PROJETO DA DISCIPLINA

1. Objetivo do projeto

Criar uma aplicação web que integre os conceitos fundamentais de HTML, CSS e JavaScript estudados na disciplina. O projeto deve evidenciar boas práticas, acessibilidade e interatividade, além de demonstrar a capacidade de desenvolver uma solução funcional e organizada.

2. Requisitos do projeto

2.1 Estrutura semântica e Acessibilidade

- Uso adequado de **elementos HTML5 semânticos** para estruturação de conteúdo (por exemplo, <header>, <nav>, <main>).
- Implementação de pelo menos 3 boas **práticas de acessibilidade**, como:
  - Texto alternativo para imagens.
  - Navegação acessível por teclado.
  - Contraste adequado entre texto e fundo.
  - Labels descritivos em formulários.

2.2 Layout responsivo

- Adaptação do layout para diferentes dispositivos usando **Flexbox, Grid e media queries**.
- **Design *mobile-first***, com otimizações para telas menores.
- Implementação de **menus responsivos** (por exemplo, o menu hambúrguer).

2.3 Interatividade com JavaScript

- Desenvolvimento de **funcionalidades interativas**, incluindo:
  - Manipulação do DOM (por exemplo, manipulação dinâmica de elementos).
  - Validação de formulários no front-end.
  - Uso de métodos avançados de arrays e objetos, tais como filter, map, reduce...

2.4 Integração de APIs

- Uso de pelo menos **2 APIs HTML5**, como Geolocation, Drag'n drop ou Web Storage, por exemplo.
- **Integração com APIs externas** (opcional) para enriquecimento da aplicação.

2.5 Organização do Código e Documentação

- **Código** modular, separado por componentes ou páginas (por exemplo, arquivos CSS e JS distintos).
- **Documentação** clara no repositório (README) contendo:
  - Propósito do projeto.
  - Como configurar e executar a aplicação.
  - Estrutura e principais funcionalidades.

3. Informações Gerais

3.1 Disponibilização do Código-fonte

- O código-fonte do projeto deverá ser disponibilizado no GitHub ou ser enviado em tarefa específica do AVA Moodle. O repositório deve incluir:
  - **Estrutura Modular:** Código organizado por componentes e módulos.
  - **Documentação:** Readme, explicando o propósito do projeto, como configurar o ambiente, etc.

3.2 Documento Descritivo do Projeto

Deve ser entregue um documento explicando o projeto, contendo:

- **Motivação e objetivo:** Problema que a aplicação busca resolver e justificativa de sua relevância.
- **Público-Alvo:** Perfil dos usuários finais.
- **Funcionalidades:** Lista das principais funcionalidades oferecidas.
- **Tecnologias utilizadas:** Tecnologias e ferramentas aplicadas no desenvolvimento.
- **Modelagem e prototipagem** (se aplicável): Incluindo wireframes e arquitetura da aplicação.

4. Critérios de Avaliação

4.1 Documentação

- **Preenchimento do *Template*:** Aderência ao template fornecido, com todas as seções devidamente preenchidas.
- **Código documentado:** Comentários claros e explicativos no código.

4.2 Apresentação

- **Explanação:** Clareza na explicação dos objetivos, público-alvo, e ferramentas usadas. Demonstração funcional da aplicação, mostrando responsividade, interatividade e acessibilidade.
- **Fluxo da Aplicação:** Explicação do fluxo de navegação e funcionamento da aplicação.

4.3 Implementação

- **Organização do projeto:** Separação clara de camadas e componentes, garantindo modularidade.
- **Semântica:** Uso correto de tags semânticas e API HTML.
- **CSS Responsivo e modular:** Aplicação de Flexbox, Grid e media queries para responsividade e separação de estilos por componentes.
- **JavaScript funcional:** Implementação de funcionalidades interativas que, por exemplo, manipulem o DOM, validem formulários e realizem operações assíncronas.
- **Segurança:** Implementação de práticas de segurança, como validação de dados e proteção contra ataques comuns.
- **Acessibilidade:** Aplicar algumas diretrizes de acessibilidade, garantindo utilização pelo maior número de usuários.

5. Checklist para orientar o desenvolvimento

5.1 Estrutura Semântica e Acessibilidade

- Uso adequado de elementos semânticos do HTML5 (<header>, <nav>, <main>, etc.).
- Aplicação de pelo menos 3 boas práticas de acessibilidade, como:
  - Texto alternativo para imagens.
  - Navegação funcional via teclado.
  - Contraste adequado entre texto e fundo.
  - Labels descritivos em formulários.
  - Feedback visual claro para elementos em foco.

5.2 Responsividade

- Uso de Flexbox e/ou CSS Grid em pelo menos 2 páginas.
- Aplicação de media queries para adaptar o design a diferentes dispositivos.
- Abordagem *mobile-first* e uso de unidades relativas (%, em, rem).

5.3 CSS Modular e Organizado

- Estilos separados por componentes ou páginas.
- Uso de pseudo-classes (e.g., :hover, :focus) para melhorar a interação.

5.4 JavaScript Interativo

- Implementação de ao menos 3 funcionalidades interativas, como:
  - Manipulação do DOM.
  - Validação de formulários.
  - Uso de métodos avançados (por exemplo, filter, map, reduce).

5.5 Validações e Feedbacks

- Validação de campos obrigatórios e formatos de entrada.
- Feedback claro ao usuário sobre ações e erros.

5.6 Uso de API

- Integração com pelo menos 2 APIs do HTML5 (e.g., Geolocation, Web Storage, Drag and Drop, Fetch, entre outras).
- Utilização opcional de APIs externas para enriquecer funcionalidades.

5.7 Boas Práticas de Código e Documentação

- Código bem organizado e comentado.
- Documento descritivo detalhado sobre o projeto (motivações, público-alvo, tecnologias, etc.).

5.8 Design e Usabilidade

- Aplicação de princípios de usabilidade (e.g., simplicidade e consistência).
- Interface visualmente agradável e intuitiva.

5.9 Elementos de Interface e navegação

- Implementação de pelo menos 2 elementos de interface, tais como:
  - Carrossel: Exibição dinâmica de imagens ou conteúdo.
  - Breadcrumbs: Indicação hierárquica de navegação para facilitar o retorno a páginas anteriores.
  - Menus suspensos ou hambúrguer: Navegação funcional e adaptável.
  - Modais: Janelas pop-up para mensagens ou formulários.
  - Tabs (abas): Organização de conteúdos por categorias.
  - Accordion: Expansão e colapso de informações para economizar espaço visual.