

Especificação Técnica — Aplicação Web React (Redux + MVC Adaptado)

■ Objetivo

Desenvolver uma aplicação web responsiva utilizando React e Redux para gerenciamento de estado global, seguindo uma arquitetura baseada em componentes e inspirada no padrão MVC. O sistema deverá permitir personalização de tema, identidade visual e ativos gráficos.

■■ Requisitos Técnicos

1. Arquitetura

- Utilizar React 18 com componentes funcionais e Hooks; - Redux Toolkit para gerenciamento de estado global; - React Router DOM para navegação entre páginas; - Padrão MVC adaptado: - Model: Definições de tipos, interfaces e acesso à API. - View: Componentes visuais e containers conectados ao Redux. - Controller: Actions, reducers e lógica de negócio encapsulada.

2. Responsividade

- O layout deve ser adaptável para Desktop, Tablet e Celular; - Utilizar CSS Grid, Flexbox e media queries; - Framework recomendado: TailwindCSS ou Styled Components.

3. Tema Central

- Criar um tema global com base em variáveis (CSS ou styled-components ThemeProvider); - Permitir troca dinâmica de cores, fontes e imagens; - Definir tokens de design reutilizáveis (spacing, colors, typography).

4. Personalização

- O site deverá permitir substituição de logo, cores e fontes por meio de um arquivo de configuração (config.json) ou endpoint remoto; - Implementar suporte a temas múltiplos (ex: tema claro e escuro).

■ Integração com API

Fonte de Dados: <https://newdiscapi-181165400335.southamerica-east1.run.app>

Endpoint Inicial: /v1/ObterPalavras Método: GET Header: X-API-Key: your_api_key

Exemplo de resposta:

```
{"palavras": [{"id": "1427", "descricao": "Simpático"}, {"id": "1428", "descricao": "Cuidadoso"}, {"id": "1429", "descricao": "Preocupado"}]}
```

■ Primeira Funcionalidade: Apresentação de Palavras-Chave

A aplicação deverá exibir as palavras obtidas da API /v1/ObterPalavras em uma interface limpa e responsiva. Cada palavra deve ser apresentada em um card estilizado, com carregamento dinâmico e tratamento de erros (mensagem amigável e retry).

■ Estrutura Sugerida de Pastas

```
src/ ■■■■ api/ ■ ■■■■ palavrasService.js ■■■■ assets/ ■■■■ components/ ■ ■■■■ PalavraCard.jsx  
■■■■ features/ ■ ■■■■ palavras/ ■ ■■■■ palavrasSlice.js ■ ■■■■ palavrasThunks.js ■■■■ hooks/  
■■■■ pages/ ■ ■■■■ PalavrasPage.jsx ■■■■ store/ ■ ■■■■ store.js ■■■■ styles/ ■ ■■■■ theme.js  
■■■■ main.jsx
```

■ Critérios de Aceite

- A aplicação deve executar corretamente em navegadores modernos (Chrome, Edge, Firefox); - Layout responsivo e fiel ao protótipo; - Integração bem-sucedida com a API; - Estado global gerido via Redux; - Tema configurável e fácil de personalizar; - Estrutura de código modular e escalável.