

Arquitectura del proyecto

Arquitectura del Proyecto

La arquitectura del proyecto estará basada en una arquitectura de cliente-servidor donde la app móvil en Flutter actúa como cliente, y el backend en Python (FastAPI) sirve como servidor y gestor de lógica de negocio. Los datos se almacenarán y sincronizarán en tiempo real en la base de datos de Firebase.

Esta arquitectura permite un procesamiento eficiente en la nube, manteniendo la app ligera y rápida en los dispositivos móviles.

Componentes Principales

1. Frontend (App Móvil en Flutter)

- Tecnología: Flutter
 - Responsabilidad: Interfaz de usuario y experiencia; gestión de transacciones, cuentas, presupuestos y reportes; comunicación con el backend y Firebase en tiempo real.
- Características clave:
 - Soporte multiplataforma (Android e iOS).
 - UI/UX minimalista para mantener el proyecto KISS.
 - Sincronización en tiempo real con Firebase.

2. Backend (API en Python con FastAPI)

- Tecnología: Python, FastAPI
 - Responsabilidad: Lógica de negocio y procesamiento en la nube, con endpoints que manejan las operaciones de contabilidad y envían datos a Firebase.

- Características clave:
 - Validación de datos.
 - Procesamiento de reportes y cálculos contables complejos.
 - Comunicación con Firebase para almacenamiento y sincronización de datos.
 - Uso de REST API para la comunicación con la app Flutter.

3. Base de Datos en la Nube (Firebase Firestore)

- Tecnología: Firebase Firestore
 - Responsabilidad: Almacenamiento de datos en tiempo real y sincronización automática entre dispositivos.
- Características clave:
 - Soporte en tiempo real para actualizaciones automáticas en la app.
 - Gestión de usuarios y autenticación con Firebase Authentication.
 - Seguridad y control de acceso mediante reglas de seguridad personalizadas.

4. Autenticación y Seguridad (Firebase Authentication y Reglas de Seguridad de Firestore)

- Tecnología: Firebase Authentication
 - Responsabilidad: Gestión de inicio de sesión de usuarios y autorización de acceso a los datos.
- Características clave:
 - Autenticación de usuarios (email y contraseña u otros métodos).
 - Reglas de seguridad para que cada usuario solo acceda a su propia información.

Diagrama Resumido de Arquitectura

Flutter (App móvil)

<->

Backend (FastAPI en Python)

<->

Firebase Firestore

Tecnologías a Utilizar

- Frontend: Flutter (con Dart)
- Backend: Python (FastAPI para la API)
- Base de Datos y Sincronización: Firebase Firestore
- Autenticación y Seguridad: Firebase Authentication
- Almacenamiento de Archivos (si es necesario): Firebase Storage
- Control de Versiones: Git (GitHub o similar)

Esta arquitectura asegura escalabilidad y sincronización en tiempo real, manteniendo una estructura ligera y fácil de mantener.