El usuario puede acceder a esta pantalla desde la lista de notas (ListNotes) al hacer clic en una nota específica para ver o editar su contenido. También puede ser dirigido aquí después de crear una nueva nota para ver y editar su contenido recién creado.

Dependencias:

* **@firebase/auth** (^1.7.2): Módulo de autenticación de Firebase para React Native.
* **@react-native-community/datetimepicker** (^8.0.1): Componente de selector de fecha y hora para React Native.
* **@react-native-firebase/app** (^19.2.2): SDK principal de Firebase para React Native.
* **@react-native-firebase/auth** (^19.2.2): Módulo de autenticación de Firebase para React Native.
* **@react-navigation/native** (^6.1.17): Biblioteca de navegación para React Native.
* **@react-navigation/native-stack** (^6.9.26): Stack Navigator para React Navigation.
* **expo** (~50.0.14): Plataforma de desarrollo de aplicaciones universales para React Native y web.
* **expo-file-system** (~16.0.9): API de sistema de archivos para aplicaciones Expo.
* **expo-image-picker** (~14.7.1): Módulo de selección de imágenes para aplicaciones Expo.
* **expo-status-bar** (~1.11.1): Barra de estado para aplicaciones Expo.
* **firebase** (^10.11.1): SDK de Firebase para JavaScript.
* **react** (18.2.0): Biblioteca principal de React.
* **react-native** (0.73.6): Marco de desarrollo para crear aplicaciones móviles utilizando React.
* **react-native-safe-area-context** (4.8.2): Contexto de áreas seguras para React Native.
* **react-native-screens** (~3.29.0): Solución de navegación nativa para React Native.
* **react-native-svg** (^15.2.0): Biblioteca para renderizar SVG en React Native.
* **react-native-vector-icons** (^10.1.0): Conjunto de iconos personalizables para React Native.

### Backend

**Firebase** es una plataforma de desarrollo de aplicaciones móviles y web creada por Google que ofrece una variedad de servicios en la nube para ayudar a los desarrolladores a construir, mejorar y hacer crecer sus aplicaciones. En nuestro proyecto de toma de notas en React Native, vamos a utilizar Firebase para gestionar la autenticación de usuarios, almacenar y sincronizar las notas en tiempo real, almacenar archivos adjuntos relacionados con las notas y añadir una capa adicional de seguridad con Firebase App Check. En resumen, Firebase nos proporcionará las herramientas necesarias para implementar funcionalidades clave como el registro de usuarios, la gestión de notas y la seguridad de la aplicación de manera eficiente y confiable.

**Authentication** es un servicio que proporciona Firebase para manejar la autenticación de usuarios en aplicaciones móviles y web de forma segura y sencilla. En nuestro proyecto de toma de notas en React Native, utilizaremos Authentication para permitir que los usuarios creen cuentas y se autentiquen en la aplicación. Esto nos permitirá garantizar que solo usuarios autorizados puedan acceder a las funcionalidades de la aplicación, como la creación, edición y eliminación de notas. Esto nos ofrece una solución robusta y escalable para gestionar la identidad de los usuarios, lo que nos permite centrarnos en la creación de una experiencia de usuario fluida y segura sin tener que preocuparnos por la implementación y gestión de sistemas de autenticación complejos.

Campos que usaremos:

* **Correo electrónico** (Email): Este campo será utilizado por los usuarios para ingresar su dirección de correo electrónico durante el proceso de registro e inicio de sesión. Proporcionará la identificación única de cada usuario en nuestra plataforma y servirá como medio de comunicación para la recuperación de cuentas y notificaciones relacionadas con la autenticación.
* **Contraseña** (Password): El campo de contraseña permitirá a los usuarios establecer una credencial segura para proteger su cuenta. Durante el proceso de registro, los usuarios deberán crear una contraseña que cumpla con ciertos criterios de seguridad, como longitud mínima y complejidad. Durante el inicio de sesión, este campo se utilizará para verificar la identidad del usuario y proporcionar acceso a la cuenta asociada con la dirección de correo electrónico especificada.

**Firestore Database** es un servicio de base de datos NoSQL en tiempo real proporcionado por Firebase, que nos permite almacenar y sincronizar datos en la nube en tiempo real. En nuestro proyecto de toma de notas en React Native, utilizaremos Firestore Database para almacenar todas las notas creadas por los usuarios de manera organizada y segura.

Firestore nos permite estructurar los datos en colecciones y documentos. Cada nota de un usuario se almacenará como un documento dentro de una colección específica, lo que nos brinda flexibilidad para realizar consultas y manipular los datos de manera eficiente.

Al utilizar Firestore, podemos garantizar que las notas de los usuarios estén siempre sincronizadas entre diferentes dispositivos y plataformas, lo que proporciona una experiencia fluida y consistente para los usuarios. Además, Firestore ofrece funcionalidades avanzadas como consultas en tiempo real, reglas de seguridad personalizadas y escalabilidad automática, lo que nos permite enfocarnos en el desarrollo de nuestra aplicación sin preocuparnos por la infraestructura subyacente. En resumen, Firestore Database será la columna vertebral de nuestro proyecto, permitiéndonos almacenar y gestionar las notas de los usuarios de manera eficiente y segura en la nube.

Campos que usaremos:

* createdAt: Este campo indica la fecha y la hora en que se creó la nota. Proporciona información sobre cuándo se agregó la nota a la base de datos, lo que puede ser útil para ordenar y visualizar las notas por fecha de creación.
* description: Es una descripción o contenido de la nota. Este campo contiene el texto principal o el cuerpo de la nota que el usuario ha escrito para capturar sus pensamientos, ideas o tareas pendientes.
* imageUrl: Representa la URL de la imagen asociada a la nota, si la hay. Algunas notas pueden contener imágenes adjuntas para complementar la descripción o el contenido de la nota. Este campo almacena la ubicación de la imagen en Firebase Storage.
* title:Es el título de la nota. Proporciona un breve resumen o encabezado que identifica el tema o el propósito de la nota. El título ayuda a los usuarios a organizar y encontrar rápidamente sus notas.
* userId: Este campo identifica de manera única al usuario que creó la nota. Cada usuario tiene un ID único asociado con su cuenta en la aplicación. Este campo nos permite vincular la nota con el usuario correspondiente, lo que facilita la gestión y la personalización de las notas según el usuario que las creó.

**Firebase Storage** es un servicio de almacenamiento en la nube proporcionado por Firebase que nos permite almacenar y servir archivos estáticos, como imágenes, videos y otros archivos multimedia, de manera segura y eficiente. En nuestro proyecto de toma de notas en React Native, utilizaremos Firebase Storage para almacenar las imágenes adjuntas relacionadas con las notas de los usuarios.

Firebase Storage nos ofrece una infraestructura escalable y confiable para almacenar archivos de cualquier tamaño, desde pequeñas imágenes hasta archivos multimedia grandes. Utilizando Firebase Storage, podremos cargar imágenes desde los dispositivos de los usuarios y almacenarlas en la nube de forma segura, lo que garantiza la disponibilidad y la integridad de los archivos asociados a las notas.

Además, Firebase Storage nos proporciona URL únicas para cada archivo almacenado, lo que nos permite acceder y compartir fácilmente los archivos multimedia con los usuarios de la aplicación. Esto facilita la visualización de las imágenes adjuntas en las notas y mejora la experiencia de usuario en general.

En nuestro proyecto, utilizaremos Firebase Storage de la siguiente manera para gestionar las imágenes adjuntas a las notas de los usuarios:

* Carga de Imágenes: Implementaremos funcionalidades en nuestra aplicación que permitan a los usuarios cargar imágenes desde sus dispositivos. Utilizaremos las API proporcionadas por Firebase para cargar estas imágenes en Firebase Storage de forma segura.
* Almacenamiento en Firebase Storage: Una vez que los usuarios seleccionen una imagen para adjuntar a una nota, nuestra aplicación enviará esa imagen al servicio de Firebase Storage. Allí, la imagen se almacenará de manera segura en la nube de Firebase y se le asignará una URL única que nos permitirá acceder a ella posteriormente.
* Obtención de URL de Imagen: Después de que la imagen se haya cargado con éxito en Firebase Storage, nuestra aplicación recibirá la URL única asociada con esa imagen. Esta URL servirá como una referencia a la imagen almacenada en la nube y se utilizará para acceder a la imagen desde cualquier lugar donde se necesite en la aplicación.
* Asociación con las Notas: Finalmente, en la creación o edición de una nota, nuestra aplicación asociará esta URL de imagen con la nota correspondiente. Esto nos permitirá mostrar la imagen adjunta junto con el contenido de la nota cuando los usuarios visualicen sus notas en la aplicación.

**Firebase App Check** es una solución de seguridad proporcionada por Firebase que ayuda a proteger las aplicaciones y sus recursos en la nube contra el abuso y el fraude. En nuestro proyecto de toma de notas en React Native, utilizaremos Firebase App Check para agregar una capa adicional de seguridad y proteger nuestra aplicación contra posibles amenazas y actividades maliciosas.

Firebase App Check funciona verificando la autenticidad de las solicitudes entrantes a nuestros recursos de Firebase, como Firestore Database y Firebase Storage. Utiliza tecnologías de seguridad avanzadas, como la autenticación de dispositivos, para asegurarse de que las solicitudes provengan de aplicaciones legítimas y no de bots, scripts automatizados u otros agentes no autorizados.

En nuestro proyecto, integraremos Firebase App Check para garantizar que solo las aplicaciones auténticas, que hayan sido debidamente registradas y verificadas por Firebase, puedan acceder a nuestros recursos en la nube. Esto nos ayudará a proteger los datos de los usuarios, prevenir el acceso no autorizado y garantizar la integridad de nuestra aplicación en general.

### Tecnologías utilizadas

* **React Native**: Es un framework de desarrollo de aplicaciones móviles que nos permite construir aplicaciones nativas para iOS y Android utilizando JavaScript y React. Con React Native, podemos compartir una base de código entre las plataformas, lo que facilita el desarrollo y el mantenimiento de la aplicación.
* **Firebase**: Firebase es una plataforma de desarrollo de aplicaciones móviles y web que proporciona una variedad de servicios en la nube, incluyendo autenticación de usuarios, bases de datos en tiempo real, almacenamiento de archivos, mensajería en la nube, análisis y más. En este proyecto, hemos utilizado varios servicios de Firebase, como:
  + Firebase Authentication: Para gestionar la autenticación de usuarios en la aplicación.
  + Firebase Firestore: Para almacenar y sincronizar datos en tiempo real entre los usuarios de la aplicación.
  + Firebase Cloud Storage: Para almacenar archivos multimedia, como imágenes o vídeos, de forma segura en la nube.

## Problemas/Dificultades y soluciones

Gestión de la autenticación

La autenticación es una parte crucial en el desarrollo de aplicaciones móviles, ya que permite verificar la identidad de los usuarios y asegurar que solo personas autorizadas puedan acceder a ciertas partes de la aplicación. En este documento, se detalla el proceso de investigación llevado a cabo para gestionar la autenticación utilizando Firebase Auth en combinación con React Native.

Objetivos de la Investigación:

* Comprender los conceptos básicos de Firebase Authentication.
* Aprender a integrar Firebase Auth en una aplicación React Native.
* Implementar métodos de autenticación comunes, como correo electrónico y contraseña, así como proveedores externos como Google y Facebook.
* Asegurar y manejar la sesión de usuario dentro de la aplicación.

Etapas de la Investigación

1. Introducción a Firebase Authentication
   1. Recursos Utilizados:
      1. Documentación oficial de Firebase: Firebase Authentication
      2. Tutoriales en línea y videos en YouTube sobre la configuración básica de Firebase Auth.
   2. Actividades Realizadas:
      1. Lectura de la documentación oficial para entender los conceptos y las capacidades de Firebase Auth.
      2. Visualización de tutoriales que explican la configuración inicial de Firebase en el proyecto.
2. Configuración del Proyecto React Native
   1. Recursos Utilizados:
      1. Documentación de React Native: React Native Getting Started
      2. Guías sobre la integración de Firebase con React Native, como la biblioteca react-native-firebase.
   2. Actividades Realizadas:
      1. Creación de un nuevo proyecto React Native utilizando npx react-native init.
      2. Instalación y configuración de las dependencias necesarias para Firebase (@react-native-firebase/app y @react-native-firebase/auth).
      3. Configuración de Firebase en el proyecto siguiendo las guías oficiales y los tutoriales.

Crear logo de la aplicación

La creación de un logo es un paso fundamental para la identidad visual de cualquier marca o proyecto. En este documento, se detalla el proceso de diseño de un logo utilizando Canva, una herramienta en línea fácil de usar que ofrece una amplia variedad de recursos y plantillas.

Objetivos de la Investigación:

* Comprender las funcionalidades y herramientas que ofrece Canva para la creación de logos.
* Aprender a utilizar Canva para diseñar un logo desde cero o personalizar una plantilla existente.
* Exportar el logo en diferentes formatos según las necesidades del proyecto.

Etapas de la Investigación

1. Introducción a Canva
   1. Recursos Utilizados:
      1. Sitio web oficial de Canva
      2. Tutoriales y videos en YouTube sobre el uso básico de Canva.
      3. Guías de diseño gráfico enfocadas en la creación de logos.
   2. Actividades Realizadas:
      1. Creación de una cuenta en Canva y exploración de su interfaz.
      2. Visualización de tutoriales para familiarizarse con las herramientas y funcionalidades básicas de Canva.
      3. Exploración de la biblioteca de plantillas de Canva para entender las opciones disponibles.
2. Definición del Concepto del Logo
   1. Recursos Utilizados:
      1. Referencias visuales de logos existentes en el mismo sector o industria.
      2. Documentación y guías sobre teoría del diseño, incluyendo el uso de colores, tipografía y formas.
   2. Actividades Realizadas:
      1. Definición de los elementos clave del logo (colores, tipografía, símbolos).
      2. Búsqueda de inspiración en otros logos y recopilación de ideas.
      3. Bocetado de conceptos iniciales en papel o digitalmente.
3. Creación del Logo en Canva
   1. Recursos Utilizados:
      1. Editor de diseño de Canva: Editor de Canva
      2. Biblioteca de elementos gráficos, fuentes y plantillas de Canva.
   2. Actividades Realizadas:
      1. Selección de una Plantilla o Creación desde Cero:
         1. Explorar las plantillas de logos disponibles en Canva.
         2. Elegir una plantilla adecuada y personalizarla o comenzar con un lienzo en blanco.
      2. Diseño del Logo:
         1. Añadir y editar textos utilizando la herramienta de texto de Canva.
         2. Incorporar formas, iconos y otros elementos gráficos desde la biblioteca de Canva.
         3. Ajustar colores y tipografía para alinearse con la identidad de la marca.
         4. Realizar pruebas y ajustes iterativos para perfeccionar el diseño.

## Resultados obtenidos

1. Interfaz de Usuario Intuitiva y Atractiva

Diseño Moderno: Se implementó un diseño moderno y minimalista, siguiendo las mejores prácticas de UX/UI para asegurar una experiencia agradable y fácil de usar.

Navegación Sencilla: La navegación entre las diferentes secciones de la aplicación (lista de notas, creación de notas) es fluida y accesible, mejorando la usabilidad.

1. Funcionalidades Principales

Creación y Edición de Notas: Los usuarios pueden crear nuevas notas, editarlas y guardarlas fácilmente. La interfaz de edición es clara y permite una entrada de texto sin distracciones.

1. Sin Micropagos

Acceso Completo sin Costos Adicionales: Todas las funcionalidades de la aplicación están disponibles sin necesidad de realizar micropagos. Esto incluye la creación ilimitada de notas, la organización y las actualizaciones futuras de la aplicación.

Transparencia y Sencillez: La ausencia de micropagos elimina distracciones y posibles frustraciones asociadas a modelos freemium o suscripciones, proporcionando una experiencia más limpia y enfocada.

1. Rendimiento y Estabilidad

Rendimiento Óptimo: La aplicación muestra un rendimiento robusto, con tiempos de carga rápidos y una navegación sin interrupciones, incluso en dispositivos con recursos limitados.

Estabilidad: Se realizaron pruebas exhaustivas para asegurar que la aplicación funcione de manera estable bajo diversas condiciones de uso. Los errores críticos fueron identificados y corregidos antes del lanzamiento.

1. Compatibilidad Multiplataforma

Android e iOS: La aplicación fue desarrollada con React Native, permitiendo su despliegue tanto en dispositivos Android como iOS. Esto garantiza una amplia accesibilidad para los usuarios.

Consistencia entre Plataformas: Se mantuvo una experiencia de usuario consistente entre ambas plataformas, respetando las guías de diseño de cada sistema operativo.

1. Privacidad y Seguridad

Datos Personales Protegidos: Se han implementado medidas de seguridad para proteger la información personal de los usuarios. Las notas se almacenan de forma segura en servidores externos.

Permisos Mínimos Necesarios: La aplicación solicita únicamente los permisos estrictamente necesarios para su funcionamiento, respetando la privacidad del usuario.

# **Conclusiones**

Este proyecto me ha permitido aprender y mejorar significativamente mis habilidades, impulsándome a seguir desarrollándome. A través del proceso, he reconocido que aún tengo mucho por aprender, lo que me motiva a continuar perfeccionando mis conocimientos y técnicas. Además, me ha inspirado a soñar con la creación de una suite de software de código abierto sin micropagos, que pueda beneficiar a la comunidad de desarrolladores y usuarios por igual, fomentando la colaboración y el acceso libre a herramientas de calidad.

# Bibliografía

Google LLC. (18 de Mayo de 2024). *Firebase*. Obtenido de https://firebase.google.com/docs?hl=es-419

Meta Platforms, Inc. (18 de Mayo de 2024). *reactnative*. Obtenido de reactnative: https://reactnative.dev/

OpenAI. (2024). *chatgpt*. Obtenido de https://chatgpt.com/