



◊ Employee Onboarding Wizard

status em desenvolvimento

tecnologia React

persistência Firebase

UI Material-UI

📖 Descrição do Projeto

Employee Onboarding Wizard é uma aplicação web desenvolvida em **React** para otimizar e simplificar o processo de cadastro de novos colaboradores. O sistema conta com um dashboard para visualização e gerenciamento de funcionários e um formulário passo a passo (wizard) que guia o usuário no preenchimento das informações básicas e profissionais do novo colaborador. A aplicação utiliza **Firebase** para persistência de dados em tempo real.

■ Estrutura do Repositório

```
employee-onboarding-wizard/
|
|   └── public/                      # Arquivos estáticos e avatares
|
|   └── src/
|       ├── assets/                  # Logos e imagens da aplicação
|       ├── components/              # Componentes React reutilizáveis
|       |   ├── common/               # Componentes genéricos (Botão, Input)
|       |   ├── dashboard/            # Componentes do painel principal
|       |   ├── layout/                # Estrutura do layout (Header, Sidebar)
|       |   └── onboarding/            # Componentes do fluxo de cadastro
|       ├── config/                  # Configuração de serviços (Firebase)
|       ├── hooks/                   # Hooks customizados para lógica de estado
|       ├── pages/                   # Páginas principais da aplicação
|       ├── services/                # Lógica de comunicação com a API/backend
|       └── theme/                   # Tema customizado do Material-UI
|
└── package.json                    # Dependências e scripts do projeto
```

◎ Funcionalidades Principais

ID	Funcionalidade	Descrição
RF01	Dashboard de Funcionários	Apresenta uma tabela com os funcionários cadastrados, permitindo uma visualização rápida e centralizada.
RF02	Wizard de Cadastro	Um formulário multi-passo para cadastrar novos funcionários, coletando informações básicas e profissionais de forma organizada.
RF03	Navegação Intuitiva	Layout com menu lateral persistente e breadcrumbs para facilitar a localização do usuário dentro do sistema.
RF04	Persistência de	As informações dos funcionários são salvas e lidas do

ID	Funcionalidade	Descrição
	Dados	Firebase Firestore, garantindo dados consistentes e em tempo real.

❖ Tecnologias Utilizadas

- **Core:** React 19, TypeScript
 - **Build Tool:** Vite
 - **Roteamento:** React Router DOM
 - **UI Framework:** Material-UI (MUI)
 - **Backend e Banco de Dados:** Firebase (Firestore)
 - **Linting:** ESLint
-

🔥 Configuração do Firebase

Para facilitar a avaliação deste desafio, as credenciais do Firebase **já estão configuradas** diretamente no arquivo `src/config.firebaseio.ts`.

Não é necessário criar arquivos `.env` ou configurar um projeto novo no Firebase Console. Basta rodar o projeto e a conexão com o banco de dados funcionará automaticamente.

Nota: Em um ambiente de produção real, estas chaves seriam armazenadas em variáveis de ambiente (`.env`) e não seriam commitadas no repositório, seguindo as boas práticas de segurança (12-Factor App). Mantive hardcoded aqui estritamente para fins de demonstração e facilidade de execução do teste.

⚠️ Pré-requisitos

- Node.js \geq 18.x

- NPM ou Yarn
-

🚀 Instalação de Dependências

Na raiz do projeto, execute o comando para instalar todas as dependências:

```
npm install
```

💻 Como Rodar o Projeto

1. Siga os passos da seção **🔥 Configuração do Firebase**.
2. Após instalar as dependências, execute o comando abaixo para iniciar o servidor de desenvolvimento:

```
npm run dev
```

A aplicação está disponível em `http://localhost:5173` (ou outra porta, se a 5173 estiver em uso).

Nota: Se houver algum problema com o vite, pode ser necessário reinstalá-lo como uma dependência de desenvolvimento, com o seguinte comando:

```
npm install --save-dev vite
```