O ambiente internet: cliente X servidor e as tecnologias

Prof. Alexandre de Oliveira Paixão

Prof. Kleber de Aguiar

Descrição

O ambiente web e seu modelo cliente X servidor, interfaces com design responsivo (abordagem mobile first) e adaptativo, tecnologias do lado cliente (HTML, CSS, páginas estáticas com JavaScript) e do lado servidor (páginas dinâmicas com PHP, acesso a banco de dados).

Propósito

Compreender a composição do ambiente web e seus componentes dos lados cliente e servidor, assim como as tecnologias inerentes a cada componente, é essencial para a formação do profissional de desenvolvimento de sistemas na web.

Objetivos

Módulo 1

O ambiente web

Reconhecer o ambiente web.

Módulo 2

O conceito de interface

Descrever o conceito de interface.

Módulo 3

Tecnologias do lado cliente

Reconhecer as tecnologias do lado cliente.

Módulo 4

Tecnologias do lado servidor

Reconhecer as tecnologias do lado servidor.

Introdução

O chamado modelo cliente X servidor, com origem na computação, é uma estrutura de aplicação distribuída em que temos tarefas partilhadas e executadas entre as duas camadas: de um lado, a origem das requisições e solicitações de recursos ou serviços — o lado cliente — e, do outro, processos sendo executados a fim de prover tais recursos ou serviços — o lado servidor. Atualmente, tal modelo é amplamente utilizado, sendo base do ambiente web e de suas aplicações.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

1 - O ambiente web

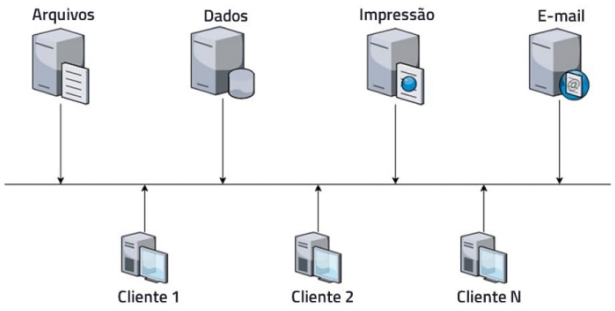
Ao final deste módulo, você será capaz de reconhecer o ambiente web.

Modelo cliente X servidor

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.

Modelo de três 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.

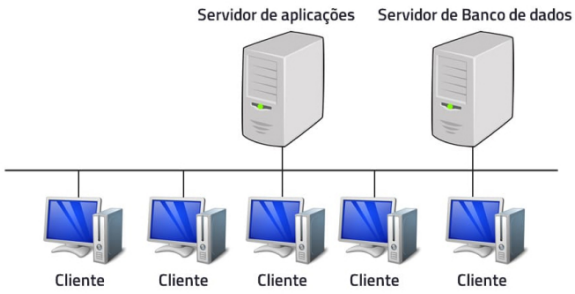
Camada de aplicação

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.

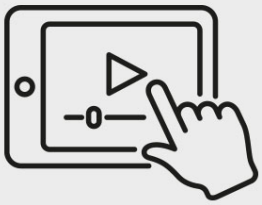
Veja um exemplo do modelo de três camadas:



Modelo de três camadas.

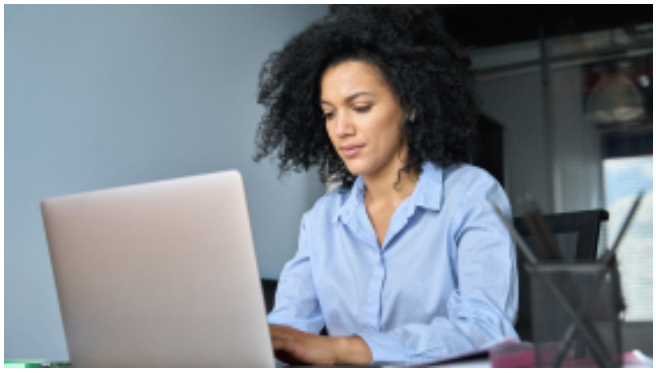
Modelo de 3 camadas

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Modelo de quatro 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, a divisão de responsabilidades ficou desta forma:



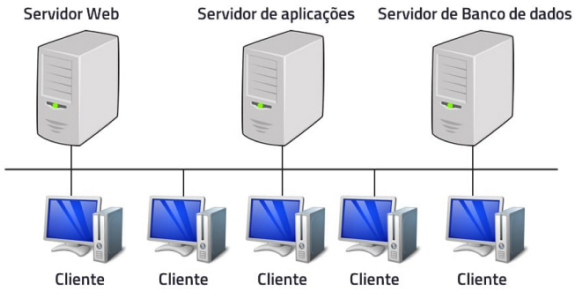
Cliente

Passou a precisar apenas de um navegador para ter acesso à aplicação.

Servidor

Composto por três servidores — o de aplicações, o de dados e o web, sendo este último o responsável pela apresentação/interface com o usuário cliente.

Veja um exemplo do modelo de quatro camadas:



Modelo de quatro camadas.

Modelo de 4 camadas

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

Ambiente web

Como vimos, inicialmente as aplicações ficavam hospedadas dentro de uma rede interna, onde estavam tanto os clientes quanto os servidores. Posteriormente, elas migraram para a internet, surgindo então 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.

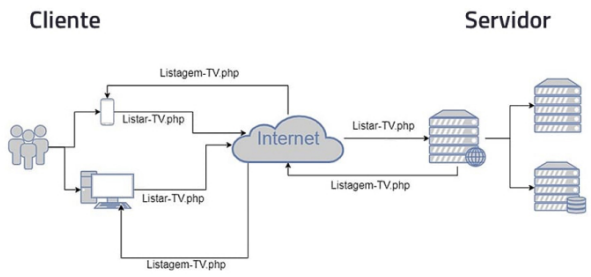
Servidor web

Sua estrutura pode comportar tanto as camadas de apresentação, aplicação e dados numa única máquina quanto em diversas máquinas, sendo essa distribuição indistinguível para o cliente.

Quando falamos de comunicação, estamos falando, 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 de um lado uma requisição sendo feita pelo cliente e, do outro, o servidor processando a requisição e respondendo ao cliente com o que foi solicitado.

Ambiente cliente X servidor

Entenda os principais conceitos sobre o fluxo do usuário requisitante.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

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 *(request)* e resposta *(response)*.

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**.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Em relação à responsabilidade de realizar as requisições no modelo cliente X servidor, assinale a alternativa correta:

A Uma das principais vantagens desse modelo é permitir a separação de responsabilidades. Com isso, caberá sempre e unicamente ao cliente realizar as requisições de serviços e/ou recursos, tendo o lado servidor um caráter sempre passivo.

B Embora inicialmente limitado a redes internas, com o surgimento da internet o modelo cliente X servidor evoluiu, tornando-se um modelo híbrido e bastante flexível, separado em N camadas, em que tanto o cliente quanto o servidor podem exercer as mesmas funções, ou seja, ambos podem requisitar e responder a solicitações.

C Para diminuir o custo de processamento no lado servidor, um cliente poderá solicitar a outros clientes recursos ou serviços já utilizados por eles. Isso é possível graças ao suporte fornecido por esse modelo à comunicação cliente X cliente.

D Nesse modelo, o lado cliente é, normalmente, o responsável por iniciar a comunicação por meio da realização de requisições ao lado servidor. Entretanto, o lado servidor

também é capaz de iniciar a comunicação, disparando notificações ou enviando mensagens para o lado cliente, por exemplo.

E Serviços PUSH são serviços que disparam notificações/mensagens para a solicitação de estabelecimento do processo de comunicação no ambiente web. Esse tipo de serviço é feito pelo cliente ao servidor.

Parabéns! A alternativa D está correta.

Normalmente é o cliente que inicia a comunicação no modelo cliente X servidor. Entretanto, o lado servidor também é capaz de realizar essa tarefa.

Questão 2

Nós vimos que o modelo cliente X servidor é a base do ambiente web. Assinale a opção que descreve corretamente o ambiente web:

A O ambiente web é composto por diversos clientes e diversos servidores. Nesse cenário, os clientes utilizam a internet e fazem requisições a diferentes servidores, localizados em diferentes partes do mundo. Os servidores então processam a requisição e devolvem a informação requisitada ou executam o serviço solicitado pelo cliente.

B No ambiente web, diferentemente do que acontecia nos primeiros modelos de camadas, há um modelo centralizado. Logo, todas as requisições são feitas a um único servidor, que as distribui para outros servidores e depois envia as respostas para os clientes.

C O avanço da tecnologia e o suporte oferecido pela internet permitiram uma importante mudança no ambiente web em relação aos modelos tradicionais de camadas. Com isso, nesse ambiente, o lado cliente tem as principais responsabilidades, incluindo manter no navegador ou em aplicativos mobile toda a lógica do negócio, facilitando assim o trabalho de processamento pelo lado servidor e agilizando a comunicação.

O ambiente web é caracterizado, sobretudo, pela transparência, diferentemente do que era visto inicialmente no modelo cliente X servidor. Com isso, um cliente sempre terá

D

controle total sobre o processo de comunicação por trás da requisição. Ele terá ciência, por exemplo, de onde se encontra o servidor ou servidores encarregados de receber e processar a sua requisição. Isso permite, por exemplo, que ele cancele a requisição a qualquer momento, caso o servidor encarregado de processá-la fique muito distante de onde ele se encontra.

E

No ambiente web, os clientes têm a possibilidade de fazer suas solicitações a qualquer um dos diversos servidores existentes no mundo, ou podem requisitar a informação ou serviço desejado diretamente a outro cliente, sem necessidade de acessar algum servidor, contribuindo a assim para um melhor fluxo de dados na internet.

Parabéns! A alternativa A está correta.

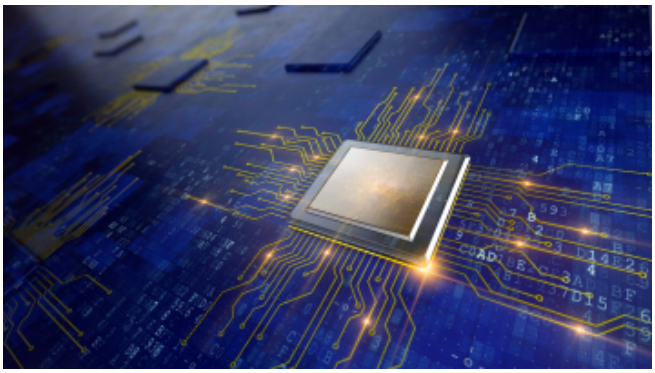
O ambiente web tem como base o modelo cliente X servidor e a evolução de seu modelo de camadas. Faz uso, portanto, de um modelo de N camadas, em que a lógica da aplicação e os dados são distribuídos em um ou mais servidores e a interface para acesso a esses servidores fica a cargo do cliente.

2 - O conceito de interface

Ao final deste módulo, você será capaz de compreender o conceito de interface.

Visão geral

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.

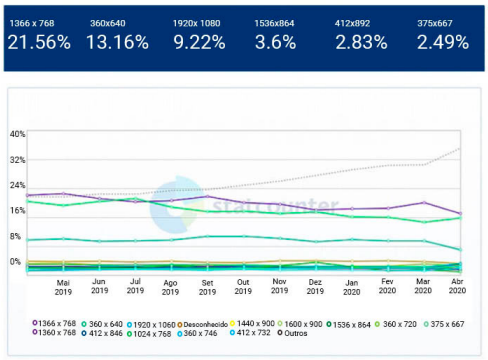
O site **StatCounter Global Stats** mantém ativa uma série de dados e estatísticas sobre dispositivos, tamanhos de tela, além de outras informações relacionadas. Sobre o tamanho de telas, e considerando o período de abril de 2019 a abril de 2020, temos os seguintes dados:

Tamanho da Tela em Pixels (Largura x Altura)	Percentual de Utilização
360 x 640	10,11%
1366 x 768	9,69%
1920 x 1080	8,4%
375 x 667	4,24%
414 x 896	3,62%

Tamanho da Tela em Pixels (Largura x Altura)	Percentual de Utilização
1536 x 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.

O conceito do design responsivo

Design responsivo

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.

Comentário

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.

Layouts fluidos

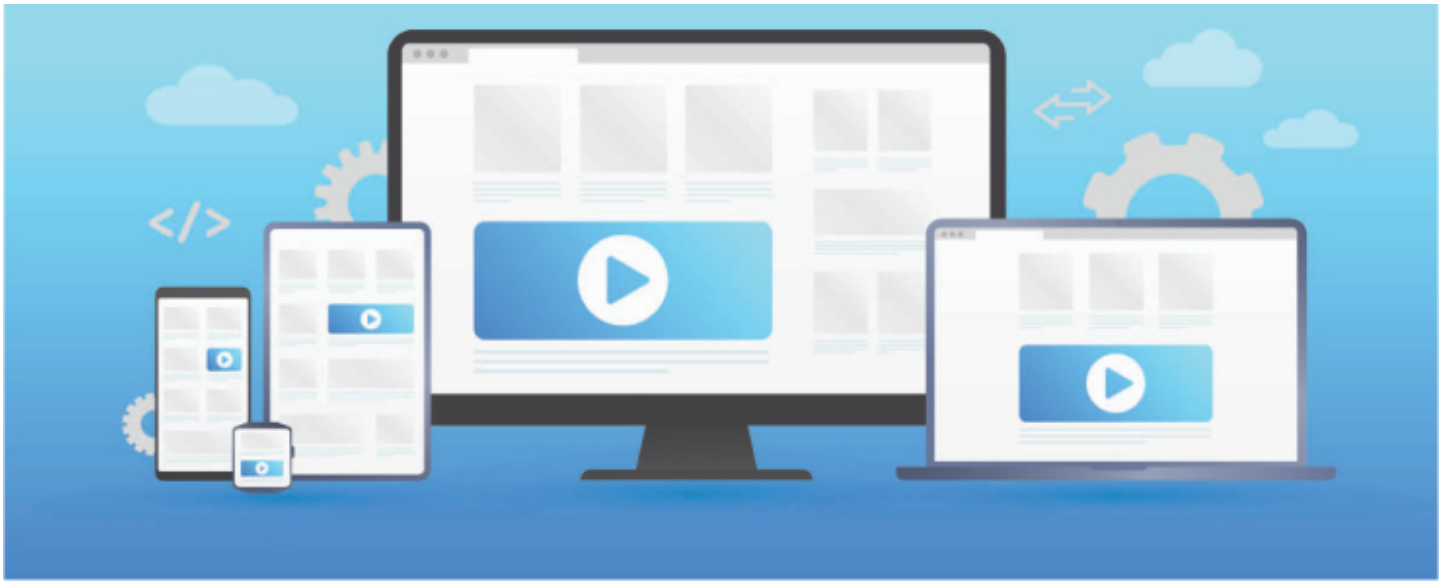
Para entender o conceito de layout fluido, é necessário entender primeiro o que seria o seu oposto, ou seja, o layout fixo.

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

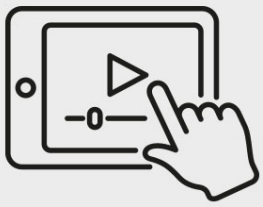
Enquanto o EM está relacionado ao tamanho do elemento de contexto (ou seja, definimos o valor EM de um elemento tomando como base o seu elemento pai), no REM definimos que o elemento de contexto, o elemento pai, será sempre a tag HTML <body>. Daí a letra “R” nessa unidade, que faz referência à raiz (root).

Saiba mais

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

Layouts fluidos

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Media query

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

Nesse contexto, 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.

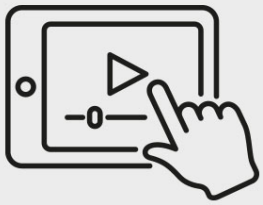
O resultado das expressões utilizadas na media query pode ser verdadeiro ou falso. No caso acima, será verdadeiro sempre que a largura da tela do dispositivo que visualiza a página seja inferior a 360px. Do contrário, será falso. Ou seja, para todos os dispositivos cuja largura de tela seja superior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

Atenção!

Outras expressões podem ser utilizadas, além da demonstrada acima, 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.

Media query

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.

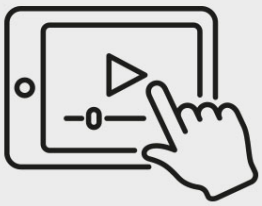
Resumindo

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

Design responsivo e design adaptativo

Assista ao vídeo e entenda mais sobre a diferença entre design responsivo e design adaptativo.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.

Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Assinale a alternativa que não corresponde ao conceito de interface:

A — A interface tem como objetivo proporcionar uma comunicação mais natural entre usuário e sistema computacional.

B — A interface é o meio pelo qual interagimos com um software, com uma aplicação, permitindo o acesso às opções e informações disponíveis.

C É o nome dado à parte de um sistema com a qual o usuário mantém contato ao usá-lo.

D A interface é a disciplina responsável pelo layout no desenvolvimento de software. Um dos seus princípios é garantir a criação de telas mais bonitas, que chamem a atenção de quem utiliza um software ou aplicativo.

E O conceito de interface está relacionado ao estudo da interação entre pessoas e computadores, também chamada de Interação Humano-Computador (IHC).

Parabéns! A alternativa D está correta.

O objetivo da interface vai além de aspectos como definir o que seria mais ou menos bonito. Seu cerne está em garantir que, sobretudo, haja uma comunicação mais natural e intuitiva entre usuário e sistema computacional.

Questão 2

Em relação ao design responsivo, assinale a opção corresponde à melhor ação a ser tomada para sua aplicação:

A Estudar os dados provenientes das visitas ou, na ausência destes, os relacionados às pesquisas de comportamento de acesso a websites para planejar a construção ou remodelação de um site, a fim de garantir que ele se adapte às características dos dispositivos que o acessam.

B Construir um site a partir de seis ou mais layouts fixos predefinidos.

C Escolher uma das três técnicas possíveis, preferencialmente o JavaScript, uma vez que sua implementação é mais simples, além de ser mais completo que as demais técnicas.

D Aplicar simultaneamente as técnicas de design responsivo e adaptativo.

E Para obter um layout que seja capaz de se adaptar a vários tamanhos de tela, define-se uma única combinação de media query.

Parabéns! A alternativa A está correta.

Como visto neste estudo, para aplicar o design responsivo, devemos fazer uso de uma combinação de técnicas a fim de garantir que uma página corresponda às preferências e características dos seus usuários com base no tamanho da tela, na plataforma e na orientação dos dispositivos por eles utilizados.

3 - Tecnologias do lado cliente

Ao final deste módulo, você será capaz de reconhecer as tecnologias do lado cliente.

As tecnologias HTML

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.

```
1  <!DOCTYPE html>
2
3  <html>
4
5      <head>
6
7      </head>
8
9      <body>
10
11     </body>
12
13 </html>
```

Estrutura obrigatória de uma página web.

De conteúdo

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

```
<!DOCTYPE html>

<html>

    <head>
        <title>Título da página</title>
    </head>

    <body>
        <h1>Minha Página</h1>
        <p><b>Conteúdo da página</b></p>
        <img src="imagens/imagem1.jpg" alt="imagem 1 da minha página">
        <hr/>

        <h2>Outros conteúdos</h2>
        <p><i>Conteúdo da página</i></p>
        <hr/>

        <ol><!--lista ordenada-->
                <li>item 1</li>
                <li>item 2</li>
                <li>item 3</li>
        </ol>

        <form method="post">
            <input type="submit" name="enviar" value="Enviar">
        </form>

    </body>
</html>
```

Tags de conteúdo de uma página web.

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:

```
<!DOCTYPE html>

<html>

    <head>
        <title>Título da Página</title>
    </head>

    <body>

        <header>
            Cabeçalho da Página
        </header>

        <nav>
            Barra de Navegador
        </nav>

        <main>
            Conteúdo da Página

            <aside>
                Barra lateral
            </aside>
        </main>

        <footer>
            Rodapé
        </footer>

    </body>

</html>
```

Tags estruturais básicas de uma página web.

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.

Saiba mais

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

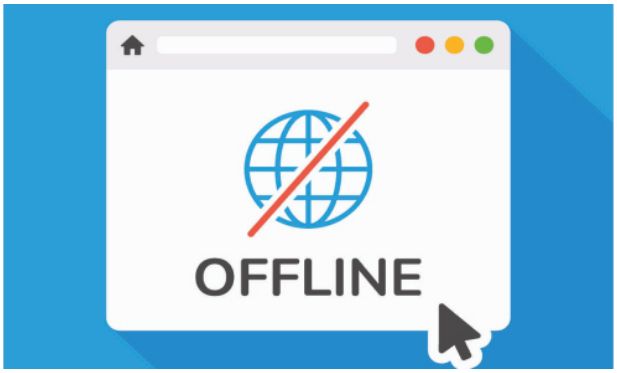
HTML5

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 off-line.

Suporte estendido a multimídia — áudio e vídeo.

Outras tecnologias: CSS e JavaScript

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

Uma cuida do conteúdo, da estruturação; a outra 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
}
```

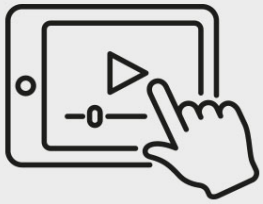
Exemplo de aplicação de CSS.

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

Tecnologias lado cliente: HTML5, CSS e JavaScript

Conheça, agora, as diferenças entre uma estrutura em HTML e em folhas de estilo.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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 <style>, dentro da tag <head> no documento.

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 <link> dentro da tag <head>.

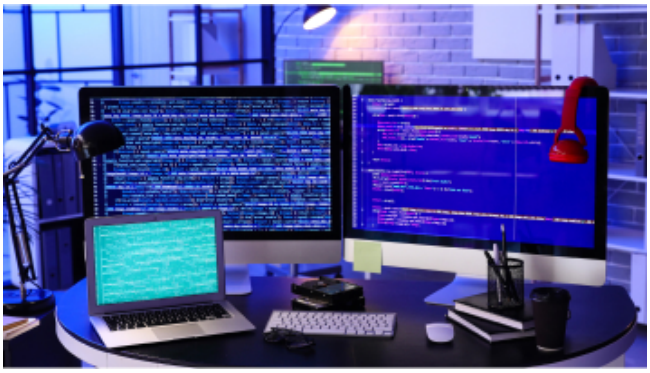
Escopo

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 <head>, 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.

Exemplo

Vamos imaginar a situação oposta: 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 agora 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.

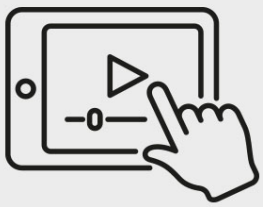
Outras considerações sobre a CSS

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.

Como inserir o CSS na página Web

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.).

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.

avaScript

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

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. Vejamos algumas características dessa linguagem:

query

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).

rototype

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).

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

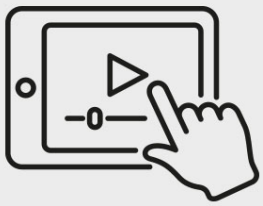
mensagem na tela. Ou ainda aumentar o tamanho de uma fonte ou diminuí-lo.

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.

Como inserir o Javascript na página Web

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

Ao desenvolvermos uma página web, devemos nos preocupar não somente com o resultado final, mas também em utilizarmos corretamente cada uma das tecnologias. Nesse contexto, assinale a opção correta quanto às boas práticas a serem seguidas.

A Utilizar os elementos HTML corretamente, tendo em mente a semântica; separar as responsabilidades entre cada tecnologia; otimizar o tempo de carregamento das

páginas; utilizar folhas de estilo a partir de arquivos externos.

B Deve-se evitar, sempre que possível, fazer uso de novas técnicas ou novas funcionalidades no que diz respeito às tecnologias client side. Isso porque as tecnologias web já possuem uma especificação própria, antiga, e, por isso, não se adaptam bem a novos recursos.

C A CSS possui um sistema de hierarquia. Com isso, ao usarmos CSS no HTML, é recomendado usar estilos e scripts inline, já que facilitam o entendimento do comportamento e também visual do elemento ao qual foram aplicados.

D Remover a CSS interna para o final da página otimiza o desempenho e acelera o tempo de carregamento da página. Logo, essa é uma das práticas mais recomendadas.

E Como forma de inserção da CSS em uma página web, é preferível utilizar a CSS internamente ao HTML, do que vincular um arquivo externo de estilos ao documento HTML.

Parabéns! A alternativa A está correta.

Como frisado nas seções de boas práticas, o ambiente web está em constante evolução. Tal fator, somado aos princípios básicos como semântica e separação de responsabilidades, define o que são as boas práticas quanto às tecnologias client side.

Questão 2

Como vimos, cada tecnologia do lado cliente possui sua própria função. Logo, a respeito da separação de funções e responsabilidades, assinale a alternativa correta.

A Com a evolução do HTML, novos elementos, como o formulário, tornaram possível a criação de páginas dotadas de interatividade, sem necessidade de recorrer ao uso de scripts provenientes de linguagens de programação, como JavaScript, por exemplo.

B O HTML é a base, a principal tecnologia do lado cliente. Apenas utilizando HTML é possível criar uma página rica em conteúdo — já que as tags servem justamente para isso — e layout — já que tudo fica dividido na estrutura semântica do HTML.

C Mais importante do que a preocupação com as funções de cada tecnologia é o resultado exibido no navegador. Logo, deve-se dar preferência ao resultado final, independentemente do que foi feito e de como foi feito, em termos de tecnologia, para se chegar a ele.

D Entre as três tecnologias do lado cliente, CSS é a mais dispensável e menos importante, já que é possível cuidar de todo o layout e apresentação fazendo uso apenas de HTML.

E O HTML cuida do conteúdo; o CSS, do layout/apresentação; o Javascript, do comportamento/interação. Com isso, ao não misturarmos as funções — embora seja possível —, obtemos vários benefícios, como o de separação de interesses e consequente facilidade para manter o código, uma vez que podemos ter diferentes pessoas trabalhando ao mesmo tempo em diferentes partes do site.

Parabéns! A alternativa E está correta.

As tecnologias do lado cliente foram desenvolvidas em momentos distintos, a começar pela HTML. Com isso, a partir do surgimento de novas necessidades, novas tecnologias foram desenvolvidas, como a CSS e o JavaScript. A utilização em conjunto dessas tecnologias, que se complementam, traz inúmeros benefícios, desde a otimização na criação das páginas ao resultado final.

4 - Tecnologias do lado servidor

Ao final deste módulo, você será capaz de reconhecer as tecnologias do lado servidor.

PHP: uma linguagem de programação server side

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.

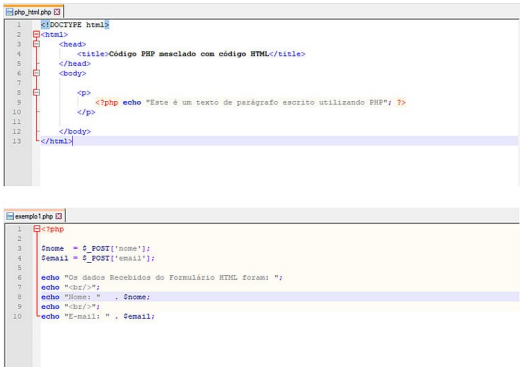
1. Etapa 1: O código PHP gerado é interpretado pelo servidor.
2. Etapa 2: Esse código é convertido em formato HTML.
3. Etapa 3: O código é exibido no navegador.

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 <?php e ?>. 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 <?php e ?>. 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 exclusão de dados.

Páginas dinâmicas e acesso a dados

Páginas dinâmicas

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, por exemplo, 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, onde 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**.

Nelas, 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.

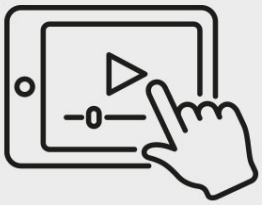
	Páginas estáticas	Páginas dinâmicas
Inclusão/Alteração/Exclusão de conteúdo	Manualmente, direto no código HTML	Automaticamente através de scripts no lado servidor, como PHP
Armazenamento do conteúdo	Na própria página HTML	Em um banco de dados

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.

Páginas dinâmicas

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



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.

GBD

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 em uma chamada/requisição ao lado servidor — script e banco de dados.

A partir do JavaScript

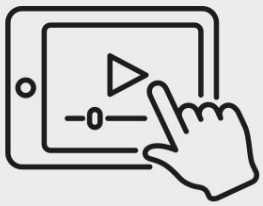
O JavaScript possui, essencialmente, duas formas para se comunicar com linguagens do lado servidor: por meio das APIs (*Application Programming Interface*) XMLHttpRequest e Fetch API. A primeira é amplamente utilizada, sendo a forma mais comum de realizar essa comunicação. É normalmente associada a uma 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, é enviada 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.

Tecnologias lado servidor: linguagem PHP e acesso a dados

Assista a este vídeo e conheça um exemplo de código PHP.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Falta pouco para atingir seus objetivos.

Vamos praticar alguns conceitos?

Questão 1

O PHP é uma linguagem de script, altamente adaptável à HTML e que lhe possibilita interatividade e dinâmica. Assinale a alternativa correta quanto a essa afirmação.

A É possível criar um script PHP que faça acesso a banco de dados utilizando apenas código HTML.

B Para recuperar informações de um banco de dados, a HTML precisa fazer uso do PHP, seja diretamente, a partir de algum elemento próprio, ou por meio de JavaScript.

C Como o PHP é altamente adaptável à HTML e esta ao JavaScript, um script escrito nesta última linguagem pode recuperar informações acessando diretamente o banco de dados.

D O PHP é altamente adaptável à HTML. Logo, assim como a HTML, um script PHP é renderizado diretamente pelo navegador.

E O PHP é uma linguagem interpretada, executável no lado cliente. O código PHP gerado é interpretado pelo cliente, convertido em formato HTML e exibido no navegador.

Parabéns! A alternativa B está correta.

O PHP é uma linguagem server side, utilizada, sobretudo, para criar páginas dinâmicas, em conjunto com a HTML e demais tecnologias do lado cliente. Embora muito adaptável à HTML, o PHP é uma linguagem de programação completa, que possui uma sintaxe específica, assim como funções e métodos nativos que lhe possibilitam o acesso tanto ao sistema de arquivos quanto a diferentes bancos de dados.

Questão 2

As páginas dinâmicas, ao contrário das páginas estáticas, proveem dinamismo ao ambiente web. Nesse contexto, assinale a opção correta.

A Uma página web completa só pode ser produzida com a utilização de páginas dinâmicas.

B As páginas dinâmicas são, resumidamente falando, uma forma de interação entre um usuário e uma página HTML. Logo, uma página que faz uso de JavaScript é uma página dinâmica.

C A única vantagem de se utilizar páginas dinâmicas é, de fato, guardar os dados do site em um lugar mais seguro.

D A utilização de linguagens de programação server side é a principal característica de uma página dinâmica.

E Páginas dinâmicas são produzidas por meio da utilização das tecnologias do lado servidor, sendo assim completamente independentes das tecnologias do lado cliente.

Parabéns! A alternativa D está correta.

Nas páginas dinâmicas, todo o conteúdo de um site pode ser gerenciado automaticamente por meio de scripts que rodam no servidor.

Considerações finais

Ao longo deste estudo, vimos como o modelo cliente X servidor, inicialmente restrito às redes internas das empresas, evoluiu tanto nos diferentes modelos de camadas quanto na migração para a internet. Novas tecnologias foram criadas no lado cliente — linguagens de marcação; folhas de estilo; linguagens de script — e também no lado servidor — linguagens de script; bancos de dados; páginas dinâmicas —, criando o ambiente web como o conhecemos atualmente, caracterizado pelas tecnologias que formam sua base e pela preocupação com a melhor experiência possível para os usuários.

Podcast

Para encerrar, ouça sobre as Tendências do Ambiente Web – Cliente X Servidor.

Para ouvir o *áudio*, acesse a versão online deste conteúdo.

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.

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.