**🧠 1. A lógica geral do projeto**

Tens um **web app simples com login e base de dados**, dividido em dois ecrãs principais:

| **Página** | **Função principal** |
| --- | --- |
| index.html | Registar ou fazer login de utilizador |
| home.html | Mostrar e editar os dados pessoais do utilizador |

**🌐 2. A interação entre HTML e JavaScript**

O HTML serve para **mostrar a interface** (inputs, botões, texto).  
O JavaScript **controla o comportamento** (guardar dados, fazer login, mostrar dados guardados).

**📦 Exemplo:**

html

CopiarEditar

<input id="email" type="email" placeholder="Email" />

Este campo é só o visual. Depois, no JavaScript, vais **buscar o valor que o utilizador escreveu** com:

js

CopiarEditar

const email = document.getElementById("email").value;

**🔗 3. O papel de cada ficheiro**

**📄 index.html + app.js**

* HTML mostra inputs de login/registo
* JS usa **Firebase Auth** para criar ou autenticar utilizadores
* Se o login for bem-sucedido, vai para home.html

**📄 home.html + home.js**

* HTML mostra campos para guardar/ver dados
* JS usa **Firebase Auth** para garantir que o utilizador está logado
* Depois, usa **Firebase Firestore** para:
  + Ler os dados do utilizador
  + Mostrar os dados guardados
  + Atualizar ou criar novos dados

**📄 firebase-config.js**

* Configura a ligação ao teu projeto Firebase
* Exporta o auth (autenticação) e db (base de dados)

**⚙️ 4. O que é JavaScript puro vs. o que é Firebase/específico do projeto**

**🟩 JavaScript puro (presente em qualquer projeto):**

| **Conceito JS** | **O que faz** |
| --- | --- |
| document.getElementById | Vai buscar elementos HTML |
| .addEventListener() | Espera por cliques ou ações do utilizador |
| async / await | Espera por operações assíncronas |
| if / else | Lógica condicional |
| console.log() | Mostrar info no terminal (debug) |
| Variáveis (const, let) | Guardar dados temporários no código |

Estes são usados **em qualquer site/app web** com JavaScript.

**🟦 Firebase / específico do teu projeto:**

| **Firebase função** | **O que faz no teu projeto** |
| --- | --- |
| createUserWithEmailAndPassword | Registar utilizador na Firebase Auth |
| signInWithEmailAndPassword | Fazer login |
| onAuthStateChanged | Verifica se o utilizador está autenticado |
| addDoc, getDocs, updateDoc | Criar, buscar ou editar dados no Firestore |
| collection, query, where | Trabalhar com coleções e filtros no Firestore |

Estas funções são da **biblioteca Firebase** e dependem do teu projeto estar ligado ao Firebase.

**🔄 5. O fluxo geral da app**

plaintext

CopiarEditar

index.html + app.js

↓

[Utilizador faz login ou registo]

↓

home.html + home.js

↓

[Carrega dados do utilizador da Firestore]

↓

[Mostra na página e permite editar]

**🧩 6. Resumo: Estrutura da lógica**

| **Camada** | **Função no projeto** |
| --- | --- |
| HTML | Apresentação visual (inputs, botões, texto) |
| JavaScript Base | Manipulação de elementos, lógica geral |
| Firebase Auth | Gestão de utilizadores (registo/login) |
| Firebase Firestore | Guardar e recuperar dados pessoais por utilizador |