

**MODELOS, MÉTODOS E TÉCNICAS DE ENGENHARIA DE SOFTWARE**  
**PROF. MARCONE DA SILVA FERREIRA**

DANIEL BRAGA ALVES DO COUTO - 62312771

GABRIEL MENDES BITENCOURT - 62321162

GIOVANNA MONTEIRO ROCHA - 62320643

HUGO GONÇALVES COSTA - 62321289

YASMIM VITÓRIA SOARES PEREIRA - 62313109

**DOCUMENTAÇÃO TÉCNICA DO SISTEMA**

***EASINGLIVING***

Betim, 24 de novembro de 2025

## **TECNOLOGIAS USADAS:**

HTML, CSS Tailwind, JavaScript (Vanilla) e Firebase v9 (Modular) com repositórios disponibilizados via GitHub e hospedados via Vercel.

### **1. VISÃO GERAL**

O **EasingLiving** é uma aplicação web SPA (Single Page Application) destinada à apresentação de projetos arquitetônicos e gestão de pedidos de compra/orçamento. O sistema conecta clientes interessados em projetos residenciais pré-definidos com a equipe de arquitetura. A infraestrutura é Serverless, utilizando o Google Firebase para autenticação e banco de dados.

### **2. ARQUITETURA DE DADOS (FIREBASE FIRESTORE)**

O banco de dados é NoSQL, estruturado em coleções de documentos. Abaixo está o esquema detalhado dos dados.

#### **2.1. COLEÇÃO: USERS**

Armazena dados de perfil e controle de acesso, vinculados ao Firebase Authentication.

Campo	Tipo	Obrigatório	Descrição
<b>uid</b>	String	Sim	Chave Primária (Mesmo ID do Auth).
<b>name</b>	String	Não	Nome completo do usuário.
<b>email</b>	String	Não	E-mail de cadastro.
<b>role</b>	String	Sim	Define o nível de acesso: client ou admin.
<b>createdAt</b>	Timestamp	Sim	Data de criação da conta.

#### **2.2. COLEÇÃO: REQUESTS**

Registra todas as solicitações de orçamento ou compras iniciadas pelos usuários.

Campo	Tipo	Obrigatório	Descrição
<b>id</b>	String	Sim	ID automático do documento.
<b>protocolo</b>	String	Sim	Código visual para o cliente (Ex: EL-X9A2-123).
<b>userId</b>	String	Sim	Referência ao UID do usuário (Chave Estrangeira).
<b>client</b>	String	Sim	Nome do cliente no momento do pedido.
<b>category</b>	String	Sim	Nome do pacote (Ex: "Casa Contemporânea").
<b>type</b>	String	Sim	Tipo de aquisição (Ex: "Lote Próprio").
<b>status</b>	String	Sim	Estado do pedido: Novo, Em Análise, Concluído.
<b>date</b>	String	Sim	Data formatada (pt-BR) para exibição.
<b>createdAt</b>	Timestamp	Sim	Data real para ordenação.
<b>address</b>	String	Sim	Endereço da obra.

### 2.3. COLEÇÃO: MESSAGES

Armazena contatos enviados via formulário "Fale Conosco".

Campo	Tipo	Obrigatório	Descrição
<b>name</b>	String	Sim	Nome do remetente.
<b>email</b>	String	Sim	E-mail de contato.
<b>message</b>	String	Sim	Conteúdo da mensagem.
<b>date</b>	Timestamp	Sim	Data de envio.

### 3. LÓGICA DE NEGÓCIO E IMPLEMENTAÇÃO

#### 3.1. AUTENTICAÇÃO E SEGURANÇA

- **Método:** Email/Password via firebase/auth.
- **Controle de UI:** Elementos de navegação (Login/Logout/Painel) alternam a visibilidade baseados na existência de um user autenticado e no campo role recuperado do Firestore.

#### 3.2. DASHBOARD E PERMISSÕES

A visualização de dados no Dashboard é condicional:

1. **Administrador (role: 'admin'):**
  - Visualiza pedidos de **todos** os usuários.
  - Possui acesso a filtros de pesquisa (Texto, Categoria, Status).
  - A consulta ao banco ordena por createdAt decrescente.
2. **Cliente (role: 'client'):**
  - Visualiza apenas **seus próprios** pedidos.
  - A query utiliza a cláusula where("userId", "==", currentAuthUser.uid).

#### 3.3. CATÁLOGO DE PROJETOS

Para otimizar a performance e reduzir leituras no banco, os dados estáticos dos projetos (título, preço, imagens, descrição) estão definidos no const dadosProjetos no código Front-end ([script.js](#)). **Categorias:** Padrão, Médio-Alto, Alto Padrão, Excelente.

## 4. FRONTEND E UX

### 4.1. NAVEGAÇÃO

O sistema utiliza **Hash Navigation** para simular uma SPA sem recarregamento de página:

- #home: Landing Page.
- #dashboard: Área logada.
- #detalhe-pacote/[categoria]: Página dinâmica que carrega dados do objeto dadosProjetos.

### 4.2. Tema (Dark/Light Mode)

- Utiliza a classe dark na tag <html> (Tailwind CSS).
- A preferência é salva no localStorage.