



# Desenvolvimento Web

**Prof. M. Marcelo Petri**

# Olá! Sejam bem vindos



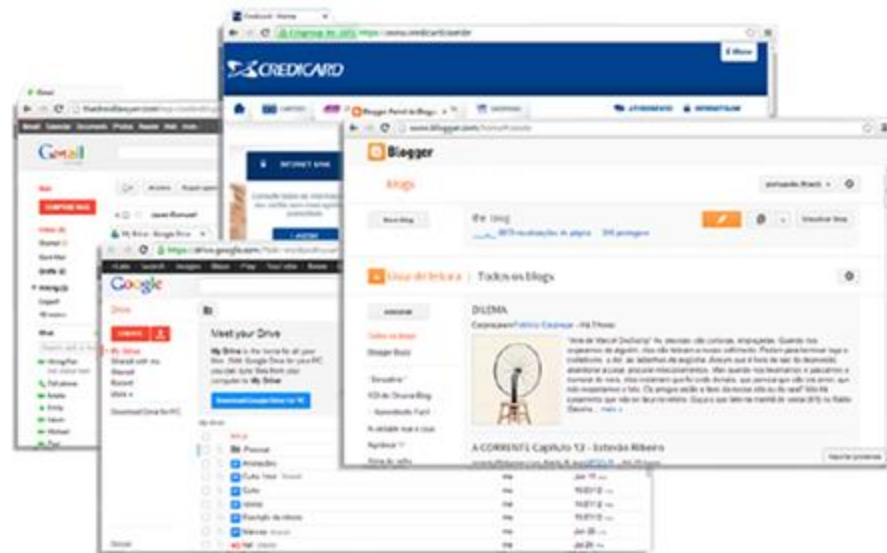
<b>DISCIPLINA/ UNIDADE CURRICULAR</b>	Desenvolvimento Web  6hs
<b>TEMA DA AULA</b>	Introdução a Web
<b>INDICADOR</b>	1. Relaciona os conceitos essenciais de infraestrutura com o desenvolvimento de uma aplicação web
<b>CRONOGRAMA</b>	1 Fundamentos da Programação para Web 1.1. Princípios de redes 1.2. Arquitetura HTTP
<b>ATIVIDADE</b>	Questionário sobre conceitos apresentados.
<b>HABILIDADES</b>	Discussões e atividades em grupo sobre as tecnologias abordadas
<b>CRITÉRIOS</b>	Análise

# O que é a Internet

- É uma estrutura em rede chamada ARPANET (Advanced Research Projects Agency), responsável por conectar computadores dos centros militares e de pesquisas, seja no intuito de compartilhar arquivos ou enviar mensagens eletrônicas.
- Internet é a “rede das redes”



# Sites X Aplicações WEB



# Sites X Aplicações WEB

- O termo **site** utilizado quando se referem a blogs, sites de notícias, sites institucionais, portais, lojas virtuais, entre outros.
- O termo **aplicação web** é muito utilizada para sistemas de gestão empresarial que são acessados através de navegadores (browsers).

# Evolução da Web

- <https://web.archive.org/>

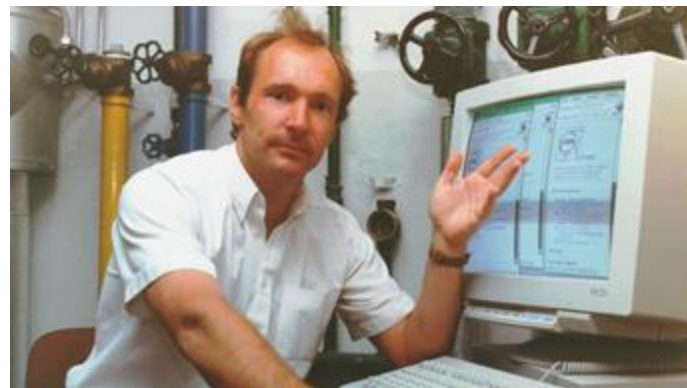


# Evolução da Web

- A história da internet é dividida em épocas marcadas pela implementação de novas tecnologias e mudanças na forma como interagimos nela.
- Conhecidas como Web 1.0, 2.0, 3.0 e 4.0, essas fases contam com características muito específicas.

# Evolução da Web - 1.0

- Proposta pelo físico e cientista da computação Tim Berners-Lee em 1989, a primeira geração da web baseava-se em hiperlinks:
  - Conteúdo estático com um design semelhante ao de páginas impressas.
  - Vídeos eram raros.
  - Internautas não podiam criar conteúdo, apenas consumir o que estava disponível.
  - As exceções a essa regra estavam nos envios de e-mail via formulários de cadastro e afins.





# Evolução da Web - 2.0

- A chegada da segunda versão da web revolucionou a experiência dos usuários:
  - possibilidade de inserção de comentários em páginas de blogs;
  - chegada das redes sociais;
  - popularização da tecnologia Ajax — utilizada na manipulação de mapas pela Google;
  - uso mais amplo dos estilos em CSS.



# Evolução da Web - 3.0

- A Web 3.0 nasce dos processos de transformação digital e baseia-se no uso de máquinas para melhorar a eficiência de atividades antes realizadas manualmente pelo usuário:
  - blockchain, tokens não-fungíveis (os NFTs) e descentralização;
  - tecnologias de automação e machine learning;
  - melhorar o desempenho de serviços da própria internet.





# Evolução da Web - 5.0

- Web 5.0: A próxima fronteira da internet
  - Inteligência artificial.



# Batalha dos Browsers (Navegadores)

- <https://www.w3schools.com/browsers/>

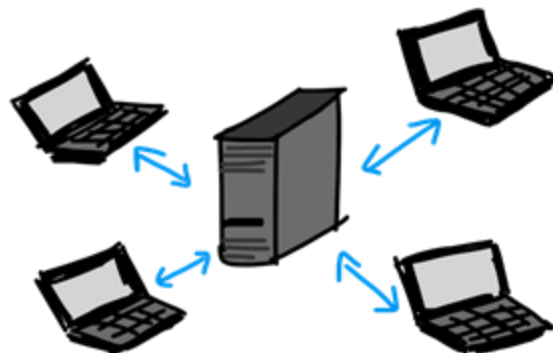
<b>2025</b>	<b><u>Chrome</u></b>	<b><u>Edge</u></b>	<b><u>Firefox</u></b>	<b><u>Safari</u></b>	<b><u>Opera</u></b>
April	79.1 %	11.4 %	3.8 %	4.2 %	0.8 %
March	78.5 %	11.6 %	3.8 %	4.2 %	0.8 %
February	78.4 %	11.7 %	3.9 %	4.3 %	0.9 %
January	79.4 %	10.7 %	3.9 %	3.9 %	1.4 %

# Design Responsivo



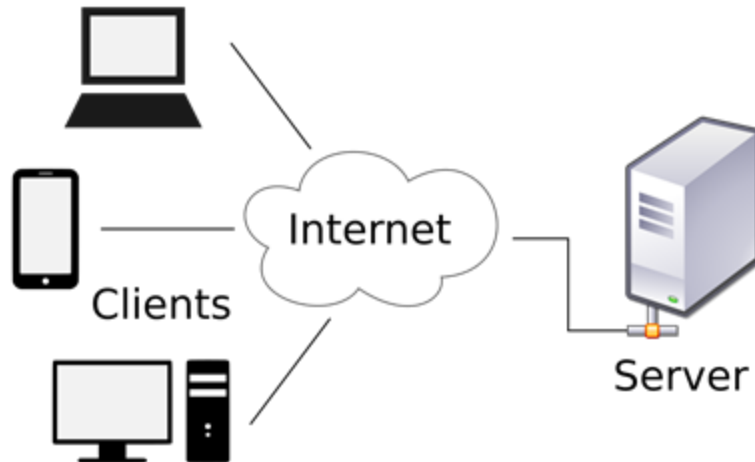
# Arquitetura Cliente-Servidor

- A arquitetura Cliente-Servidor é um modelo de comunicação entre dispositivos na rede.
- É uma das arquiteturas mais comuns na web e em muitos outros sistemas distribuídos.
- Consiste em dois elementos principais: o cliente e o servidor.



# Arquitetura Cliente-Servidor - Cliente

- O cliente é o dispositivo ou aplicação que faz solicitações ao servidor.
- Exemplos de clientes: navegadores web, aplicativos mobile, etc.
- Os clientes enviam requisições ao servidor e esperam pelas respostas.





# Arquitetura Cliente-Servidor - Servidor

- O servidor é o dispositivo ou aplicação que recebe as requisições dos clientes e fornece respostas.
- Os servidores podem armazenar e gerenciar dados, executar lógica de negócio e fornecer recursos para os clientes.
- Exemplos de servidores: servidores web, servidores de aplicação, bancos de dados, etc.



# Arquitetura Cliente-Servidor

- Diferenças entre processamento no cliente e no servidor.
- Vantagens e desvantagens de cada abordagem.
- Exemplos de aplicações que utilizam cada abordagem.

# Tipos de Protocolos

- Na Internet, esse protocolo é chamado de TCP/IP (Transmission Control Protocol / Internet Protocol):
  - **TCP** (Transmission Control Protocol): comunicação.
  - **IP** (Internet Protocol): identificador.
  - **FTP** (File Transfer Protocol): transferência de arquivos.
  - **HTTP** (Hyper Text Transfer Protocol): hipertextos.
  - **HTTPS** (Hyper Text Transfer Protocol Secure): Criptografia.
  - **DNS** (Domain Name System): converte nomes de domínios.

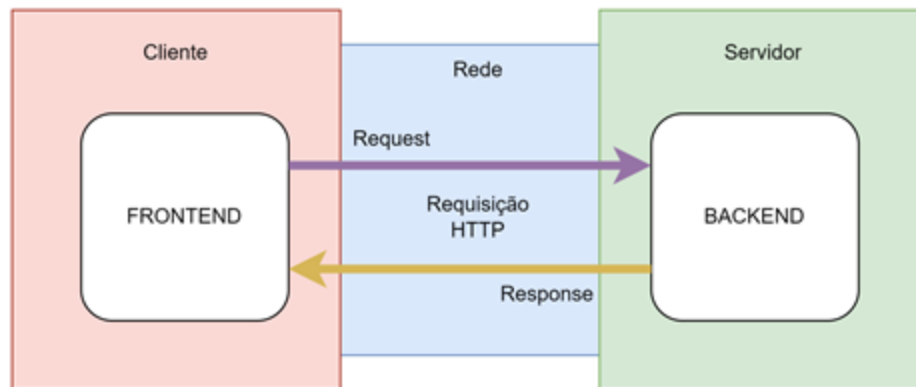
# Protocolo HTTP

- O Protocolo de Transferência de Hipertexto (HTTP) é um protocolo de comunicação utilizado na web.
- Ele define as regras para a comunicação entre clientes e servidores.
- O HTTP é a base para a comunicação entre navegadores web e servidores web.



# Requisições e Respostas HTTP

- As comunicações HTTP são realizadas através de requisições e respostas.
  - **Requisição:** Uma mensagem enviada pelo cliente ao servidor, solicitando um recurso específico.
  - **Resposta:** Uma mensagem enviada pelo servidor em resposta à requisição do cliente, contendo o recurso solicitado ou uma mensagem de erro.



# Métodos HTTP

- O HTTP define vários métodos (ou verbos) que indicam a ação a ser realizada na requisição.
- Alguns métodos comuns são:
  - GET: Obter informações do servidor.
  - POST: Enviar dados ao servidor para serem processados (por exemplo, formulários).
  - PUT: Atualizar um recurso no servidor.
  - DELETE: Remove um recurso no servidor.

# Códigos de Status HTTP

- O servidor retorna um código de status HTTP para indicar o resultado da requisição.
- Alguns códigos de status comuns são:
  - 200 OK: Requisição bem-sucedida.
  - 404 Not Found: O recurso solicitado não foi encontrado.
  - 500 Internal Server Error: Erro interno no servidor.



<https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Status>

# Exemplo de Requisição HTTP

- Método: GET
- URL: <https://www.exemplo.com/pagina.html>

## Exemplo de URL

http:// 32.143.12.1 :80 /images ?title=gatinhos



# Exemplo de Resposta HTTP

- Código de Status: 200 OK
- Conteúdo da Resposta: (conteúdo da página solicitada)



**404.** That's an error.

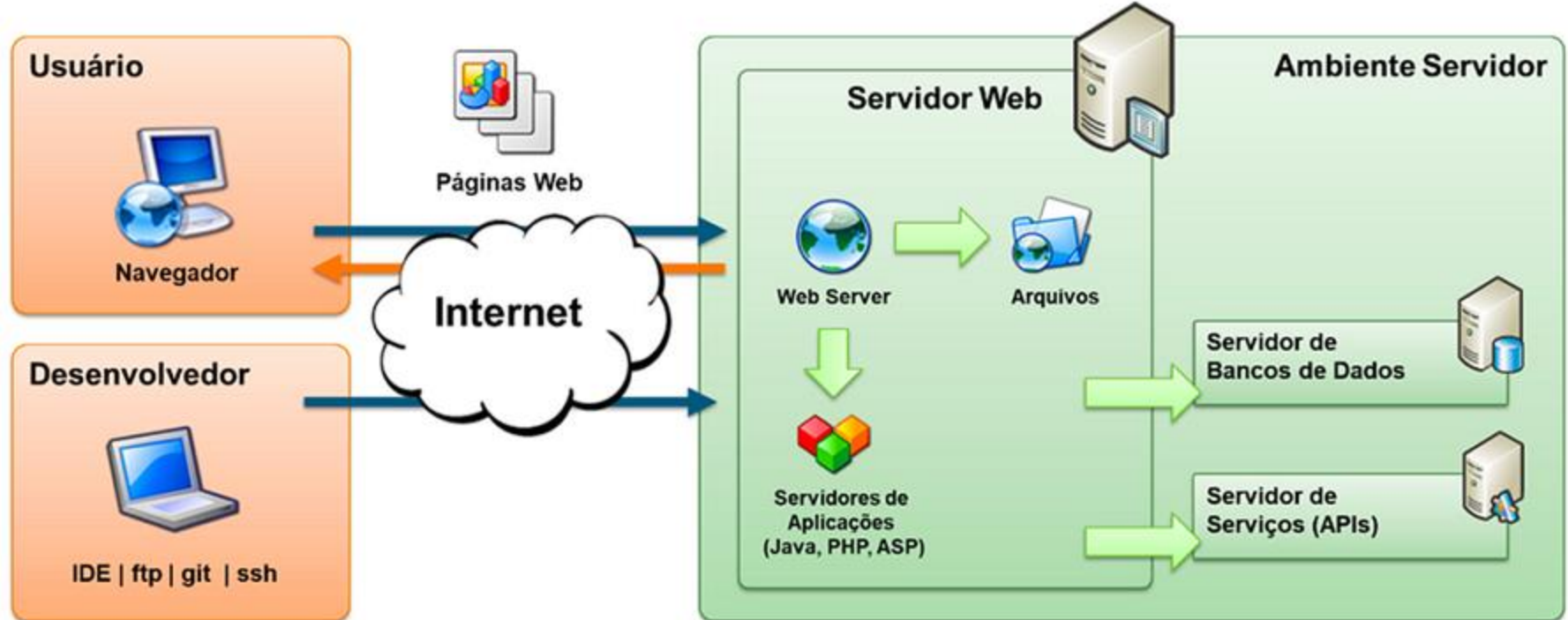
The requested URL `/does_not_exist` was not found on this server. That's all we know.



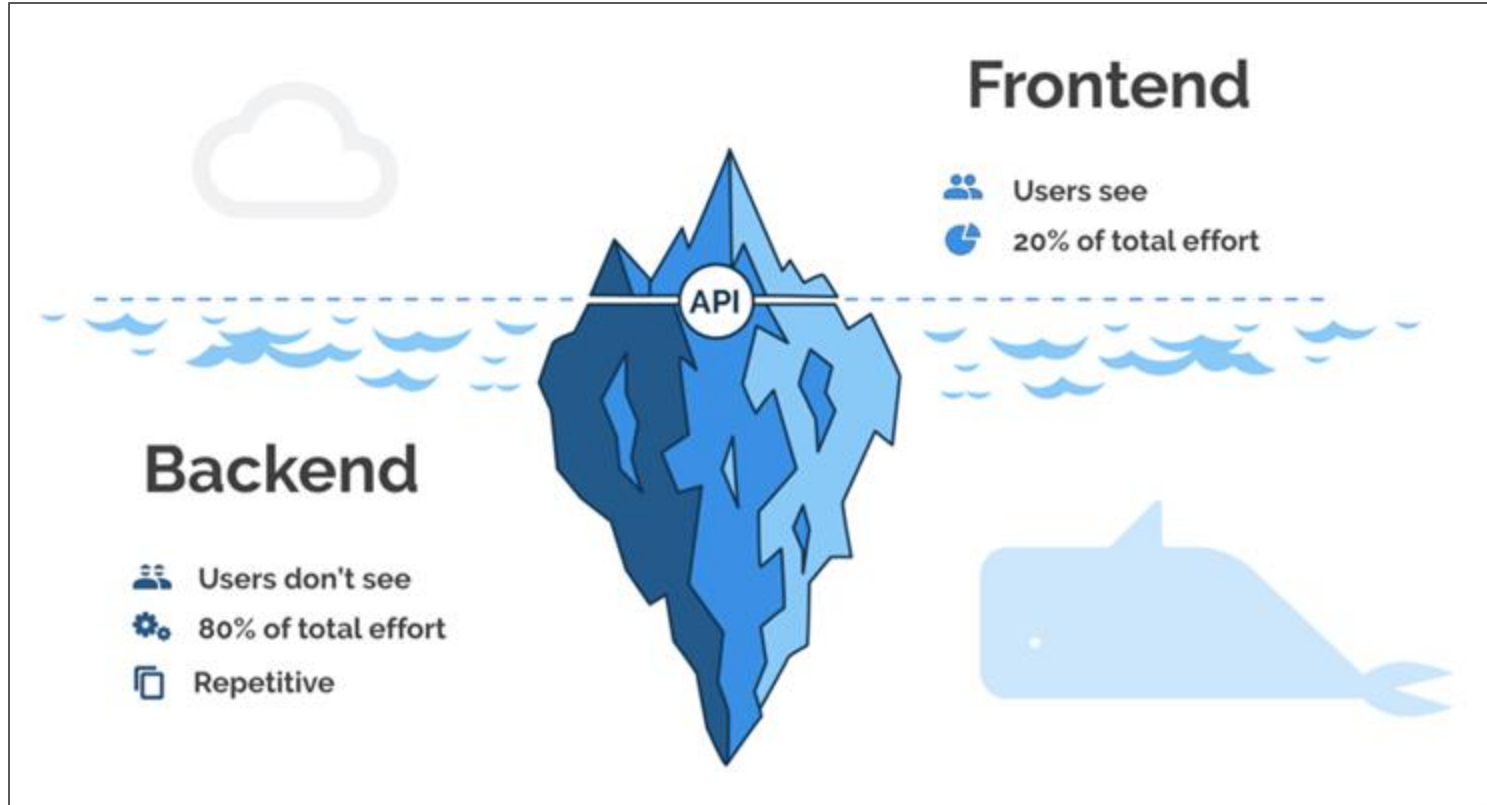
# Arquitetura Web

- O que é desenvolvimento front-end?
- A importância do front-end para a experiência do usuário (UX).
- Diferença entre front-end e back-end.

# Arquitetura Web



# Arquitetura Web



# Tecnologias



# Introdução ao Desenvolvimento Web

- Quais sites ou aplicações web vocês mais utilizam e qual a importância da interface visual na sua opinião?

# Introdução ao Desenvolvimento Web

- O papel do front-end na comunicação com o back-end.
- Tecnologias utilizadas no back-end (visão geral).
- Como o front-end consome dados do back-end: APIs e JSON.
- Protocolos utilizados (HTTP, REST).

# Introdução ao Desenvolvimento Web

- DevTools do Chrome para inspecionar uma requisição HTTP.
- Estrutura de uma API simples e como o front-end a consome.





# Tecnologias Client-Side

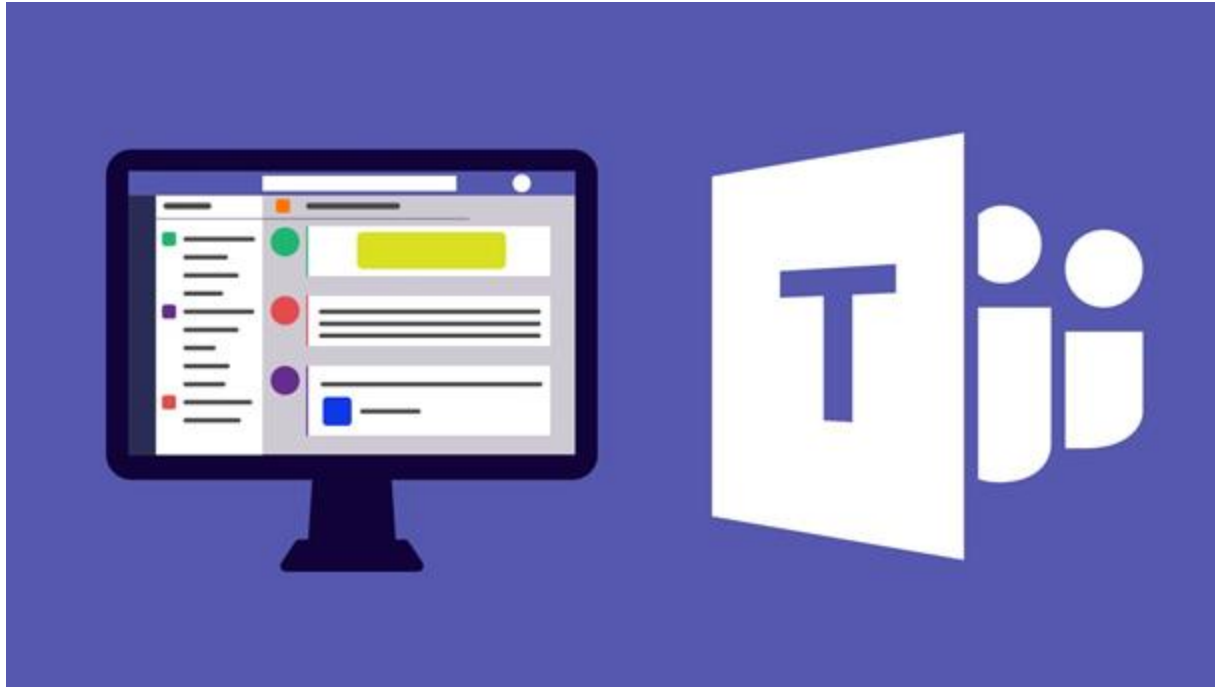
- Estrutura básica do front-end: HTML, CSS e JavaScript.
- Frameworks e bibliotecas populares (React, Angular, Vue.js).
- A importância da responsividade e acessibilidade.

# Ferramentas de Desenvolvimento

- Uso do VS Code para desenvolvimento front-end.
- Inspeção de elementos e debug no DevTools do Chrome.
- Versionamento de código com Git e GitHub.

# Atividades

- Responder questionário TEAMs



# Referências

- **Básicas: [OBRIGATORIAMENTE 3 BIBLIOGRAFIAS]**
- ARAÚJO, Everton Coimbra de. Desenvolvimento para web com Java. Florianópolis: Visual Books, 2010.
- QIAN et al. Desenvolvimento Web Java. LTC, 2010.
- GOMES, Yuri Marx Pereira. Java na web com JSF, Spring, Hibernate e Netbeans 6. Rio de Janeiro, RJ: Ciência Moderna, 2008.
  
- **Complementar: [OBRIGATORIAMENTE 5 BIBLIOGRAFIAS]**
- GONÇALVES, Edson. Desenvolvendo aplicações web com netbeans ide 6. Rio de Janeiro, RJ: Ciência Moderna, 2008.
- LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair de. Programação Java para Web. 2ª Ed. São Paulo: Novatec, 2015.
- SOUZA, Thiago Hernandez de. Java mais primefaces mais ireport: desenvolvendo um CRUD para web. Rio de Janeiro, RJ: Ciência Moderna, 2013.
- KALIN, Martin. Java Web Services: implementando. Rio de Janeiro, RJ: Alta Books, 2010.
- EIS, Diego. Guia Front-End: o caminho das pedras para ser um Dev Front-End. São Paulo: Casa do Código, 2015.
- SOUZA, Alberto. Spring MVC: domine o principal framework web Java. São Paulo: Casa do Código, 2015.

