

# INTRODUÇÃO

Em nossas aulas, serão abordados os conhecimentos necessários para a utilização das tecnologias que envolvem o desenvolvimento de sistemas web e a possibilidade de reaproveitamento dessas tecnologias no ambiente móvel. As aulas buscam relacionar os aspectos principais do desenvolvimento de sistemas que são acessados via navegador e disponibilizados na internet, como esses sistemas têm evoluído sua interface com o usuário e quais tecnologias contribuíram para isso.

Nesta aula, serão abordados importantes conceitos que facilitarão o entendimento de algumas tecnologias incorporadas ao desenvolvimento dos sistemas, as quais serão apresentadas nas aulas seguintes. Também será apresentada a linguagem de marcação HTML, que será utilizada como base na construção de interfaces nas demais aulas.

## TEMA 1 – DESENVOLVIMENTO WEB

O desenvolvimento de sistemas foi se transformando à medida que as interfaces usadas para o acesso às informações dos sistemas foram evoluindo: de telas de texto puro para interfaces gráficas.

A incorporação das redes de computadores pelas empresas impulsionou a redistribuição do processamento executado pelos programas, podendo ser feito em diferentes dispositivos e retornando o resultado para o usuário que pode visualizá-lo, estando em uma localização diferente de onde é feito esse processamento.

Esse novo contexto alterou o conjunto de conhecimentos que o desenvolvedor precisava ter ao projetar um sistema. Agora é preciso avaliar o ambiente em que será executado, como será a comunicação entre o usuário e a interface e, também, a comunicação entre a interface e o conjunto de regras necessárias para a execução do sistema.

Para desenvolver um sistema, uma das primeiras coisas que são percebidas como necessárias pelo desenvolvedor é a

linguagem de programação. Apesar de uma linguagem de programação ser indispensável, o entendimento e o uso de tecnologias de rede são a base de apoio para o uso da maioria dos sistemas atualmente, principalmente ao considerar sistemas web, como também aplicativos em dispositivos móveis (smartphones, tablets e notebooks).

Um sistema web utiliza as tecnologias e ferramentas associadas à internet como base para seu desenvolvimento, e a navegação nesse tipo de sistema tem como base o Uniform Resource Locator (URL), que possibilita a localização do recurso que contém a informação, podendo esse recurso ser uma figura, uma página com diferentes conteúdos, um vídeo ou um sistema de informação.

Como exemplo de localização de um recurso usando uma URL, temos

“<http://www.empresa.com.br/pasta/programa.php?opcao=12>”, em que o recurso é um programa que receberá um parâmetro, vai ser executado e retornará a informação; e

“<http://www.site.org/pasta/imagem.png>”, em que, nesse caso, o recurso é uma imagem que será retornada para o usuário.

A forma como o usuário interage com um sistema web foi evoluindo com o tempo, e com a aplicação de novas características e componentes na criação de interfaces, foi possível a implementação de aplicações padrão RIA (Rich Internet Applications, ou “aplicações de internet ricas”), em que a interação do usuário se assemelha ao tipo de interação a que ele estava acostumado nos programas desktop (com isso, a interface fica mais “rica” em detalhes) (Deitel; Deitel, 2009, p. 313). Essas aplicações são chamadas de “aplicações web” (web applications) pelo órgão internacional que padroniza o ambiente web, chamado de W3C (W3C, 2021b).

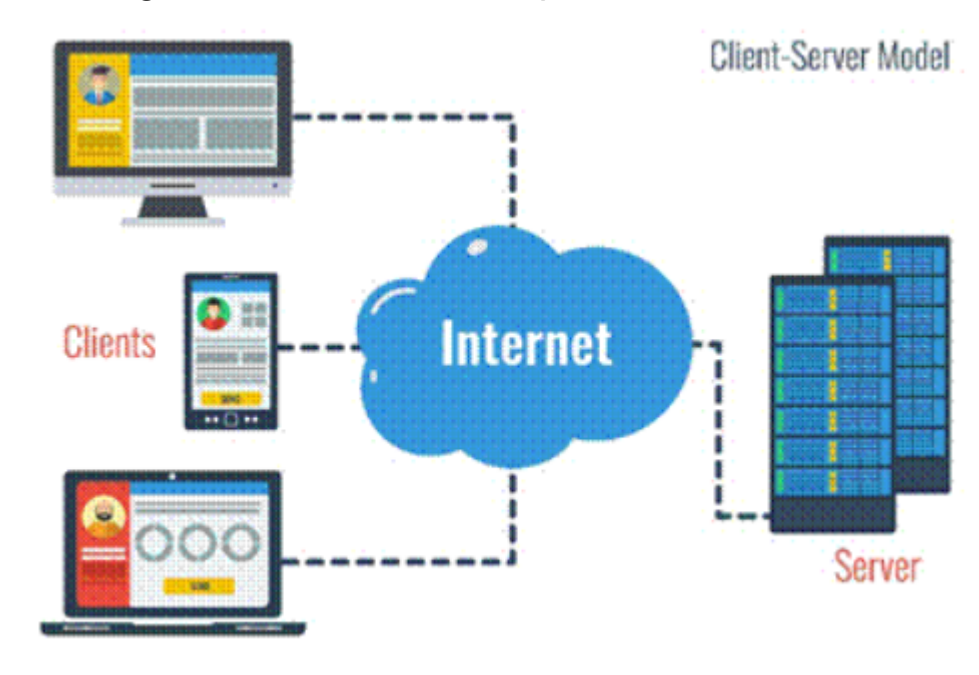
Para entender melhor a parte técnica que será apresentada futuramente, devemos antes detalhar alguns conceitos específicos, como arquitetura cliente-servidor, tipos de aplicações no ambiente web e aplicações web em ambiente móvel.

## 1.1 CLIENTE-SERVIDOR

Tanenbaum e Bos (2015, p. 577) afirmam que “a web é fundamentalmente um sistema cliente-servidor, no qual o usuário é um cliente e um website é o servidor”. Considerando que a web é um ambiente cliente-servidor e que trataremos de sistemas que vão trabalhar nesse ambiente, vamos entender melhor esse mecanismo.

Uma arquitetura cliente-servidor estabelece de início uma forma de comunicação entre duas entidades: uma que solicita uma informação (cliente) e outra que responde a essa solicitação (servidor). Os dispositivos não precisam estar fisicamente na mesma localização: basta que se tenha um meio de transmissão das informações que possibilitem que um dispositivo encontre o outro (Tanenbaum; Bos, 2015, p. 68; Connolly; Begg, 2005, p. 58).

Figura 1 – Diferentes dispositivos em rede



Créditos: sergeyvasutin/Adobe Stock.

Como exemplificado na Figura 1, essa comunicação pode acontecer entre diferentes dispositivos. A rede de computadores cria um caminho padronizado para a troca de informações, mas também pode acontecer no mesmo dispositivo, em que uma

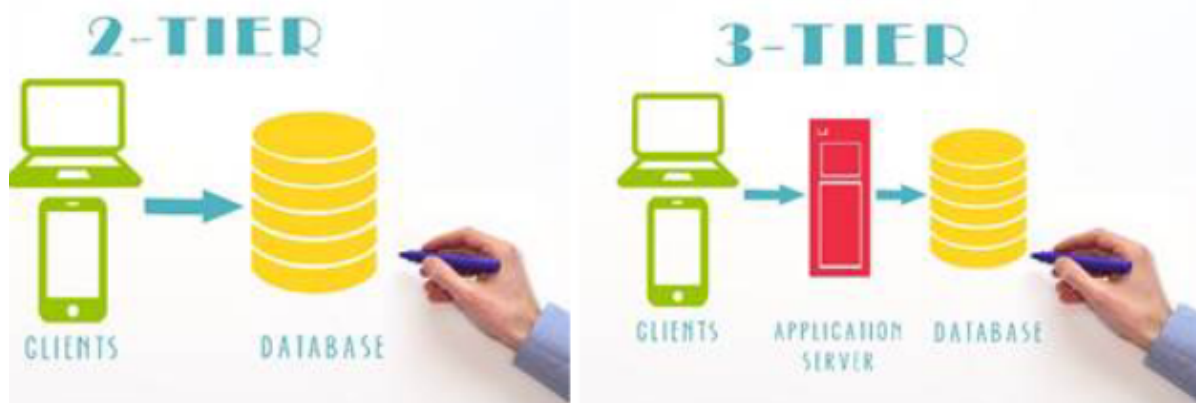
aplicação cliente solicita uma informação para uma aplicação servidora que está em execução no mesmo dispositivo.

Segundo Connolly e Begg (2005, p. 57), a estruturação dessas aplicações pode ser feita de duas formas.

a. Tier 2 (duas camadas): o aplicativo cliente (camada 1) recebe, processa e apresenta os dados do resultado da solicitação e o armazenamento dos dados e validação de acesso estão localizados no servidor (camada 2).

b. Tier 3 (três camadas): o processamento fica distribuído em três componentes, podendo ser feito em parte pelo aplicativo no cliente (camada 1) que fica responsável pela apresentação dos dados, por um sistema intermediário (middleware) localizado em um servidor que realiza o processamento das regras de negócio (camada 2) e pela base de dados em outro servidor (camada 3).

Figura 2 – Sistemas em 2 e 3 camadas



Créditos: A Stefanovska/Adobe Stock.

O uso de camadas depende do tipo de sistema que está sendo desenvolvido, da capacidade de processamento requerida e da quantidade de usuários que precisam acessar o sistema de forma simultânea.

Em nosso tutorial, a arquitetura cliente-servidor será utilizada associada aos protocolos e às formas de comunicação utilizadas na internet, sendo o protocolo HTTP (HyperText Transfer Protocol

ou “protocolo de transferência de hipertexto”) utilizados como base da comunicação entre os programas exemplificados nas demais aulas.

### 1.1.1 HYPERTEXT TRANSFER PROTOCOL (HTTP)

O HTTP não é um conceito, mas um dos mais utilizados protocolos de comunicação na internet. Ele é a base do ambiente web, sendo usado na comunicação das solicitações dos navegadores e nas respostas dos servidores.

Conforme Tanenbaum e Wetherall (2011, p. 664), no início da web, os conteúdos armazenados nos servidores eram estáticos, ou seja, arquivos contendo páginas de texto ou figuras eram transferidos para o cliente usando um aplicativo chamado browser (navegador) e a forma básica de construir esses conteúdos era utilizando uma linguagem de marcação chamada HTML (a qual veremos mais adiante).

Para disponibilizar um sistema web no servidor, um serviço HTTP precisa ser instalado e configurado, para que seja executado indefinidamente, aguardando alguma solicitação via rede. Essa solicitação pode ser criptografada ou não, e utiliza como padrão as portas 443 (criptografado) e 80 (normal).

### 1.1.2 PROGRAMAÇÃO NO SERVIDOR

Um dos sistemas que implementam o HTTP mais utilizados é o sistema Apache, que, além de receber e responder às solicitações HTTP, também tem mecanismos para incorporar uma linguagem de programação que pode ser usada no desenvolvimento de sistemas no lado servidor. Outros serviços HTTP também podem ser utilizados (por exemplo, Nginx, Nodejs), mas um fator que vai tornar esses serviços propícios para o desenvolvimento de sistemas no lado servidor é a integração com alguma linguagem de programação.

Com a utilização de uma linguagem de programação incorporada ou integrada ao um serviço HTTP, o desenvolvedor terá acesso ao banco de dados, a outros servidores, a sistemas de arquivos e demais recursos de forma centralizada. Essa centralização pode ser muito importante para compartilhar informações atualizadas entre vários clientes.

## 1.2 APLICAÇÕES WEB

No início da internet, ao desenvolver um sistema em um ambiente web, o processamento era feito somente no servidor e o cliente apenas solicitava a nova página que traria outras informações. Nesse método, tínhamos as “páginas dinâmicas”: sites em que as páginas eram construídas na medida em que eram solicitadas. Assim, o usuário navegava nos menus dos sistemas e clicava nas opções. Ao selecioná-las (clicando em um hiperlink, por exemplo), uma solicitação era enviada ao servidor e uma nova página chegava ao navegador com todas as informações. Esse contexto era chamado de MultiPage Applications – MPA (ou “aplicação de múltiplas páginas”).

Todo o processamento era feito no servidor, e conforme a quantidade de usuários ia aumentando, multiplicava-se a capacidade do servidor. Em sistemas menores, com poucos usuários, essa situação não chegava a impactar no ambiente do servidor e muitos sistemas ainda têm essa estrutura. No entanto, em sistemas com milhões de acessos, a infraestrutura necessária para manter o sistema com um desempenho aceitável era muito grande.

Uma questão que pesava nesse tipo de processamento é que toda a página (layout, figuras, estrutura, dados) precisava ser refeita na tela do navegador. Isso tinha um impacto no servidor (processava toda a tela novamente), no cliente (redesenhava toda a tela novamente) e na comunicação (todos os bytes da tela eram transmitidos novamente).

Para criar uma alternativa para essa estrutura e diminuir a crescente demanda por mais processamento no servidor, a atenção voltou-se para a forma como uma aplicação de desktop (em duas camadas) era desenvolvida. A tela e o processamento eram manipulados pelo lado do cliente e apenas a informação específica que precisava ser atualizada era solicitada ao servidor. Em um sistema web, esse método é chamado de Single Pages Applications – SPA (ou “aplicação de página única”).

Na SPA, o servidor envia para o cliente um código que será executado no navegador e, quando a informação precisar ser atualizada, esse código no cliente envia uma solicitação contendo apenas essa necessidade para o servidor.

Dessa forma, a primeira transmissão enviada para o usuário conterá o código principal e, assim, uma boa parte do processamento será transferida para o cliente e o servidor centralizará os dados. Com o aumento da quantidade de usuários acessando o servidor, como parte do processamento vai para cada usuário (cliente), os recursos do servidor (memória, processamento) podem ser menores.

Como vantagem adicional, esse código que será executado no cliente poderá trabalhar as interações com o usuário de forma mais rápida e poderá criar personalizações no layout de forma mais dinâmica.

Tanto as aplicações MPA quanto as aplicações SPA são consideradas aplicações web (web applications), desde que tornem a experiência do usuário semelhante ao uso de aplicações desktop.

Para desenvolver tanto uma aplicação MPA quanto uma aplicação SPA, o desenvolvedor precisará conhecer ferramentas e tecnologias usadas no navegador do cliente (por exemplo, HTML, CSS, JavaScript) e no servidor (por exemplo, banco de dados, plataforma HTTP). Para aprofundar o contexto de processamento no lado cliente, incluiremos aqui, futuramente, um detalhamento maior sobre o Cascading Style Sheets – CSS (ou “folha de estilo em cascata”), criado com o propósito de prover um mecanismo mais flexível para organizar a estética dos documentos HTML e do JavaScript, linguagem de programação inserida no navegador para manipular o documento HTML e interagir com este.

### 1.3 PROGRESSIVE WEB APP (PWA)

Outro ambiente que tem sido alvo do desenvolvimento de sistemas atuais é o móvel (mobile), em que o smartphone é o principal dispositivo. Esse ambiente tem algumas características próprias, por exemplo, sensores, sistema operacional e interface e, com isso, tem um desenvolvimento diferenciado.



No entanto, o desenvolvimento web tem também algumas tecnologias em comum com o ambiente móvel, em que podemos citar: ambos usam a rede de comunicação para acessar bases de dados, um navegador web também está presente nos dispositivos móveis e as linguagens de programação são as mesmas. Essa percepção vai orientar a possibilidade de poder reutilizar o código dessas aplicações entre esses ambientes.

Uma opção que foi criada para o reaproveitamento de código de uma aplicação web e que possibilita ter a experiência de uso semelhante a um aplicativo usado em aparelhos móveis é o Progressive Web App – PWA (ou “aplicativo web progressivo”). Com esse tipo de configuração da aplicação web, o sistema pode ser instalado no smartphone de forma semelhante a um aplicativo, podendo ser identificado na tela inicial do aparelho ou na lista de aplicativos. Nesse caso, a instalação não é feita pela loja de aplicativos dos sistemas Android ou IOS, mas por meio de uma interface no navegador do smartphone.

Algumas características fazem esse tipo de configuração ser diferente de um simples atalho para um site: a aplicação web deve ter um arquivo de configuração chamado de manifesto, que indica diversos parâmetros a serem utilizados na sua instalação (por exemplo, nome, ícone, local), deve poder interceptar a conexão com a rede e desviar o acesso ao dispositivo local nos casos de estar sem rede (off-line), o acesso externo deve ser criptografado (https), deve ser usável tanto em navegadores com menos recursos (usando menos funcionalidades) quanto em navegadores mais avançados e deve ser utilizável independentemente do tamanho da tela (responsivo).

Deve-se considerar que esse tipo de alternativa, apesar de melhor integrada com o sistema operacional, continua sendo uma aplicação web, que é instalada no dispositivo móvel e automatiza a execução de um navegador reconfigurado para não mostrar seus controles e menus; entretanto, para o usuário, a utilização é semelhante à de um aplicativo nativo (MDN Web Docs, 2021).