



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.



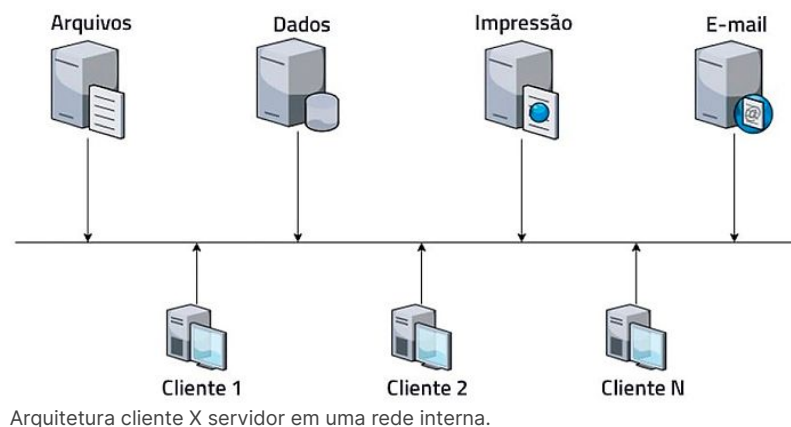
Conteúdo interativo

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.



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:



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

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.

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.

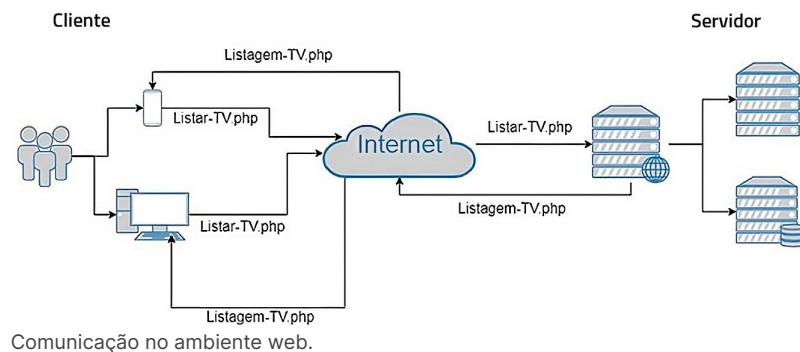
Servidor web

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.



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.



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 descryptografia 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.

Name:	<input type="text" value="José Oliveira"/>
E-mail:	<input type="text" value="joseoliveira@xpto.com.br"/>
<input type="button" value="Enviar"/>	

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 Moraes (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.

```
1 <style type="text/css">
2 @media (max-width: 360px)
3 {
4   .menu_lateral
5   {
6     display: none;
7   }
8 }
9 </style>
```

Exemplo de declaração de media query.

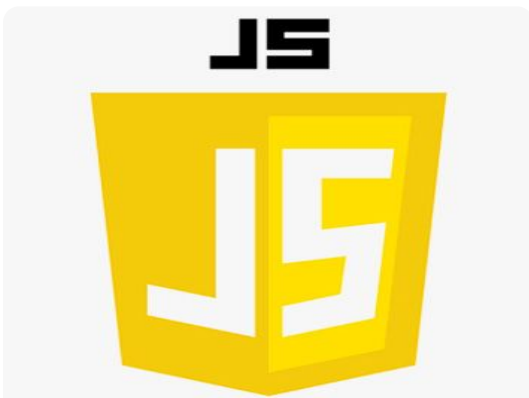
O resultado das expressões utilizadas na media query pode ser verdadeiro ou falso. No caso de nosso exemplo, será verdadeiro sempre que a largura da tela do dispositivo que visualiza a página seja inferior a 360px. Do contrário, será falso. Ou seja, para todos os dispositivos cuja largura de tela seja 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, no ambiente web, estamos falando de linguagens de programação que rodam no navegador e cujo exemplo mais comum é o **JavaScript**.

Essa linguagem adiciona interação a uma página web, permitindo, por exemplo, a atualização dinâmica de conteúdos, o controle de multimídia, a animação de imagens e muito mais. No contexto do design responsivo, sua faceta mais importante é a de atualização dinâmica de conteúdo — e não só do 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.

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.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Título da página</title>
5   </head>
6   <body>
7     <header>
8       Cabeçalho da página
9     </header>
10    <nav>
11      Barra de navegação
12    </nav>
13    <main>
14      Conteúdo da página
15    </main>
16    <aside>
17      Barra lateral
18    </aside>
19    <footer>
20      Rodapé
21    </footer>
22  </body>
23 </html>
```

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.



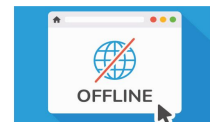
Melhorias de conectividade.



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

Atividade

4 oooooo Ooooooooooooo SS

Texto de parágrafo: Estamos estuda

body 1

Com o PC, há mais liberdade de estilo

(Nota: este formulário deve ser preenchido no GSS interno para facilitar a visualização).

and $\tilde{\sigma}_i = \sigma_i$ for $i \in \mathcal{I}$.

Página web resultante do código anterior.

O texto do parágrafo.
 Estamos estudando o
 uso de CSS em
 páginas web
 (contendo o código em left: 40px;
 e o código em uma
 página web).

margin-left: 20px; color: maroon; text-align: center; font-size: 1.2em; font-weight: bold; padding: 10px 0;">
 Título
 paraspágina
 pixelfont-size: 0.8em; text-align: justify;

Com o CSS, podemos alterar o estilo de páginas web. Podemos inserir os códigos CSS de diferentes maneiras, seguindo

A word cloud visualization of the text, with the most prominent words being "Consultoria", "em", "tecnologia", "de", "comput", and "em nuvem". Other visible words include "arquitetura", "soluções", "treinam", "computação", "nuvem", "CSS", "facilitar", "experiência", "problema", "você", "faria", "para", "serviços", "lo?", "consultoria", and "treinam". The words are arranged in a circular pattern, with the largest words at the top and bottom.

Segura

em
compu
em nu

O código não tem
segurança em computação em
nuvem, contemplam os recursos
tecnológicos mais avançados do
mercado.

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.

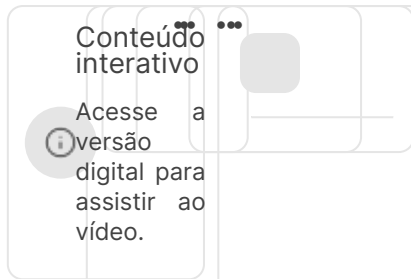
Conforme foi mencionado na resposta, trata-se da inserção de código CSS internamente na página HTML, que deve ocorrer com o uso da tag `<style>` no início do código CSS e com `</style>`, indicando o fechamento das linhas de código CSS, delimitando as linhas de

Inserindo JavaScript em código HTML

Código de CSS
separando o
código de estilo do
código HTML.
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, o usuário não precisa mais lidar com o problema do navegador.

acontece quando o usuário não clica no botão ao HTML formata o conteúdo da página para a

fundamentalmente, a linguagem de programação

que os desenvolvedores usam para criar o código CSS externo, mas não foi feita referência ao arquivo estudado, o que pode ser resolvido inserindo o seguinte código no arquivo CSS externo.

realizar a implementação de um sistema de

atividades (como a de trabalhar com

práticas de ensino e aprendizagem, o

consolidando a aprendizagem e a

conhecimento e a prática de programação CSS interno, mas não se utilizou a tag (style). Portanto, de qualquer forma, para resolver o problema, basta inserir a tag (style) no início da linha 9.

exploração, mostrando a importância

o não é, a importância de uma página

desse ponto de vista, a importância de

que, a documentação de uma página HTML está misturada com código JavaScript, o

combinação de HTML e JavaScript é possível. Para resolver o problema, basta criar dois arquivos

possíveis de usar. Eles podem ser para HTML e outro para JavaScript.

criação de uma página web, o usuário

páginas de uma página web, o usuário

interativo, ou seja, a resolução de um problema que altera o tamanho da fonte para 35

eficiente, a resolução de um problema que altera o tamanho da fonte para 35

com a utilização de um botão. Entretanto, não foi especificado o

usabilidade de uma página web, o usuário

A aplicação de uma página web, o usuário

mostrada a seguir, é de dados, é

a seguir é uma página web, o usuário

adaptação de uma página web, o usuário

contida no site do W3schools.

A alternativa está correta.

É possível alterar a parte comportamental 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, um serviço web para funcionar.

Outro

Conteúdo¹ interativo

Acesse a versão digital para assistir ao vídeo.

[illegible]

que vai se tornando cada vez mais comum a utilização de scripts no lado do servidor, e também a utilização de scripts no lado do cliente. Além disso, a linguagem PHP é uma das mais utilizadas para a criação de páginas web. Neste último caso, é possível utilizar o PHP para a criação de páginas web, e também para a criação de scripts no lado do cliente. O PHP é uma linguagem de programação que pode ser utilizada para a criação de páginas web, e também para a criação de scripts no lado do cliente. O PHP é uma linguagem de programação que pode ser utilizada para a criação de páginas web, e também para a criação de scripts no lado do cliente.

um pouco mais ao longo deste extenso período.

GTK), embora não seja um melhor apenas em linguagem para PHP e outro isso mesclado.

SO mesclado
com HTML.

Ao
 analisarmos
 os códigos,
 inicialmente

é importante

notar primeiro

ambos códigos de

possuem imagem,

extensões as

“.php”. tags de um

Outras arquivos

extensão HTML

possível comum,

mas com

atualmente execução

em do código

são “.php4”

dentro das

“.phtml” tags. Aqui

temos a

função

“echo”

Exemplos de código PHP.

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

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Código PHP mesclado com código HTML</title>
5 </head>
6 <body>
7
8 <?php echo "Este é um texto de parágrafo escrito utilizando PHP"; ?>
9
10 </body>
11 </html>
```

```
1 <?php
2
3 $nome = $_POST['nome'];
4 $email = $_POST['email'];
5
6 echo "Os dados Recebidos do Formulário HTML foram: ";
7
8 echo "Nome: " . $nome;
9
10 echo "E-mail: " . $email;
```


PHPNa em
questão
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

Você já viu esse
cartãozinho
No FIME, os
vendedores IP
são os melhores
do Brasil. Mais
utilizam o melhor
do sistema de
descontos e
descontos de
descontos de
descontos de
descontos de

[illegible]

uma. Você precisa
são. Para fazer uma análise
booleana antes de
interrogá-la no podcast.
números lógicos estão
em movimento nas
flutuantes
temos opções a seguir.
strings
um e Qual delas está
arrays
programática, de modo
interáveis
diversas de você possa
(iterables)
classes usadas no podcast?
classes, e
reconhecem
em
NULL a inclusão
diferentes à
call backs. A

Logo, de dados. PHP é um sempre que framework de precisarmos desenvolvimento fazer uso de uma de frontend que dessas gera o código classes, de HTML seus automaticamente a métodos ou partir do layout atributos, escolhido basta incluir previamente. la no script Portanto, atua no desejado lado do cliente.

Para incluir um script em outro, o PHP é um framework de desenvolvimento algumas funções: backend que possibilita a integração com

Atividade

7.

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

Páginas dinâmicas e acesso a dados

Nos primórdios da web, de as páginas eram essencialmente estáticas, ou seja, não havia injeção de conteúdo, de outros problemas personalizados.

Com o passar do tempo e a evolução tecnológica, surgiu a demanda por uma alternativa de geração de conteúdo está correta.

Diversas dinamicamente. Basta observar as aplicações web que utilizamos, como redes sociais, plataformas de e-commerce, bancos online, streaming, etc. Inserimos nossos dados nas aplicações de login e senha e o servidor nos fornece os códigos (HTML, CSS e JavaScript) que são executados a integração com o banco de dados. Nesse contexto, PHP oferece todas as possibilidades. Nesse tipo de abordagem, é fundamental o acesso a bancos de



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

dados a partir de fontes HTML? +
de linguagem complementar? +
como PHP? importância próxima
da linguagem que fazer a
PHP que nos dá um exemplo de
contexto de resposta para
evolução, teremos um
tecnológica, mas imagine o
em HTML básica de páginas.
necessidade de páginas
de geração de conteúdo que esse site
de conteúdo para a
dinâmica de páginas seja
e graças a isso, a mesma
integradas ao mesmo
bandas também composto
dados um pouco por páginas
estilo, este seria
devido ao CSS pontos em
Além disso, muito
sabemos que é um
possível enviar, por
dados do exemplo, no
HTML para o qual o que
PHP mediante os
um formulário. conteúdos
Para dos posts. No
prossequirmos site estático,
é importante teríamos que
definirmos escrever dez
que são diferentes
páginas HTML, arquivos
dinâmicas. A modificando o
melhor forma conteúdo em
de fazer isso, um
porém, eles, um
definindo diretamente
que seria o seu tags
antônimo, ou HTML, e só
seja, as conseguiríamos
páginas reaproveitar
estáticas. os estilos e a
interatividade
de navegador
utilizando CSS
e JavaScript
externos.
Entretanto,

Até então, as páginas eram criadas manualmente, com o uso de ferramentas de edição de texto.

Com o advento das tecnologias de desenvolvimento web, a criação de páginas dinâmicas tornou-se possível. Isso significa que as páginas podem ser geradas automaticamente, com base em dados armazenados em um banco de dados.

Essa abordagem oferece várias vantagens, como a possibilidade de atualizar o conteúdo das páginas sem a necessidade de intervenção manual.

Além disso, as páginas dinâmicas permitem a personalização do conteúdo com base no perfil de cada usuário, melhorando a experiência do usuário.

Por fim, a utilização de bancos de dados facilita a manutenção e a escalabilidade do sistema, permitindo o armazenamento de grandes volumes de dados.

Em resumo, a criação de páginas dinâmicas é uma prática essencial para o desenvolvimento de sistemas web modernos e eficientes.

Essa abordagem não apenas otimiza o processo de desenvolvimento, mas também garante a flexibilidade necessária para lidar com mudanças e crescimento.

Portanto, a adoção de páginas dinâmicas é um passo fundamental para qualquer projeto web que vise a oferecer uma experiência de usuário de alta qualidade.

Assim, a combinação de tecnologias de desenvolvimento web e bancos de dados resulta em soluções web mais robustas e escaláveis.

Essa abordagem é amplamente utilizada em diversos setores, desde e-commerce até sistemas corporativos.

Em conclusão, a criação de páginas dinâmicas é uma prática essencial para o sucesso de qualquer projeto web moderno.

Essa abordagem não apenas otimiza o processo de desenvolvimento, mas também garante a flexibilidade necessária para lidar com mudanças e crescimento.

Portanto, a adoção de páginas dinâmicas é um passo fundamental para qualquer projeto web que vise a oferecer uma experiência de usuário de alta qualidade.

Assim, a combinação de tecnologias de desenvolvimento web e bancos de dados resulta em soluções web mais robustas e escaláveis.

Essa abordagem é amplamente utilizada em diversos setores, desde e-commerce até sistemas corporativos.

Em conclusão, a criação de páginas dinâmicas é uma prática essencial para o sucesso de qualquer projeto web moderno.

Essa abordagem não apenas otimiza o processo de desenvolvimento, mas também garante a flexibilidade necessária para lidar com mudanças e crescimento.

Portanto, a adoção de páginas dinâmicas é um passo fundamental para qualquer projeto web que vise a oferecer uma experiência de usuário de alta qualidade.

Assim, a combinação de tecnologias de desenvolvimento web e bancos de dados resulta em soluções web mais robustas e escaláveis.

Essa abordagem é amplamente utilizada em diversos setores, desde e-commerce até sistemas corporativos.

Em conclusão, a criação de páginas dinâmicas é uma prática essencial para o sucesso de qualquer projeto web moderno.

Até então, as páginas eram criadas manualmente, com o uso de ferramentas de edição de texto.

Com o advento das tecnologias de desenvolvimento web, a criação de páginas dinâmicas tornou-se possível. Isso significa que as páginas podem ser geradas automaticamente, com base em dados armazenados em um banco de dados.

Essa abordagem oferece várias vantagens, como a possibilidade de atualizar o conteúdo das páginas sem a necessidade de intervenção manual.

Além disso, as páginas dinâmicas permitem a personalização do conteúdo com base no perfil de cada usuário, melhorando a experiência do usuário.

Por fim, a utilização de bancos de dados facilita a manutenção e a escalabilidade do sistema, permitindo o armazenamento de grandes volumes de dados.

Em resumo, a criação de páginas dinâmicas é uma prática essencial para o desenvolvimento de sistemas web modernos e eficientes.

Essa abordagem não apenas otimiza o processo de desenvolvimento, mas também garante a flexibilidade necessária para lidar com mudanças e crescimento.

Portanto, a adoção de páginas dinâmicas é um passo fundamental para qualquer projeto web que vise a oferecer uma experiência de usuário de alta qualidade.

Assim, a combinação de tecnologias de desenvolvimento web e bancos de dados resulta em soluções web mais robustas e escaláveis.

Essa abordagem é amplamente utilizada em diversos setores, desde e-commerce até sistemas corporativos.

Em conclusão, a criação de páginas dinâmicas é uma prática essencial para o sucesso de qualquer projeto web moderno.

Essa abordagem não apenas otimiza o processo de desenvolvimento, mas também garante a flexibilidade necessária para lidar com mudanças e crescimento.

Portanto, a adoção de páginas dinâmicas é um passo fundamental para qualquer projeto web que vise a oferecer uma experiência de usuário de alta qualidade.

Assim, a combinação de tecnologias de desenvolvimento web e bancos de dados resulta em soluções web mais robustas e escaláveis.

Essa abordagem é amplamente utilizada em diversos setores, desde e-commerce até sistemas corporativos.

Em conclusão, a criação de páginas dinâmicas é uma prática essencial para o sucesso de qualquer projeto web moderno.

Essa abordagem não apenas otimiza o processo de desenvolvimento, mas também garante a flexibilidade necessária para lidar com mudanças e crescimento.

Portanto, a adoção de páginas dinâmicas é um passo fundamental para qualquer projeto web que vise a oferecer uma experiência de usuário de alta qualidade.

Assim, a combinação de tecnologias de desenvolvimento web e bancos de dados resulta em soluções web mais robustas e escaláveis.

Essa abordagem é amplamente utilizada em diversos setores, desde e-commerce até sistemas corporativos.

Em conclusão, a criação de páginas dinâmicas é uma prática essencial para o sucesso de qualquer projeto web moderno.



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, o **banco de dados**.

De forma resumida, podemos defini-lo como um repositório em que diversas informações podem ser armazenadas e posteriormente recuperadas.

Essas informações são gerenciadas por bancos de dados. Se, por um lado, o SGBD é responsável por montar a estrutura do banco de dados — já que, entre outras funções, precisa, por outro lado, para recuperarmos uma informação guardada em um banco de dados e exibí-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.

Formas de acesso aos dados

A partir do
SQL HTML
tecnologias
vistas até
agora, há
SQL para as
mais variadas
formas de
necessidades,
acessar os
dados, com
opções
gratuitas ou
guardados
pagas. Entre
em um
banco de
dados, são
comumente
utilizados em
conjunto com
o PHP:
MySQL e
PostgreSQL.

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

Atividade

2

Considere que você
feita em
PHP pode
recuperar
dados do
banco de
dados da empresa
vez que a
carregada. Você deve
isso comunicando com
linguagens do
lado do servidor.
maior acesso aos dados.
sites. O meio das
página seguir, você poderia
visualizada, na sua
é composta. Marque a
por opção correta. A
conteúdo.
armazenado é
em amplamente
utilizada, sendo
de dados e
código forma mais
comum de
HTML páginas web
produzidos. É
por comunicação. É
possível a
linguagem a uma de
do lado de
servidor.
Com isso, para
web, chamadas
cada página por
que abrimos acesso aos
em segundos usuários,
dinâmicos e em
implantação de dados.
chamada,
requisição não seja
ao suportada por
servidor, os
script dinâmicos são
banco de dados.
consistência, podem
interagir algum
evento, no por
navegador. Ou seja,
normalmente têm
origem em
uma ação no
uso das aplicações.
disparada pelo
usuário, enviar
uma requisição
ao lado servidor,
páginas web
como recuperar
dinâmicas e o termo
algum dado, por
exemplo, tratar o
seu retorno e o web
implementadas com
PHP.
Isso tudo sem
linguagem seja de
processado tendo em
visar a facilidade e
a página 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.

Conclusão: Explore o tópico

Design Interativo no site da maioria das interfaces humanas e comece a tecnologia humana responsável.

sempre em um computador completa da CSS possui um guia de unidades de

especificação de unidades de CSS. referência Pearson

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

• Conceito de interface

CSS Assim como CSS

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências

Referências