

**Universidade Estadual do Piauí - UESPI**  
**Curso de Tecnologia em Sistemas de Computação**  
**Disciplina: Programação Web**  
**Professor: Eyder Rios**

**3<sup>a</sup> Avaliação**  
**Trabalho de Implementação**  
**Data de Entrega: 02/12/2025**

## Tema

### **Explorador de Países e Culturas com REST Countries**

Desenvolver uma aplicação Web interativa que permita ao usuário explorar informações culturais, geográficas e políticas de países ao redor do mundo. Os dados devem ser obtidos exclusivamente da **REST Countries API**, sem necessidade de autenticação.

O uso de qualquer tecnologia de front-end (HTML, CSS, JavaScript ou frameworks) é **livre e opcional**.

## API Utilizada

- **Nome:** REST Countries API
- **URL:** <https://restcountries.com/>
- **Autenticação:** Nenhuma (acesso direto)
- **Dados disponíveis:** nome, capital, bandeira, população, área, continentes, línguas, moedas, localização geográfica, domínios de internet, entre outros.

## Resumo da Aplicação

Criar uma aplicação que permita:

1. Buscar países por nome
2. Filtrar países por continente
3. Visualizar detalhes: bandeira, capital, população, idiomas, moedas, etc.
4. Salvar países favoritos no navegador
5. Interface moderna, responsiva e intuitiva

## Requisitos Funcionais

1. Campo de busca por nome do país
2. Filtro por continente (ex: América, Europa, Ásia)
3. Listagem dos países com:
  - Nome, bandeira, capital, região, população
4. Página ou modal de detalhes com:

- Nome oficial, área, idiomas, moedas, domínios da internet, fronteiras (quando houver), mapa (embed)
5. Sistema de favoritos com armazenamento local ( `localStorage` )
  6. Tratamento de erros (ex: país não encontrado, campos ausentes)
  7. Spinner de carregamento durante requisições
  8. Página "Sobre" com informações dos autores
  9. **Rodapé em todas as páginas contendo os nomes dos membros da equipe** (obrigatório)

## Requisitos Não Funcionais

1. HTML5 semântico
2. Design responsivo (mobile-first)
3. Código modular e reutilizável
4. Boas práticas de UX/UI
5. Tratamento robusto de exceções
6. Versionamento progressivo e bem distribuído com Git

## Fator de Autenticidade (FA)

### Propósito

Promover desenvolvimento contínuo, evitar “entrega de última hora” e incentivar boas práticas de versionamento.

### Definição

$$FA = \min(1, (C/5) \times (I/6))$$

Onde:

- C: commits válidos (conteúdo real, mensagem clara, contribuição concreta)
- I: intervalo em horas entre primeiro e último commit (mínimo 6h para FA ideal)

### Exemplo

4 commits válidos

8 horas entre o primeiro e o último commit

$$\begin{aligned} FA &= \min(1, (4/5) \times (8/6)) \\ FA &= \min(1, 0.8 \times 1.33) \\ FA &= \min(1, 1.064) \\ FA &= 1 \end{aligned}$$

Commits concentrados em poucas horas ⇒ FA baixo

Evolução distribuída com mensagens descritivas ⇒ FA alto

### Nota Final

$$\text{Nota Final} = \text{Nota Técnica} \times FA$$

## Entrega

- Re却orio GitHub p煤blico com:
  - README.md contendo:
    - Descrição do projeto
    - Integrantes (até 4)
    - Tecnologias utilizadas
    - API utilizada
    - Link enviado via SIGAA até **02/12/2025**

## Critérios de Avaliação (Nota Técnica)

Nº	Critério	Peso
1	Interface + Responsividade	25%
2	Integração com API + Tratamento de erros	25%
3	JS modular + DOM + Assíncrono	20%
4	Documentação e estrutura do projeto	15%
5	Estilo, estética e criatividade	15%