

Desenvolvimento Front-End

Prof.^a Ma. Jessica Oliveira



Aula 06 – 01/04/2025

Construção de *Layouts* Responsivos.



Conceitos.

- A construção de *layouts* responsivos é uma técnica fundamental no desenvolvimento *web* moderno, que visa criar interfaces que se ajustam automaticamente a diferentes tamanhos de tela e dispositivos. Isso inclui desde *smartphones* e *tablets* até monitores *widescreen*.
- A responsividade não se refere apenas à adaptação do tamanho dos elementos, mas também ao **rearranjo**, ao **redimensionamento** e, muitas vezes, à **reformulação completa** da interface para **garantir uma boa experiência do usuário**.

A evolução do *Design Web*.

- Antes da popularização dos dispositivos móveis, os sites eram projetados com *layouts* fixos, ou seja, definidos em *pixels*.
- Isso resultava em uma experiência ruim quando visualizados em telas menores, pois exigia o uso de rolagem horizontal ou deixava os conteúdos distorcidos e mal alinhados.
- Com o aumento do acesso à internet por meio de dispositivos móveis, surgiu a necessidade de desenvolver sites capazes de se adaptar automaticamente ao ambiente em que eram visualizados.
- Isso levou ao conceito de *design* responsivo, um termo cunhado por Ethan Marcotte em 2010, que introduziu práticas como o uso de *media queries*, *layouts* fluidos e imagens flexíveis para garantir uma adaptação harmoniosa.

Design Responsivo vs. Design Adaptativo



Design Responsivo.

- Utiliza um *layout* fluido, onde **os elementos se reorganizam automaticamente de acordo com o tamanho da tela**, utilizando unidades relativas (como porcentagem ou *viewport units*) e *media queries* para ajustar estilos.
- Características:
 - Layouts flexíveis e fluidos.
 - Adaptação contínua e dinâmica, sem pontos de quebra rígidos.
 - Usa *media queries* para ajustar o *layout* em diferentes larguras de tela.

Design Adaptativo.

- Utiliza *layouts* fixos em diferentes pontos de interrupção (*breakpoints*).
- Em vez de mudar continuamente, ele "salta" de um *layout* para outro, dependendo do dispositivo ou da resolução.
- Características:
 - Utiliza pontos de quebra específicos para telas de diferentes tamanhos.
 - Permite maior controle sobre cada *layout* individual.
 - Não é fluido, ou seja, a adaptação ocorre de maneira abrupta.

Flexbox: flexibilidade e organização.



Definição.

- O Flexbox (*Flexible Box Layout*) é uma tecnologia usada para criar layouts flexíveis e alinhados de forma simples e eficiente.
- Ele foi desenvolvido para facilitar o alinhamento de elementos em um eixo principal e um eixo secundário, permitindo que os itens se adaptem ao espaço disponível.
- Isso é especialmente útil quando temos elementos que precisam se reorganizar em diferentes tamanhos de tela.

O que é um Container Flexível?

- O Flexbox funciona em um elemento pai (container) que possui diversos elementos filhos (itens).
- Ao aplicar o estilo **display: flex;** no contêiner, todos os itens internos passam a se comportar de maneira flexível.

Eixos do Flexbox.

- **Eixo Principal (*Main Axis*):** por padrão, o eixo principal é horizontal (da esquerda para a direita). Isso pode ser alterado com a propriedade **flex-direction**.
- **Eixo Secundário (*Cross Axis*):** é perpendicular ao eixo principal, geralmente de cima para baixo.



Propriedades essenciais do Container.

flex-direction: define a direção dos itens.

- **row** (padrão): alinha os itens na horizontal.
- **column**: alinha os itens na vertical.
- **row-reverse** e **column-reverse**: invertem a direção.

justify-content: alinha os itens no eixo principal.

- **flex-start**: alinha à esquerda.
- **center**: alinha ao centro.
- **flex-end**: alinha à direita.
- **space-between**: espaçamento uniforme entre os itens.
- **space-around**: espaçamento ao redor de cada item.

align-items: Alinha os itens no eixo secundário.

- **stretch** (padrão): itens esticados para preencher o contêiner.
- **center**: alinha os itens ao centro verticalmente.
- **flex-start** e **flex-end**: alinha no início ou no final.

Propriedades dos Itens Flexíveis.

- **flex-grow**: controla quanto um item pode crescer em relação aos outros.
- **flex-shrink**: define quanto um item pode encolher.
- **flex-basis**: define o tamanho inicial do item.

Grid Layout: organização bidimensional.



Definição.

- Enquanto o Flexbox funciona bem para alinhamentos lineares (um eixo), o *Grid Layout* é ideal para criar *layouts* complexos com várias linhas e colunas.
- Ele é perfeito para montar galerias, *dashboards* e qualquer estrutura onde os elementos precisem se organizar tanto na horizontal quanto na vertical.



Estrutura básica do Grid.

- O Grid é aplicado no elemento pai (container) com a propriedade **display: grid;**. A partir daí, você define as linhas e colunas para distribuir os elementos filhos (itens).
- Propriedades principais do Grid:
 - **grid-template-columns**: define quantas colunas o grid terá.
 - **grid-template-rows**: define quantas linhas o grid terá.
 - **gap**: espaçamento entre as células do grid.
- Ajustando a posição dos itens:
 - **grid-column**: controla em quais colunas o item ocupará espaço.
 - **grid-row**: controla as linhas ocupadas.

Mobile-First e Media Queries.



Definição.

- A abordagem *Mobile-First* significa desenvolver o site primeiramente para dispositivos móveis e, depois, adaptá-lo para telas maiores.
- Isso garante que o *layout* seja leve e eficiente para *smartphones* e *tablets* antes de ser expandido para *desktops*.
- Isso evita que o *design* se torne pesado ou carregado em dispositivos móveis, onde o processamento e a velocidade de internet são menores.
- As *media queries* são os blocos de CSS que aplicam estilos específicos com base em características da tela, como largura e altura, auxiliando na implementação do conceito de *mobile-first*.

Por que usar *Mobile-First*?

- Desempenho otimizado: *sites mobile-first* são mais rápidos e eficientes.
- Manutenção simples: código CSS mais enxuto e de fácil manutenção.
- SEO melhorado: o Google prioriza sites responsivos nos resultados de busca.
- Experiência do Usuário: garantia de uma boa experiência desde o começo.

Vamos para a prática?



Dúvidas?

jessica.oliveira@p.ucb.br