



Universidad Autónoma
de San Luis Potosí

Universidad Autónoma de San Luis Potosí



FACULTAD DE
INGENIERÍA

Facultad de Ingeniería

Fundamentos de Desarrollo Móvil

MANUAL DEL PROGRAMADOR

Alumno:

- *Jesús Antulio Jiménez Delgado*
- *Stephanie Monserrat Hervert Morales*

Profesor:

- *Francisco Everardo Estrada Velázquez*

06 / 12 / 2022

Esta aplicación móvil busca proveer las funcionalidades necesarias para la gestión de un listado de tareas. El presente manual explica la distribución y el desarrollo del código fuente, para la creación del proyecto y cada una de sus funcionalidades.

Tecnología

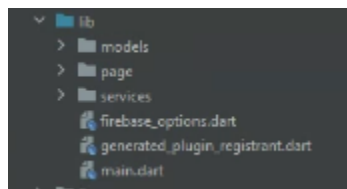
Para el desarrollo de este proyecto se utilizó Android Studio el cual es un entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y está basado en IntelliJ IDE. Las tecnologías utilizadas son:

- Firebase: uso de bases de datos no relacionales.
- Android Studio: framework empleado para desarrollar todo el Front-End de la aplicación.
- Node.JS: framework empleado para el desarrollo del Back-End, desde la implementación de módulos como Express para realizar el enrutamiento, como el manejo de middlewares, procesamiento de peticiones GET y POST, manejo del filesystem para almacenar archivos (particularmente imágenes), y lo utilizamos para el setup de Android Studio.

Distribución

Todo el código del proyecto está almacenado en un repositorio en GitHub, al cual todos los miembros del equipo de desarrollo tienen acceso. En dicho repositorio, cada miembro tiene su rama para hacer los cambios que requiera y poder hacer pruebas sin alterar el producto sin errores que está en la rama principal (main).

Dentro del repositorio, se encuentran todos los archivos necesarios para la ejecución de la aplicación, pero la principal que se utiliza es la carpeta lib. La cual contiene las carpetas: models, page, services y los archivos: firebase y el main.



models

Esta carpeta contiene los modelos necesarios para que la aplicación funcione correctamente y se reutilice la menor cantidad de código posible, además de que se mantenga la estructura de los objetos. El archivo dentro de esta carpeta es:

- response.dart:
 - Modelo generado para recibir objetos de la base de datos, específicamente de la tabla task, cada objeto tiene los siguientes datos:
 - code: integer.

- message: string.
- task.dart:
 - Modelo generado con el módulo de firebase para recibir objetos de la base de datos, específicamente de la tabla task, cada objeto tiene los siguientes datos:
 - uid: string.
 - title: string.
 - subtitle: string.

page

Esta carpeta contiene prácticamente todo el código necesario para el funcionamiento correcto del Front-End de la aplicación, desarrollado en Android Studio, como fue mencionado anteriormente.

service

Esta carpeta contiene todo el código necesario para que el Back-End funcione correctamente, y el framework sobre el que está desarrollado es Node.JS, con conexión a la base de datos de firebase.

Back-End

firebase_options.dart

Este archivo contiene todo lo necesario para conectarse a la base de datos de firebase. Hay que tomar en cuenta que si se lleva a mover algo aquí podría llegar a no funcionar la base de datos, la forma en la que se instala es mediante terminal.

firebase_crud.dart

Se localiza en la carpeta service, este archivo cuenta con las funciones necesarias para modificar la base de datos. Estas funciones son:

- addTask: recibe a través de un objeto de request los valores necesarios para registrar una tarea en la aplicación. Retorna un mensaje de éxito si se generó correctamente, y si no, retorna un mensaje de error.
- updateTask: recibe el id específico de una tarea y como objeto de request los valores necesarios para registrar una tarea en la aplicación, si un valor no es enviado, pide que llene el espacio faltante. Retorna un mensaje de éxito si se actualizó correctamente, y si no, retorna un mensaje de error.
- deleteTask: recibe el id específico de una tarea y hace la búsqueda del mismo para eliminar el registro de la base de datos en firebase.
- readTask: no recibe nada y devuelve a quien hace la petición una lista donde cada elemento es un objeto Task.

Front-End

La carpeta de mayor relevancia para el front-end, dentro de la carpeta “lib” es “page”. Se encuentran los archivos necesarios para el funcionamiento del proyecto, como mostrar las tareas y las pantallas de edición y creación.

Los archivos de mayor relevancia en el front-end son:

- main.dart: contiene una promesa que arrancará nuestra aplicación cuando lo ejecutamos en el emulador y llamará a la página “Login”.
- login.dart: es una maquetación de una pantalla para entrar dentro de la aplicación de forma “autenticada”, esta pantalla es simplemente maquetación y desde ella se puede llamar a la página “Register” o “List”.
- register.dart: es una maquetación de una pantalla para registrarse dentro de la aplicación, esta pantalla es simplemente maquetación y desde ella se puede llamar a la página “Login” o “List”.
- listtask.dart: esta pantalla visualizar todas las tareas que se encuentran en la base de datos, de esta pantalla se pueden llamar a la página “Add” o “Edit” o llamar a la función “Delete” la cual elimina la tarea de la base de datos.
- addtask.dart: esta pantalla es para agregar tareas, en caso de que se agregue correctamente se mostrará un mensaje de éxito, en caso contrario se le pedirá que llene el espacio faltante. Desde ella se puede llamar a la página “List”.
- edittask.dart: esta pantalla es para actualizar tareas, se le mostrará una pantalla similar que la página “Add” con los campos ya llenos y el id que se había generado cuando se creó la tarea, en caso de que se actualizo correctamente se mostrará un mensaje de éxito, en caso contrario, el cual falte alguno de los campos, se le pedirá que llene el espacio faltante. Desde ella se puede llamar a la página “List”.