

Documento Teórico - Desenvolvimento Web

1. Estruturação do HTML e Justificativa do Uso de Tags Semânticas

Quando começamos a montar uma página web, o HTML é como o esqueleto do site - é ele que dá estrutura e forma ao que vai aparecer na tela. Mas mais do que simplesmente posicionar os elementos, é importante usar as tags certas para que o conteúdo faça sentido não só visualmente, mas também para leitores de tela, buscadores como o Google e até outros desenvolvedores que forem dar manutenção no código depois.

Por isso, a gente usa o que chamamos de tags semânticas. A `<header>`, por exemplo, indica que aquele trecho é o cabeçalho da página. Dentro dela, é comum colocar o menu de navegação, com a tag `<nav>`, e o título do site. Já o `<main>` serve para destacar o conteúdo principal da página, enquanto `<section>` ajuda a dividir esse conteúdo em blocos organizados. O `<footer>`, como o nome sugere, é o rodapé, onde colocamos informações como formulários, redes sociais ou links complementares.

Tudo isso ajuda muito quem está acessando o site com tecnologias assistivas, porque essas ferramentas conseguem entender melhor o que é cada parte da página. E como bônus, os motores de busca também conseguem indexar melhor o conteúdo, o que ajuda o site a aparecer mais nas pesquisas.

2. Organização do CSS

Depois de estruturar o HTML, vem a parte de deixar tudo bonito com o CSS. E aqui a ideia foi manter tudo o mais organizado possível. Criamos um arquivo externo de estilo, chamado `style.css`, onde centralizamos todas as regras de visual. Isso facilita muito a manutenção e evita que o HTML fique poluído com códigos de estilo espalhados.

No CSS, usamos bastante classes reutilizáveis. Por exemplo, criamos uma classe chamada `.botao` que estiliza todos os botões do site do mesmo jeito - assim eles ficam padronizados e com uma aparência profissional. Também usamos nomes de classe descritivos como `.navbar` para o menu de navegação e `.card` para os blocos de conteúdo, o que deixa tudo mais intuitivo.

Além disso, cuidamos para que as imagens sejam responsivas. Criamos uma classe chamada `.img-fluid` que garante que elas se ajustem ao tamanho da tela do dispositivo, evitando aquele problema de imagem estourada ou cortada em telas menores. No geral, a ideia foi deixar o CSS limpo, modular e fácil de mexer depois.

Documento Teórico - Desenvolvimento Web

3. Responsividade e Quebras de Layout (Breakpoints)

Hoje em dia, é impossível saber de onde alguém vai acessar um site. Pode ser de um celular pequeno, de um tablet, de um notebook ou de uma tela grande. Então, para garantir que o site fique bom em todos esses cenários, a gente precisa torná-lo responsivo. Isso significa que o layout do site se adapta automaticamente ao tamanho da tela.

A maneira de fazer isso é usando media queries no CSS. Elas funcionam como 'condições' que dizem: 'se a tela tiver tal tamanho, aplique esse estilo'. Por exemplo, se a tela tiver menos de 768 pixels de largura (como acontece na maioria dos smartphones), a gente pode reorganizar o conteúdo em uma única coluna, aumentar o espaçamento entre os elementos e deixar o texto maior para facilitar a leitura.

Para telas intermediárias, como tablets, usamos outro breakpoint, geralmente até 1024 pixels. E para telas maiores, liberamos o espaço para mostrar mais colunas, colocar imagens maiores e assim por diante. Com isso, a navegação fica confortável em qualquer dispositivo, o que é essencial para uma boa experiência do usuário.

4. Acessibilidade

A acessibilidade é um aspecto que muitas vezes é deixado de lado, mas faz toda a diferença. O objetivo é garantir que qualquer pessoa, independentemente de limitações físicas ou cognitivas, consiga navegar pelo site.

No nosso projeto, tomamos alguns cuidados importantes. Um deles foi usar o atributo `aria-label` em botões e links, o que dá uma descrição mais precisa do que aquele elemento faz - isso é muito útil para quem usa leitores de tela. Também adicionamos o atributo `alt` em todas as imagens, descrevendo o que elas mostram. Assim, mesmo quem não consegue ver a imagem consegue entender seu conteúdo.

Nos formulários, usamos a tag `<label>` junto com o atributo `for`, que associa corretamente o texto ao campo de entrada. Isso melhora a usabilidade e torna o preenchimento mais intuitivo. Por fim, como toda a estrutura do HTML foi feita com tags semânticas, isso também facilita bastante a navegação por teclado e o uso de tecnologias assistivas.

A ideia é que, mesmo sem enxergar a tela ou sem conseguir usar um mouse, a pessoa consiga navegar tranquilamente pelo site. E isso é algo que todo bom projeto web precisa considerar.