

# Programação Front End – AULA 10

Matheus Moresco

Engenharia de Software – 3 Período

2025/01

# Introdução

- Media Queries
- Design Responsivo

# O que é Design Responsivo?

- O **design responsivo** é uma abordagem do desenvolvimento web que garante que um site ou aplicação se adapte automaticamente a diferentes tamanhos de tela e dispositivos, proporcionando uma experiência de usuário otimizada.



# Design Responsivo

## Principais características:

- Layouts **flexíveis** que se ajustam dinamicamente.
- Uso de **Media Queries** para adaptar estilos.
- Imagens e elementos que **redimensionam proporcionalmente**.
- Priorização da **usabilidade e acessibilidade**.

# Design Responsivo

## Benefícios:

- ✓ Melhor experiência para usuários em dispositivos móveis e desktops.
- ✓ Maior tempo de permanência no site e menor taxa de rejeição.
- ✓ SEO aprimorado (Google prioriza sites responsivos nos resultados de busca).

# O que são Media Queries?

- As **Media Queries** são um recurso do CSS que permite aplicar estilos diferentes com base nas características do dispositivo, como **largura da tela, altura, orientação, resolução**, entre outros. Elas são fundamentais para o **design responsivo**.
- **Como funcionam?**
  - Permitem definir estilos específicos para diferentes tamanhos de tela.
  - Usam **breakpoints** para determinar quando as regras devem ser aplicadas.
  - Podem ser usadas para adaptar **layout, fontes, cores e imagens**.

# Como funcionam as Media Queries?

- Uma Media Query segue esta estrutura básica:

```
1  @media tipo-de-mídia and (condição) {  
2      /* Regras CSS aplicadas quando a condição for verdadeira */  
3      }  
4
```

# Como funcionam as Media Queries?

- O layout começa com um estilo padrão para dispositivos móveis.
- Para telas maiores que 768px, o fundo muda para cinza e o tamanho da fonte aumenta.
- Para telas maiores que 1024px, o fundo fica azul e o texto maior ainda.

```
1
2  body {
3    background-color: white;
4    font-size: 16px;
5  }
6
7  /* Mudança para telas maiores que 768px (Tablets e Desktops) */
8  @media screen and (min-width: 768px) {
9    body {
10      background-color: lightgray;
11      font-size: 18px;
12    }
13  }
14
15  /* Mudança para telas maiores que 1024px (Desktops) */
16  @media screen and (min-width: 1024px) {
17    body {
18      background-color: lightblue;
19      font-size: 20px;
20    }
21  }
22
```



# Tipos de Media Queries

Os Medias Queries podem se basear em várias informações para gerar uma condição personalizada. Os principais tipos de Media Queries são baseadas em:

- Largura da tela
- Orientação da tela
- Tipo de dispositivo

## Baseadas no tipo de dispositivo

- Podemos aplicar estilos específicos para telas, impressoras e outros dispositivos.  
Exemplo de estilo para impressão:

```
1
2  @media screen { /* Aplica apenas em telas */ }
3  @media print { /* Aplica apenas ao imprimir */ }
4
5
```

# Baseadas em largura da tela

- Definem regras conforme a largura da tela do dispositivo.
  - **max-width**: Aplica estilos quando a largura da tela for menor ou igual ao valor.
  - **min-width**: Aplica estilos quando a largura da tela for maior ou igual ao valor.

```
1
2  @media screen and (max-width: 600px) {
3      /* Para telas pequenas */
4  }
5
6  @media screen and (min-width: 1024px) {
7      /* Para telas grandes */
8  }
9
```

## Baseadas na orientação da tela

- Definem regras conforme o dispositivo está no modo retrato (**portrait**) ou paisagem (**landscape**).

```
1
2  @media screen and (orientation: portrait) {
3      body {
4          background-color: pink;
5      }
6  }
7
8  @media screen and (orientation: landscape) {
9      body {
10         background-color: lightblue;
11     }
12 }
13
```

# Mobile-First vs. Desktop-First

- As Media Queries podem ser usadas com duas abordagens principais:

Mobile-First (Recomendado)	Desktop-First
Começa com o layout para móveis e expande para telas maiores	Começa com o layout para desktops e reduz para telas menores
Usa min-width nas Media Queries	Usa max-width nas Media Queries
Melhor performance em dispositivos móveis	Pode resultar em mais ajustes para telas pequenas

# Mobile-First

- Começa com estilos para dispositivos **móveis** e expande para telas maiores.

```
1  
2  body {  
3      font-size: 16px; /* Estilo padrão para celulares */  
4  }  
5  
6  @media screen and (min-width: 768px) { /* Tablets e desktops */  
7      body {  
8          font-size: 18px;  
9      }  
10 }  
11
```

# Desktop-First

- Começa com estilos para **desktops** e adapta para telas menores.

```
1  
2  body {  
3      font-size: 20px; /* Estilo padrão para desktops */  
4  }  
5  
6  @media screen and (max-width: 768px) { /* Ajuste para telas menores */  
7      body {  
8          font-size: 16px;  
9      }  
10 }
```

# Mobile-First vs. Desktop-First

- O **Mobile-First** é mais recomendado, pois garante melhor desempenho em dispositivos móveis.
- O **Desktop-First** pode ser útil em projetos onde a maioria dos usuários acessa pelo computador.



# Boas práticas ao usar Media Queries

- **Use breakpoints estratégicos:** Defina **poucos pontos de quebra** baseados no conteúdo, não em dispositivos específicos.
- **Comece com Mobile-First:** Aplique estilos primeiro para telas menores e expanda conforme necessário.
- **Combine Media Queries com técnicas flexíveis:** Use flexbox, grid layout e unidades relativas (% , em, rem, vw, vh) para melhorar a responsividade.
- **Teste em diferentes dispositivos:** Utilize ferramentas como:
  - Chrome DevTools (F12 → aba "Toggle Device Toolbar")
  - Responsively App (testa em vários tamanhos simultaneamente)

# Conclusão

- **Media Queries** são essenciais para um **design responsivo eficiente**.
- A abordagem **Mobile-First** é a mais recomendada.
- Sempre teste em diferentes dispositivos e use breakpoints estratégicos.

## Exemplo prático

- Na página do Blog, aplicar Media Queries para esconder o menu lateral em telas menores.

## Exercício prático

- Use media Queries para ajustar o estilos dos elementos de acordo com o tamanho de tela na pagina dos sites favoritos.