

Topologia e Organização do Armazenamento

O projeto utiliza o **Firebase Firestore** como banco de dados. O Firestore funciona com uma **estrutura de coleções e documentos**, o que facilita organizar os dados de cada usuário separadamente e de forma segura. Cada coleção representa um tipo de dado, e cada documento dentro dessa coleção guarda informações individuais.

A topologia do banco de dados do projeto foi planejada da seguinte forma:

Coleção usuario

Armazena os dados de cada usuário registrado, como nome, email e data criacao. Cada documento tem como ID gerado pelo Firebase . Isso garante que cada usuário tenha suas informações separadas e protegidas.

Coleção transacoes

Cada documento representa uma transação financeira do usuário, contendo campos como:

tipo → receita ou despesa

descricao → descrição do gasto ou entrada

valor → valor da transação

data → data da transação

categoria → categoria da despesa ou receita

Usuario id → ID do usuário dono da transação

criado em → data/hora do registro

Essa organização permite recuperar todas as transações de um usuário rapidamente, usando consultas filtradas pelo usuário id.

Coleção relatorios

Cada documento representa um relatório financeiro gerado pelo usuário, contendo:

nome → título do relatório

criado em → data de criação

usuario id → ID do usuário

Possíveis campos adicionais para métricas ou filtros do relatório

Coleção projecoes

Guarda projeções financeiras do usuário, como estimativas de receitas e despesas futuras.

Essa estrutura mantém os dados **organizados por tipo**, garantindo que consultas e atualizações sejam rápidas e seguras.

2. Justificativa da Estrutura

Usar coleções separadas para transacoes ,relatorios e projecoes facilita a manutenção do projeto e evita confusão de dados.

Associar cada documento a um usuario id garante **segurança e privacidade**, permitindo que cada usuário acesse apenas seus próprios dados.

A estrutura permite **escalabilidade**, caso seja necessário adicionar novos tipos de dados, como metas financeiras ou categorias personalizadas.

3. Evidências do Banco de Dados

No **Firebase Console**, os dados ficam visíveis na aba **Firestore Database**. Cada coleção aparece listada, e ao clicar nela, é possível ver os documentos com seus campos e valores.

Exemplo de evidências para o relatório:

Coleção transacao → mostrando transações registradas, cada documento com descricao, valor,tipo , categoria,data e usuario id.

Coleção relatorios → mostrando relatórios gerados por usuários, com nome, criado em e usuario id.

Coleção usuario → mostrando os dados básicos dos usuários registrados.

Firebase

financiaFacil ▾ Cloud Firestore > Banco de dados

Visão geral do projeto | Configurar o App Check | Visualização do painel | Criador de consultas | Mais no Google Cloud

Alinhos de projetos

Authentication

Firestore Database

Categorias dos produtos

Criação Executar Analytics AI

Ferramentas de desenvolvimento relacionadas

Firebase Studio

Spark Sem custos (US\$ 0/mês) Fazer upgrade (NOVO)

(default) categorias Y50euwzbpo58Yf4dpKZh

+ Iniciar coleção

analises categorias >

contas economias metas projecoes relatorios transacoes usuarios

+ Adicionar documento

Y50euwzbpo58Yf4dpKZh

+ Iniciar coleção

+ Adicionar campo

cor: "#667eea"
criadoEm: 27 de novembro de 2025 às 19:38:28 UTC-3
nome: "kk"
tipo: "despesa"
userId: "4ohdlP8GeIvQbuGNREOEorB3Ny1"

Mais no Google Cloud

Firebase

financiaFacil ▾ Cloud Firestore > Banco de dados

Visão geral do projeto | Configurar o App Check | Visualização do painel | Criador de consultas | Mais no Google Cloud

Alinhos de projetos

Authentication

Firestore Database

Categorias dos produtos

Criação Executar Analytics AI

Ferramentas de desenvolvimento relacionadas

Firebase Studio

(default) metas UzPqSQnoQ7gp0Ag98Rsw

+ Iniciar coleção

analises categorias >

contas economias metas projecoes relatorios transacoes usuarios

+ Adicionar documento

UzPqSQnoQ7gp0Ag98Rsw

+ Iniciar coleção

+ Adicionar campo

categoria: "profissional"
criadoEm: 27 de novembro de 2025 às 20:17:24 UTC-3
data: "2025-12-27"
nome: "gustavo"
userId: "4ohdlP8GeIvQbuGNREOEorB3Ny1"
valor: 1234
valorAtual: 122

Mais no Google Cloud

EXPLORADOR ...

EDITORES ABERTOS

- WORKSPA...
- htmls
 - js
 - firebase-config.js**
 - duvidas.js
 - economias.js
 - login.js
 - metas.js
 - orcamentos.js
 - pagina inicial.js

ESTRUTURA DO CÓDIGO

LINHA DO TEMPO firebase-co...

- Arquivo Salvo 1 hora
- Desfazer / Refazer
- Arquivo Salvo
- Arquivo Salvo
- Arquivo Salvo 2 horas
- Arquivo Salvo 3 horas
- Arquivo Salvo
- Desfazer / Refazer
- Arquivo Salvo

```

firebase-config.js x JS projecoes.js JS cadastro.js ⇢ categorias.html ⇢ configuracoes.html JS configuracoes.js JS contas.js

htmls > js > firebase-config.js > ...
1 import { initializeApp } from "https://www.gstatic.com/firebasejs/10.7.0/firebase-app.js";
2 import { getFirestore } from "https://www.gstatic.com/firebasejs/10.7.0.firebaseio-firebase.js";
3 import { getAuth } from "https://www.gstatic.com/firebasejs/10.7.0/firebase-auth.js";
4
5 const firebaseConfig = {
6   apiKey: "AIzaSyCDdn4EMsqEbX9f0Q0TISiwH0AvCORRUo",
7   authDomain: "financiafacil-c9238.firebaseioapp.com",
8   projectId: "financiafacil-c9238",
9   storageBucket: "financiafacil-c9238.firebaseiostorage.app",
10  messagingSenderId: "490435881150",
11  appId: "1:490435881150:web:599d421d69747256ec16ef",
12  measurementId: "G-S2CHBRQGG3"
13 };
14
15 const app =.initializeApp(firebaseConfig);
16 export const db = getFirestore(app);
17 export const auth = getAuth(app);
18

```

Arquivo **Editar** **Seleção** **Ver** **Acessar** **Executar** ... **← →** **workspace (Workspace)**

EXPLORADOR ...

EDITORES ABERTOS

- WORKSPA...
- htmls
 - js
 - duvidas.js
 - economias.js
 - firebase-config.js**
 - login.js**
 - metas.js
 - orcamentos.js
 - pagina inicial.js

ESTRUTURA DO CÓDIGO

LINHA DO TEMPO login.js

- Arquivo Salvo 3 horas
- Arquivo Salvo
- Desfazer / Refazer
- Arquivo Salvo
- Arquivo Salvo
- Arquivo Salvo
- Arquivo Salvo

```

login.js x JS transacoes.js ⇢ transacoes.html ⇢ analise-gastos.html JS analise-gastos.js JS categorias.js # cadastro.css JS login.js

htmls > js > login.js > ...
1 import { auth } from "./firebase-config.js";
2 import { signInWithEmailAndPassword } from "https://www.gstatic.com/firebasejs/12.6.0/firebase-auth.js";
3
4 document.getElementById('formLogin').addEventListener("submit", async (e) => {
5   e.preventDefault();
6
7   const email = document.getElementById("inputEmail").value;
8   const senha = document.getElementById("inputSenha").value;
9
10  if (!email.includes("@")) return alert("Digite um email válido.");
11  if (senha.length < 6) return alert("A senha precisa ter pelo menos 6 caracteres.");
12
13  try {
14    await signInWithEmailAndPassword(auth, email, senha);
15    alert("Login realizado com sucesso!");
16    window.location.href = "pagina inicial.html";
17  } catch (error) {
18    alert("Erro no login: " + error.message);
19  }
20 });
21

```