

# Layouts responsivos e design adaptativo

# Layouts Responsivos e Design Adaptativo

- Objetivo da Aula:
- Entender a diferença entre layouts responsivos e design adaptativo.
- Aprender técnicas para criar interfaces que funcionem em diferentes dispositivos.

## Introdução

- Por que designs responsivos são importantes?
- Acesso à internet por dispositivos móveis supera desktops.
- Necessidade de uma experiência consistente em diferentes telas.
- SEO (Google prioriza sites mobile-friendly).

# Layout Responsivo

- Definição:
- Adapta-se fluidamente ao tamanho da tela usando CSS (media queries, grids flexíveis).
- Características:
- Único código para todas as telas.
- Redimensionamento contínuo.

## Design Adaptativo (Adaptive Design)

- Definição:
- Versões específicas para diferentes tamanhos de tela (breakpoints fixos).
- Características:
- Vários layouts pré-definidos.
- Melhor controle para cada dispositivo.

# Diferenças Chave

<u><i>Layout Responsivo</i></u>	<u><i>Design Adaptativo</i></u>
Fluido e contínuo	Layouts fixos por breakpoint
Único código	Múltiplos códigos/detecção de dispositivo
Menos controle em telas específicas	Maior otimização por dispositivo

# Técnicas para Layouts Responsivos

- Viewport Meta Tag:

html

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Media Queries (CSS):

- css

```
@media (max-width: 768px) { ... }
```

- Unidades Relativas (%, rem, vw/vh)
- Grids Flexíveis (Flexbox, CSS Grid)

Breakpoints comuns:

Mobile: < 768px

Tablet: 768px – 1024px

Desktop: > 1024px

- Exemplo de implementação:

Css:

```
/* Mobile */
```

```
@media (max-width: 767px) { ... }
```

```
/* Tablet */
```

```
@media (min-width: 768px) and (max-width: 1023px)  
{ ... }
```

## Ferramentas Úteis

- Frameworks CSS: Bootstrap, Tailwind, Foundation.
- Testes: Chrome DevTools (Modo Responsivo), BrowserStack.
- Design: Figma (Auto Layout), Adobe XD (Responsive Resize).

## Boas Práticas

- Mobile-First: Comece pelo mobile e depois adapte para telas maiores.
- Otimização de Imagens: Use srcset para carregar imagens adequadas.
- Performance: Evite excesso de media queries e JavaScript desnecessário.



## Exemplo Prático

Código de um Layout Responsivo Simples:

```
html<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <style>
    .container { width: 100%; max-width: 1200px; margin: 0 auto; }
    @media (max-width: 600px) { .container { padding: 10px; } }
  </style>
</head>
<body>
  <div class="container">Conteúdo responsivo!</div>
</body>
</html>
```

Resumo:

Responsivo: Fluido, único código.

Adaptativo: Vários layouts, mais controle.

Use media queries, grids e mobile-first para melhores resultados.