



Documentación detallada del proyecto

Descripción Detallada del Proyecto - Take Note

La aplicación Take Note es una plataforma diseñada para facilitar la gestión y organización de notas personales en dispositivos móviles con sistema operativo Android. Take Note ofrece a los usuarios una interfaz intuitiva y funcionalidades robustas que les permiten crear, editar, eliminar y restaurar notas de manera eficiente. El objetivo principal de Take Note es proporcionar una experiencia de usuario fluida y satisfactoria, manteniendo un alto nivel de seguridad y accesibilidad.

La aplicación se desarrollará utilizando Android Studio como Entorno de Desarrollo Integrado (IDE), aprovechando las herramientas específicas proporcionadas para el desarrollo de aplicaciones Android. La elección de Android Studio se basa en su compatibilidad, enfoque específico para Android, amplia comunidad de desarrolladores, integración de herramientas esenciales y soporte para los principales lenguajes de programación para Android.

Funcionalidades Principales:

1. Registro y Autenticación de Usuarios:

- Los usuarios podrán registrarse en la aplicación proporcionando información básica como contraseña segura y email.
- La autenticación se realizará mediante credenciales válidas, permitiendo a los usuarios iniciar sesión de manera segura y confiable.

```

if (email.isEmpty() || password.isEmpty()) {
    Toast.makeText(context: Login.this, text: "Por favor, completa todos los campos.", Toast.LENGTH_SHORT).show();
} else {
    mAuth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(activity: Login.this, task -> {
            if (task.isSuccessful()) {
                // Inicio de sesión exitoso
                Toast.makeText(context: Login.this, text: "Inicio de sesión exitoso.", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(context: Login.this, Home.class);
                startActivity(intent);
                finish(); // Finaliza la actividad de login solo si el inicio de sesión es exitoso
            } else {
                // Error durante el inicio de sesión
                Toast.makeText(context: Login.this, text: "Error al iniciar sesión. Verifica tus credenciales.", Toast.LENGTH_SHORT).show();
            }
        });
}
});

```

2. Gestión de Notas:

- Los usuarios tendrán la capacidad de crear nuevas notas ingresando un título y contenido relevante.

```

private void guardarNota() {
    String titulo = editTextTitulo.getText().toString();
    String cuerpo = editTextCuerpo.getText().toString();

    //validaciones.
    if (titulo.isEmpty()) {
        Toast.makeText(context: this, text: "El título no puede estar vacío", Toast.LENGTH_SHORT).show();
        return;
    }
    if (cuerpo.isEmpty()) {
        Toast.makeText(context: this, text: "El cuerpo no puede estar vacío", Toast.LENGTH_SHORT).show();
        return;
    }
    //Revisar que el usuario este autenticado.
    FirebaseUser user = mAuth.getCurrentUser();
    if (user != null) {
        String user_id = user.getId();

        try {
            dbHelper.insertarNota(user_id, titulo, cuerpo);
            Toast.makeText(context: this, text: "Nota guardada correctamente", Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            Toast.makeText(context: this, text: "Error al insertar la nota", Toast.LENGTH_SHORT).show();
        }
    }
}

```

- Se permitirá la edición de notas existentes, lo que brinda flexibilidad para actualizar y modificar la información según sea necesario.

```

editTextTitulo = findViewById(R.id.editTituloNota);
editTextCuerpo = findViewById(R.id.editCuerpoNota);

// Obtén los datos de la nota desde el Intent
Intent intent = getIntent();
if (intent != null) {
    idNota = intent.getIntExtra( name: "ID_NOTA", defaultValue: -1);
    String tituloNota = intent.getStringExtra( name: "TITULO_NOTA");
    String cuerpoNota = intent.getStringExtra( name: "CUERPO_NOTA");

    // Asigna los valores de la nota a las variables
    editTextTitulo.setText(tituloNota);
    editTextCuerpo.setText(cuerpoNota);
}

Button buttonGuardar = findViewById(R.id.btnGuardar);

//Oír el click en el boton de guardar y realizar las acciones correspondientes
buttonGuardar.setOnClickListener(v -> {
    String titulo = editTextTitulo.getText().toString();
    String cuerpo = editTextCuerpo.getText().toString();

```

- Los usuarios podrán eliminar notas que ya no necesiten, y estas se moverán a la papelera para su posible restauración en el futuro.

```

public void enviarNotaAPapelera(String userId, int notaId) {
    SQLiteDatabase db = this.getWritableDatabase();

    try {
        Nota nota = obtenerNota(db, tableName: "notas", notaId);

        if (nota != null) {
            ContentValues values = new ContentValues();
            values.put("user_id", obtenerUsuarioActual());
            values.put("idNota", notaId);
            values.put("titulo", nota.getTitulo());
            values.put("cuerpo", nota.getCuerpo());

            long result = db.insertOrThrow( table: "papelera", nullColumnHack: null, values);
            if (result != -1) {
                eliminarNota(userId, notaId);
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        if (db.isOpen()) {
            db.close();
        }
    }
}

```

- La función de restauración de notas desde la papelera garantizará que los usuarios puedan recuperar información importante accidentalmente eliminada.

```
//Funcion para traer una nota de la tabla Papelera a la tabla Notas.
public void restaurarNotaDesdePapelera(String userId, int position) {
    DatabaseHelper.Nota nota = listaDeNotas.get(position);
    int notaId = nota.getId();
    Log.d( tag: "NotaAdapter", msg: "Restaurando nota desde la papelera: ID=" + notaId + ", Titulo=" + nota.getTitulo());
    dbHelper.restaurarNotaDesdePapelera(context, obtenerUsuarioActual(), notaId);
    listaDeNotas.remove(position);
    notifyItemRemoved(position);
}
```

3. Configuración de la Aplicación:

- Los usuarios podrán personalizar la apariencia de la aplicación cambiando entre modos de visualización, como modo claro y modo oscuro, según sus preferencias.

```
boolean isDarkTheme = loadThemeState( activity: this);
temaUtils.applyThemeToActivity( activity: this, isDarkTheme);

if(isDarkTheme){
    mNotaAdapter.setTextColor(Color.WHITE);
    mNotaAdapter.setImageColor(Color.WHITE);
} else {
    mNotaAdapter.setTextColor(Color.BLACK);
    mNotaAdapter.setImageColor(Color.BLACK);
}
```

- Se ofrecerá a los usuarios la opción de gestionar su cuenta, lo que incluye la capacidad de cambiar el email y/o contraseña, o eliminar la cuenta por completo si así lo desean.

Requerimientos No Funcionales Destacados:

1. Seguridad:

- **Almacenamiento Seguro de Contraseñas:**
 - Firebase Authentication almacena las contraseñas de los usuarios de manera segura utilizando algoritmos de hash y métodos criptográficos avanzados.

- **Inicio de Sesión Seguro:**

- Durante el inicio de sesión, Firebase Authentication compara la contraseña ingresada por el usuario con la versión encriptada almacenada en los servidores de Firebase.

- **Ventajas de Firebase Authentication:**

- Simplifica la gestión de usuarios y la autenticación en la aplicación.
- Proporciona una capa de seguridad robusta sin requerir codificación adicional.
- Se integra fácilmente con otras funciones y servicios de Firebase.

2. Accesibilidad:

- La aplicación se compromete a seguir las pautas de accesibilidad web para asegurar que todos los usuarios, independientemente de sus habilidades o discapacidades, puedan acceder y utilizar la aplicación de manera efectiva. Para lograr esto, la aplicación implementará los siguientes principios:

1. Cumplimiento legal:

- La aplicación cumplirá con las leyes y regulaciones pertinentes que rigen la accesibilidad web, incluidas, entre otras, las directrices del World Wide Web Consortium (W3C) y los estándares de la Ley de Estadounidenses con Discapacidades (ADA, por sus siglas en inglés).

2. Contraste y legibilidad:

- Se garantizará un buen contraste entre el texto y el fondo para mejorar la legibilidad, permitiendo que los usuarios con discapacidades visuales puedan navegar y comprender el contenido de manera más efectiva.

3. Adaptación para dispositivos móviles:

- La aplicación será completamente adaptable a dispositivos móviles, asegurando que tanto el texto como las imágenes se ajusten adecuadamente al tamaño de la pantalla y sean fácilmente accesibles y legibles en dispositivos móviles de diferentes dimensiones.

4. Etiquetas y descripciones de imágenes:

- Todas las imágenes incluidas en la aplicación contarán con etiquetas descriptivas adecuadas, permitiendo que los usuarios con discapacidades visuales puedan comprender el contenido visual a través de lectores de pantalla u otras tecnologías de asistencia.

5. Navegación clara y sencilla:

- Se implementará una navegación clara y coherente en toda la aplicación, facilitando a los usuarios con discapacidades cognitivas o de movilidad la ubicación y el acceso a las diferentes secciones y funciones.

3. Rendimiento:

- Se pondrá un énfasis especial en garantizar que la aplicación pueda manejar grandes cantidades de datos de manera eficiente, asegurando tiempos de carga rápidos y una experiencia fluida incluso en dispositivos con recursos limitados.

4. Usabilidad:

- La interfaz de usuario será diseñada para ser intuitiva y fácil de usar, con el objetivo de proporcionar una experiencia agradable para usuarios de diferentes niveles de habilidad y experiencia tecnológica.

5. Mantenibilidad:

- El código de la aplicación seguirá buenas prácticas de programación y estará debidamente documentado para facilitar el mantenimiento y la extensión futura del software, asegurando la sostenibilidad a largo plazo de la aplicación.

Otros Elementos Relevantes:

1. Restricciones Tecnológicas:

- Se considerará la compatibilidad con versiones específicas de Android y los requisitos de hardware mínimos para garantizar un rendimiento óptimo en una amplia variedad de dispositivos móviles.

2. Documentación:

- Se generará una documentación detallada que incluirá diagramas de arquitectura, manuales de usuario y cualquier otra información

relevante para facilitar la comprensión y el uso efectivo de la aplicación por parte de los usuarios finales y los desarrolladores.

La aplicación Take Note se desarrollará con un enfoque centrado en el usuario que necesite organizar sus ideas o notas sin depender del papel y sin importar edad, priorizando la seguridad, accesibilidad, rendimiento, usabilidad y mantenibilidad para ofrecer una experiencia de usuario excepcional y confiable.

Estructura de la Base de Datos:

Se utiliza una base de datos SQLite que depende de 2 tablas las cuales son Notas, que como su nombre lo indica se encarga de guardar las notas creadas y sus valores correspondientes mientras que la otra es la de Papelera cumpliendo la función que su nombre indica de guardar las notas que el usuario descarta aunque no se eliminan aun de manera definitiva.

DER

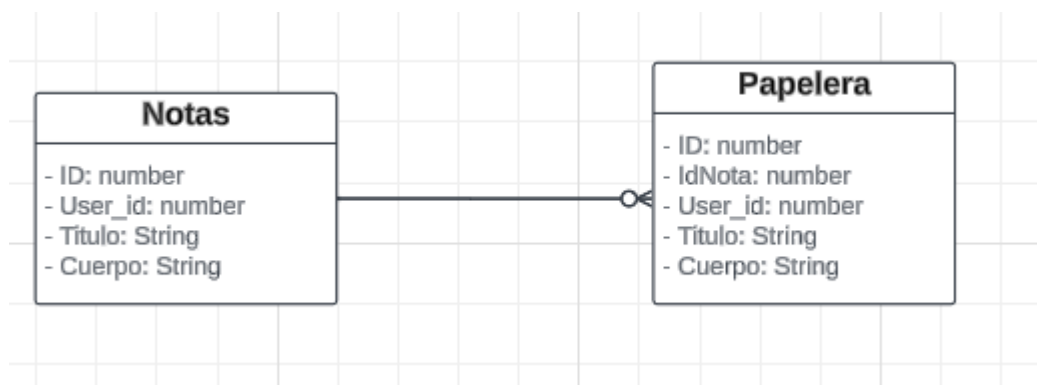
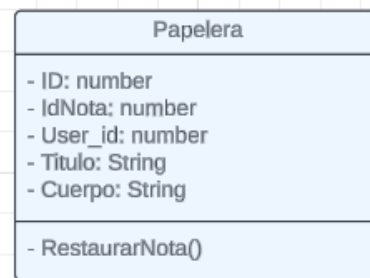
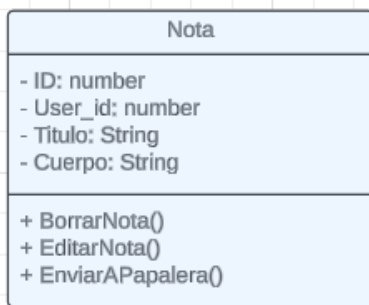


Diagrama de casos UML



Configuración Importante:

- Version utilizada: Android Studio Hedgehog | 2023.1.1 Patch 1
- Dependencias de Gradle utilizadas

```
dependencies {
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.10.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.navigation:navigation-fragment:2.7.5")
    implementation("androidx.navigation:navigation-ui:2.7.5")
    implementation("androidx.preference:preference:1.1.1")
    implementation("androidx.preference:preference:1.2.0")
    implementation(platform("com.google.firebase:firebase-bom:32.7.1"))
    implementation("com.google.firebase:firebase-auth:21.0.1")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}
```

- **Autenticación con Firebase:**

La aplicación utiliza Firebase Authentication para gestionar la autenticación de los usuarios. Se han implementado varios métodos de inicio de sesión, incluyendo correo electrónico y contraseña, así como también inicio de sesión con Google y Facebook. Esta integración permite a los usuarios acceder de manera segura a sus cuentas y protege sus datos personales.

- **Seguridad de la Aplicación:**

Se han tomado medidas importantes para garantizar la seguridad de la aplicación y la protección de los datos del usuario. Esto incluye la implementación de comunicaciones seguras utilizando HTTPS para la transferencia de datos entre la aplicación y los servidores de Firebase. Además, se han aplicado prácticas recomendadas para prevenir posibles ataques de inyección de SQL y otras vulnerabilidades de seguridad en la base de datos local SQLite.

- **Almacenamiento Local de Notas:**

Las notas de los usuarios se almacenan de forma segura en una base de datos SQLite local. Se han implementado mecanismos de encriptación y almacenamiento seguro para proteger la confidencialidad de los datos del usuario. La base de datos se gestiona de manera eficiente para garantizar la integridad de las notas y proporcionar una experiencia de usuario fluida.

```
//Creacion de la base de datos.
+ tomimiramone"
@Override
public void onCreate(SQLiteDatabase db) {
    String createTableQuery = "CREATE TABLE notas (_id INTEGER PRIMARY KEY AUTOINCREMENT, user_id TEXT, titulo TEXT, cuerpo TEXT)";
    db.execSQL(createTableQuery);

    String createPapelerTableQuery = "CREATE TABLE papeler (_id INTEGER PRIMARY KEY AUTOINCREMENT, idNota INTEGER, titulo TEXT, cuerpo TEXT, user_id TEXT)";
    db.execSQL(createPapelerTableQuery);
}
```

Flujo de Trabajo y Desarrollo del Proyecto Take Note:

El proceso de desarrollo del proyecto Take Note sigue un enfoque metodológico integral que abarca diversas etapas, desde la planificación inicial hasta el mantenimiento continuo del software. A continuación, se detallan las diferentes fases del ciclo de vida del proyecto y las metodologías, herramientas y tecnologías utilizadas en cada una de ellas:

1. Planificación:

- En esta etapa, se definieron los objetivos del proyecto, los requisitos del usuario y se elaboró un plan detallado para guiar el desarrollo.
- Se establecieron las funcionalidades principales de la aplicación, y se dividió en diferentes puntos para ir desarrollándolo de manera prolija.

2. Análisis:

- Se identificaron los casos de uso clave, los requisitos funcionales y no funcionales, y se elaboró una documentación detallada que sirvió como guía para el diseño y desarrollo. VER IEEE

3. Diseño:

- En esta etapa, se elaboraron los diseños de la interfaz de usuario (UI) y la arquitectura del sistema.

4. Implementación:

- La implementación de Take Note se llevó a cabo utilizando Android Studio como IDE principal y los lenguajes de programación Java y SQLite de base de datos.
- Se utilizó Firebase Authentication para la parte de gestión de usuarios.

5. Pruebas:

- Se realizaron pruebas exhaustivas en todas las etapas del desarrollo para garantizar la calidad y la fiabilidad del software.
- Se implementaron pruebas de integración y de aceptación para validar el funcionamiento correcto de las funcionalidades principales y detectar posibles errores y fallos.

```

tommiramone
@Test
public void testAgregarNota() {
    Log.d(TAG, "Iniciando prueba de agregar nota...");
    // Ingresar texto en el campo de título
    onView(withId(R.id.editTituloNota)).perform(replaceText(stringToBeSet: "Titulo de prueba"), closeSoftKeyboard());

    onView(withId(R.id.editTituloNota)).check(matches(withText("Titulo de prueba")));

    // Ingresar texto en el campo de cuerpo
    onView(withId(R.id.editCuerpoNota)).perform(replaceText(stringToBeSet: "Cuerpo de prueba"), closeSoftKeyboard());

    onView(withId(R.id.editCuerpoNota)).check(matches(withText("Cuerpo de prueba")));

    // Hacer clic en el botón de guardar
    onView(withId(R.id.btnGuardar)).perform(click());

    Log.d(TAG, "La prueba de agregar nota ha finalizado.");
}

```

6. Mantenimiento:

- El mantenimiento de la aplicación Take Note implica la corrección de errores, la optimización del rendimiento y la implementación de nuevas funcionalidades según las necesidades del usuario. Claramente se oirá a cada usuario que tenga una queja o recomendación y luego de meditarlo y

analizarlo se haran las correcciones correspondientes en caso de creerlo necesario más allá de lo obvio.

Metodologías de Desarrollo:

- Se aplicó una metodología ágil de desarrollo de software, adoptando principios de Scrum y Kanban para gestionar el proceso de desarrollo de manera eficiente y colaborativa.

Herramientas y Tecnologías Utilizadas:

- Android Studio: IDE principal para el desarrollo de aplicaciones Android.
- Java : Lenguaje de programación utilizados para la implementación de la lógica de la aplicación.
- Git y GitHub: Plataforma de control de versiones y colaboración para gestionar el código fuente y facilitar la colaboración entre los miembros del equipo.
- Firebase: Plataforma de desarrollo de aplicaciones móviles de Google utilizada para la autenticación de usuarios.
- JUnit: Frameworks de pruebas unitarias y de integración utilizados para garantizar la calidad del software mediante pruebas automatizadas.

El proceso de desarrollo del proyecto Take Note se basa en una combinación de buenas prácticas de ingeniería de software y metodologías ágiles, lo que permite entregar un producto final de alta calidad que cumple con las expectativas del usuario y los estándares de la industria.

Consideraciones de Seguridad:

- Con Firebase Authentication pudimos tener controlado de manera sencilla la autenticación de los usuarios dejando la seguridad en manos de dicha plataforma lo cual nos ahorró bastante tiempo de desarrollo.
- Por otro lado para la inserción de notas y edición sobre todo se utilizaron consultas parametrizadas, se tuvo en cuenta la revisión de los campos evitando campos vacíos y también la verificación de usuario registrado que permitía saber si la persona estaba logueada antes de

realizar alguna acción para que no se produzcan errores o problemas que pueda aprovechar algún otro con malicia.

- La base de datos que utilizo es SQLite que de naturaleza es persistente lo que le agrega otra cuota de seguridad a nuestra aplicación.

Documentación Técnica:

- La aplicación "TakeNote" sigue una arquitectura MVC (Modelo-Vista-Controlador), donde cada componente desempeña un papel crucial en la funcionalidad general de la aplicación.

Modelo (Model):

- El componente principal del modelo es `DatabaseHelper`. Este maneja la creación y la gestión de la base de datos SQLite subyacente. Administra operaciones como la inserción, actualización, eliminación y recuperación de notas.
- `DatabaseHelper` también gestiona la lógica de negocio relacionada con las notas y los usuarios, incluida la interacción con Firebase Authentication para obtener el usuario actual.

Vista (View):

- La actividad principal es `Home`, que actúa como la vista principal de la aplicación. En esta actividad, los usuarios pueden ver sus notas, buscarlas, editarlas, eliminarlas y enviarlas a la papelera.
- `Home` interactúa con el adaptador de notas (`NotaAdapter`) para mostrar las notas en un `RecyclerView` de manera eficiente. La interfaz de usuario se actualiza dinámicamente según las interacciones del usuario y los datos del modelo.
- Además, `Config` y `Papelera` también son vistas importantes que permiten al usuario configurar la aplicación y ver las notas archivadas respectivamente.

Controlador (Controller):

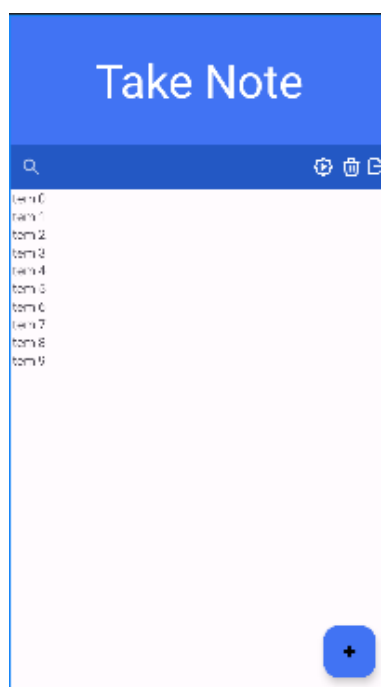
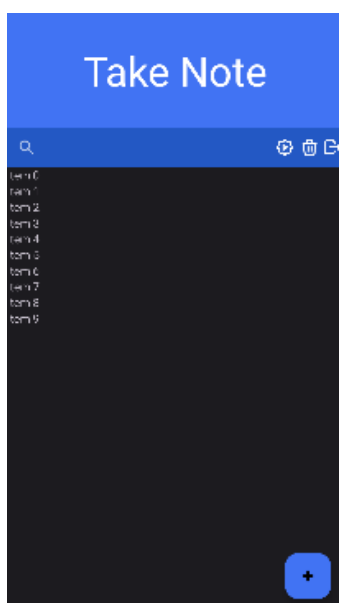
- La lógica de control y coordinación se implementa principalmente en las actividades (`Home`, `Config`, `Papelera`), donde se manejan los eventos de UI, como clics de botones y acciones de búsqueda.

- El adaptador de notas (`NotaAdapter`) actúa como un puente entre la vista y el modelo al proporcionar la lógica para mostrar y manipular las notas en el `RecyclerView` . Maneja eventos como clics en ítems, ediciones y eliminaciones de notas.

El flujo de la aplicación comienza con la actividad `Home` , donde se cargan las notas del usuario desde la base de datos utilizando `DatabaseHelper` . Estas notas se muestran en un `RecyclerView` a través del `NotaAdapter` . Los usuarios pueden interactuar con las notas, lo que desencadena acciones que se manejan tanto en la vista como en el modelo. Además existen las pantallas que permiten una vista en detalle de cada nota. Y una de configuración con sus correspondientes opciones tanto cambiar el modo de oscuro a claro o viceversa como de gestionar la cuenta del usuario actual.

Accesibilidad:

Se cumple con la accesibilidad necesaria en una aplicación móvil. Los colores generan un contraste que facilitan la lectura teniendo dos esquemas de colores según el modo elegido.



Ademas la aplicación esta diseñada y preaprada para que si el usuario tiene un tamaño predeterminado de letra en su dispositivo la aplicación se adapte a ello cambiando el tamaño de la letra.

Por ultimo la aplicación es compatible con lectores de pantalla para usuarios con discapacidades visuales. Esto implica que nos hemos tomado el tiempo de etiquetar adecuadamente los elementos de la interfaz de usuario y proporcionar una estructura lógica y coherente para la navegación.

Seguridad:

En cuanto a seguridad los datos sensibles que son contraseña, email y demás estan cubiertos gracias a la utilización de Firebase Authentication. En cuanto a las notas, no creemos necesario el cifrar y agregar dificultad a la comunicación entre base de datos y usuario para eso ya que todo se mantiene de manera local. Además intentamos tener recaudos evitar la inyeccion SQL

Casos de prueba:

1. Caso de Prueba: Inicio de Sesión

Objetivo:

Probar la funcionalidad de inicio de sesión en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Se dispone de un usuario registrado y válido.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla de inicio, introducir el email en el campo "Nombre de Usuario".
3. Introducir la contraseña en el campo "Contraseña".
4. Tocar el botón "Iniciar Sesión".

Resultados Esperados:

- Se espera que la aplicación inicie sesión correctamente si se proporcionan credenciales válidas.
- Se espera que se muestre un mensaje de error si se proporcionan credenciales inválidas o si hay un problema de conexión.

Criterios de Éxito:

- La aplicación muestra la pantalla principal después de iniciar sesión exitosamente.
- La aplicación muestra un mensaje de error si las credenciales son inválidas.
- La aplicación muestra un mensaje de error si hay un problema de conexión.

Datos de Prueba:

- Usuario válido: test@example.com
- Contraseña válida: password123

2. Caso de Prueba: Registro de usuario

Objetivo:

Probar la funcionalidad de registro de usuario en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Se dispone de una base de datos que acepte usuarios nuevos.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. Introducir la contraseña en el campo "Contraseña".
3. Introducir el email en el campo "Email".

4. Tocar el botón "Registrarse".

Resultados Esperados:

- Se espera que la aplicación registre al nuevo usuario correctamente si se proporcionan credenciales válidas.
- Se espera que se muestre un mensaje de error si se proporcionan credenciales inválidas o si hay un problema de conexión.

Criterios de Éxito:

- La aplicación muestra la pantalla de inicio de sesión después de registrarse exitosamente.
- La aplicación muestra un mensaje de error si las credenciales son inválidas.
- La aplicación muestra un mensaje de error si hay un problema de conexión.

Datos de Prueba:

- Usuario: test@example.com
- Contraseña: password123

3. Caso de Prueba: inserción de nota:

Objetivo:

Probar la funcionalidad de inserción de notas en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Se dispone de una base de datos que tome las notas.

Pasos de Prueba:

1. Abrir la aplicación móvil.

2. En la pantalla de inicio hacer click en el boton con el logo "+".
3. En la nueva pantalla ingresar la un titulo en el campo "Titulo"
4. En la nueva pantalla ingresar la un cuerpo en el campo "Cuerpo"
5. Tocar en agregar nota.
6. Al volver a la pantalla principal se deberia visualizar la nota creada.

Resultados Esperados:

- Se espera que el botón de la pantalla principal envíe a una pantalla para crear la nota.
- Se espera que el usuario pueda guardar la nota creada
- Se espera que el usuario pueda visualizar la nota en la pantalla principal tras guardarla.

Criterios de Éxito:

- La aplicación muestra la pantalla de personalización de nota.
- La aplicación muestra un mensaje en caso de que sea correcta la inserción de la nota.
- La aplicación muestra un mensaje si hay un problema con la nota.
- La aplicación muestra un mensaje de error si hay un problema de conexión.
- La aplicación muestra la nota en la pantalla principal de manera correcta.

Datos de Prueba:

- Título válido: Título de prueba
- Cuerpo válido: Cuerpo de prueba

4. Caso de Prueba: Edición de notas.

Objetivo:

Probar la funcionalidad de edición de notas en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Se dispone de una base de datos que contenga las notas.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla de inicio hacer click en el boton de edición junto a la nota deseada.
3. En la nueva pantalla ingresar modificar el campo que contiene "Titulo"
4. En la nueva pantalla ingresar modificar el campo que contiene "Cuerpo"
5. Tocar en editar
6. Al volver a la pantalla principal se debería visualizar la nota editada.

Resultados Esperados:

- Se espera que el botón de la nota envíe a una pantalla para editar la nota.
- Se espera que el usuario pueda guardar la nota editada
- Se espera que el usuario pueda visualizar la nota en la pantalla principal tras editarla.

Criterios de Éxito:

- La aplicación muestra la pantalla de personalización de nota.
- La aplicación muestra un mensaje en caso de que sea correcta la edición de la nota.
- La aplicación muestra un mensaje si hay un problema con la nota.
- La aplicación muestra un mensaje de error si hay un problema de conexión.
- La aplicación muestra la nota en la pantalla principal de manera correcta.

Datos de Prueba:

- Título válido: Título de receta 2
- Cuerpo válido: Receta. 2

5. Caso de Prueba: Eliminación de notas.

Objetivo:

Probar la funcionalidad de eliminación de notas en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Se dispone de una base de datos que contenga las notas.
- Se dispone de una base de datos que represente la papelera.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla de inicio hacer click en el boton de eliminación junto a la nota deseada.
3. La pantalla de inicio debería mostrarse sin la nota que se decidió eliminar.
4. Al hacer click en la barra de la pantalla de sesión sobre el icono de "basura" debería dirigirse a la papelera.
5. La pantalla papelera debería mostrar la nota previamente borrada.

Resultados Esperados:

- Se espera que el botón de la nota la envíe a la papelera.
- Se espera que el usuario pueda visualizar en la papelera la nota.
- Se espera que el usuario pueda visualizar la nota en la pantalla principal tras editarla.

Criterios de Éxito:

- La aplicación envía la nota a la papelera .
- La aplicación redirige de pantalla principal a papalera correctamente.
- La aplicación muestra la nota eliminada en la papelera.

6. Caso de Prueba: Restauración de notas.

Objetivo:

Probar la funcionalidad de restauración de notas en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Se dispone de una base de datos que contenga las notas.
- Se dispone de una base de datos que contenga las notas en la papelera.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla de papelera hacer click en el boton de flecha doblada junto a la nota deseada.
3. La pantalla de papelera deberia mostrarse sin la nota que se decidió restaurar.
4. Al hacer click en la barra de la pantalla de papelera sobre el icono de "Home" deberia dirigirse a la pantalla principal.
5. La pantalla principal deberia mostrar la nota previamente restaurada.

Resultados Esperados:

- Se espera que el botón de la nota la envíe a la pantalla principal.
- Se espera que el usuario pueda visualizar en la pantalla principal la nota.

Criterios de Éxito:

- La aplicación envía la nota a la pantalla principal.
- La aplicación redirige de papelera a pantalla principal correctamente.
- La aplicación muestra la nota restaurada en la pantalla principal.

7. Caso de Prueba: Configuración

Objetivo:

Probar la funcionalidad de modo de color en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla principal hacer click en el boton de engranaje de la barra.
3. Deberia mostrarse la pantalla de Configuración.
4. Al hacer click en el slider de "Modo oscuro" deberia alternar entre colores claros u oscuros según lo previamente elegido.
5. Si se hace click en "Gestion de perfil" deberia redigir al usuario a otra pantalla.

Resultados Esperados:

- Se espera que el slider cambie el modo de imagen de la aplicación.
- Se espera que el usuario acceda a la pantalla de gestion de usuario.

Criterios de Éxito:

- La aplicación cambia su modo de imagen con el slider
- La aplicación redirige a la gestion correctamente.

8. Caso de Prueba: Gestion de usuario

Objetivo:

Probar las funcionalidades de la gestion de usuario en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Una base de datos para los usuarios.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla principal hacer click en el boton de engranaje de la barra.
3. Deberia mostrarse la pantalla de Configuración.
4. Si se hace click en "Gestion de perfil" deberia redirigir al usuario a otra pantalla.
5. Si se hace click en "Cambiar email" deberia enviar a una pantalla que permite la modificación del email del usuario.
6. Si se hace click en "Cambiar contraseña" se deberia enviar a una pantalla que pida una nueva contraseña.
7. Si se hace click en "Borrar usuario" se deberia eliminar el usuario, cerrar la sesión actual y enviar a la pantalla de Login.

Resultados Esperados:

- Se espera que al acceder en "Gestión de perfil" aparezcan 3 opciones para el usuario
- Se espera que el usuario pueda acceder y cambiar su email con la opción "Cambiar email"
- Se espera que el usuario pueda acceder y cambiar su contraseña con la opción "Cambiar contraseña"

- Se espera que el usuario pueda borrar su usuario con "borrar usuario"

Criterios de Éxito:

- La aplicación muestra la pantalla "gestion de usuario"
- La aplicación cambia el email del usuario
- La aplicación cambia la contraseña del usuario
- La aplicación borra al usuario de la base de datos.

9. Caso de Prueba: Cerrar sesión

Objetivo:

Probar las funcionalidades de cerrar sesión en la aplicación móvil.

Precondiciones:

- La aplicación está instalada en el dispositivo móvil.
- El dispositivo tiene acceso a internet.
- Una base de datos para los usuarios.

Pasos de Prueba:

1. Abrir la aplicación móvil.
2. En la pantalla principal hacer click en el boton de puerta de la barra.
3. Deberia enviar automaticamente a la pantalla de inicio de sesión

Resultados Esperados:

- Se espera que al hacer click en el icono indicado se cierre la sesión y rediriga a iniciar sesión

Criterios de Éxito:

- La aplicación cierra la sesión
- La aplicación redirige a iniciar sesión luego.