

CSS

RESPONSIVIDADE PARA CELULAR

Desktop First, medidas flexíveis, imagens responsivas e media queries



Fundação Arco-Íris de Araputanga
FCARP
FACULDADE CATÓLICA
RAINHA DA PAZ

Professor Thiago Dias Negrão

Aula 04-2 2026

```
@media (max-width: 768px) {  
  .pagina {  
    flex-direction:  
    column;  
  }  
}
```

Mobile



```
Style.css  
@media (max-width: 575.98px) {  
  ...  
}
```

OBJETIVOS DA AULA

- Entender o conceito de layout responsivo
- Aplicar a estratégia Desktop First
- Utilizar medidas flexíveis e limites de largura
- Criar media queries para telas menores
- Adaptar menu, conteúdo e cartões para celular

Resultado esperado: o mesmo projeto funcionando bem no computador e no celular.

REVISÃO DO PROJETO DESKTOP

- Container centralizado com max-width
- Cabeçalho organizado com Flexbox
- Área principal dividida em duas colunas
- Cartões posicionados na mesma linha
- Espaçamentos definidos com margin, padding e gap

Agora vamos reorganizar o layout quando a largura da tela diminuir.

LAYOUT FIXO E RESPONSIVO

Fixo

- Depende de larguras rígidas
- Pode criar rolagem horizontal
- Textos e imagens podem estourar
- Exige zoom no celular

Responsivo

- Aproveita a largura disponível
- Reorganiza os elementos
- Mantém o conteúdo legível
- Adapta-se a diferentes telas

ESTRATÉGIA DESKTOP FIRST

```
/* Estilo padrão para computador */  
.servicos {  
  display: flex;  
  gap: 24px;  
}
```

```
/* Alteração para telas menores */  
@media (max-width: 768px) {  
  .servicos {  
    flex-direction: column;  
  }  
}
```

- Primeiro construímos o desktop
- Depois definimos as exceções
- max-width limita a aplicação
- O CSS dentro do media substitui regras

VIEWPORT

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

- width=device-width usa a largura real
- initial-scale define o zoom inicial
- A tag fica dentro do head
- Sem viewport, o celular simula uma tela larga

MEDIDAS FIXAS E RELATIVAS

Fixas

- px mantém um tamanho definido
- Útil para bordas e detalhes
- Pode limitar a adaptação
- Exige cuidado em larguras

Relativas

- % acompanha o elemento pai
- rem acompanha a fonte raiz
- vw acompanha a largura da tela
- Permite maior flexibilidade

PORCENTAGEM E MAX-WIDTH

```
.container {  
  width: 90%;  
  max-width: 1200px;  
  margin: 0 auto;  
}
```

```
.imagem {  
  width: 100%;  
  max-width: 560px;  
}
```

- 90% cria margem lateral
- max-width impede crescimento excessivo
- margin auto centraliza
- A combinação funciona em várias telas

UNIDADES REM, VW E VH

```
html { font-size: 16px; }  
  
h1 { font-size: 3rem; }  
  
.hero { min-height: 70vh; }  
  
.banner { width: 100vw; }
```

- 1 rem usa o tamanho da fonte raiz
- vw representa a largura da tela
- vh representa a altura da tela
- Use com moderação e teste

Teste cada alteração e observe o resultado no navegador.

IMAGENS RESPONSIVAS

```
img {  
  max-width: 100%;  
  height: auto;  
  display: block;  
}
```

```
.hero-imagem {  
  width: 50%;  
}
```

- max-width evita estouro
- height auto preserva a proporção
- display block remove espaço indesejado
- A largura específica pode mudar no media

MEDIA QUERIES

```
@media (max-width: 768px) {  
  body {  
    font-size: 16px;  
  }  
  
  .container {  
    width: 92%;  
  }  
}
```

- A condição observa a largura da tela
- As regras valem até 768px
- Coloque o media no final do CSS
- Evite repetir regras que não mudam

PONTOS DE QUEBRA

- 1200px ou mais: monitores amplos
- Entre 769px e 1199px: notebooks e tablets horizontais
- Até 768px: tablets verticais e celulares maiores
- Até 480px: celulares menores
- Escolha o ponto em que o layout começa a falhar

O conteúdo define o ponto de quebra. Os números servem como referência, não como regra absoluta.

FLEXBOX NO CELULAR

```
@media (max-width: 768px) {  
  .hero,  
  .servicos {  
    flex-direction: column;  
  }  
  
  .hero > div {  
    width: 100%;  
  }  
}
```

- Os elementos passam para uma coluna
- Cada bloco ocupa a largura disponível
- O HTML permanece igual
- A ordem segue a estrutura do documento

MENU PARA CELULAR

```
@media (max-width: 768px) {  
  header .container {  
    flex-direction: column;  
    padding: 20px 0;  
  }  
  
  nav {  
    flex-wrap: wrap;  
    justify-content: center;  
  }  
}
```

- Logo e menu ficam empilhados
- flex-wrap permite quebra de linha
- Links ficam centralizados
- Mantenha área confortável para toque

CARTÕES EM COLUNA

```
@media (max-width: 768px) {  
  .servicos {  
    display: flex;  
    flex-direction: column;  
  }  
  
  .cartao {  
    width: 100%;  
  }  
}
```

- Um cartão por linha
- Largura total do container
- Leitura vertical natural
- Espaçamento continua com gap

FONTES RESPONSIVAS

```
h1 {  
  font-size: 48px;  
}  
  
@media (max-width: 768px) {  
  h1 {  
    font-size: 34px;  
    line-height: 1.2;  
  }  
}
```

- Títulos grandes podem quebrar
- Reduza sem prejudicar a hierarquia
- Ajuste a altura da linha
- Evite textos pequenos demais

ESPAÇAMENTO NO CELULAR

```
section {  
  padding: 80px 0;  
}
```

```
@media (max-width: 768px) {  
  section { padding: 48px 0; }  
  .hero { gap: 28px; }  
  .cartao { padding: 20px; }  
}
```

- Reduza espaços muito amplos
- Mantenha separação entre conteúdos
- Preserve áreas de toque
- Evite elementos encostados nas bordas

TESTE NO NAVEGADOR

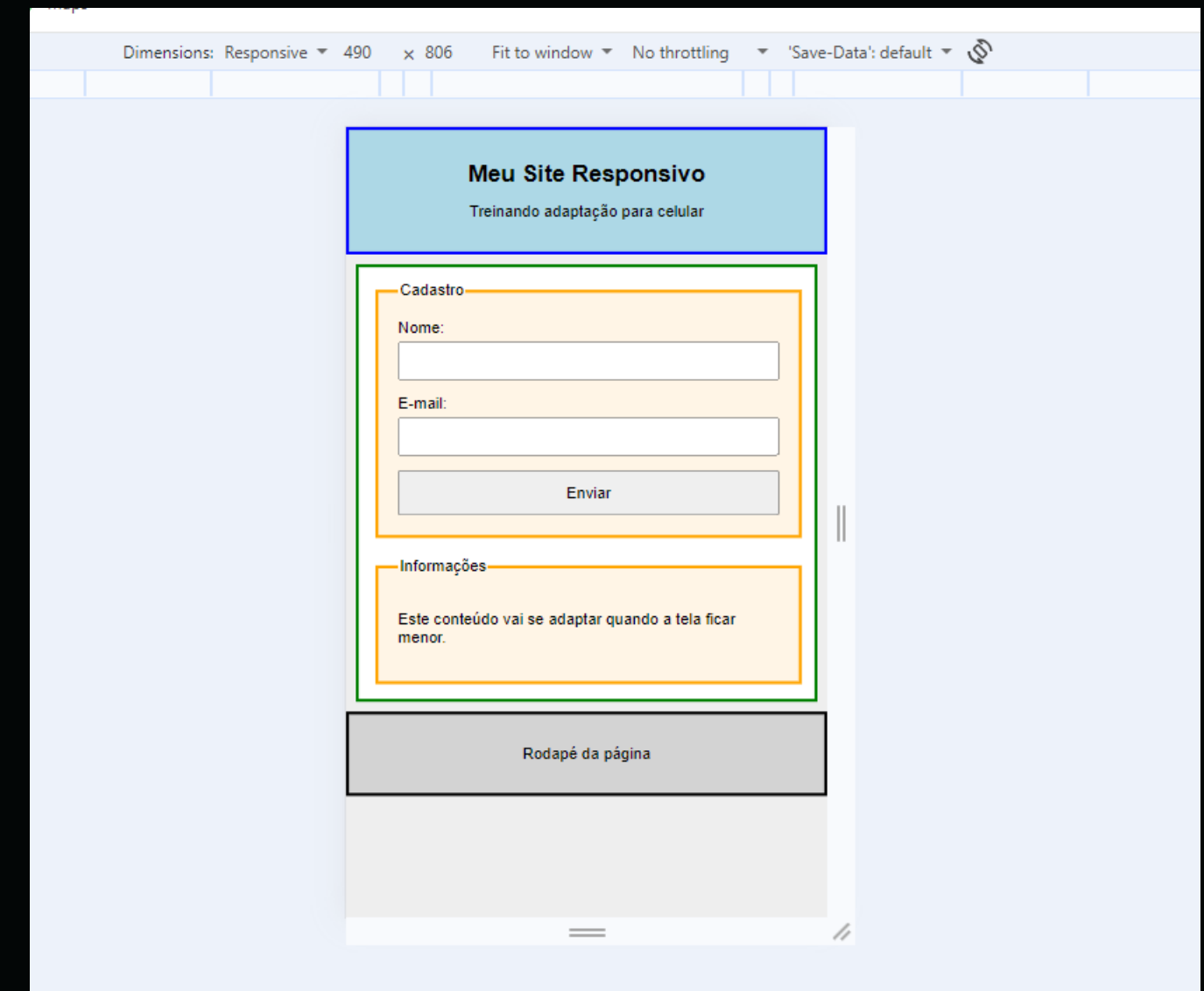
- Abra as ferramentas do desenvolvedor com F12
- Ative o modo de dispositivo móvel
- Teste diferentes larguras de tela
- Confira rolagem horizontal, textos, imagens e menu
- Faça também um teste no celular real

Teste o layout em várias larguras. Não confie apenas em um modelo de aparelho.

ADAPTAÇÃO DO PROJETO

ATIVIDADE PRÁTICA

- Confirme a presença da tag viewport
- Mantenha o desktop como estilo padrão
- Crie um media query para até 768px
- Empilhe hero e cartões com flex-direction
- Ajuste menu, fontes, imagens e espaçamentos



REVISÃO DA AULA

RESPONDA SEM CONSULTAR

- O que significa responsividade?
- Para que serve a tag viewport?
- Qual é a diferença entre px e %?
- O que uma media query faz?
- Como transformar uma linha Flexbox em coluna?

Próxima aula: aprofundamento em CSS e construção de interfaces mais completas.