



PROYECTO

FINAL

Desarrollo de Aplicaciones Multiplataforma

" P A L U E G O "

Jaime Rodríguez Benítez



ÍNDICE

PROYECTO FINAL ANDROID

BREVE RESUMEN	01
<ul style="list-style-type: none">• Introducción• Descripción• Viabilidad y factor diferencial	
TECNOLOGÍAS USADAS	03
<ul style="list-style-type: none">• Entorno de desarrollo• Persistencia de datos y registros• Diseño gráfico	
DESARROLLO DEL PROYECTO	05
<ul style="list-style-type: none">• Planificación<ul style="list-style-type: none">- Desarrollo y material necesario- Planificación temporal- Modelado de datos- Target objetivo• Librerías externas	
RESULTADO	09
<ul style="list-style-type: none">• Utilidad y usabilidad<ul style="list-style-type: none">- Identificación de subsistemas de análisis- Establecimiento de requisitos- Estructura del proyecto• Aspecto Gráfico<ul style="list-style-type: none">- Principios generales de la interfaz- Navegabilidad y descripción individual	
APÉNDICE	15
<ul style="list-style-type: none">• Conclusiones• Bibliografía• Glosario	

BREVE RESUMEN

PROYECTO FINAL ANDROID

INTRODUCCIÓN

En este documento vamos a ver el su o que podemos hacer de la aplicación que se va exponer, las tecnologías que se han usado y cómo se ha hecho además de algunos otros detalles del proyecto.

DESCRIPCIÓN

"Paluego" es una aplicación de Android diseñada para brindar a los usuarios una herramienta completa y organizada para su vida diaria. Con su función de creación de notas, los usuarios pueden guardar de manera conveniente cualquier cosa que necesiten recordar, ya sea una lista de compras o una idea para un proyecto personal.

Una de las características destacadas de la aplicación es su función de búsqueda, que permite a los usuarios filtrar y encontrar fácilmente las notas que necesitan en un instante. Además, gracias a la persistencia de datos en la nube, los usuarios pueden tener la tranquilidad de que sus notas estarán seguras y accesibles desde cualquier dispositivo Android con la aplicación instalada. Esto hace que sea increíblemente conveniente mantenerse al día con todas las tareas pendientes, ideas y proyectos personales, ya que los usuarios pueden acceder a su información desde cualquier lugar.

La capacidad de cambiar entre diferentes cuentas en la aplicación brinda a los usuarios la flexibilidad de mantener sus notas organizadas según el contexto en el que se necesiten. Cada cuenta tiene su propia información de inicio de sesión y notas guardadas, lo cual resulta especialmente útil para perfiles de trabajo, personales o incluso colaborativos.

En resumen, "Paluego" es una aplicación de notas versátil que ofrece recordatorios, múltiples cuentas y almacenamiento en la nube para ayudar a los usuarios a mantenerse organizados en su vida diaria y no olvidar detalles importantes. Con una interfaz intuitiva y características prácticas, esta aplicación se convierte en un compañero indispensable para aquellos que buscan tener todas sus notas importantes al alcance de su mano, sin importar dónde se encuentren.

BREVE RESUMEN

PROYECTO FINAL ANDROID

VIABILIDAD Y FACTOR DIFERENCIAL

Una aplicación de notas con capacidad para grabar y reproducir audios ofrece beneficios significativos para los usuarios, también la usabilidad de la misma y la interfaz intuitiva son esenciales para que esta sea funcional para todo tipo de perfil. Por los siguientes motivos entre otros la funcionalidad adherida de la grabación de audio es importante:

1. **Precisión y riqueza de detalles:** La grabación de audio permite capturar información con mayor precisión, incluyendo tono de voz, entonación y emociones. Esto enriquece la comprensión y conserva el contexto original de las ideas y discusiones.
2. **Ahorro de tiempo y esfuerzo:** La grabación de audios evita la necesidad de escribir extensos fragmentos de texto, lo cual resulta especialmente útil en situaciones que requieren una captura rápida de información, como entrevistas, reuniones o clases. Los usuarios pueden simplemente grabar el audio y revisarlo más tarde.
3. **Accesibilidad y conveniencia:** Al almacenar los audios en la aplicación de notas, los usuarios pueden acceder a ellos en cualquier momento y desde cualquier dispositivo. Esto brinda flexibilidad y conveniencia, ya que no dependen de un único dispositivo o ubicación física para acceder a la información grabada.
4. **Complemento a las notas escritas:** La capacidad de combinar audios y notas escritas en una sola aplicación ofrece una experiencia más completa y enriquecedora. Los usuarios pueden utilizar las notas escritas para resaltar puntos clave, agregar estructura y resumir la información capturada en los audios, facilitando la revisión y el entendimiento posterior.

En resumen, una aplicación de notas con grabación y reproducción de audios proporciona una forma precisa, eficiente y conveniente de capturar y revisar información. Esta funcionalidad complementa las notas escritas, ofreciendo una experiencia más completa y enriquecedora para los usuarios.

TECNOLOGÍAS USADAS

PROYECTO FINAL ANDROID

ENTORNO DE DESARROLLO



Android Studio es un entorno de desarrollo integrado (IDE) de Google para crear aplicaciones móviles para dispositivos Android. Cuenta con un editor de código altamente personalizable, un depurador integrado, un emulador de dispositivos y una amplia gama de herramientas para el diseño de interfaces de usuario, la gestión de versiones, la integración con Gradle, la creación de pruebas automatizadas y la optimización del rendimiento. Android Studio es la herramienta recomendada por Google para el desarrollo de aplicaciones móviles en Android, ofreciendo una solución completa y poderosa para los desarrolladores que buscan crear aplicaciones de alta calidad para dispositivos Android.

Kotlin es un lenguaje de programación moderno y conciso que se ejecuta en la JVM y es completamente interoperable con Java. Ofrece soporte para la programación orientada a objetos y funcional, así como características avanzadas, como evitar los nulos en tiempo de compilación. Kotlin se ha convertido en el lenguaje preferido para el desarrollo de aplicaciones móviles Android debido a su excelente compatibilidad con Android Studio y su capacidad para escribir código legible y eficiente.



TECNOLOGÍAS USADAS

PERSISTENCIA DE DATOS Y REGISTRO



Firebase es una plataforma de desarrollo móvil que ofrece una amplia gama de herramientas y servicios para crear aplicaciones móviles de alta calidad. Sus características clave incluyen almacenamiento en la nube, autenticación de usuarios, notificaciones push, analítica y funciones en la nube. Firebase también ofrece herramientas para mejorar la calidad de la aplicación, como pruebas de usuario beta, pruebas de rendimiento y herramientas de monitoreo en tiempo real. Con Firebase, los desarrolladores pueden crear aplicaciones móviles seguras, escalables y confiables en un tiempo más corto y con menos recursos. La plataforma es fácil de integrar en aplicaciones existentes y proporciona una solución todo en uno para las necesidades de desarrollo móvil.

DISEÑO

Canva es una herramienta en línea que permite a los usuarios crear diseños gráficos atractivos sin habilidades avanzadas de diseño. Ofrece una amplia variedad de plantillas y herramientas fáciles de usar para agregar y editar texto, imágenes, formas y otros elementos gráficos. Es gratuito con opciones de pago para acceder a características avanzadas y se puede utilizar para crear diseños para redes sociales, presentaciones, infografías, folletos y más. En resumen, Canva es una herramienta en línea accesible y fácil de usar para crear diseños de calidad profesional.



DEARROLLO

PROYECTO FINAL ANDROID

PLANIFICACIÓN

DESARROLLO Y MATERIAL NECESARIO

El proyecto de "Paluego" tiene como único creador y responsable el mismo que edita este documento, encargado de todas las etapas de diseño (análisis, diseño, programación, tests..).

El desarrollo ha constado de unas 180 horas de trabajo, divididas de manera imparcial en los tres meses que ha durado la FCT y las dos semanas posterior a esto.

El **look and feel** ha sido diseñado con la herramienta online Canva como se detalla anteriormente, el **diseño** dentro del IDE está compuesta por archivos XML enlazados a sus clases y como también se ha comentado anteriormente el **lenguaje de programación** escogido ha sido Kotlin por su versatilidad y por las muchas ayudas que se encuentra en las redes. Tras este proyecto tengo mayor manejo y soltura con el **IDE** (Android Studio).

El anteriormente mencionado IDE necesita unos mínimos para poder funcionar (Microsoft® Windows® 8/10/11 de 64 bits, arquitectura de CPU x86_64, procesador Intel Core de segunda generación o posterior, o CPU AMD compatible con un hipervisor de Windows, 8 GB de RAM o más), aunque es recomendable algo superior porque la experiencia ha resaltado que con equipos capaces pero justos puede dejarte tirado a veces.

DESSARROLLO

PROYECTO FINAL ANDROID

PLANIFICACIÓN

PLANIFICACIÓN TEMPORAL

13/03/2023

Preparación y diseño de pantallas, enlace entre los elementos gráficos y su parte en código.



01/06/2023

Autenticación y autologin. Implantación de Firebase en el proyecto.



04/06/2023

Configuración de la bases de datos online con Firebase y guardado primitivo de los datos de la aplicación en la nube



08/06/2023

Programación del código



14/06/2023

Vista de notas, grabado de voz y reproducción de audios grabados



20/06/2023

Testing y elaboración de manuales.

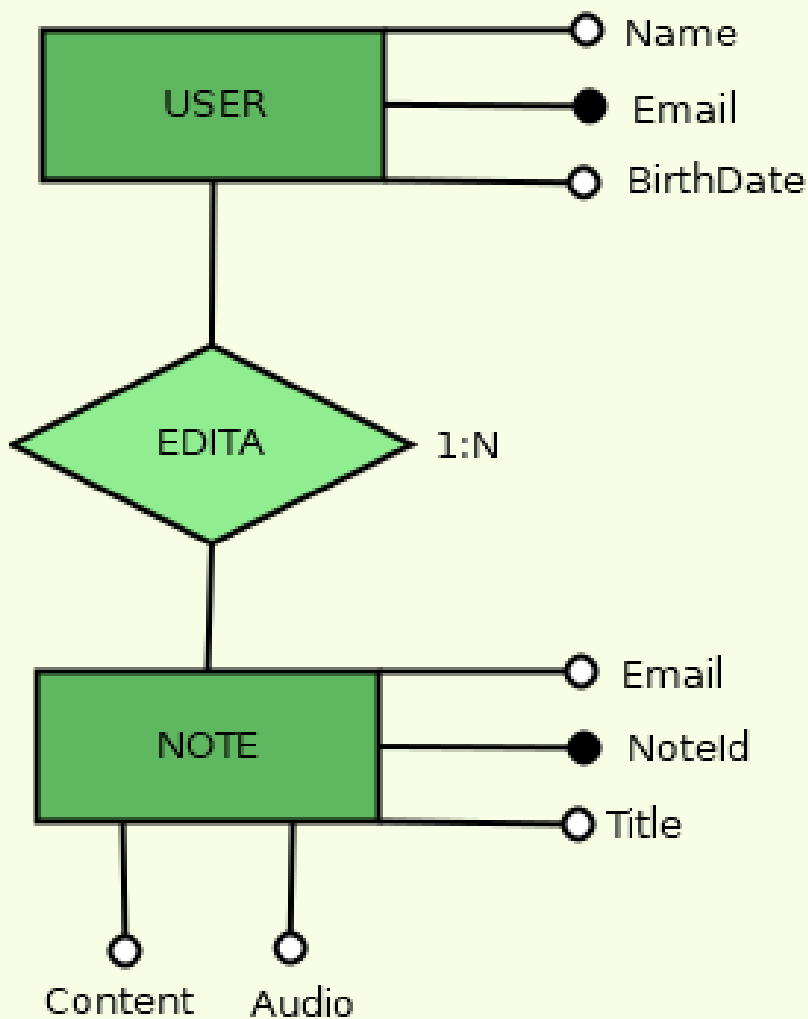


DEARROLLO

PROYECTO FINAL ANDROID

PLANIFICACIÓN

MODELADO DE DATOS



La **base de datos** que nos crea y nos proporciona Firebase, consta únicamente por dos entidades, las cuales creamos y damos sus atributos en sus respectivas partes de código. Son los siguientes:

User:

- Name : String
- Email (Primary Key) : String
- BirthDate : Date

Note:

- Email : String
- Noteld (Primary Key) : String
- Title : String
- Content : String
- Audio : ByteArray?

Las notas pueden ser instanciadas por cualquier usuario y y cada nota con su id único solo puede ser instanciada y editada por un usuario.

DEARROLLO

PROYECTO FINAL ANDROID

PLANIFICACIÓN

TARJET OBJETIVO

El público objetivo de la app son personas que desean gestionar de manera eficiente sus tareas pendientes, ideas y proyectos personales, así como aquellos que buscan una forma conveniente de recordar y acceder a detalles importantes en cualquier momento y desde cualquier lugar. La aplicación está dirigida a usuarios de dispositivos Android, independientemente de su nivel de experiencia tecnológica, y busca ofrecer una experiencia de uso intuitiva y accesible.

LIBRERÍAS EXTERNAS

Para poder acceder a características que no son propias de nuestro IDE, necesitamos importar **librerías externas**, estas son algunas de las utilizadas y su función:

- Firebase-Auth: Para la autenticación del usuario.
- Firebase-FireStore: Para poder guardar los datos de la aplicación en la nube.

RESULTADO

PROYECTO FINAL ANDROID

UTILIDAD Y USABILIDAD

IDENTIFICACIÓN DE SUBSISTEMAS DE ANÁLISIS

Para hacer más sencilla la comprensión de la aplicación la dividimos en subsistemas que se diferencian a simple vista porque están encargados de la creación y configuración de una parte específica de la app. Son los siguientes:

- Subsistema de registro y login, gestiona el proceso de la creación de usuario, la autenticación de este en la aplicación y la capacidad de guardar el usuario que se ha logeado para que no tenga que volver a hacerlo en su dispositivo.
- Subsistema de guardado y recuperación de notas, es el encargado de reconocer la nota y si es genuina, guardarla en la base de datos para posteriormente poder recuperarla.
- Subsistema de preferencias, es el encargado de guardar las preferencias de la app y ponerlas en ejecución una vez esta se inicie.

ESTABLECIMIENTO DE REQUISITOS

Requisitos Funcionales

Los usuarios pueden:

- Registrarse y autenticarse en la aplicación
- Crear, editar y borrar notas.
- Filtrar las notas por búsqueda manual.
- Configurar las preferencias de la app

Requisitos de Usabilidad

La interfaz es fácil de usar e intuitiva para la gran mayoría de usuarios de android.

Requisitos de Seguridad

Firebase (Google) proporciona seguridad para los datos de los usuarios.

Requisitos de Rendimiento

La aplicación funciona sin congelaciones ni cortes haciendo las gestiones ágilmente.

RESULTADO

PROYECTO FINAL ANDROID

UTILIDAD Y USABILIDAD

ESTRUCTURA DEL PROYECTO

En el proyecto se ha tenido la intención en todo momento de que la visión del mismo sea comprensible y modular, se podría dividir en dos grandes grupos:

- Estructura del código (**kotlin**)

El código en un estado primigenio de la aplicación estaba pensado en tener una estructura MVVM, pero al no hacer falta el ViewModel gracias al guardado que hacemos de los datos se queda en una arquitectura VM (**View** - **Model**), en dónde la vista la componen las clases asociadas a los elementos gráficos, las activities, customviews y fragments y en el paquete model guardamos aquellas clases que sirven como recursos de estas anteriormente citadas. No es necesario pero es más limpio y ordenado si tenemos un paquete que contenga los **adapters** de nuestros RecyclerViews.

- Estructura de los recursos gráficos (**XML** y **PNG**)

La organización de los elementos gráficos te la ejecuta android studio a medida que vas incorporando estos elementos, los directorios más importantes son la carpeta de **drawable**, dónde guardamos los elementos gráficos que usamos en nuestra aplicación como imágenes o iconos definidos con vectores.

La siguiente carpeta importante y que debe ser conocida por el desarrollador es la de **layout**, que es donde se encuentran los XML que conforman la vista de las pantallas que vamos a usar o items gráficos.

Es importante conocer los archivos que componen el directorio **values**, alberga valores esenciales para nuestra aplicación y nos previene de usar hard codes en nuestro código con archivos que nos guardan colores, strings o temas como constantes para que la aplicación pueda usarlos.

RESULTADO

PROYECTO FINAL ANDROID

ASPECTO GRÁFICO

PRINCIPIOS GENERALES DE LA INTERFAZ

La interfaz de la aplicación sigue unas directrices claras, tiene un paleta de colores escueta y concisa, los elementos gráficos son fácilmente reconocible por los usuarios y simples y el diseño se rige por formas redondeadas y desenfadas siendo amigables para el usuario.

El diseño de la aplicación está disponible en [este enlace](#).

NAVEGABILIDAD Y DESCRIPCIÓN INDIVIDUAL

Pantalla Splash

Es la encargada de comprobar si el usuario inició sesión anteriormente y de ver si los datos guardados son válidos para autologear al usuario y mandarlo a la pantalla principal o mandarlo a la de login.

Pantalla de Login

Es la encargada de autenticar al usuario y pasarlo a la aplicación. Guarda el usuario si se loguea exitosamente para poder usarlo durante la ejecución de la aplicación.

Pantalla de Registro

Se encarga de crear el usuario. Controla que no haya ciertos campos vacíos o erróneos.

Pantalla Principal

Contiene la lista de notas, con posibilidad de filtrarlas. Puede acceder a una nueva nota, alguna existente y las puede borrar haciendo swipe. Nos desloguea si pulsamos el botón para ello y nos puede llevar a las preferencias.

Pantalla de la nota

Nos da el detalle de la nota y podemos editarla.

Pantalla de Preferencias

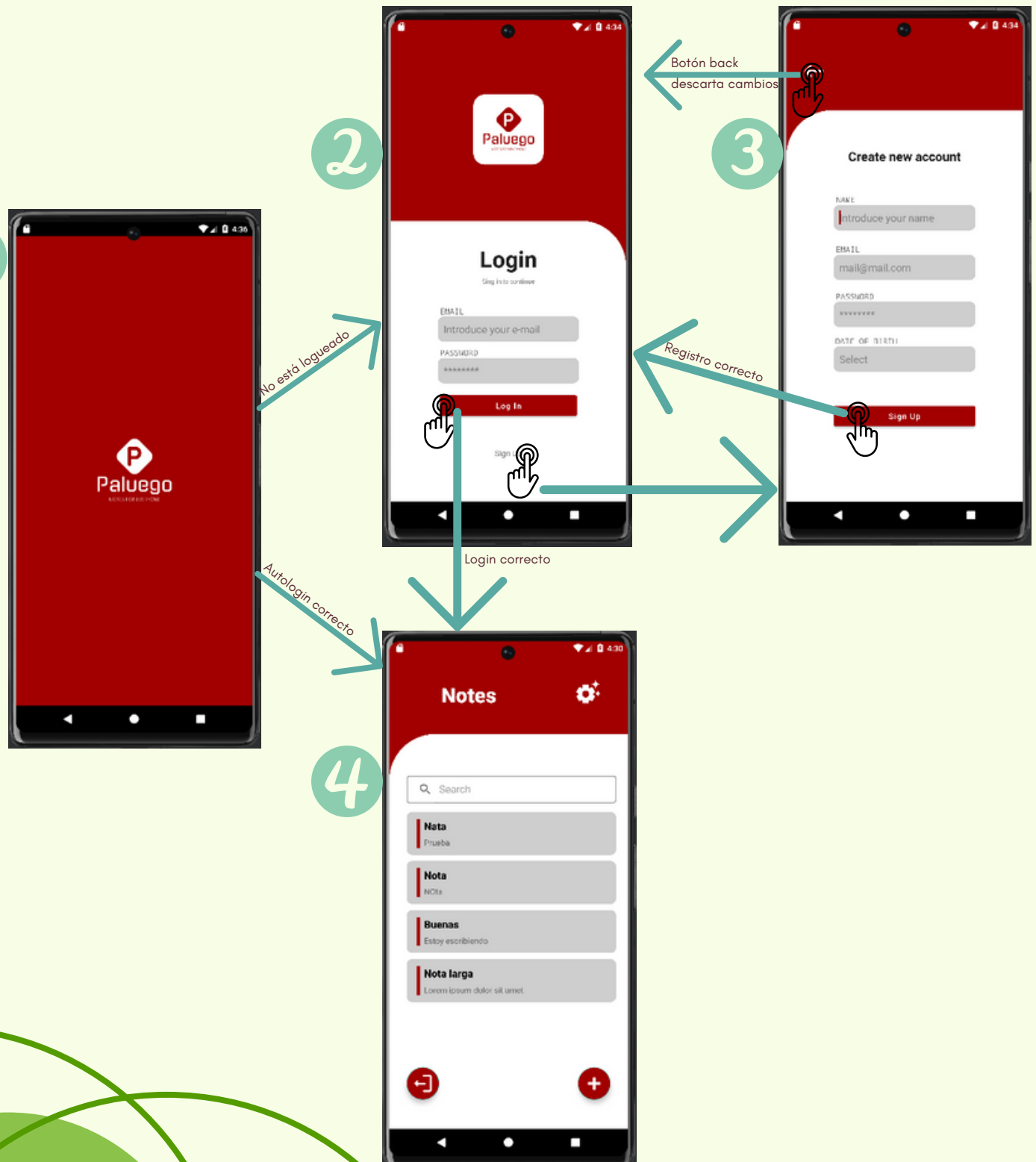
Nos da opciones a cambiar ciertas opciones para comodidad del usuario.

RESULTADO

PROYECTO FINAL ANDROID

ASPECTO GRÁFICO

NAVEGABILIDAD Y DESCRIPCIÓN INDIVIDUAL

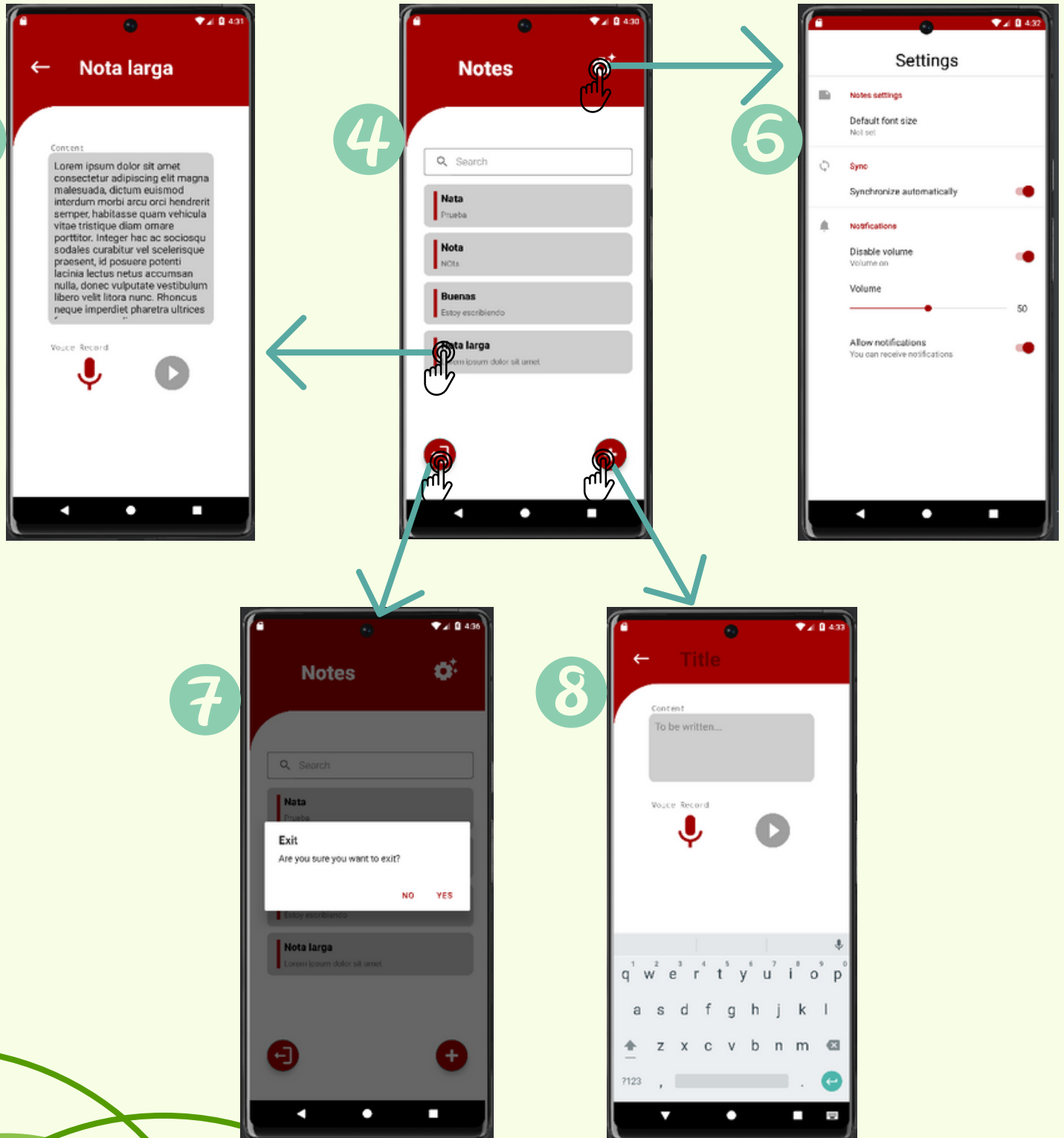


RESULTADO

PROYECTO FINAL ANDROID

ASPECTO GRÁFICO

NAVEGABILIDAD Y DESCRIPCIÓN INDIVIDUAL



RESULTADO

PROYECTO FINAL ANDROID

ASPECTO GRÁFICO

NAVEGABILIDAD Y DESCRIPCIÓN INDIVIDUAL

Tal como hemos visto en el grafo presentado en las dos anteriores páginas la primera pantalla que siempre se ve es la splash (1) que muestra el logo y verifica los datos para ir hacia la pantalla principal donde hay un listado de notas (4), o bien si no es posible autenticar al usuario viaja hacia la pantalla de login que te da la opción de introducir un usuario y contraseña que si son válidos te llevan a la pantalla principal (4) o bien si el usuario pulsa en registrase (sign up), accedes a la pantalla de registro (3) en dónde puede crear un nuevo usuario y cuando lo creas correctamente te cierra la pantalla y te deja de vuelta en la del login.

Una vez la autenticación está completa, la actividad principal actúa de matriz y desde ella se puede ir a crear una nueva nota (8), a la pantalla de ajustes (6), puede entrar a editar las notas (5) o bien puede desloguearse (7) e ir de nuevo a la pantalla de login(2).

APÉNDICE

PROYECTO FINAL ANDROID

CONCLUSIONES

Remitiéndome al documento aportado quiero dejar como conclusión que no hay nada mejor que coger una idea que ya existe y mejorarla para que a los usuarios les sea más fácil y les haga el día más sencillo y esta app lo cumple, es versátil y fácil de usar.

La aplicación se encuentra en fase alfa, están definidos los pilares de la misma pero para nada está acotada toda su funcionalidad, hay muchas mejoras e implementaciones posibles para esta.

BIBLIOGRAFÍA

Canva

<https://www.canva.com>

Documentación oficial de android

<https://developer.android.com/>

Guía del lenguaje Kotlin

<https://kotlinlang.org>

Documentación oficial de Firebase

<https://firebase.google.com/docs>

Documentación de StackOverflow

<https://stackoverflow.com>

APÉNDICE

PROYECTO FINAL ANDROID

GLOSARIO

Un **IDE** (Integrated Development Environment) es un entorno de desarrollo completo que combina un editor de código, un compilador o intérprete, y herramientas para ayudar a los desarrolladores a escribir, depurar y compilar su código de manera más eficiente.

El **target** es el grupo específico de personas a las que se dirige un producto, servicio o marca en particular.

En informática, una **librería** es un conjunto de funciones y recursos predefinidos que los programadores utilizan para facilitar el desarrollo de software. Proporcionan funcionalidades específicas y reutilizables, lo que ahorra tiempo y esfuerzo en la escritura de código desde cero.

En programación, **importar** consiste en agregar una librería externa al código para acceder a sus funciones y recursos. Esto permite reutilizar código existente y ampliar las capacidades del programa al aprovechar las funcionalidades ofrecidas por la librería importada.

Los **subsistemas de análisis** son componentes más pequeños en un sistema que se dividen para facilitar su comprensión y análisis. Cada subsistema cumple una función específica y se examina de manera independiente para identificar problemas y tomar decisiones.

MVVM (**Model-View-ViewModel**) es una arquitectura de software que se utiliza en el desarrollo de aplicaciones para separar las preocupaciones y mejorar la modularidad. En MVVM, el "Modelo" representa los datos y la lógica de negocio, la "Vista" se encarga de la interfaz de usuario y la "Vista de Modelo" actúa como intermediario entre el Modelo y la Vista, proporcionando los datos necesarios y manejo de eventos. Esta arquitectura promueve la separación de la lógica de presentación de los datos y facilita las pruebas unitarias, el mantenimiento y la escalabilidad del código.