

Documentación del Código del Proyecto

Manuel Vargas

4 de marzo de 2025

Índice

1. Introducción	1
2. Estructura del Proyecto	2
3. Dependencias	4
4. Descripción del Código	4
4.1. screens	4
4.1.1. Photos	4
4.1.2. Tables	4
4.1.3. Reports	5
4.1.4. Notes	5
4.1.5. Litologic	5
4.1.6. StructuralDatas	6
4.2. components	6
4.2.1. Reports	6
4.2.2. Otros	6
4.3. database	7
4.4. navigation	7

1. Introducción

La geología es una disciplina fundamental para el estudio y comprensión de los procesos que conforman la Tierra. A lo largo de la historia, los geólogos han utilizado diversas herramientas y metodologías para investigar los materiales que componen nuestro planeta, así como para entender los fenómenos naturales que ocurren a lo largo del tiempo. En la actualidad, la tecnología

y la digitalización han revolucionado el trabajo de los geólogos, permitiendo un enfoque más preciso, eficiente y accesible para la recopilación y análisis de datos.

Este proyecto tiene como objetivo desarrollar una aplicación móvil para geólogos, diseñada específicamente para asistir en la recopilación, organización y análisis de datos de campo en tiempo real. La aplicación integrará varias funcionalidades, tales como el almacenamiento de notas, fotos, tablas y la gestión de informes geológicos. A través de su interfaz, los geólogos podrán registrar información relacionada con las muestras recolectadas, sus coordenadas geográficas, las características de las rocas analizadas y otras variables de interés. Además, la aplicación permitirá la creación de informes con plantillas específicas para diferentes tipos de rocas y otros elementos geológicos, facilitando así la generación de informes técnicos de forma rápida y eficiente.

La importancia de este proyecto radica en la optimización del trabajo de campo, donde la recopilación de datos de manera eficiente y organizada es esencial para el éxito de cualquier investigación geológica. La aplicación propuesta no solo proporcionará un sistema de almacenamiento de datos robusto, sino también herramientas de análisis y presentación de los mismos, mejorando la productividad y precisión de los geólogos durante sus estudios.

En este contexto, el uso de herramientas como React Native, Expo y SQLite, junto con librerías especializadas, permitirá que la aplicación sea funcional, accesible y adaptada a las necesidades específicas del entorno de trabajo de los geólogos. Este proyecto, por tanto, busca no solo facilitar el trabajo de campo, sino también contribuir al avance en la investigación geológica mediante el uso de tecnologías innovadoras.

2. Estructura del Proyecto

La estructura del proyecto está organizada de manera que favorece la modularidad, el mantenimiento y la escalabilidad. A continuación se describe la disposición de los principales directorios y archivos en el proyecto:

- **/src:** Este es el directorio principal que contiene todo el código fuente de la aplicación. A continuación se detallan los subdirectorios y archivos más relevantes dentro de esta carpeta:
 - **/assets:** Aquí se almacenan todas las imágenes y recursos estáticos utilizados en la aplicación, tales como iconos, imágenes de fondo y otros elementos visuales.

- **/components:** Contiene los componentes reutilizables de la aplicación, tales como botones personalizados, formularios, cuadros de texto y otros elementos de interfaz de usuario que se pueden utilizar en múltiples pantallas.
 - **/database:** Este directorio incluye toda la lógica relacionada con la base de datos SQLite, como la creación, modificación y acceso a las tablas que almacenan los datos de la aplicación (notas, fotos, tablas y otros elementos), básicamente se encuentra todo el CRUD relacionado a cada entidad utilizada en la aplicación, además de que están las estructuras de cada elemento.
 - **/screens:** Contiene las pantallas principales de la aplicación. Cada una de estas pantallas está organizada de manera independiente para mejorar la claridad y facilidad de mantenimiento.
 - **/navigation:** Aquí se encuentra la configuración de la navegación de la aplicación, como el 'TabsNavigator' para la navegación entre pantallas. Este directorio gestiona el enrutamiento de la app para facilitar el desplazamiento entre diferentes secciones, todas las nuevas componentes o pantallas que se quieran ingresar en la aplicación deben estar en el navigator.
- **App.tsx:** Este archivo es el punto de entrada principal de la aplicación. Se encarga de inicializar la aplicación, configurar la navegación y establecer el entorno necesario para que la aplicación funcione correctamente.
 - **package.json:** Contiene las dependencias del proyecto, así como la configuración de scripts y otros detalles relevantes para la gestión del proyecto mediante npm o yarn.
 - **README.md:** Este archivo proporciona la documentación general del proyecto, incluyendo la descripción del propósito de la aplicación, las instrucciones para la instalación y configuración, y cualquier otra información relevante para los desarrolladores o colaboradores que trabajen en el proyecto.

La estructura modular de la aplicación facilita tanto el desarrollo como el mantenimiento de la misma, asegurando que cada componente y funcionalidad esté organizada de manera clara y accesible. Esta organización también permite una fácil expansión de la aplicación en el futuro, ya que nuevas funcionalidades o componentes pueden ser añadidos sin afectar la estructura general del proyecto.

3. Dependencias

Para poder utilizar el código del proyecto, es necesario instalar las siguientes dependencias:

- **Node.js:** Es un entorno de ejecución para JavaScript que permite ejecutar código JavaScript en el servidor, es obligatorio tenerlo para poder ejecutar la aplicación. Se puede descargar e instalar desde el sitio web oficial de Node.js.
- **Android Studio:** Posee un emulador de Android que permite probar la aplicación en un entorno virtual. Se puede descargar e instalar desde el sitio web oficial de Android Studio.
- **Expo CLI:** Es una herramienta no obligatoria de línea de comandos que facilita el desarrollo de aplicaciones móviles con React Native y Expo. Se puede instalar mediante npm o yarn.

4. Descripción del Código

4.1. screens

4.1.1. Photos

Descripción detallada de la pantalla de fotos, incluyendo la lógica y la interfaz de usuario.

- **PhotosScreen.tsx:** Archivo principal de la pantalla de fotos, aquí se permite tomar fotos, ver las fotos existentes y eliminarlas, además de que hay una opción de ver la ubicación de la imagen a través de Google Maps.

4.1.2. Tables

Descripción detallada de la pantalla de tablas, incluyendo la lógica y la interfaz de usuario.

- **TablesScreen.tsx:** Archivo principal de la pantalla de tablas, aquí se permite crear tablas, ver las tablas existentes y eliminarlas, además de poder ordenarlas según el filtro que convenga.
- **TableEditorScreen.tsx:** Pantalla de edición de tablas, donde se pueden modificar los contenidos de una tabla existente. Dentro de esta pantalla se pueden agregar filas y columnas, así como editar los valores de las celdas.

4.1.3. Reports

Descripción detallada de la pantalla de informes, incluyendo la lógica y la interfaz de usuario.

- **ReportsScreen.tsx**: Archivo principal de la pantalla de informes, aquí se permite crear informes, ver los informes existentes y eliminarlos, además de poder ordenarlos según el filtro que convenga.
- **ReportEditorScreen.tsx**: Pantalla de edición de informes, donde se pueden modificar los contenidos de un informe existente. Dentro de esta pantalla se puede escoger una de las plantillas propuestas para acelerar el proceso de creación de informes, también se pueden agregar imágenes y modificar la tabla escogida según el tipo de informe que se este creando.

4.1.4. Notes

Descripción detallada de la pantalla de notas, incluyendo la lógica y la interfaz de usuario.

- **NotesScreen.tsx**: Archivo principal de la pantalla de notas, aquí se permite crear notas, ver las notas existentes y eliminarlas.
- **NoteEditorScreen.tsx**: Pantalla de edición de notas, donde se pueden modificar los contenidos de una nota existente. Dentro de esta pantalla se puede agregar imágenes y tablas previamente ya creadas en la aplicación.

4.1.5. Litologic

Descripción detallada de la pantalla de litología, incluyendo la lógica y la interfaz de usuario.

- **LithologicListScreen.tsx**: Archivo principal de la pantalla de litología, aquí se permite crear litologías, ver las litologías existentes y eliminarlas, además de poder ordenarlas según el filtro que convenga.
- **LithologicFormScreen.tsx**: Pantalla de edición de litologías, donde se pueden modificar los contenidos de una litología existente. Dentro de esta pantalla se puede agregar imágenes y tablas previamente ya creadas en la aplicación.

4.1.6. StructuralDatas

Descripción detallada de la pantalla de datos estructurales, incluyendo la lógica y la interfaz de usuario.

- **StructuralDatasScreen.tsx**: Archivo principal de la pantalla de datos estructurales, aquí se permite crear datos estructurales, ver los datos estructurales existentes y eliminarlos, además de poder ordenarlos según el filtro que convenga.
- **PhotoSelectorScreen.tsx**: Pantalla de selección de fotos, donde se pueden elegir una foto existente para asociarla a un dato estructural. Dentro de esta pantalla se pueden ver las fotos existentes y seleccionar una de ellas.

4.2. components

4.2.1. Reports

- **ReportForm.tsx**: Componente relacionado con la creación y edición de informes, permite al usuario ingresar los datos necesarios para crear un informe, que cambia dependiendo del tipo de roca seleccionada.
- **TableEditor.tsx**: Componente relacionado con la edición de tablas, permite al usuario modificar los contenidos de una tabla existente, agregando o eliminando filas y columnas según sea necesario.
- **PhotoSection**: Componente relacionado con la visualización de fotos en un informe, permite al usuario ver una foto y añadir en el informe.
- **StreckeisenDiagram.tsx**: Componente relacionado con la visualización de un diagrama de Streckeisen, permite al usuario ver un diagrama de clasificación de rocas en base a los valores dados en la tabla creada previamente.
- **PDFGenerator.tsx**: Componente relacionado con la generación de archivos PDF, permite al usuario generar un archivo PDF con todos los datos del reporte creado.

4.2.2. Otros

- **ImageSelectionDialog.tsx**: Componente relacionado con la selección de imágenes, permite al usuario elegir una imagen creada con anterioridad en la aplicación, donde se puede ordenar por varios parámetros existentes.

- **InsertOptionsDialog.tsx:** Componente relacionado con la inserción de elementos (tablas o imagenes) en las notas, permite al usuario elegir entre insertar una imagen o una tabla en la nota que se esta creando o editando.
- **OfflineMapScreen.tsx:** Componente relacionado con la visualización de mapas sin conexión, permite al usuario ver la ubicación de una imagen, incluso si no hay conexión a Internet.
- **PdfViewerScreen.tsx:** Componente relacionado con la visualización de archivos PDF, permite al usuario ver un archivo PDF en la aplicación.
- **PhotoComponent.tsx:** Componente relacionado con la visualización de fotos, permite al usuario ver una foto en la aplicación.
- **TableComponent.tsx:** Componente relacionado con la visualización de tablas, permite al usuario ver una tabla en la aplicación.
- **TableSelectionDialog.tsx:** Componente relacionado con la selección de tablas, permite al usuario elegir una tabla creada con anterioridad en la aplicacion, donde se puede ordenar por varios parametros existentes.

4.3. database

- **database.ts:** Archivo principal de la base de datos, donde se establece la conexión con la base de datos SQLite y se definen las tablas y entidades utilizadas en la aplicación.

4.4. navigation

- **TabsNavigator.tsx:** Archivo principal de la navegación de la aplicación, donde se definen las pestañas y las pantallas principales de la aplicación.
- **types.ts:** Archivo de tipos de datos, donde se definen los tipos de datos utilizados en la aplicación.