

Desenvolvimento Front-End

Prof.^a Ma. Jessica Oliveira



Apresentação da Disciplina

Plano de ensino disponível no AVA.

Avaliações

Composição das notas bimestrais.

- Os dois bimestres seguirão a seguinte divisão de pontos:
 - Prova individual teórica: **1,5 ponto**;
 - Exercícios pré-aula e práticas individuais realizadas em sala: **1,0 ponto**;
 - Projetos, em equipes, compostos pelas entregas via AVA: **2,0 pontos**.
- Além disso, há o ponto do Programa Protagonismo Discente (PPD), que soma mais **1,0 ponto** na nota do semestre.

Metodologia de Ensino

Proposta para o semestre.

Avaliação Diagnóstica

Orientações:

- Criar uma página HTML simples com um pequeno CSS *inline* ou interno.
- A página deve conter:
 - Um título (h1).
 - Um parágrafo (p).
 - Uma lista (ul ou ol).
 - Uma imagem (img).
 - Um botão (button).

Aula 01 – 18/02/2025

Introdução ao desenvolvimento front-end: a evolução da *web*.

Web 1.0: A Era Estática.

- 1991 até início dos anos 2000;
- Páginas estáticas escritas em HTML puro, sem interação dinâmica.
- Pouco uso de CSS, estilização mínima (muitas vezes *inline*).
- Estrutura baseada em tabelas (<table>) para organização do layout.
- Sites majoritariamente informativos, sem suporte a colaboração do usuário.
- Comunicação unidirecional (do servidor para o usuário, sem possibilidade de resposta direta).

Tecnologias da *Web* 1.0.

- HTML (*HyperText Markup Language*): estruturação de páginas *web*.
- CSS (*Cascading Style Sheets*) rudimentar: estilos básicos aplicados *inline* ou em arquivos externos.
- *JavaScript* (início tímido): usado de forma limitada para pequenas interações.

Web 2.0: A Web Dinâmica e Interativa.

- Meados dos anos 2000 até o presente;
- Páginas dinâmicas e interativas, com atualização de conteúdo em tempo real.
- Separação entre HTML para estrutura, CSS para estilo e *JavaScript* para interatividade.
- Uso extensivo de *AJAX (Asynchronous JavaScript and XML)* para carregamento assíncrono de dados.
- Sites colaborativos, como redes sociais e fóruns.
- Aplicações web complexas substituindo *softwares desktop*.

Tecnologias da Web 2.0.

- HTML5: novas tags semânticas (<header>, <footer>, <article>).
- CSS3: novos seletores, flexbox, grid, animações.
- JavaScript (ECMAScript 5 e 6+): manipulação avançada do DOM.
- Frameworks e bibliotecas: jQuery, AngularJS, ReactJS, Vue.js.
- APIs e AJAX: comunicação assíncrona entre cliente e servidor.
- Pré-processadores de CSS: Sass e Less.

Web 3.0: A Web Inteligente e Descentralizada.

- Atualidade e futuro.
- Inteligência Artificial e *Machine Learning* para personalização de conteúdo.
- Internet das Coisas (IoT), *blockchain* e descentralização de dados.
- Uso de APIs REST e GraphQL para comunicação eficiente entre sistemas.
- Interfaces altamente responsivas e acessíveis, utilizando PWA (*Progressive Web Apps*).

Tecnologias da Web 3.0.

- JavaScript ES6+ e TypeScript: código mais robusto e tipado.
- *Frameworks* modernos: React, Vue, Angular, Svelte.
- *WebAssembly* (WASM): permite rodar código de outras linguagens no navegador.
- *Blockchain* e descentralização: aplicações como contratos inteligentes e NFTs.
- Computação em nuvem e *edge computing*: processamento de dados distribuído.

Para realizar em sala...

- ~~Divisão da turma em **quatro equipes**;~~
- Definição do **projeto a ser desenvolvido**;
- Definição do **nome do repositório no GitHub** (e do nome da equipe, se quiserem);
- Estabelecer um canal de comunicação eficiente para a equipe;
- Criar um quadro de tarefas (Trello, Notion...) para auxiliar na documentação e divisão de tarefas;
- Criação do **rascunho para o README.md inicial**, descrevendo brevemente: nome do projeto, o propósito do sistema e a equipe responsável (nome dos integrantes);

- Organização das funções na equipe: planejem como irão dividir as tarefas dentro da equipe e escolham **quem ficará responsável por cada papel**, lembrando que todos podem, e devem, contribuir com o código:
 - **Gerente do Projeto:** responsável por organizar tarefas e prazos da equipe, além de fazer as entregas no AVA pela equipe.
 - **Líder de Versionamento:** responsável por garantir que o Git/GitHub esteja sendo usado corretamente.
 - **Desenvolvedores:** programam as funcionalidades do sistema.
 - **Documentador:** atualiza o README.md e auxilia na escrita dos relatórios.
 - **Tester/Validador:** testa o código e verifica se as funcionalidades estão corretas.

Para o pós-aula...

- Criar e configurar o repositório no GitHub.
- Criar um arquivo **README.md** com informações básicas do projeto.
- Criar a estrutura inicial do projeto em Java no VS Code.
- Demonstrar o uso do Git para versionamento (commit inicial, configuração do **.gitignore**).
- Relatório técnico explicando a configuração do ambiente e a importância do versionamento.

Dúvidas?

jessica.oliveira@p.ucb.br