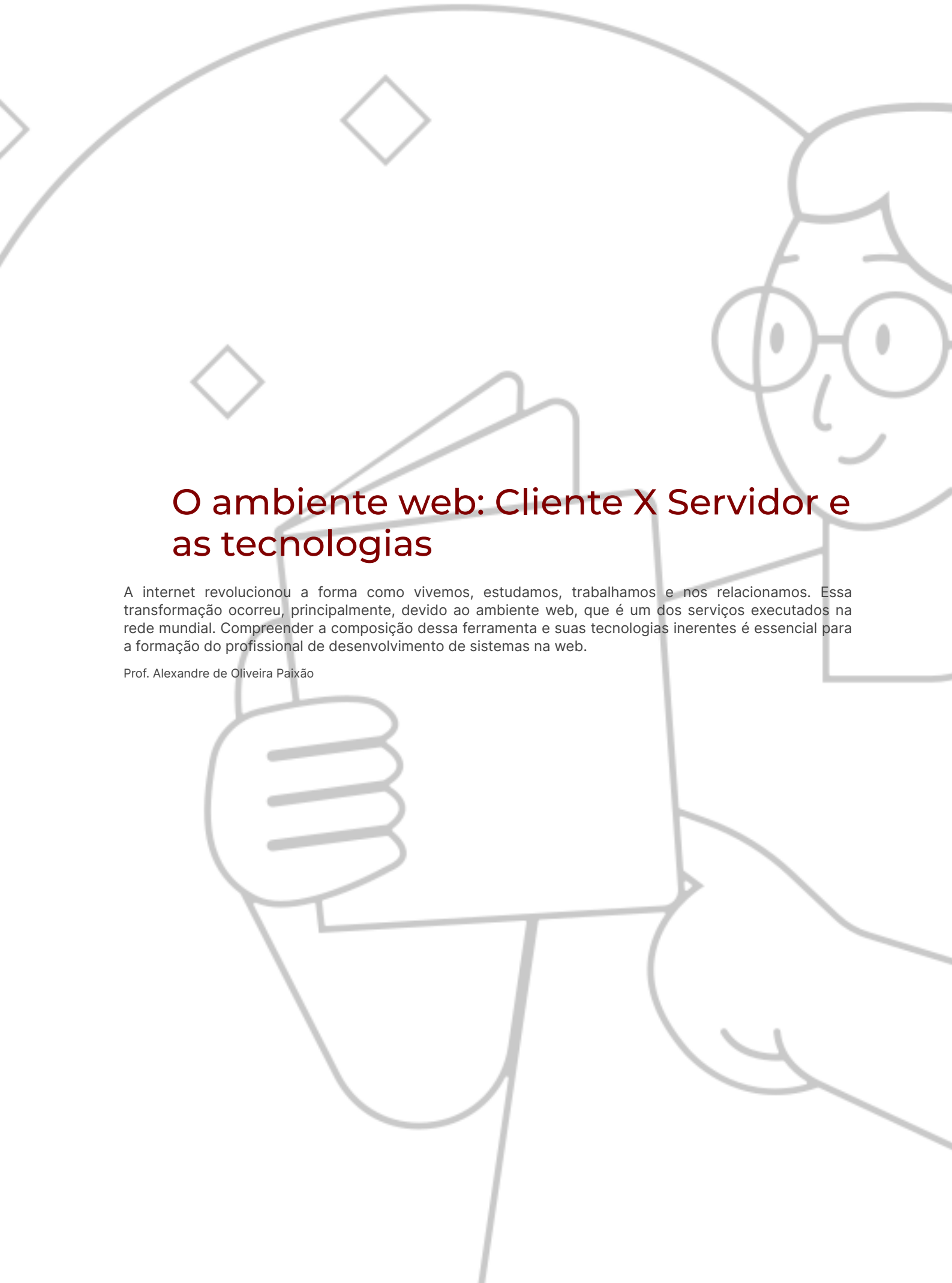


O ambiente web: Cliente X Servidor e as tecnologias

A internet revolucionou a forma como vivemos, estudamos, trabalhamos e nos relacionamos. Essa transformação ocorreu, principalmente, devido ao ambiente web, que é um dos serviços executados na rede mundial. Compreender a composição dessa ferramenta e suas tecnologias inerentes é essencial para a formação do profissional de desenvolvimento de sistemas na web.

Prof. Alexandre de Oliveira Paixão

Objetivos

- Reconhecer o ambiente web.
- Descrever o conceito de interface.
- Reconhecer as tecnologias do lado cliente.
- Reconhecer as tecnologias do lado servidor.

Introdução

Neste vídeo, veremos os principais assuntos que serão abordados ao longo do conteúdo, destacando o conceito de ambiente web, com foco nos modelos da arquitetura Cliente x Servidor e nas tecnologias de ambos os lados dessa arquitetura.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Modelo cliente X servidor

Na arquitetura cliente x servidor, os clientes (ou usuários) são aqueles que solicitam serviços (requisições) executados nas aplicações armazenadas nos servidores, o que envolve o ciclo requisição e resposta. Os clientes podem usar dispositivos como computadores, tablets, smartphones, por exemplo, e são responsáveis por solicitar algum serviço, com processos de início e fim bem definidos. Os servidores, normalmente, são mais robustos e funcionam continuamente e, quando recebem a requisição de algum cliente, processam-na e retornam uma resposta, que é vista pelo usuário. A arquitetura cliente x servidor apresenta vários benefícios, especialmente no modelo de quatro camadas, separando os papéis dos usuários, das aplicações, dos dados e da web.

O vídeo vai abordar a arquitetura cliente x servidor, relacionando-a ao ciclo requisição e resposta e demonstrando os benefícios que essa arquitetura apresenta, especialmente no modelo de quatro camadas.

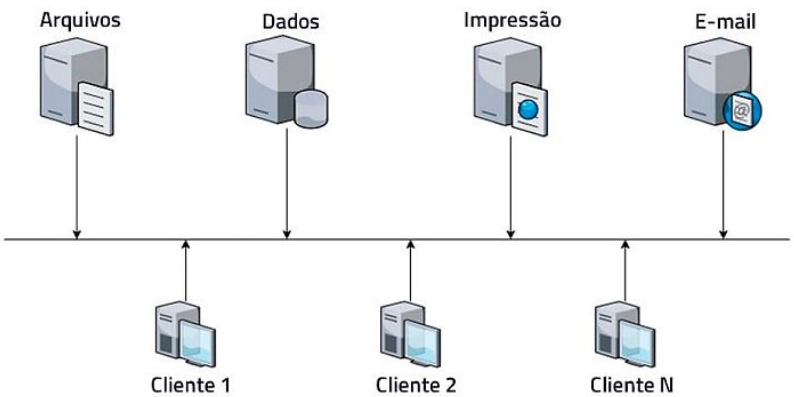
Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

O **modelo cliente X servidor** foi criado pela Xerox PARC nos anos 1970, tendo como principal premissa a separação entre dados e recursos de processamento, ao contrário do modelo predominante à época — conhecido como modelo centralizado, em que tanto o armazenamento dos dados quanto o seu processamento ficavam a cargo dos computadores de grande porte: mainframe.

Ambiente cliente X servidor

O ponto de partida para entendermos a arquitetura do modelo cliente X servidor é tomarmos como exemplo a rede interna de computadores de uma empresa, em que temos máquinas exercendo a função de servidores — provendo serviços como armazenamento de arquivos ou dados, impressão, e-mail etc. — e máquinas exercendo a função de clientes — consumindo os recursos fornecidos pelos servidores. Essa arquitetura pode ser vista na imagem a seguir.



Arquitetura cliente X servidor em uma rede interna.

Aplicações no modelo cliente X servidor

Esse modelo tornou possível o desenvolvimento de aplicações que fizessem uso de sua **arquitetura distribuída**. Tais aplicações foram desenvolvidas tendo como base o conceito de desenvolvimento em camadas. Logo, surgiram os modelos de duas, três e quatro (ou N) camadas.

Modelo de duas camadas

Nesse modelo, temos as camadas cliente e servidor, sendo função da primeira tratar a lógica do negócio e fazer a interface com o usuário, enquanto a segunda é responsável por tratar os dados — normalmente fazendo uso de **sistemas gerenciadores de bancos de dados** (SGDB). São exemplos desse modelo as aplicações desktop instaladas em cada computador cliente que se comunicam com um servidor na mesma rede. A imagem que segue exemplifica esse tipo de rede.

Modelo de duas camadas.

Esse modelo foi criado para resolver alguns problemas do modelo anterior, entre eles a necessidade de reinstalação/atualização da aplicação nos clientes a cada mudança de regra ou lógica. Logo, foi incluída uma camada a mais, a **camada de aplicação**. Com isso, as responsabilidades de cada camada ficaram assim divididas:

Camada de apresentação

Representada pela aplicação instalada na máquina cliente. Era responsável pela interface com o usuário e passou a acessar o servidor de aplicação, perdendo o acesso direto ao servidor de dados.

Camada de aplicação

Representada por um servidor responsável pela lógica e pelas regras de negócio, assim como pelo controle de acesso ao servidor de dados.

Camada de dados

Representada por um servidor responsável pelo armazenamento dos dados.

Veja um exemplo do modelo de três camadas:

Modelo de três camadas.

O grande avanço obtido nesse modelo foi **tirar da máquina cliente a responsabilidade pela interface com o usuário**, passando a centralizá-la em um único ponto, normalmente em um servidor web. Com isso, no lugar de aplicações instaladas em cada máquina cliente, passamos a ter os clientes acessando aplicações hospedadas em servidores web a partir de navegadores. Nesse modelo, um servidor é composto por três servidores — o de aplicações, o de dados e o web. A divisão de responsabilidades ficou desta forma:

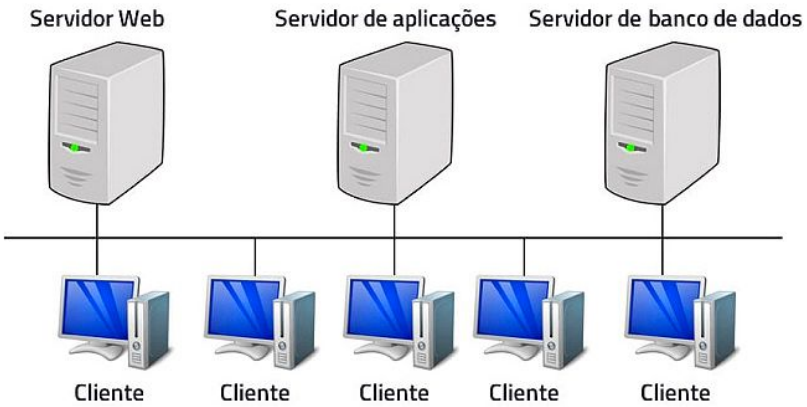
Cliente

Necessita apenas de um navegador para ter acesso à aplicação.

Servidor web

Responsável pela apresentação/interface com o usuário cliente.

Veja um exemplo do modelo de quatro camadas:



Modelo de quatro camadas.

Atividade 1

Considere que você trabalhe no setor de TI de uma grande corporação e esteja participando de uma reunião com profissionais de diferentes áreas. Você precisa argumentar sobre as características da arquitetura cliente x servidor para gestores que não são da área técnica de computação. Entre as opções a seguir, qual delas seria a mais indicada nesse contexto?

A arquitetura cliente x servidor separa os papéis dos clientes (usuários) e das aplicações que são hospedadas em servidores. Os clientes solicitam serviços (requisição) e as aplicações hospedadas nos servidores fornecem serviços (respostas). Normalmente, os clientes são responsáveis por iniciar a comunicação com os servidores.

A

B A arquitetura cliente x servidor separa os papéis dos clientes (usuários) e das aplicações. Os servidores solicitam serviços (requisição) e as aplicações hospedadas nos clientes fornecem serviços (respostas). Normalmente, os servidores são responsáveis por iniciar a comunicação com os clientes.

C A arquitetura cliente x servidor é utilizada para separar os bancos de dados da lógica de negócio de aplicações web. Os clientes representam os bancos de dados enquanto os servidores representam a lógica de negócio da aplicação.

D A arquitetura cliente x servidor é utilizada para separar as aplicações corporativas dos sistemas de e-mail. Os clientes são as aplicações corporativas enquanto os e-mails representam os servidores.

E A arquitetura cliente x servidor foi utilizada nos primórdios da web, mas não é utilizada atualmente por conta do advento da internet das coisas, em que cada dispositivo pode atuar tanto como cliente quanto servidor.

A alternativa A está correta.

De fato, a arquitetura cliente x servidor possibilita a separação dos papéis dos clientes e dos servidores. Os clientes, normalmente, iniciam a comunicação por meio da solicitação de algum serviço aos servidores.

Ambiente WEB

Com o avanço tecnológico da arquitetura cliente x servidor, passamos pelos modelos de duas, três e quatro camadas. Nessa última abordagem, de quatro camadas, não há necessidade de instalação de softwares adicionais nas máquinas dos clientes, ou seja, até mesmo a interface fica por conta do servidor web, que fornece os códigos que serão processados por navegadores nas máquinas dos clientes. Além disso, há outros servidores com tarefas específicas como os servidores de aplicações e de dados. Portanto, o que se conhece como servidor, nesta abordagem, é composto por outros servidores, aumentando a robustez e portabilidade das aplicações.

O vídeo vai explorar a evolução da web até chegar ao modelo de quatro camadas, caracterizando o aumento da robustez e da portabilidade das aplicações nessa abordagem.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Como vimos, inicialmente, as aplicações ficavam hospedadas dentro de uma rede interna, onde estavam os clientes e os servidores. Posteriormente, elas migraram para a internet, surgindo o **ambiente web**, cuja base é justamente prover aos clientes, usuários, o acesso a várias aplicações a partir de diversos dispositivos, como navegadores em desktops e smartphones ou a partir de aplicações mobile.

Comentário

É importante destacar um aspecto quando tratamos do ambiente web: a comunicação.

Até aqui, vimos que esse ambiente é composto pelo:

Cliente

Utiliza um navegador ou aplicativo e consome serviços hospedados em um servidor web.

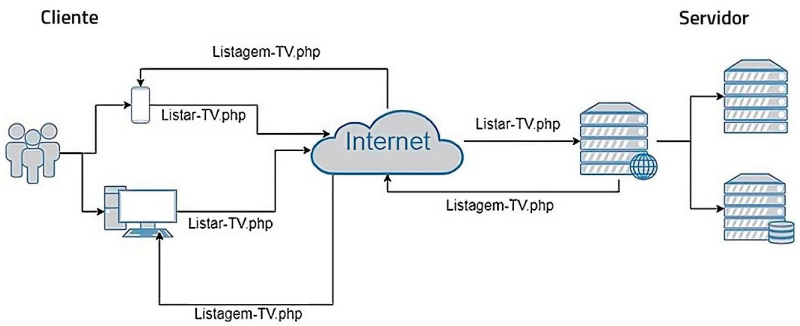
Servidor web

Pode comportar as camadas de apresentação, aplicação e dados em uma única máquina ou em diversas máquinas, sendo essa distribuição indistinguível para o cliente.

Quando falamos de comunicação, estamos tratando mais especificamente de como trafegam os dados entre a requisição enviada por um cliente e a resposta provida, por um servidor.

Comunicação no ambiente web

A comunicação, nesse ambiente, é feita sobre a internet, com o uso dos seus protocolos de comunicação, sendo o principal protocolo o **HTTP** *(HyperText Transfer Protocol)*, que é um protocolo para transferência de hipertexto. Na imagem seguinte, podemos ver um exemplo de comunicação no ambiente web.



Comunicação no ambiente web.

No exemplo apresentado, temos, de um lado, o cliente que, com um desktop ou smartphone, faz a requisição, através da internet, de um serviço — representada pelo arquivo Listar-TV.php — a um servidor. O servidor web, após processar a requisição, retorna a informação solicitada, representada pelo arquivo Listagem-TV.php. Com isso, podemos entender como funcionam as aplicações disponíveis no ambiente web, como websites de notícias, comércio eletrônico, e-mail, redes sociais etc. Em cada um desses casos, há uma requisição sendo feita pelo cliente e o servidor processando a requisição e respondendo ao cliente com o que foi solicitado.

Solicitação e resposta

O processo de comunicação no ambiente web é conhecido como solicitação *(request)* e resposta *(response)*. Normalmente, a solicitação é iniciada pelo cliente, mas é possível que também o servidor a inicie, como em serviços PUSH — serviços que disparam notificações/mensagens para os clientes que fizeram opção por recebê-las.

Processo de solicitação e resposta .

Client side X Server side

Essas duas expressões são muito comuns quando falamos de aplicações rodando no ambiente web. Ambas se referem a tecnologias e códigos disponibilizados no **lado cliente** (nesse caso, o dispositivo utilizado por um usuário para fazer uma requisição) e no **lado servidor**.

Atividade 2

Você foi convidado para ser um palestrante sobre a evolução da web em um evento da empresa na qual trabalha. Você recebeu várias dicas sobre o que você deveria falar. Então, após uma análise crítica das opções a seguir, qual delas você escolheria?

A Desde que a web foi criada, o modelo cliente x servidor não sofreu alterações porque se trata de uma tecnologia muito robusta e eficiente, sempre adotando duas camadas, ou seja, de apresentação e de web.

B Ao longo da evolução da web, observamos diferentes abordagens na adoção do modelo cliente x servidor, passando por duas, três e quatro camadas. Nessa última abordagem, o cliente não precisa de nenhum software adicional instalado no seu computador porque o servidor web fornece os códigos que serão executados nos navegadores.

C Ao longo da evolução da web, observamos diferentes abordagens na adoção do modelo cliente x servidor, passando por duas, três e quatro camadas. Nessa última abordagem, basta que o cliente instale o software de interface, desenvolvido especificamente para cada aplicação, que o possibilita comunicar com a aplicação hospedada no servidor.

D O modelo cliente x servidor de quatro camadas se refere às aplicações que utilizam criptografia, a saber: camada de visualização, camada de criptografia, camada de descriptografia e camada de processamento.

E O modelo cliente x servidor utiliza apenas uma camada, que concentra toda aplicação, dados, web, entre outros.

A alternativa B está correta.

O modelo de quatro camadas da arquitetura cliente x servidor possibilita melhor separação dos conceitos, de modo que os clientes não precisam sequer instalar nenhum software adicional nas suas máquinas, nem mesmo para a interface, bastando usar os navegadores comumente utilizados.

Demonstração prática da Arquitetura cliente X Servidor

Com a evolução da web, a geração de páginas web dinâmicas se tornou fundamental. Mas o que são páginas web dinâmicas? Em poucas palavras, são páginas geradas com conteúdos personalizados. Há vários exemplos de geração de páginas web dinâmicas: redes sociais, bankline, streaming, entre tantos outros. Portanto, seja qual for a aplicação, você acessa um navegador de sua preferência, insere o endereço (exemplo fictício: www.xpto.com.br) e aciona a tecla enter.

Ao fazer isso, a sua solicitação (requisição) chegará no servidor que hospeda a aplicação que, por sua vez, devolve uma resposta contendo os códigos (HTML, CSS, JavaScript) que serão executados no seu próprio navegador. A página que você visualiza tem formulários de login e senha, que você preenche e, depois, aciona a tecla enter novamente. Os seus dados de login e senha serão enviados ao servidor da aplicação, que fará o processamento deles e, depois, devolverá os códigos da página que você visualizará no seu navegador, contendo os seus dados da sua rede social, da sua conta bancária etc. E isso se repetirá com cada usuário que acessar tais serviços, ou seja, cada um deles verá um conteúdo específico, destinado a eles.

O vídeo vai demonstrar de forma prática o funcionamento das páginas web dinâmicas, apresentando como fazê-las exibir conteúdos personalizados a cada usuário que acessar seus serviços.

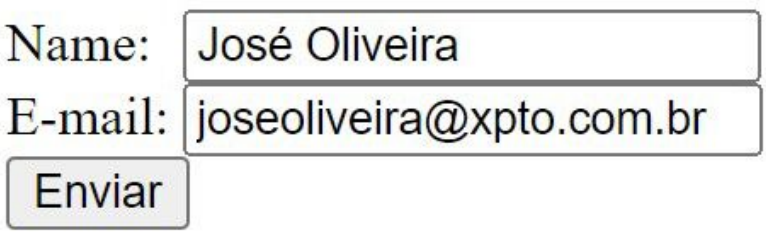
Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

Para materializar a ideia, vamos considerar o exemplo disponibilizado pelo W3schools em Formulários PHP na aba PHP.

Vamos analisar a execução do exemplo e, depois, analisaremos seus códigos. Há uma página de frontend que contém dois campos de formulário: Name e E-mail, conforme mostra a imagem.



Formulário do frontend.

Ao inserir os dados de nome igual a José Oliveira e de e-mail igual a joseoliveira@xpto.com.br e clicar no botão Enviar, os dados serão levados ao servidor que hospeda a aplicação, que fará o processamento desses dados, gerando a página que será devolvida ao usuário, contendo uma mensagem personalizada, conforme mostra a seguir.

Welcome Jonh

Your email adress is jonh.doe@exemplo.com

Esse exemplo ilustra o ciclo requisição e resposta. Na requisição, o nome e o e-mail foram enviados ao servidor que, por sua vez, envia a resposta com a mensagem personalizada ao usuário. Trata-se de um exemplo bem simples, mas que ilustra, em linhas gerais, o que acontece com exemplos mais complexos.

Depois da visão geral que foi mostrada, vamos analisar os códigos tanto do frontend quanto do backend. Começaremos com os códigos da página do frontend, conforme mostra a seguir.

```
plain-text



Name:

E-mail:
```

Observe que a primeira linha de código do formulário tem dois atributos: action e method, conforme mostra a imagem.

```
plain-text
```

O atributo action contém o nome da aplicação que processará os dados do formulário no servidor e, nesse caso, trata-se de uma página PHP denominada welcome.php.

O atributo method especifica o método de envio dos dados que, neste caso, é o post, o que quer dizer que os dados serão enviados no corpo da mensagem. Há outros métodos de envio, como o GET, que anexa os dados no endereço.

Em seguida, há os códigos dos campos de formulário, definidos por meio das tags input, conforme mostra a imagem.

```
plain-text
```

As tags input do tipo text possibilitam criar campos de formulário em que os caracteres digitados ficam visíveis. O atributo name possibilita dar um nome aos campos, o que é importante para que a aplicação do servidor possa extrair e processar adequadamente os dados contidos nestes campos. Finalmente, é criado o botão que envia os dados ao servidor, por meio da tag input com tipo submit, conforme mostra a imagem.

```
plain-text
```

E os códigos da página welcome.php podem ser semelhantes aos que estão mostrados a seguir.

```
php



Welcome

Your email address is:
```

Essa página extrai os dados provenientes da requisição e mostra uma mensagem personalizada utilizando a instrução echo.

A imagem mostra o trecho de código PHP que extrai o conteúdo do campo name que chegou na requisição via método Post.

```
php
```

E a imagem mostra o trecho de código PHP que extrai o conteúdo do campo e-mail que também chegou na requisição via método Post.

```
php
```

Atividade 3

Considere que você chegou para trabalhar como desenvolvedor em uma empresa e se deparou com certos códigos utilizados na integração de sistemas. Baseado nesse contexto, você está em uma atividade que envolve o desenvolvimento de uma aplicação com PHP que recebe dois dados (nome e curso) como parâmetros de requisição via método POST. A aplicação deve extrair os dados da requisição e mostrar uma mensagem de boas-vindas personalizada. Como você implementaria esse código?

A

```
<html>
<body>

Olá, <?php echo $_POST["nome"]; ?> <br>
Seja bem-vindo ao curso de <?php echo $_POST["curso"]; ?>

</body>
</html>
```

B

```
<html>
<body>
Olá, <?php echo $_GET["nome"]; ?> <br>
Seja bem-vindo ao curso de <?php echo $_GET["curso"]; ?>
</body>
</html>
```

C

```
<html>
<body>
Olá, <?php echo nome; ?> <br>
Seja bem-vindo ao curso de <?php echo curso; ?>
</body>
</html>
```

D

```
<html>
<body>
Olá, <?php printf(&nome); ?> <br>
Seja bem-vindo ao curso de <?php printf(&curso); ?>
</body>
</html>
```

E

```
<html>
<body>
Olá, <?php request.getParameter[“nome”]; ?> <br>
Seja bem-vindo ao curso de <?php request.getParameter[“curso”]; ?>
</body>
</html>
```

A alternativa A está correta.

O uso de $_POST["nome"] possibilita coletar o dado "nome" que chega em uma requisição via método POST. E a instrução echo imprime o conteúdo desse dado na tela. Situação semelhante ocorre com o dado intitulado "curso".

Visão geral de interface de usuários

As interfaces de usuário podem ser entendidas como sendo a ponte de ligação entre os usuários e as aplicações. E grande parte das aplicações são acessadas por usuários que não têm conhecimento das tecnologias utilizadas no projeto de software. Sendo assim, é de fundamental importância que as interfaces sejam claras e tenham usabilidade adequada, de modo a proporcionar boa experiência aos usuários. Além disso, atualmente, há computadores, tablets e smartphones com diferentes tamanhos de telas e, dessa forma, é importante que as interfaces possam funcionar adequadamente em todos os dispositivos.

O vídeo vai abordar a interação entre as aplicações de computador e os seres humanos que os utilizam, destacando a importância da clareza e usabilidade da interface.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

O conceito de interface está ligado à área de Interação Humano-Computador (IHC), que pode ser resumida como o estudo da interação entre pessoas e computadores. Nesse contexto, a interface, muitas vezes chamada de interface do utilizador, é quem provê a interação entre o ser humano e o computador. No início da utilização dos computadores, tal interação era realizada por meio de linha de comando e, posteriormente, também mediante **interfaces gráficas** *(Graphical User Interface - GUI)*. Segundo Morais (2014), no início, a interação foi, de certo modo, primária, deixando um pouco de lado o ser humano, por não existir um estudo aprofundado desses aspectos.

Dessa forma, o foco do estudo da interface envolvia principalmente o hardware e o software, e o ser humano simplesmente tinha que se adaptar ao sistema criado. Posteriormente, com o avanço da tecnologia e do acesso a computadores, e mais recentemente a outros dispositivos, sobretudo os smartphones, a necessidade de melhorar a interação tem crescido continuamente.

A interface do lado cliente

Como Silva (2014) explica, a evolução tecnológica levou a uma crescente utilização de dispositivos móveis que possuem os mais **variados tamanhos de tela e funcionalidades**.

Sobre essa variedade nas características dos dispositivos utilizados como interface para o acesso a aplicações no ambiente web, é necessário garantir a usabilidade, ou seja, que sejam desenvolvidos sistemas fáceis de usar e de aprender, além de flexíveis. Em complemento a esse conceito, e partindo do ponto de vista da usabilidade, esta deve estar alinhada ao conceito de design responsivo, o qual deverá permitir que as páginas web e consequentemente as aplicações web respondam a qualquer dispositivo sem perda de informações por parte do usuário.

O site **StatCounter Global Stats** mantém ativa uma série de dados e estatísticas sobre dispositivos, tamanhos de tela, além de outras informações relacionadas. Sobre o tamanho de telas, e considerando o período de abril de 2019 a abril de 2020, temos os seguintes dados:

Tamanho da tela em pixels (largura x altura)	Percentual de utilização
360 × 640	10,11%
1366 × 768	9,69%
1920 × 1080	8,4%
375 × 667	4,24%
414 × 896	3,62%
1536 × 864	3,57%

Tabela: Estatísticas mundiais sobre resolução de telas de dispositivos.
Alexandre Paixão.

Quando consideramos essas mesmas estatísticas, mas levando em conta especificamente os dados de navegação do Brasil, temos um cenário diferente, conforme pode ser visto na imagem a seguir.

Estatísticas sobre resoluções de telas de dispositivos - Brasil.

Atividade 1

Considere que você trabalhe como desenvolvedor frontend e foi convidado para participar de uma reunião com um cliente. Nessa reunião, você precisa argumentar sobre as interfaces de usuário. Entre as opções mostradas a seguir, qual delas você utilizaria nessa reunião?

A As interfaces de usuários só devem ser acessadas por profissionais da área de TI, de modo que não precisam ser tão claras e objetivas.

B As interfaces de usuários são utilizadas por profissionais de TI que trabalham com o desenvolvimento e a integração de bancos de dados relacionais e não relacionais.

C As interfaces de usuários estão caindo em desuso por conta do uso de smartphones, tendo em vista que as aplicações desenvolvidas para esses dispositivos não necessitam destes recursos.

D As interfaces de usuário possibilitam que usuários de diferentes áreas e backgrounds acessem aplicações que, frequentemente, são complexas. Tudo isso de forma clara e objetiva, com boa usabilidade.

E As interfaces de usuários, embora sejam muito importantes, encontram limitações para lidar com dispositivos com telas de tamanhos diferentes.

A alternativa D está correta.

Não é necessário ser da área de tecnologia para utilizar e acessar aplicações por meio de interfaces de usuários. Por isso, devem ser implementadas de forma clara, objetiva e com boa usabilidade, facilitando a utilização dos usuários.

O conceito do design responsivo

No passado, as telas dos monitores utilizados nos computadores tinham tamanhos semelhantes. Entretanto, com o avanço da tecnologia e o surgimento de outros dispositivos, como notebooks, tablets e smartphones, há diversos tamanhos de telas e, sendo assim, é importante que os usuários possam acessar as páginas web de forma adequada independentemente do dispositivo que estejam utilizando. Diante disso, surgiram alguns conceitos como design responsivo, em que as páginas web se adaptam ao tamanho de tela do dispositivo utilizado, proporcionando melhor experiência aos usuários. Essa abordagem implica layouts mais fluidos, em que os conteúdos podem ser exibidos de formas diferentes.

O vídeo vai explicar a necessidade que surgiu de layouts mais fluidos, que garantam a possibilidade de acesso às páginas web de forma adequada independentemente do dispositivo que os usuários estejam utilizando.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Segundo Knight (2011), o **design responsivo** é a abordagem que sugere que o design e o desenvolvimento devam responder ao comportamento e ao ambiente do usuário com base no tamanho da tela, na plataforma e na orientação do dispositivo por ele utilizado.

Essa definição, na prática, implica que a página web/aplicação acessada deve ser capaz de, automaticamente, responder às preferências do usuário e, com isso, evitar que seja necessário construir diferentes versões de uma mesma página/aplicação para diferentes tipos e tamanhos de dispositivos.

A origem do design responsivo

O conceito de design responsivo teve sua origem no projeto arquitetônico responsivo. Tal projeto prega que uma sala ou um espaço deve se ajustar automaticamente ao número e fluxo de pessoas dentro dele. Para tanto, é utilizada uma combinação de robótica e tecnologia, como: sensores de movimento; sistemas de controle climático com ajuste de temperatura e iluminação; juntamente com materiais — estruturas que dobram, flexionam e expandem.

Da mesma forma que no Projeto Arquitetônico Responsivo, arquitetos não refazem uma sala ou um espaço de acordo com o número, fluxo e as características de seus ocupantes, no ambiente web não devemos ter que precisar construir uma versão de uma mesma página de acordo com as características dos seus visitantes. Isso traria ainda outros custos, como identificar uma enorme combinação de tamanhos de tela e tecnologia, entre outros fatores, para criar uma mesma quantidade de páginas correspondentes.

Design responsivo na prática

Na prática, ao aplicarmos o conceito de design responsivo, fazemos uso de uma combinação de técnicas, como **layouts fluidos, media query e scripts**. A seguir veremos cada uma dessas técnicas em detalhes.

Para entender o conceito de layout fluido, é necessário entender primeiro o que seria o seu oposto, ou seja, o layout fixo.

Layout fixo

As dimensões (largura e altura) dos elementos de uma página web são definidos com a utilização de unidades de medidas fixas, como os pixels (menor ponto que forma uma imagem digital). Com isso, tais elementos não se adaptam às alterações no tamanho do campo de visão dos dispositivos que os visualiza.

Layout fluido

Já os layouts fluidos fazem uso de unidades flexíveis — no lugar de definir as dimensões com o uso de quantidades fixas são utilizados valores flexíveis. Isso permite, por exemplo, que em vez de definir que o cabeçalho de uma página tenha 1366 pixels de largura, possamos definir que ele ocupe 90% do tamanho da tela do dispositivo que o visualiza. Daí o conceito de fluido, ou seja, de adaptabilidade ao campo de visão conforme dimensões do dispositivo que visualiza a página.

Além dos valores percentuais, há outras unidades de medidas flexíveis, por exemplo:

EM

Unidade de medida tipográfica, estando relacionada à letra “M”. O tamanho base dessa unidade equivale à largura da letra “M” em maiúscula.

REM

Enquanto o EM está relacionado ao tamanho do elemento de contexto (ou seja, definimos o valor EM de um elemento tomando como base o seu elemento pai), no REM definimos que o elemento de contexto, o elemento pai, será sempre a tag HTML . Daí a letra “R” nessa unidade, que faz referência à raiz (root).

Além das unidades, fixas e flexíveis já mencionadas, há ainda outras disponíveis. A listagem completa pode ser acessada no site do W3C – CSS Units.

A função de apresentação, de estruturar o layout de uma página, no ambiente web, cabe às **folhas de estilo (CSS)**. Trataremos mais a fundo do CSS ao longo do nosso estudo. Por ora, para definir o que é media query, falaremos um pouco também sobre CSS.

Com base na afirmação de que cabe ao CSS estruturar o layout de uma página web, temos normalmente associada a uma página web uma ou mais folhas de estilo — que são códigos que definem aspectos de toda a página, como as dimensões dos elementos, cores de fundo, as cores e os tipos de fonte etc.

Comentário

Media query é a utilização de media types (tipos de mídia) a partir de uma ou mais expressões para definir formatações para dispositivos diversos. Com o seu uso podemos, por exemplo, definir que determinado estilo de um ou mais elementos seja aplicado apenas a dispositivos cuja largura máxima de tela seja igual ou menor que 600px.

A imagem seguinte mostra um fragmento de código em que uma media query é utilizada para impedir que um menu lateral (o elemento HTML cuja classe equivale a “menu_lateral”) seja exibido caso a largura da tela do dispositivo seja menor que 360px.

```
<style type="text/css">
@media (max-width: 360px)
{
  .menu_lateral
  {
    display: none;
  }
}
</style>
```

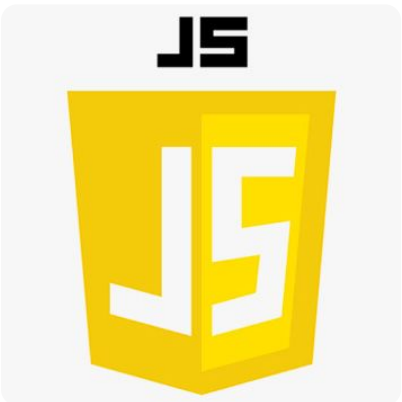
Exemplo de declaração de media query.

O resultado das expressões utilizadas na media query pode ser verdadeiro ou falso. No caso de nosso exemplo, será verdadeiro sempre que a largura da tela do dispositivo que visualiza a página seja inferior a 360px. Do contrário, será falso. Ou seja, para todos os dispositivos cuja largura de tela seja superior a 360px, o código CSS em questão será ignorado.

Atenção

Na media query, podemos utilizar expressões como a definição do tipo de mídia (media type) — ou seja, um estilo que se aplica apenas a um ou mais tipos de documento, como a versão para impressão de uma página web, por exemplo — e a combinação entre escalas de valores.

Scripts



Quando falamos em scripts no lado cliente,Essa linguagem adiciona interação a uma no ambiente web,página web, permitindo, por exemplo, a estamos falando deatualização dinâmica de conteúdos, o controle linguagens dede multimídia, a animação de imagens e muito programação quemais. No contexto do design responsivo, sua rodam no navegador efaceta mais importante é a de atualização cujo exemplo maisdinâmica de conteúdo — e não só do comum é o **JavaScript**. conteúdo, mas também da apresentação dele.

Design responsivo X Design adaptativo

O conceito de **design adaptativo**, muitas vezes, confunde-se com o de design responsivo. Enquanto o segundo, como já visto anteriormente, consiste na utilização de uma combinação de técnicas para ajustar um site automaticamente em função do tamanho da tela dos dispositivos utilizados pelos usuários, no design adaptativo são usados layouts estáticos baseados em pontos de quebra (ou de interrupção), em que, após o tamanho de tela ser detectado, é carregado um layout apropriado para ele. Em linhas gerais, no design adaptativo, são criados layouts com base em seis tamanhos de tela mais comuns. A aplicação desses dois conceitos na prática acontece da seguinte forma:

Design responsivo

Medias queries são utilizadas, em conjunto com scripts, para criar um layout fluido que se adapte — por meio, sobretudo, da adequação das dimensões de seus elementos — ao tamanho da tela do dispositivo utilizado pelo visitante.

Design adaptativo

Um site é planejado e construído com a definição de seis layouts predefinidos, em que são previstos pontos de quebra para que a página se adapte às seis diferentes dimensões utilizadas.

Poderíamos ainda dizer que o design responsivo é mais complexo, porém mais flexível. Já o adaptativo, mais trabalhoso, embora menos flexível.

Como dito, no design responsivo é preciso criar uma série de combinações de media query para que o layout se adapte aos mais variados tamanhos de tela. Já no adaptativo, imaginemos uma situação em que foram definidos os seguintes layouts e quebras: 360px, 720px, 900px, 1080px, 1440px e 1800px. Caso a largura da tela do dispositivo seja superior a 360px e inferior a 720px — por exemplo, 700px —, será carregado o layout de 360px, que equivale, praticamente, à sua metade. É possível imaginar que, nesse caso, o resultado não seja visualmente muito agradável ou otimizado.

Mobile first

Uma das abordagens de design responsivo mais utilizadas atualmente é a mobile first. Tal abordagem está centrada no crescente uso de dispositivos móveis na navegação no ambiente web e defende que em primeiro lugar seja pensado o design para telas menores e, posteriormente, para telas maiores. Trata-se de um enfoque progressivo (*progressive enhancement*), no qual se parte dos recursos e tamanhos de tela disponíveis nos dispositivos menores, progredindo com a adição de recursos e conteúdo tendo em vista as telas e os dispositivos maiores.

A partir da definição de mobile first podemos identificar o seu contraponto com o desenvolvimento web tradicional, em que temos o conceito de degradação graciosa (*graceful degradation*):

As páginas web são projetadas tendo em vista dispositivos desktop e telas maiores e, posteriormente, adaptadas para dispositivos móveis e telas menores.

> A aplicação prática do mobile first consiste em planejar o desenvolvimento de um site priorizando os recursos e as características presentes nos dispositivos móveis, como o tamanho de tela, a largura de banda disponível e até mesmo recursos específicos, como os de localização, por exemplo.

Atividade 2

Você foi convidado para participar de um podcast sobre tecnologia para o público em geral. Sendo assim, você deve explicar o conceito de design responsivo. Entre as opções mostradas a seguir, qual delas seria a adequada nesse contexto?

A Design responsivo está relacionado com a responsabilidade do usuário em utilizar adequadamente os recursos das páginas web, ou seja, trata-se do comprometimento do usuário em fazer bom uso dos recursos ofertados a ele.

B Design responsivo está relacionado com o desenvolvimento de aplicações web seguras, que utilizem criptografia dos dados, entre outros recursos.

C Design responsivo está relacionado ao fato de que as páginas web serão exibidas adequadamente em dispositivos com tamanhos de telas diferentes, ou seja, o layout das páginas se adapta aos tamanhos de telas dos dispositivos.

D O conceito de design responsivo é obsoleto e foi utilizado nos primórdios da web com páginas web estáticas.

E Design responsivo está relacionado às páginas web que utilizam formulários de login e senha, que precisam de recursos de segurança cibernética para garantir a confidencialidade dos dados.

A alternativa C está correta.

De fato, design responsivo possibilita que as páginas web sejam exibidas adequadamente em dispositivos com tamanhos de telas diferentes, proporcionando melhor experiência aos usuários.

As tecnologias HTML

HTML, sigla para linguagem de marcação de hipertexto, é utilizada para a estruturação de páginas web. Objetivando materializar a ideia, vamos fazer uma comparação com a construção civil. O terreno pode ser comparado ao servidor web, que hospeda as aplicações. A estrutura de vigas e pilares da edificação pode ser comparada ao HTML, que define a estrutura de páginas web. O acabamento pode ser comparado a outras tecnologias, como CSS (Cascading Style Sheet). HTML tem várias tags predefinidas, como para a definição de títulos, parágrafos, formulários, inserção de imagens, vídeos, entre outras.

O vídeo vai demonstrar, por meio de uma comparação com a construção civil, como a tecnologia HTML é fundamental, permitindo estruturar páginas da web.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

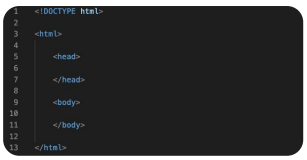
HTML

A **HTML** é considerada a tecnologia fundamental da web, pois sua função é definir a estrutura de uma página web. Essa linguagem de marcação, criada por Tim Berners-Lee na década de 1990, inicialmente objetivava permitir a disseminação de pesquisas entre Lee e seus colegas pesquisadores, mas foi rapidamente difundida até formar a rede que, posteriormente, veio a se tornar a World Wide Web como a conhecemos atualmente.

Em linhas gerais, a HTML é uma linguagem de marcação simples, composta por elementos, chamados tags, que são relacionados a textos e outros conteúdos a fim de lhes dar significado. Por exemplo: podemos marcar um texto como sendo um parágrafo, uma lista ou uma tabela. É possível, ainda, inserir vídeos e imagens. Além disso, também utilizamos essa marcação para definir a estrutura de um documento de forma lógica: menu de navegação, cabeçalho, rodapé etc. As tags podem ser agrupadas nos seguintes tipos:

Estruturais

Juntamente com o elemento de definição do DocType, como pode ser visto na imagem seguinte, compõem a estrutura obrigatória de uma página web.



De conteúdo

Como nome sugere, têm o papel de marcar o conteúdo pelo seu tipo.

Semânticas

Relacionadas ao tipo de conteúdo e à criação de seções para agrupá-lo de acordo com sua função no documento. Para melhor entender esse conceito, veja a imagem a seguir:

Como visto na imagem apresentada, as tags < header >, < nav >, < main > e < footer > desempenham papel semântico, uma vez que estruturam a página em seções. Como seus nomes indicam, elas separam o conteúdo em partes lógicas que formam o esqueleto da maioria das páginas HTML, ou seja: cabeçalho, menu de navegação, conteúdo principal e rodapé. Logo, tags de parágrafo, imagem, entre outras, são inseridas dentro de cada uma dessas seções, formando assim um documento HTML completo.

Uma listagem completa de tags e atributos (usados para adicionar características a uma tag) pode ser encontrada no site do W3C.

HTML5

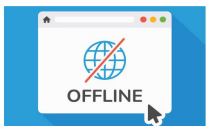
A versão mais recente da **HTML** é a **5**, que trouxe algumas importantes evoluções em relação às anteriores. Entre tais novidades destacam-se:

Novos atributos e elementos, com foco sobretudo na semântica.

Melhorias de conectividade.

Possibilidade de armazenamento de dados no lado cliente.

Otimização nas operações offline.



Suporte estendido a multimídia — áudio e vídeo.

Atividade 1

Você foi designado para o treinamento de novos estagiários na empresa em que trabalha. Um dos temas do treinamento envolve tecnologias de frontend. Diante disso, qual das opções mostradas a seguir poderia ser utilizada nesse treinamento?

A HTML é utilizada para a estruturação de páginas web por meio do uso de diversas tags pré-definidas.

B HTML é utilizada para a definição de estilo de páginas web, enquanto outras tecnologias, como o CSS, são utilizadas para a estruturação de páginas web.

C HTML é utilizada para alteração da parte comportamental de páginas web, ou seja, ao clicar em um botão, HTML define as ações que serão executadas.

D HTML é utilizada para a integração de códigos escritos em PHP e banco de dados desenvolvido em MySQL.

E HTML está caindo em desuso e está sendo substituída por XML, que quer dizer linguagem de marcação extensível.

A alternativa A está correta.

HTML tem uma função fundamental, que é a estruturação de páginas web e contém diversas tags predefinidas que auxiliam na criação de títulos, parágrafos, inserção de imagens, vídeos, entre outros recursos.

Outras tecnologias: CSS e JavaScript

Embora HTML 5 até tenha recursos que podem ser utilizados para alterar o estilo de certos elementos de páginas web, o CSS é muito mais eficiente nesse propósito, especialmente em portais que tenham várias páginas que guardem a mesma identidade visual. Ou seja, por meio de CSS externo, pode-se otimizar o trabalho de alteração do estilo de várias páginas web de uma só vez. JavaScript, por outro lado, pode alterar a parte comportamental de páginas web, ou seja, pode-se definir o que acontece quando o usuário clica em um botão, por exemplo.

O vídeo vai mostrar que cada tecnologia se adapta melhor a um propósito, sendo CSS utilizado em casos em que é necessária a alteração do estilo de várias páginas web de uma só vez e JavaScript, para alterar a parte comportamental de páginas web.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

CSS

O **CSS** corresponde à segunda camada no tripé de tecnologias que formam o lado cliente, no ambiente web. Trata-se de uma linguagem declarativa cuja função é controlar a apresentação visual de páginas web. Com isso, têm-se a separação de funções em relação à HTML.

CSS

Sigla de Cascading Style Sheets. Em português, folhas de estilo em cascata.

Dica

Em um website, o HTML cuida do conteúdo, da estruturação e o CSS cuida da apresentação, do layout.

Sintaxe

A sintaxe da CSS consiste em uma declaração em que são definidos o(s) elemento(s) e o(s) estilo(s) que desejamos aplicar a ele(s) ou, em outras palavras:

O seletor

Um elemento HTML (body, div, p etc.) ou o seu identificador (atributo ID) ou classe (atributo class).

A propriedade

Característica do elemento (cor, fonte, posição etc.).

O valor

Novo parâmetro a ser aplicado à característica do elemento.

Por exemplo, para alterar a cor da fonte de um texto inserido em um parágrafo, poderíamos utilizar uma das variações apresentadas na imagem a seguir.

```
<p id="paragrafo_exemplo">Texto do parágrafo que será estilizado com CSS</p>
```

```
#paragrafo_exemplo{
    color: #ff0000; //vermelho
}

OU

p{
    color: #ff0000; //vermelho
}
```

Exemplo de aplicação de CSS.

No exemplo apresentado, vimos duas formas para definir o estilo de uma tag de parágrafo. Na primeira, o elemento ao qual o estilo foi aplicado foi definido com a utilização de seu atributo ID. Na segunda, dos seletores, propriedades existentes e mais detalhes sobre a CSS, é recomendado ler o Guia de Referência do próprio W3C.

Como inserir o CSS na página web

Há quatro formas de inserir o CSS em um documento:

Inline

Os estilos, neste caso, são aplicados com a utilização do atributo “style” seguido de uma ou mais propriedades/valores.

Interno

Os estilos são definidos com a utilização da tag , dentro da tag <head> no documento.</p>

Externo

Essa é a forma preferencial de inserir estilos. Nela, é utilizado um arquivo externo, com extensão “.css”, contendo apenas estilos. Para vincular esse arquivo externo ao documento, é utilizada a tag dentro da tag .

Escopo

Essa forma foi introduzida pela HTML5. Com ela, um estilo pode ser definido em nível de escopo, ou seja, declarado em seções específicas do documento. Sua declaração é feita da mesma forma que na inline. Entretanto, no lugar de ser declarada no, é declarada dentro da tag à qual se quer aplicar os estilos.

Seletores CSS

A CSS permite uma série de combinações para a aplicação de estilos. Pode-se usar aplicações simples, como as vistas até aqui, nas quais apenas um elemento foi selecionado, ou combinações mais complexas, em que vários elementos podem ser agrupados a fim de receberem um mesmo estilo.

Boas práticas relacionadas à CSS

É boa prática e fortemente recomendado utilizar a **forma externa** para incluir CSS em uma página web. Entre outras vantagens, como uma melhor organização do código, separando o HTML do estilo, devemos ter em mente que um mesmo arquivo CSS pode ser usado em várias páginas de um site.

Agora imagine uma situação oposta.

Exemplo

Temos um site cujo layout (topo, rodapé e menu, por exemplo) é comum em todas as suas páginas – arquivos .html independentes. Ao usarmos as formas inline e interna, precisaríamos replicar um mesmo código em todas as páginas. Imagine ter que fazer alguma alteração ou inclusão, tal operação precisaria ser repetida inúmeras vezes. Em contrapartida, se usarmos um arquivo externo e o linkarmos em todas as páginas, precisaremos trabalhar apenas em um único código, tornando-o muito mais fácil de ser mantido e ainda diminuindo consideravelmente nosso trabalho.

Outra **boa prática**, tendo em vista o **desempenho do carregamento da página web é compactar o arquivo** — normalmente chamamos este processo de minificação. Existem softwares e até mesmo sites que fazem esse trabalho, que consiste em, resumidamente, diminuir os espaços e as linhas no arquivo .css, reduzindo assim o seu tamanho final.

Outras considerações sobre a CSS

Uma nova funcionalidade tem ganhado bastante espaço ultimamente no que diz respeito à CSS: os **pré-processadores**, como Sass, Less, Stylus etc. Em linhas gerais, um pré-processador é um programa que permite gerar CSS a partir de uma sintaxe — própria de cada pré-processador —, que inclui inúmeras facilidades não suportadas naturalmente pelo CSS, como variáveis, funções, regras aninhadas, entre outras.

O fluxo de gerar CSS por meio de um pré-processador consiste na escrita do código contendo as regras a serem aplicadas, fazendo uso da sintaxe de cada pré-processador. Ao final, esse código será compilado, gerando então o código CSS normal.

JavaScript

O **JavaScript** completa o tripé de tecnologias web que rodam no lado cliente. Trata-se de uma linguagem de programação que, assim como o CSS, é interpretada pelo navegador. Entre suas principais características, destaca-se o fato de ser multiparadigma (orientação a objetos, protótipos, funcional etc.).

JavaScript

Costuma-se abreviar o seu nome como “JS”, que é também a extensão de seus arquivos, quando vinculados externamente ao documento HTML.

Sua função é, sobretudo, fornecer interatividade a páginas web, e foi criada com o intuito de diminuir a necessidade de requisições ao lado servidor, permitindo a comunicação assíncrona e a alteração de conteúdo sem que seja necessário recarregar uma página inteira.

Sintaxe

O JavaScript é, ao mesmo, amigável, mas também completo e poderoso. Embora criado para ser leve, uma vez que é interpretado nativamente pelos navegadores, trata-se de uma linguagem de programação completa e, como já mencionado, multiparadigma. Logo, seus códigos podem ser tanto estruturados quanto orientados a objetos. Além disso, permitem que bibliotecas, como Jquery, Prototype etc. sejam criadas a partir de seu core, estendendo assim a sua funcionalidade.

Jquery

É uma biblioteca JavaScript rápida, pequena e rica em recursos que simplifica processos como a manipulação de documentos HTML, eventos, animação, além do AJAX (JQuery).

Prototype

É um framework JavaScript de código aberto, modular e orientado a objetos que provê extensões ao ambiente de script do navegador, fornecendo APIs para manipulação do DOM e AJAX (PrototypeJs).

Vejamos algumas características dessa linguagem:

Eventos e manipulação DOM

Essa linguagem oferece amplo suporte à manipulação de eventos relacionados a elementos HTML. É possível, por exemplo, utilizar um elemento < button> (botão) que, ao ser clicado, exiba uma mensagem na tela. Ou ainda aumentar o tamanho de uma fonte ou diminuí-lo.

Mensagem e entrada de dados

O JavaScript possui suporte a funções nativas para a exibição de caixas de diálogo para entrada de dados ou exibição de mensagens, como alertas, por exemplo.

Atividade 2

Considere que você seja um estudante de ciência da computação e esteja estudando tecnologias utilizadas no frontend. Você acessou uma sala virtual de estudo em que há vários estudantes mostrando seus pontos de vista sobre o assunto. Assinale a opção que esteja correta diante desse contexto.

A HTML, CSS e JavaScript são tecnologias muito utilizadas no projeto de páginas web. CSS é utilizad para estruturar as páginas, JavaScript é utilizada para alteração do estilo e HTML é utilizada para a alteração da parte comportamental de páginas web.

B HTML, CSS e JavaScript são tecnologias muito utilizadas no projeto de páginas web. JavaScript é utilizada para estruturar as páginas, HTML é utilizada para alteração do estilo e CSS é utilizada para a alteração da parte comportamental de páginas web.

C HTML, CSS e JavaScript são tecnologias muito utilizadas no projeto de páginas web. HTML é utilizada para estruturar as páginas, CSS é utilizada para alteração do estilo e JavaScript é utilizada para a alteração da parte comportamental de páginas web.

D Com o advento do HTML 5, as páginas web atualmente só utilizam HTML para realizar a estruturação, definição de estilo e parte comportamental de páginas web.

E Com o avanço tecnológico, HTML, CSS e JavaScript estão sendo substituídas por XML, que quer dizer Linguagem de Marcação Extensível.

A alternativa C está correta.

Em projeto de páginas web, deve-se explorar o que as tecnologias HTML, CSS e JavaScript têm de melhor. HTML é utilizada para estruturar as páginas, definindo títulos, subtítulos, parágrafos, etc. CSS é utilizada para a alteração de estilo, alterando as cores, os tipos e os tamanhos de fontes etc. JavaScript é utilizada para a alteração da parte comportamental das páginas, por meio de eventos quando o usuário clica no mouse.

Utilizando CSS em página HTML

Como foi estudado anteriormente, HTML quer dizer linguagem de marcação de hipertexto e tem diversas tags pré-definidas. Além disso, CSS tem vários recursos que podem ser utilizados para alterar o estilo de páginas web. Consequentemente, para se ter o domínio adequado de tantos recursos, é fundamental que os estudantes pratiquem por meio de atividades práticas, de modo que possam explorar os detalhes de tantas possibilidades. Por exemplo, como implementar uma página web que demande as alterações das cores do título e subtítulo, cor de fundo da página, alinhamento do parágrafo? Como reunir os recursos de CSS integrados em páginas HTML?

Essas são perguntas pertinentes e, para respondê-las, é muito importante conhecer as características de cada uma dessas tecnologias, como a sintaxe. Podemos utilizar CSS de diferentes formas. Será que opção de

CSS inline é a mais eficiente? Ou então devemos utilizar sempre o CSS interno? E se utilizarmos o CSS externo, isso altera a página final visualizada pelos usuários? Qual opção escolher? Portanto, o bom conhecimento sobre esses recursos auxilia na escolha mais adequada das abordagens tecnológicas dos projetos de frontend.

O vídeo vai demonstrar na prática como usar os recursos do CSS para alterar vários estilos de páginas HTML. Você verá como implementar uma página web com alterações das cores do título e subtítulo, cor de fundo da página, alinhamento do parágrafo e reunir os recursos de CSS integrados em páginas HTML.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

Observe o exemplo do código mostrado a seguir. Trata-se de um exemplo de código CSS interno em página HTML. Inserimos o código CSS utilizando a tag

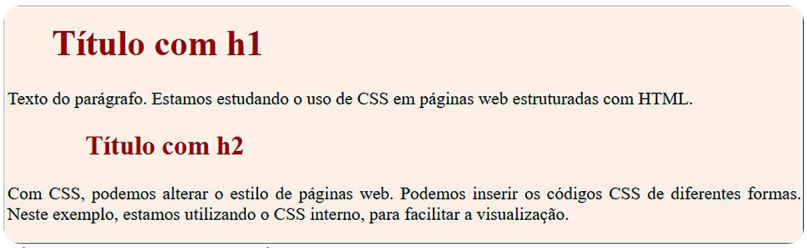
Titulo com h1

Texto do parágrafo. Estamos estudando o uso de CSS em páginas web estruturadas com HTML.

Titulo com h2

Com CSS, podemos alterar o estilo de páginas web. Podemos inserir os códigos CSS de diferentes formas. Neste exemplo, estamos utilizando o CSS interno, para facilitar a visualização.

A página web resultante do código anterior está mostrada a seguir. Observe!



Página web resultante do código anterior.

A seguir, mostraremos o código CSS do seletor body. A cor de fundo (background-color) da página é alterada para linen.

```plain-text
            body {
                background-color: linen;
            }
```

Adiante, mostraremos o código CSS do seletor h1. Nesse caso, duas propriedades são alteradas: a cor do texto (color) é alterada para maroon; e a margem esquerda (margin-left) é alterada para 40 pixels.

```
plain-text


            h1 {
                color: maroon;
                margin-left: 40px;
            }
```

A seguir, mostraremos o código CSS do seletor h2. Duas propriedades são alteradas também: a cor do texto (color) é alterada para maroon; e a margem esquerda (margin-left) é alterada para 70 pixels.

```
plain-text


            h2 {
                color: maroon;
                margin-left: 70px;
            }
```

Adiante, mostraremos o código CSS do seletor de parágrafo (p). Nesse caso, o alinhamento do texto (text-align) é definido para justificado (justify).

```
plain-text


            p {
                text-align: justify;
            }
```

Observando os códigos CSS dos seletores body, h1, h2 e parágrafo, podemos concluir que podemos alterar uma ou mais propriedades dos seletores. As propriedades podem ser repetidas para os seletores conforme mostram os códigos de h1 e h2.

Se utilizássemos o CSS externamente, com os mesmos códigos mostrados nesse exemplo com CSS interno, o resultado da página não seria alterado, ou seja, os usuários veriam a mesma página. Entretanto, o CSS externo proporciona melhor separação de conceitos, ou seja, um arquivo exclusivo com HTML e outro arquivo exclusivo com CSS.

Atividade 3

Considere que você trabalha em uma fábrica de software. Um colega de trabalho está enfrentando dificuldade com um código e pediu sua ajuda tendo em vista que o resultado não é o que se espera. O código está mostrado a seguir. Depois de analisar o código, qual é o problema e o que você faria para solucioná-lo?

```
plain-text
```

```
body {
   background-color: linen;
}

h1 {
   color: maroon;
   margin-left: 40px;
}

h2 {
   color: maroon;
   margin-left: 70px;
}

p {
   text-align: justify;
}
```


```
Serviços prestados: consultoria e treinamento em computação em nuvem.
```


```
Nossos treinamentos sobre segurança em computação em nuvem contemplam os recursos
tecnológicos mais avançados do mercado.
```

A O código não tem erros. Deve ser um problema do navegador.

B Trata-se de um código que utiliza CSS externamente, mas não foi feita a referência ao arquivo CSS. Portanto, para resolver o problema, basta referenciar o arquivo CSS externo dentro da tag ⟨body⟩.

C Podemos concluir que as tags HTML estão nos locais errados. Para resolver o problema, as tags que estão dentro de ⟨body⟩ devem ser colocadas dentro da tag ⟨head⟩.

D Trata-se de um código que utiliza CSS inline, mas não há tag ⟨inline⟩ sendo utilizada. Para resolver o problema, deve-se inserir a tag ⟨inline⟩ em cada linha de código contido dentro da tag ⟨body⟩.

E Trata-se de um código que utiliza CSS internamente, mas não foram inseridas as tags de abertura ⟨style⟩ de início e de fechamento ⟨/style⟩. Portanto, para resolver o problema, basta inserir as tags nos locais especificados.

A alternativa E está correta.

Conforme foi mencionado na resposta, trata-se da inserção de código CSS internamente na página HTML, que deve ocorrer com o uso da tag ⟨style⟩ no início do código CSS e com ⟨/style ⟩, indicando o fechamento das linhas de código CSS, delimitando as linhas de código de CSS, separando o código de estilo do código HTML.

Inserindo Javascript em código HTML

Como foi estudado anteriormente, por meio de JavaScript, podemos alterar a parte comportamental de páginas web. JavaScript tem diversos recursos que podem ser explorados como eventos que ocorrem quando o usuário clicar no mouse, entre outros. JavaScript é uma linguagem de programação multiparadigma e o seu uso não se limita à alteração da parte comportamental de páginas web. Entretanto, o foco do nosso estudo se refere aos recursos tecnológicos que podem ser empregados no frontend. Portanto, com JavaScript, podemos alterar imagens que são exibidas na tela, estilo dos seletores, entre várias outras possibilidades. E, assim como aconteceu com HTML e CSS, é fundamental que os estudantes realizem atividades práticas para consolidar o conhecimento, de modo a explorarem todo o potencial desses recursos que, combinados, possibilitam a criação de páginas web interativas, eficientes e com boa usabilidade.

O vídeo vai demonstrar de forma prática como explorar os recursos de JavaScript, com foco nos recursos que podem ser empregados no frontend, lembrando que essa linguagem não se limita apenas a alterar a parte comportamental de páginas web.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Roteiro de prática

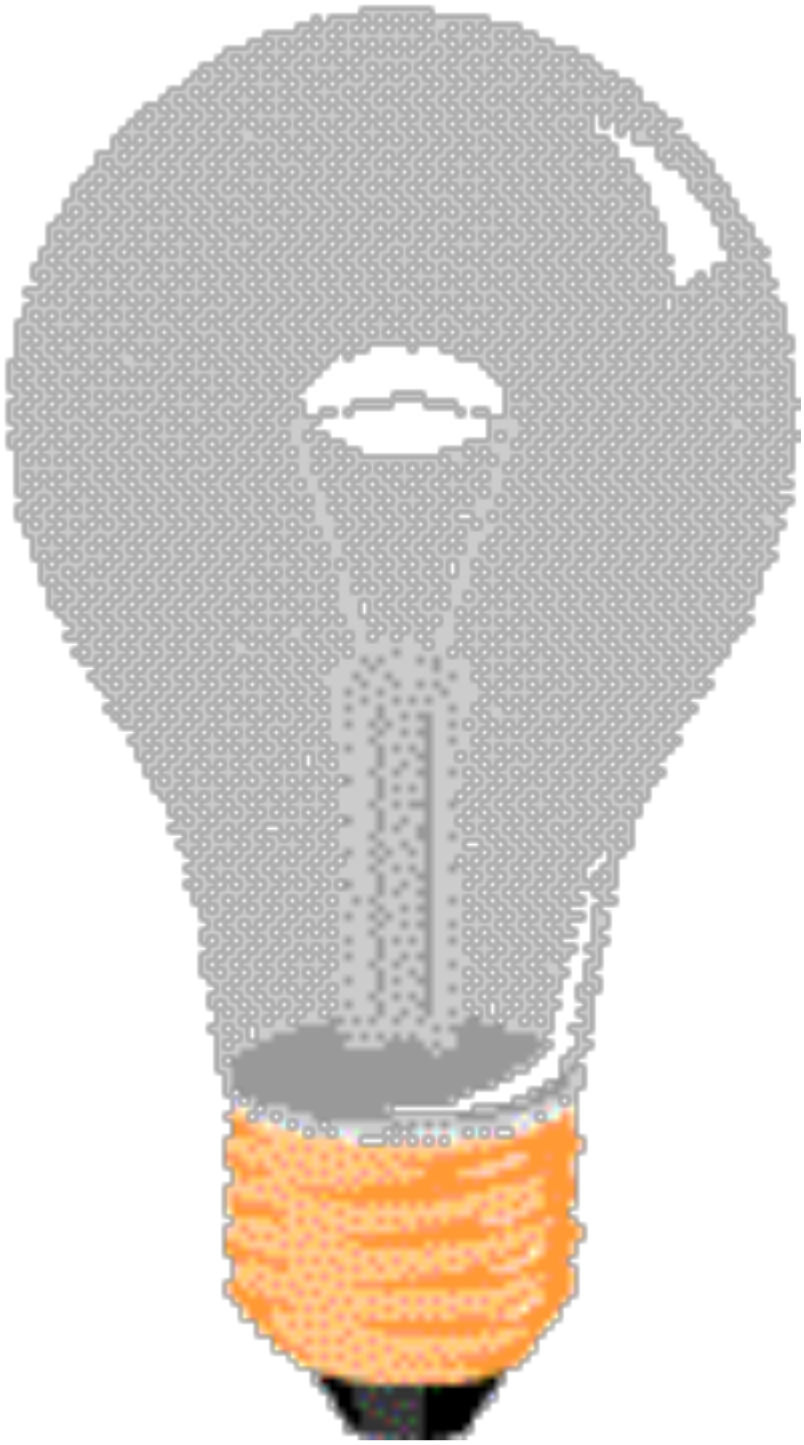
Vamos considerar uma aplicação que utilize JavaScript para alterar a imagem que é exibida em uma página web. A partir de dois botões, o usuário pode definir se acende ou apaga a luz de uma lâmpada. A aplicação mostrada a seguir é uma adaptação de uma contida no site do W3schools.

Considere que ilustramos uma página web com a imagem de uma lâmpada e dois botões: Acender e Apagar. Nesta imagem, a lâmpada está apagada e permanecerá assim até que o usuário clique no botão Acender.

O que o JavaScript pode fazer?

JavaScript pode alterar valores de atributos HTML.

Neste exemplo, o JavaScript altera o valor do atributo src (source) de uma imagem.



Quando o usuário clicar no botão ACENDER, a página web mudará conforme exposto a seguir.

O que o JavaScript pode fazer?

JavaScript pode alterar valores de atributos HTML.

Neste caso, o JavaScript altera o valor do atributo src (source) de uma imagem.

Agora, vamos analisar o código da página web, mostrado adiante.

```
javascript



        What Can JavaScript Do?


JavaScript can change HTML attribute values.


In this case JavaScript changes the value of the src (source) attribute of an image.


Turn on the light


Turn off the light
```

Inicialmente, vamos analisar a linha de código relacionada ao carregamento da imagem da lâmpada na página, conforme mostraremos agora.

```
plain-text
```

A tag tem alguns atributos que serão descritos a seguir:

- **id:** é o identificador (myImage) da imagem, que será utilizado pelo código JavaScript;
- **src:** se refere source ou origem da imagem. Neste caso, foi inserido o nome da imagem (pic_bulboff.gif);
- **sytle:** neste caso, está definindo a largura (width) da imagem para 100 pixels.

Agora, vamos analisar a linha de código relacionada ao clique no botão de Acender.

```
javascript
Acenda a luz
```

A tag <button> cria o botão e também utiliza o evento onclick, que define a ação que será realizada quando o usuário clicar no botão denominado Acender. Vamos analisar mais detalhadamente:

- **document.getElementById('myImage'):** vincula a ação do clique do botão ao identificador myImage, que se refere à imagem estudada anteriormente.

- **src:** altera o conteúdo de src do identificador myImage, ou seja, altera a figura que será exibida na tela.

- **pic_bulbon.gif:** é o nome do arquivo da figura que será carregado na tela.

De forma semelhante ao que acabamos de ver, quando o usuário clicar no botão Apagar, a figura será alterada novamente para pic_bulboff.gif, conforme mostraremos a seguir.

```
javascript

Apague a luz
```

Atividade 4

Considere que você trabalhe em um fábrica de software e um colega de trabalho está enfrentando problemas em um código de uma página web em que está trabalhando. Ele pede a sua ajuda objetivando resolver o problema. O código está mostrado a seguir. Qual é o problema e o que você faria para resolvê-lo?

```
javascript
```

What Can JavaScript Do?

```
JavaScript can change the style of an HTML element.

Click Me!
```

A Não há erro de código. Deve ser um problema do navegador.

B Trata-se de código CSS externo, mas não foi feita referência ao arquivo CSS. Portanto, para resolver o problema, basta referenciá-lo adequadamente.

C Trata-se de código CSS interno, mas não se utilizou a tag ⟨style⟩. Portanto, para resolver o problema, basta inserir a tag ⟨style⟩ no início da linha 9.

D O problema é que o código HTML está misturado com código JavaScript, o que não é possível. Para resolver o problema, basta criar dois arquivos separados, um para HTML e outro para JavaScript.

E A linha 9 utiliza código JavaScript que altera o tamanho da fonte para 35 pixels quando o botão for clicado. Entretanto, não foi especificado o evento onclick nesta linha de código. Portanto, para resolver o problema, deve-se substituir a linha 9 pelo seguinte código:

<button type="button" onclick="document.getElementById('demo').style.fontSize='35px'">Clicar</button>

A alternativa E está correta.

É possível alterar a parte comportalmental de páginas web com JavaScript, definindo ações que são realizadas a partir de eventos acionados pelo usuário, como clique no botão do mouse. No exercício abordado, trata-se de código HTML, contendo JavaScript, que altera o conteúdo da propriedade fontSize do seletor do parágrafo, que tem id igual a demo, para 35 pixels. Entretanto, não há o evento onclick na linha 9.

PHP: uma linguagem de programação server side

PHP é uma linguagem de programação que possibilita a criação de scripts que atuam no lado do servidor, assim como outras linguagens, como Java, Python, entre outras. Com PHP, podemos extrair dados provenientes de requisições de clientes, fazer integração com banco de dados, é multiparadigma e, dessa forma, é possível trabalhar com paradigma orientado a objetos, entre outros recursos. PHP é interpretada e, sendo assim, necessita de um servidor web para funcionar. Outro aspecto importante é que os scripts em PHP são convertidos para HTML. Então, os usuários veem apenas o código HTML gerado, não vendo o código PHP de origem.

O vídeo vai abordar a linguagem PHP e mostrar sua importância para permitir a integração com o banco de dados e possibilitar a criação de scripts que atuam do lado do servidor.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Uma das principais funções das linguagens de programação server side é permitir o acesso a informações armazenadas em bancos de dados. Uma vez que apenas utilizando HTML e JavaScript isso não é possível, faz-se necessária a utilização de uma linguagem no lado servidor. Entre as diversas linguagens disponíveis no lado servidor estão o Java, o Python, o ASP, o .NET e o PHP, que conheceremos um pouco mais ao longo deste estudo.

PHP

PHP (*Hypertext Preprocessor*) é uma linguagem de programação baseada em script, open source e destinada, sobretudo, ao desenvolvimento web. Trata-se de uma linguagem criada para ser simples, tendo nascida estruturada e, posteriormente, adotado o paradigma de orientação a objetos — apenas 10 anos depois da sua criação.

Saiba mais

O principal foco do PHP são os scripts do lado servidor, dando suporte a funções como coleta e processamento de dados oriundos de formulários HTML, geração de conteúdo dinâmico com o acesso a bancos de dados, entre outras. Além do foco nos scripts no lado servidor, é possível também utilizar o PHP por meio de scripts em linha de comando e na criação de aplicações desktop (com a utilização da extensão PHP-GTK), embora não seja a melhor linguagem para isso.

Como o PHP funciona

O PHP é uma **linguagem interpretada**, ou seja, ela precisa “rodar” sobre um servidor web. Com isso, todo o código gerado é interpretado pelo servidor, convertido em formato HTML e então exibido no navegador.

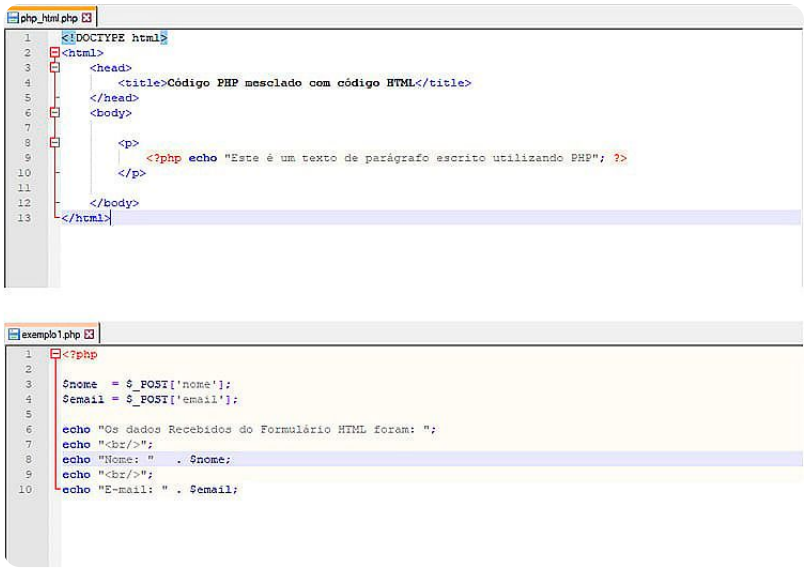
1. Etapa 1: O código PHP gerado é interpretado pelo servidor.
2. Etapa 2: Esse código é convertido em formato HTML.
3. Etapa 3: O código é exibido no navegador.

Logo, o código fonte não pode ser visto no lado cliente, mas apenas o HTML gerado.

Outra característica importante do PHP é poder ser utilizado na maior parte dos sistemas operacionais, assim como em vários servidores web diferentes, como o Apache, o IIS e o Nginx, entre outros.

Anatomia de um script PHP

Um **script PHP** é composto por código delimitado pelas tags . A última, de fechamento, não é obrigatória. Devido à sua simplicidade, um mesmo script PHP pode conter tanto código estruturado quanto orientado a objetos. Pode até conter código de marcação HTML. Neste último caso, o código próprio do PHP deverá ficar entre as tags de abertura e fechamento. A imagem a seguir mostra dois exemplos de código, um apenas em PHP e outro mesclado com HTML. Ao analisarmos os códigos, inicialmente é importante notar que ambos possuem a extensão ".php". Outras extensões possíveis, mas atualmente em desuso, são ".php3", ".php4", ".phtml".



Exemplos de código PHP.

No primeiro código da imagem, temos as tags de um arquivo HTML comum, com exceção do código inserido dentro das tags . Aqui temos a função "echo", que serve para imprimir algo na tela, associado a uma frase. Quando visualizado no navegador, o código será renderizado como HTML normal. Caso exibamos a fonte, só será possível ver as tags HTML e o conteúdo, sem o código PHP em questão.

Na segunda parte da imagem, temos um exemplo de código em que são definidas duas variáveis, $nome e $email, que recebem dois valores enviados de um formulário HTML, por meio do método POST. Daí a utilização do array superglobal $_POST — cujos índices 'nome' e 'email' correspondem ao atributo 'name' dos campos input do formulário. A seguir, é utilizada a função "echo" para a impressão de uma frase e do conteúdo das variáveis recebidas. Repare ainda na utilização, mais uma vez, de uma tag html, a
, em meio ao código PHP.

Sintaxe

Veja a seguir um resumo sobre a sintaxe do PHP.

Variáveis

No PHP, as variáveis são criadas com a utilização do símbolo de cifrão ($). Além disso, PHP é case sensitive, ou seja, sensível a letras maiúsculas e minúsculas, pois faz diferença quando utilizamos uma e outra.

Tipos de dados

O PHP é uma linguagem fracamente tipada. Logo, embora possua suporte para isto, não é necessário definir o tipo de uma variável em sua declaração. Os tipos de dados disponíveis em PHP são: booleanos, inteiros, números de ponto flutuante, strings, arrays, interáveis (*iterables*), objetos, recursos, NULL e call-backs.

Operadores condicionais

No PHP, há suporte às condicionais if, else, if e else ternários, if else e switch.

Laços de repetição

No PHP estão disponíveis os laços for, foreach, while e do-while.

Funções e métodos

O código PHP possui uma grande quantidade de funções e métodos nativos.

Inclusão de scripts dentro de scripts

O PHP permite a **inclusão de um script dentro de outro script**. Isso é muito útil, sobretudo se pensarmos no paradigma de orientação a objetos, em que temos, em um programa, diversas classes, codificadas em diferentes scripts. Logo, sempre que precisarmos fazer uso de uma dessas classes, de seus métodos ou atributos, basta incluí-la no script desejado. Para incluir um script em outro, o PHP disponibiliza algumas funções:

- Include
- Require
- Include once
- Require_once

Acesso ao sistema de arquivos

Por meio do PHP, é possível ter acesso ao sistema de arquivos do servidor web. Com isso, pode-se por exemplo, manipular arquivos e diretórios, desde a simples listagem à inclusão ou à exclusão de dados.

Atividade 1

Considere que você foi convidado para participar de um podcast sobre computação. Você foi questionado sobre PHP, quais são suas características etc. Você recebeu algumas dicas de colegas e precisa fazer uma análise crítica antes de usá-la no podcast. As dicas estão mostradas nas opções a seguir. Qual delas está correta, de modo que você possa usá-la no podcast?

A PHP é um framework de desenvolvimento de frontend que gera o código HTML automaticamente a partir do layout escolhido previamente. Portanto, atua no lado do cliente.

B PHP é um framework de desenvolvimento backend que possibilita a integração com banco de dados, ou seja, faz o denominado mapeamento objeto relacional entre a aplicação e o banco de dados.

C PHP é um framework de desenvolvimento de frontend que gera o código CSS automaticamente a partir do layout escolhido previamente. Portanto, atua no lado do cliente.

D PHP é uma linguagem multiparadigma, baseada em scripts, que atua no lado do servidor, possibilitando a extração de dados provenientes das requisições dos clientes, integração com bancos de dados, entre outros recursos.

E PHP é uma linguagem de frontend que atua na geração de códigos automatizados de segurança cibernética, prevenindo problemas como injeção de SQL, entre outros problemas.

A alternativa D está correta.

Diversas aplicações web que utilizamos demandam a persistência de dados em bancos de dados. Diante de situações como essa, é importante que as aplicações do servidor tenham recursos para extrair dados provenientes das requisições, como login e senha, por exemplo, possibilitando, inclusive, a integração com banco de dados. Nesse contexto, PHP oferece todas essas possibilidades.

Páginas dinâmicas e acesso a dados

Nos primórdios da web, as páginas eram essencialmente estáticas, ou seja, não havia a geração de conteúdo personalizado. Com o passar do tempo e a evolução tecnológica, surgiu a demanda por geração de conteúdo dinamicamente. Basta observar o acesso às redes sociais, plataformas de e-commerce, bankline, streaming etc. Inserimos nossos dados de login e senha e o servidor nos fornece os códigos (HTML, CSS e JavaScript) que são executados nos navegadores, mostrando os nossos dados na interface de usuário. Nesse tipo de abordagem, é fundamental o acesso a bancos de dados a partir de linguagens como PHP.

O vídeo vai mostrar a importância da linguagem PHP no contexto da evolução tecnológica, em que há necessidade de geração de conteúdo dinamicamente e com integração a bancos de dados.

Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Se fôssemos implementar em uma página web tudo o que estudamos até aqui, teríamos uma página **HTML básica**, com um pouco de interação no próprio navegador, graças ao JavaScript, e também com um pouco de estilo, este devido ao CSS. Além disso, já sabemos que é possível enviar dados do HTML para o PHP mediante um formulário. Para prosseguirmos, é importante definirmos o que são **páginas dinâmicas**. A melhor forma de fazer isso, porém, é definindo o que seria o seu antônimo, ou seja, as páginas estáticas.

> HTML + JavaScript + CSS, sem conexão com uma linguagem de programação, formam o que podemos chamar de páginas estáticas. Embora seja até possível termos um site inteiro composto por páginas estáticas, isso seria muito trabalhoso e também nada usual.

Mas e agora? Qual o próximo nível? O que fazer a seguir? A resposta para essas perguntas está no que abordaremos a seguir: **páginas dinâmicas** e **acesso a dados**.

Exemplo

Imagine um site que tenha dez páginas. Agora imagine que esse site tenha a mesma estrutura visual, o mesmo cabeçalho, menu, rodapé e outros pontos em comum. Pense em um blog, por exemplo, no qual o que muda são os conteúdos dos posts. No site estático, teríamos que escrever dez diferentes arquivos HTML, modificando o conteúdo em cada um deles, diretamente nas tags HTML, e só conseguiríamos reaproveitar os estilos e a interatividade de navegador utilizando CSS e JavaScript externos. Entretanto, todo o conteúdo precisaria ser digitado e muito código HTML repetido. Todo esse trabalho nos ajuda a entender o que são páginas estáticas.

Ainda utilizando o exemplo de um blog, imagine que você deseja receber comentários em seus posts, deseja que seus visitantes possam interagir com você e vice-versa. Como fazer isso? A resposta, como você já deve imaginar, é: **páginas dinâmicas**.

> A combinação das tecnologias do lado cliente com as tecnologias do lado servidor produzem as páginas dinâmicas.

Nas páginas dinâmicas, é possível receber as informações provenientes do cliente, processá-las, guardá-las, recuperá-las e utilizá-las sempre que desejarmos. E não é só isso: podemos guardar todo o conteúdo do nosso blog no banco de dados. Com isso, teríamos apenas uma página PHP que recuperaria nosso conteúdo no banco e o exibiria no navegador. A tabela a seguir apresenta um pequeno resumo comparativo entre as páginas estáticas e dinâmicas.

	Páginas estáticas	Páginas dinâmicas
Inclusão/Alteração/ Exclusão de conteúdo	Manualmente, direto no código HTML	Automaticamente através de scripts no lado servidor, como PHP
Armazenamento do conteúdo	Na própria página HTML	Em um banco de dados

Tabela: Comparativo entre páginas estáticas e dinâmicas.

Alexandre Paixão

Outra importante característica de um site dinâmico é possibilitar a utilização de ferramentas de gestão de conteúdo (CMS) para manter as informações do site sempre atualizadas. Com isso, depois de pronto, não será mais necessário mexer nos códigos-fonte, basta acessar a ferramenta e gerenciar o conteúdo por meio dela. Já no site estático, será preciso modificar diretamente o código HTML, tornando necessário alguém com conhecimento técnico para isso.

Acesso a dados

Como mencionado anteriormente, o ambiente web é composto por tecnologias que rodam do lado cliente e do lado servidor. Complementando o que vimos até aqui, temos ainda, do lado servidor, o **banco de dados**. De forma resumida, podemos defini-lo como um repositório em que diversas informações podem ser armazenadas e posteriormente recuperadas.

Para realizar a gestão desses dados, existem os **SGBD**, ou sistemas gerenciadores de bancos de dados. Se, por um lado, o SGBD é responsável por montar a estrutura do banco de dados — entre outras funções —, por outro lado, para recuperarmos uma informação guardada em um banco de dados e exibi-la em uma página

web, é necessário utilizar uma linguagem do lado servidor, como o PHP. Em outras palavras, não é possível acessar o banco de dados utilizando apenas HTML ou mesmo JavaScript. Sempre será necessária a utilização de uma linguagem server side para o acesso aos dados.

SGBD

Há diversos tipos de SGBD para as mais variadas necessidades, com opções gratuitas ou pagas. Entre os gratuitos, dois são comumente utilizados em conjunto com o PHP: MySQL e PosgreSQL.

Formas de acesso a dados

A partir das tecnologias vistas até aqui, há algumas formas de acessar os dados guardados em um banco de dados.

A partir do HTML

Uma das maneiras mais comuns de enviar e recuperar dados a partir do HTML é fazendo uso de formulários. Com eles, é possível submetermos nossos dados para uma linguagem no lado servidor/PHP. Este, então, recebe as informações e as armazena no banco de dados. Da mesma forma acontece o caminho inverso. Podemos ter um formulário em nossa página HTML que solicite dados ao PHP e este as envie de volta, após recuperá-las do banco de dados.
Vale lembrar ainda o que vimos sobre o PHP: ele permite a utilização de códigos HTML diretamente em seus scripts. Logo, uma página web feita em PHP pode recuperar dados do banco de dados toda vez que é carregada. É isso o que acontece na maioria dos sites. Cada página visualizada é composta por conteúdo armazenado em banco de dados e código HTML produzidos por uma linguagem do lado servidor. Com isso, cada página que abrimos em sites dinâmicos implica uma chamada/requisição ao lado servidor — script e banco de dados.

A partir do JavaScript

O JavaScript possui, essencialmente, duas formas para se comunicar com linguagens do lado servidor: por meio das APIs (*Application Programming Interface*) XMLHttpRequest e Fetch API. A primeira é amplamente utilizada, sendo a forma mais comum de realizar essa comunicação. É normalmente associada a uma técnica de desenvolvimento web chamada AJAX. Já a segunda é mais recente e oferece algumas melhorias, embora não seja suportada por todos os navegadores.
A comunicação em ambas consiste em, mediante algum evento no navegador, normalmente originado em uma ação disparada pelo usuário, enviar uma requisição ao lado servidor, como recuperar algum dado, por exemplo, tratar o seu retorno e o exibir na tela. Isso tudo sem que seja necessário recarregar toda a página.

Atividade 2

Considere que você foi designado para proferir uma palestra para estudantes que visitam a empresa de TI na qual você trabalha. Você deve falar sobre páginas web dinâmicas e acesso aos dados. Qual das opções a seguir você poderia utilizar na sua palestra? Marque a opção correta.

A Páginas web dinâmicas possibilitam a geração de conteúdo personalizado para cada usuário, por meio do acesso aos dados dos usuários, armazenados em bancos de dados.

B Páginas web dinâmicas são aquelas nas quais os usuários podem interagir por meio de botões, por exemplo. Ou seja, os usuários têm maior dinamismo no uso das aplicações.

C Páginas web dinâmicas é o termo para designar páginas web implementadas com PHP como linguagem de frontend, tendo em vista a facilidade e o dinamismo no desenvolvimento das interfaces de usuários.

D Páginas web dinâmicas foram usadas nos primórdios da web. Entretanto, caíram em desuso por conta da web 3.0.

E Páginas web dinâmicas são aquelas que têm vídeos, como as plataformas de streaming de filmes, por exemplo.

A alternativa A está correta.

Com a evolução tecnológica, tornou-se necessária a geração de conteúdo personalizado e dinamicamente, por meio de acesso a bancos de dados, sendo fundamental, portanto, utilizar linguagens de programação que possibilitem a integração de aplicações com bancos de dados.

Considerações finais

- Modelo cliente x servidor.
- Ambiente web.
- Conceito de interface.
- Conceito de design responsivo.
- HTML, CSS e JavaScript.
- Linguagem PHP.
- Páginas dinâmicas e acesso a dados.

Explore +

Confira as indicações que separamos especialmente para você!

Leia o livro **Design de interação: além da interação homem-computador**, escrito por Jennifer Preece, Yvonne Rogers e Helen Sharp, da editora John Wiley e Sons.

Leia o livro **Interação humano-computador**, escrito por Simone Diniz Junqueira Barboza e Bruno Santana da Silva, da editora Elsevier.

Visite o tópico **CSS Units** no site do W3C e conheça a lista completa das unidades de medidas da CSS. Conforme mencionado, a CSS possui uma série de unidades de medidas para expressar tamanho, sendo esse associado a propriedades como “width”, “margin”, entre outros. Estão disponíveis unidades de tamanho absoluto, como centímetros, milímetros, pixels etc. e também unidades de tamanho relativo, como “em”, “ex”, entre outros.

Assim como a maioria das tecnologias, a CSS possui um guia de referência oficial. Como esse guia é mantido pelo W3C, entre no site para acessá-lo e conhecer a sua especificação e todos os detalhes a ela relacionados.

Referências

BENYON, D. **Interação humano-computador**. São Paulo: Pearson Prentice Hall, 2011.

STAT COUNTER GLOBAL STATS. **Screen resolution stats Brazil**. S. d.

FRIEDMAN, V. **Responsive web design**: what it is and how to use it. Smashing Magazine, 11 ago. 2018. Consultado na internet em: 13 jun. 2022.

MOZILLA. **MDN web docs**: CSS Preprocessor. S. d.

PAIXÃO, A. **Notas de aula sobre ambiente web cliente X servidor e as tecnologias do professor alexandre paixão**. Disponível sob licença Creative Commons BR Atribuição – CC BY, 2020.

SILVA, M. S. **Web design responsivo**: aprenda a criar sites que se adaptam automaticamente a qualquer dispositivo, desde desktop até telefones celulares. São Paulo: Novatec, 2014.

WROBLEWSKI, L. **Mobile first**. S. d.

W3SCHOOLS. **CSS reference**. S. d.