

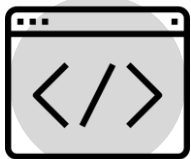
# Profissão: Engenheiro Front-End



# BOAS PRÁTICAS



# Projeto 4



Texto explicando o que o  
estudante verá neste  
documento

- **Crie o Hero**
- **Importe uma fonte externa**
- **Crie a lista de planos**
- **Crie a seção “Dispositivos disponíveis”**
- **Crie o cabeçalho**
- **Realize o Deploy**



# Crie o Hero

- **Imagemin:**  
Use a biblioteca Imagemin para melhorar o desempenho do carregamento de páginas da web, reduzir o consumo de largura de banda e economizar espaço de armazenamento. Entretanto, use-a com cuidado para evitar perdas de qualidade excessivas nas imagens. É recomendado testar as configurações e garantir que a qualidade visual das imagens otimizadas seja aceitável para o uso pretendido.
- **Hero:**  
Ao projetar e implementar um hero (banner) em um site, existem algumas boas práticas que você pode seguir para garantir uma experiência positiva para os usuários veja a seguir.



# Crie o Hero



## **Clareza e simplicidade:**

O hero deve transmitir uma mensagem clara e concisa. Evite sobrecarregar a área com excesso de texto ou elementos visuais complexos. Mantenha a mensagem simples e direta para que os usuários possam entender rapidamente o propósito do hero.



## **Chamar à ação:**

Se apropriado para o contexto, inclua uma chamada à ação (CTA) no hero. Uma CTA bem projetada e posicionada pode incentivar os usuários a realizar a ação desejada, como clicar em um botão, inscrever-se em um serviço ou explorar mais o site.



# Crie o Hero



## **Imagem impactante:**

Use uma imagem de alta qualidade e impactante no hero para chamar a atenção dos usuários. Certifique-se de que a imagem esteja relacionada ao conteúdo e ao propósito do site. Considere o uso de imagens que despertem emoções positivas e estejam alinhadas com a identidade visual da marca.



## **Responsividade:**

Certifique-se de que o hero seja responsivo e se ajuste bem em diferentes tamanhos de tela e dispositivos. Teste o design em dispositivos móveis, tablets e desktops para garantir que a experiência seja consistente e agradável em todas as plataformas.

# Crie o Hero



## **Velocidade de carregamento:**

O tempo de carregamento do hero é crucial para uma boa experiência do usuário. Otimize as imagens e o código para garantir que o hero seja carregado rapidamente, evitando assim a frustração dos usuários devido a um carregamento lento.



## **Contraste adequado:**

Certifique-se de que o texto e os elementos do hero sejam legíveis e destacados da imagem de fundo. Use cores contrastantes e escolha uma fonte legível para garantir uma boa legibilidade em todos os dispositivos.



# Crie o Hero

- **Coerência com o restante do site:**  
O hero deve se integrar harmoniosamente ao design geral do site. Mantenha a coerência visual com o estilo, paleta de cores e tipografia utilizados nas outras partes do site para criar uma experiência coesa e profissional.
- **Teste e iteração:**  
Realize testes com usuários e analise o desempenho do hero para identificar áreas de melhoria. Esteja disposto a iterar e ajustar o design do hero com base nos feedbacks recebidos e nas métricas de desempenho.





# Crie o Hero



## **last-child:**

Essa pseudo-classe seleciona apenas o último elemento dentro de seu contêiner pai, independentemente de seu tipo ou classe. Se você precisar selecionar o último elemento de um tipo específico, independentemente de sua posição, você pode usar a pseudo-classe :last-of-type.

# Importe uma fonte externa

## • @font-face {:

Acompanhe os principais componentes da regra @font-face:

- font-family: Especifica o nome da família de fontes personalizada que será usada para referenciar a fonte no restante do CSS.
- src: Define o caminho para a fonte personalizada. Pode ser uma URL para um arquivo de fonte externo (por exemplo, um arquivo .woff2) ou uma declaração local() que faz referência a uma fonte instalada no computador do usuário.
- format: Especifica o formato da fonte. Existem vários formatos de fonte comumente usados, como WOFF, WOFF2, TTF e EOT. É importante fornecer diferentes formatos de fonte para garantir a compatibilidade em diferentes navegadores.



# Crie a lista de planos

## • **`nth-child()`**

Acompanhe algumas dicas sobre o uso dessa pseudo-classe:

- Entenda a contagem de elementos: A pseudo-classe `:nth-child()` seleciona elementos com base em sua posição em relação aos seus irmãos. Lembre-se de que a contagem começa em 1, não em 0. Portanto, o primeiro elemento é representado por `:nth-child(1)`, o segundo por `:nth-child(2)`, e assim por diante.
- Experimente fórmulas matemáticas: A fórmula  $an+b$  permite que você selecione elementos em uma sequência específica. Você pode usar diferentes valores para  $a$  e  $b$  para ajustar a seleção de elementos desejada. Por exemplo, `:nth-child(2n+1)` seleciona todos os elementos ímpares, enquanto `:nth-child(3n+2)` seleciona elementos que seguem a sequência 3, 6, 9, etc.



# Crie a lista de planos



## **Combine com outras pseudo-classes:**

Você pode combinar a pseudo-classe `:nth-child()` com outras pseudo-classes para refinar ainda mais a seleção de elementos. Por exemplo, `:nth-child(odd)` seleciona todos os elementos ímpares, e `:nth-child(even)` seleciona todos os elementos pares.



## **Use com cuidado em layouts dinâmicos:**

Se você estiver usando a pseudo-classe `:nth-child()` em um layout que pode mudar dinamicamente, certifique-se de que a fórmula seja flexível o suficiente para lidar com diferentes cenários. Uma alteração no número de elementos ou na ordem pode afetar a seleção.



# Crie a lista de planos



## **Combine com seletor de tipo ou classe:**

Para limitar a seleção a um determinado tipo de elemento ou classe, você pode combinar a pseudo-classe `:nth-child()` com um seletor de tipo ou classe. Por exemplo, `ul li:nth-child(3n+1)` seleciona os primeiros elementos de cada grupo de três elementos dentro de uma lista não ordenada.



## **Teste em vários cenários:**

Ao usar a pseudo-classe `:nth-child()`, teste seu código em vários cenários para garantir que os elementos estejam sendo selecionados corretamente. Verifique se os estilos aplicados estão de acordo com suas expectativas e se adaptam corretamente às alterações no conteúdo ou no layout.



# Crie a seção "Dispositivos disponíveis"

## • **Fração (fr):**

Você pode combinar frações com outras unidades de medida, como pixels ou porcentagens, para criar layouts mais complexos.

Por exemplo:

```
.container {
  display: grid;
  grid-template-columns: 1fr 200px 30%;
}
```

Nesse exemplo, a primeira coluna ocupa 1 fração do espaço disponível, a segunda coluna tem uma largura fixa de 200 pixels e a terceira coluna ocupa 30% do espaço disponível.



# Crie a seção "Dispositivos disponíveis"



- Acompanhe algumas dicas sobre o uso da unidade fr no sistema de grid:
  - Pense em proporções: As frações (fr) permitem definir proporções entre as colunas ou linhas do grid. Considere cuidadosamente como você deseja que o espaço seja distribuído entre os itens. Por exemplo, 1fr 2fr significa que a segunda coluna terá o dobro do tamanho da primeira.
  - Use frações de forma responsiva: As frações se adaptam automaticamente ao tamanho do contêiner. Isso é particularmente útil em layouts responsivos, onde você deseja que as colunas ou linhas se ajustem conforme a largura da tela muda. Experimente diferentes combinações de frações para criar layouts flexíveis.

# Crie a seção "Dispositivos disponíveis"



## **Use frações para preenchimento flexível:**

Se você quiser que uma coluna ou linha preencha o espaço restante após as outras colunas ou linhas terem ocupado seu tamanho, você pode usar uma fração com o valor auto. Por exemplo, 1fr auto significa que a primeira coluna ocupará uma fração do espaço disponível e a segunda coluna preencherá o espaço restante.



## **Tenha cuidado com o uso excessivo de frações:**

Embora as frações sejam uma ferramenta poderosa para criar layouts flexíveis, é importante usá-las com moderação. O uso excessivo de frações pode levar a layouts complexos e difíceis de manter. Avalie se o uso de frações é realmente necessário em cada caso específico.





# Crie a seção "Dispositivos disponíveis"

- **Teste em diferentes tamanhos de tela:**

Ao usar frações no sistema de grid, teste seu layout em diferentes tamanhos de tela para garantir que as proporções e o comportamento responsivo estejam funcionando conforme o esperado. Verifique se as colunas ou linhas se ajustam corretamente e se o layout permanece legível e esteticamente agradável em diferentes dispositivos.



# Crie o cabeçalho



## **Darken:**

Ao contrário da função lighten, essa função permite ajustar a escuridão de uma cor. Essa função é usada para tornar uma cor mais escura ou "escurecer" uma cor existente.



# Realize o Deploy

## • **Nowrap:**

Acompanhe algumas dicas para usar a propriedade white-space: nowrap de forma eficaz:

- Limitar o uso: Aplique white-space: nowrap somente quando necessário, como em títulos, links ou outros casos em que é importante manter o conteúdo em uma única linha. Evite usá-lo em grandes blocos de texto, pois pode dificultar a leitura e a compreensão.
- Verificar a legibilidade: Certifique-se de que o conteúdo em white-space: nowrap seja legível e não fique cortado ou truncado. Verifique se há espaço suficiente no elemento para a exibição do texto sem quebras de linha automáticas.



# Realize o Deploy

- Considerar a responsividade:**  
 Em layouts responsivos, verifique como o conteúdo em white-space: nowrap se comporta em dispositivos de tela menor. Em alguns casos, pode ser necessário aplicar um tratamento especial para garantir que o texto seja exibido adequadamente em telas menores.
- Ajustar largura do elemento:**  
 Certifique-se de que o elemento que contém o texto em white-space: nowrap tenha uma largura adequada para a exibição completa do conteúdo. Se o elemento for muito estreito, o texto pode ficar oculto ou fora de proporção.



# Realize o Deploy

- **Verificar a interação:**  
Considere a interação do usuário com o conteúdo em white-space: nowrap. Verifique se há necessidade de implementar algum mecanismo de rolagem horizontal ou outras formas de exibir o conteúdo que excede a largura do elemento.
- **Testar em diferentes navegadores:**  
Verifique a compatibilidade do white-space: nowrap em diferentes navegadores para garantir que o comportamento seja consistente e esperado em todos eles.



# Bons estudos!

