

Firebase es una plataforma de desarrollo de aplicaciones creada por Google. Proporciona una serie de servicios en la nube que permiten a los desarrolladores construir aplicaciones

Para trabajar con firebase primero se tiene que crear un nuevo proyecto dentro de la pagina o en consola, asociarlo a algún servicio de autenticacion como google, facebook o con correo y contraseña y copiar o descargar el archivo con las credenciales, en mi caso:

```
import { initializeApp } from "firebase/app";
import { getAnalytics } from "firebase/analytics";

const firebaseConfig = {
  apiKey: "AIzaSyDlyrO3fuH06CjIeNk2mA4jpwjda8DFa3Q",
  authDomain: "proyecto-prueba-fe0c9.firebaseio.com",
  projectId: "proyecto-prueba-fe0c9",
  storageBucket: "proyecto-prueba-fe0c9.firebaseio.com",
  messagingSenderId: "922986580561",
  appId: "1:922986580561:web:f3864e3019712fbdbbc670",
  measurementId: "G-EGY8DZ1RT1"
};

const app = initializeApp(firebaseConfig);
const analytics = getAnalytics(app);
```

Después de eso se necesitan instalar las dependencias para trabajar con firebase dentro del visual con npm en la consola:

-npm init -y

-npm install express mysql2 body-parser firebase-admin cors

Y se escribe un inicio de sesion básico con un middleware para la verificación del token en firebase y la conexión con la base de datos mysql

```
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const mysql = require("mysql2");
const admin = require("firebase-admin");

const app = express();
app.use(cors());
app.use(bodyParser.json());

// ♦ Inicializar Firebase con credenciales
const serviceAccount = require("../firebase/serviceAccountKey.json");
```

```
admin.initializeApp({
  credential: admin.credential.cert(serviceAccount),
});

const db = mysql.createConnection({
  host: "localhost",
  user: "root",
  password: "password",
  database: "base_datos",
});

db.connect((err) => {
  if (err) {
    console.error("Error conectando a MySQL:", err);
  } else {
    console.log("Conectado a MySQL ✅");
  }
});

// ♦ Middleware para verificar token de Firebase
async function verificarToken(req, res, next) {
  const token = req.headers.authorization?.split(" ")[1];
  if (!token) return res.status(401).json({ error: "No autorizado" });

  try {
    const decoded = await admin.auth().verifyIdToken(token);
    req.user = decoded;
    next();
  } catch (err) {
    res.status(401).json({ error: "Token inválido" });
  }
}

// ♦ Rutas
app.get("/", (req, res) => {
  res.send("Servidor Express con Firebase + MySQL");
});

// Ejemplo: insertar usuario autenticado en MySQL
app.post("/usuarios", verificarToken, (req, res) => {
  const { nombre } = req.body;
  const uid = req.user.uid; // ID de Firebase
```

```

const email = req.user.email;

db.query(
  "INSERT INTO usuarios (uid, nombre, email) VALUES (?, ?, ?)",
  [uid, nombre, email],
  (err, result) => {
    if (err) return res.status(500).json({ error: err });
    res.json({ message: "Usuario guardado en MySQL", id:
result.insertId });
  }
);
});

// ♦ Iniciar servidor
app.listen(3000, () => console.log("Servidor corriendo en
http://localhost:3000"));

```

Dentro del código el usuario inicia sesión con alguna de las formas que se dio en firebase como por ejemplo google y se verifica si se tiene el token dado con jwt y si es correcto la consulta se guarda en firebase

Firebase podría funcionar para <la seguridad del programa ya que facilita el uso de jwt y cuenta distintas formas de inicio de sesión ayudando a la forma en la que se trabaja dentro del código y aligerando el peso aquí obviamente no se cuentan los otros usos que se le pueden dar como podrían ser el hosting y notificaciones y sobretodo storage para guardar los archivos.