

Server-Side – Entendendo os conceitos

Católica de Santa Catarina – Centro Universitário
Jaraguá do Sul - SC, Brasil

Professor: Ph.D. Andrei Carniel (Prof. Andrei)

Contato: andrei.carniel@gmail.com

linktr.ee/andrei.carniel



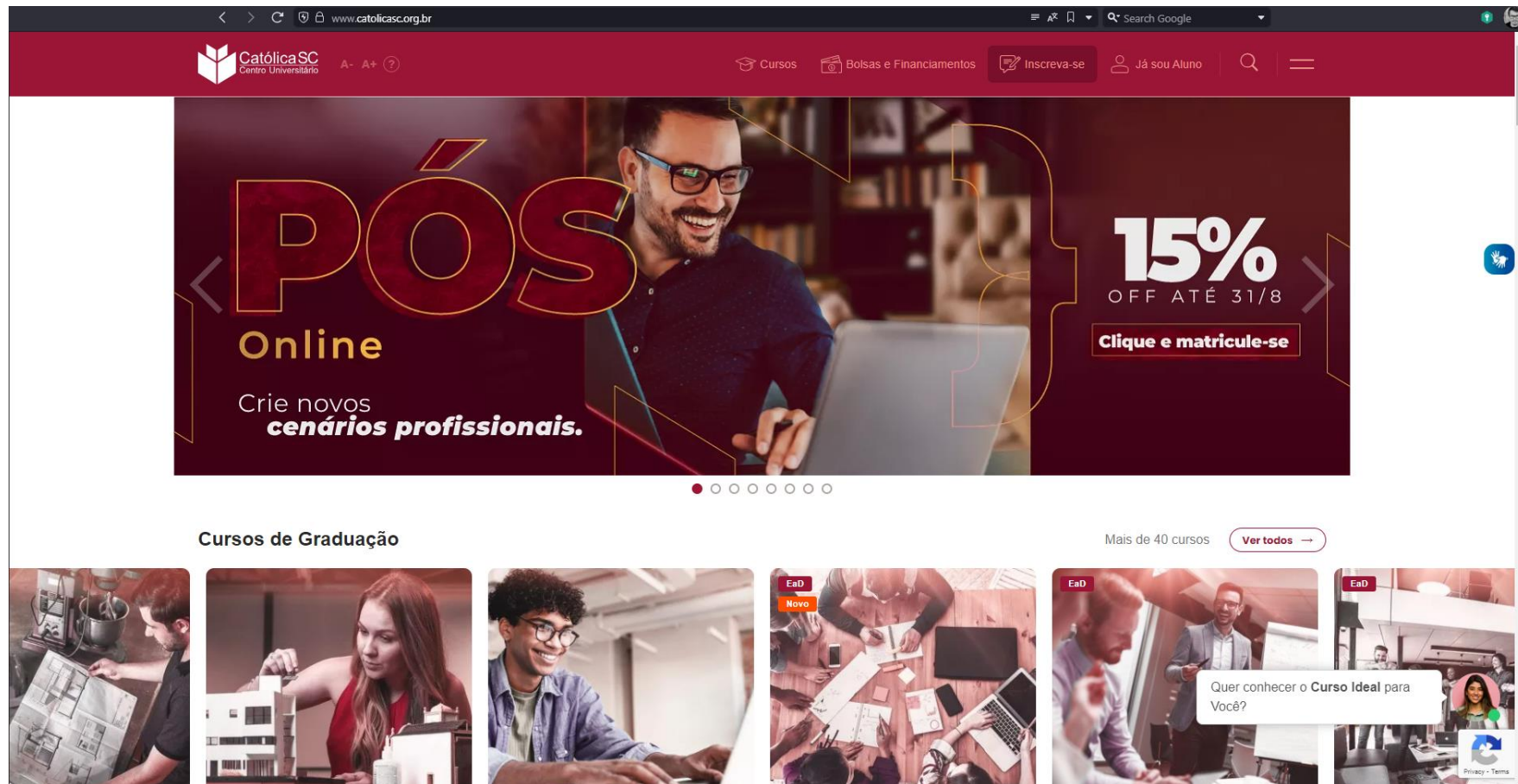
Retomando conceitos

Funcionamento básico da WEB

- Navegadores comunicam-se com web servers utilizando o HyperText Transfer Protocol (HTTP).
- Quando você clica em um link em uma página da web, envia um formulário ou faz uma pesquisa, uma HTTP request (solicitação HTTP) é enviada do seu navegador para o servidor de destino.
- Os servidores da web esperam por mensagens de solicitação do cliente, processam-nas quando chegam e respondem ao navegador da web com uma mensagem HTTP response.
 - A resposta contém uma linha de status indicando se a solicitação foi bem-sucedida ou não (ex. “HTTP/1.1 200 OK” para sucesso).

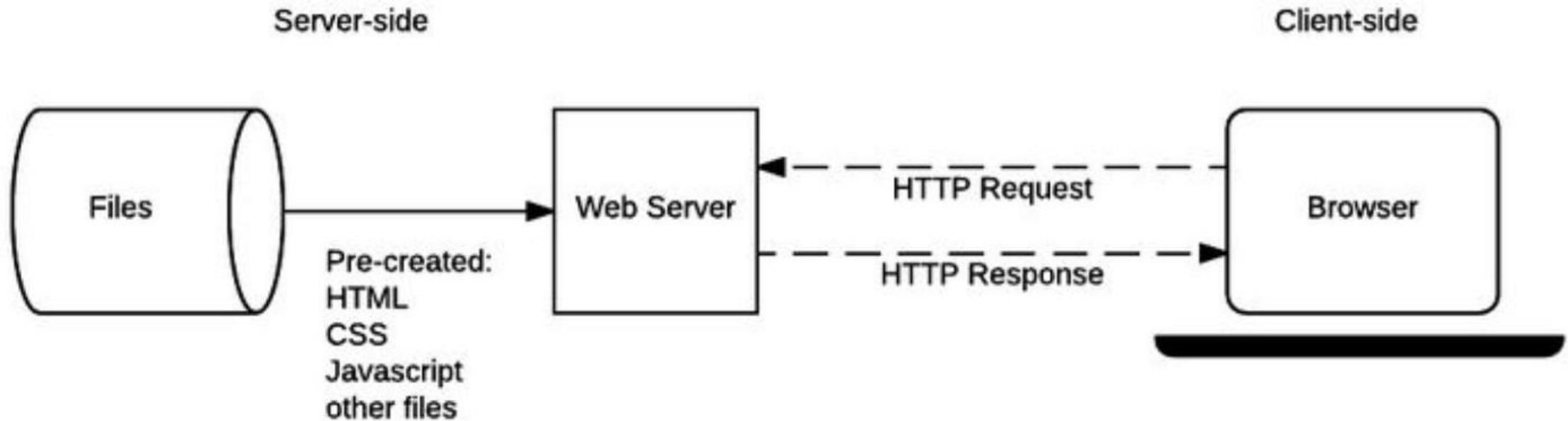
Quando a solicitação é bem-sucedida

- O corpo de uma resposta bem-sucedida a uma solicitação conteria o recurso solicitado (por exemplo, uma nova página HTML ou uma imagem, etc ...), que poderia então ser exibido pelo navegador.



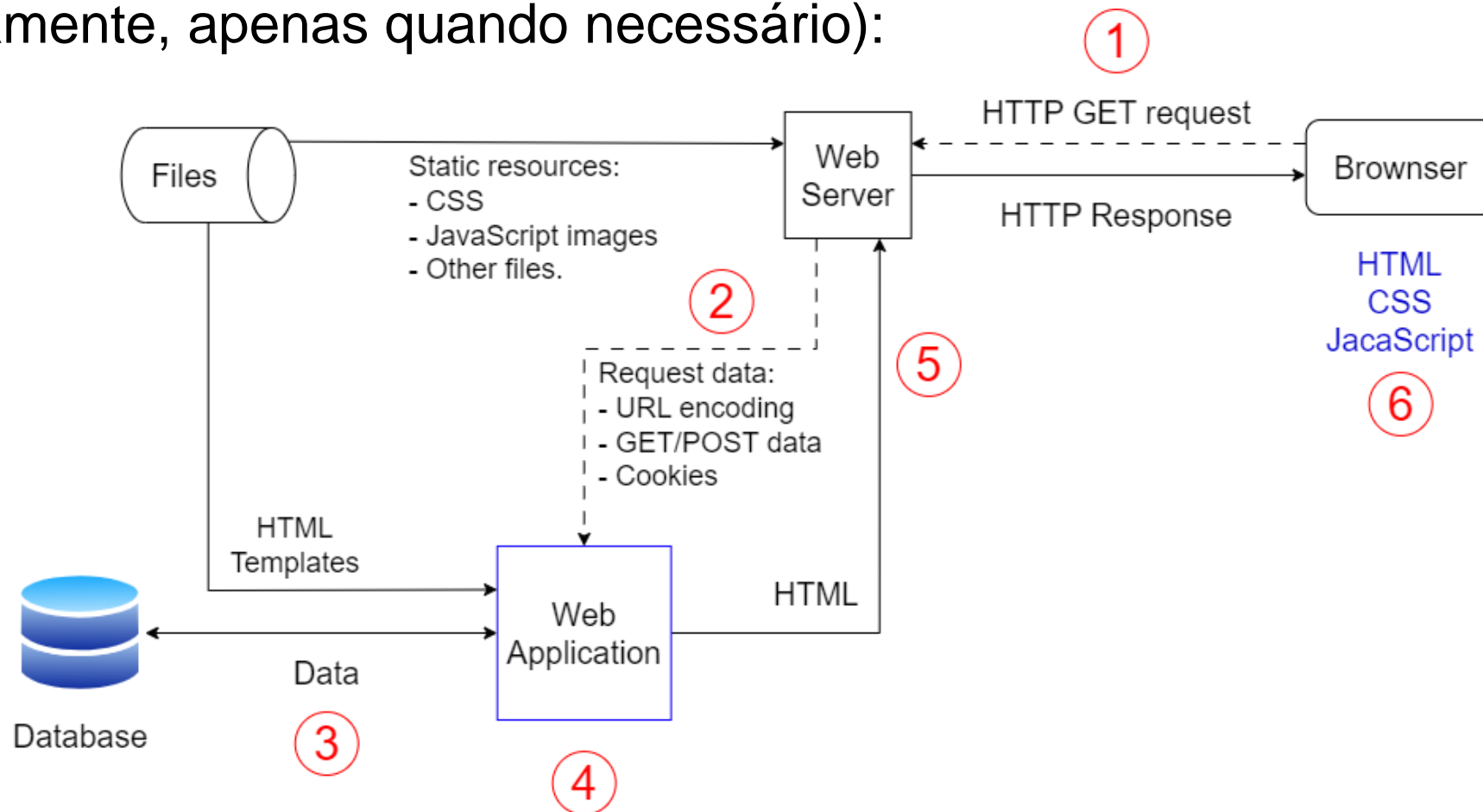
Exemplo de aplicação

- Exemplo de site estático:



Exemplo de aplicação

- Site dinâmico (aquele em que parte do conteúdo da resposta é gerado dinamicamente, apenas quando necessário):



Desenvolvimento WEB

Existem dois tipos de linguagens para desenvolvermos para internet:

- As linguagens client-side (front-end)
 - As linguagens server-side são linguagens que o SERVIDOR entende.
 - Por exemplo: se criarmos um script em linguagem back-end (PHP, Asp, Java, Ruby, etc) que apenas calcula a soma de $2 + 2$, será o SERVIDOR (ou back, ou server) que calculará este resultado.
- As linguagens server-side (back-end).
 - As linguagens client-side são linguagens onde apenas o seu NAVEGADOR vai entender.
 - Se fizermos esse cálculo em alguma linguagem front-end, como o JavaScript, quem calculará essa conta é o BROWSER do usuário.

Desenvolvimento WEB



Cálculo no Servidor

=



Back-end ou Server-side



Cálculo no Browser

=



Client-side ou Front-end

Cargos

- Profissionais que trabalham na interface do usuário => Desenvolvedores front-end.
- Profissionais que trabalham no core da aplicação, fazendo uma programação que somente o servidor irá entender => Desenvolvedores back-end.
- Existem ainda os Desenvolvedores full-stack, estes atuam na aplicação toda de forma menos aprofundada normalmente.

Linguagens Server-Side

São as linguagens com que os programadores back-end normalmente trabalham:

- Python;
- Java;
- PHP;
- ASP;
- Ruby;
- JavaScript (no caso do Node.js).

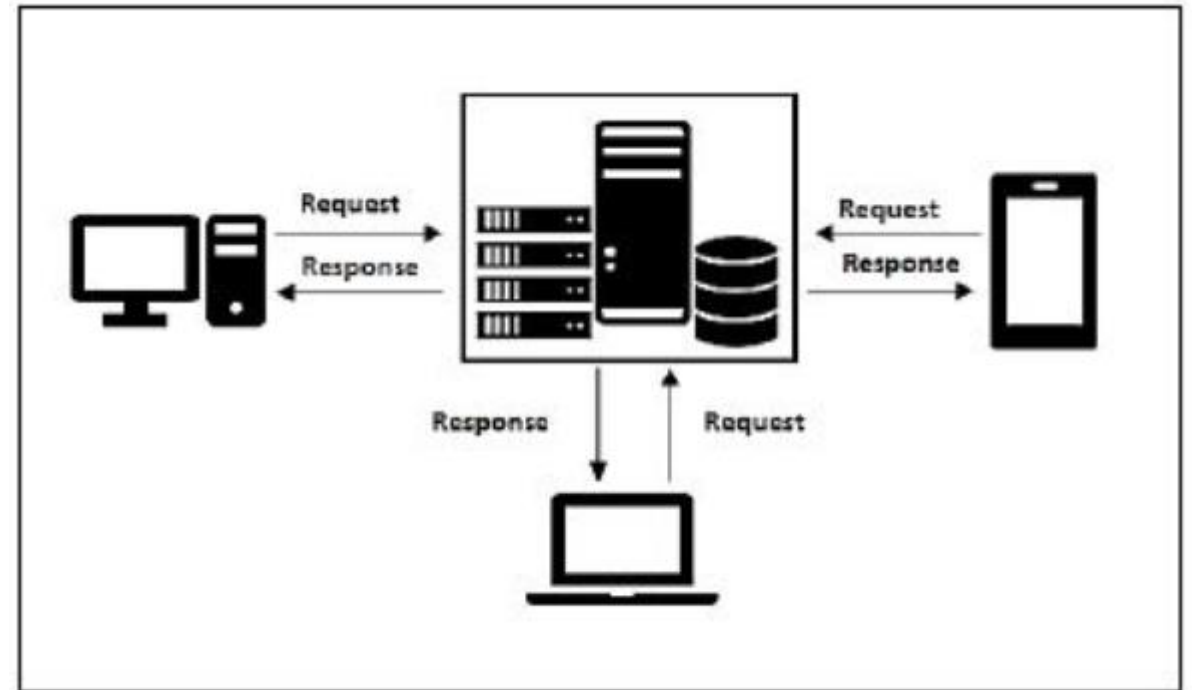
Perguntas para debate

1. Por que usar aplicações com processamento centralizado ao invés de processamento descentralizado (na máquina do usuário)?
2. Cite três desafios para tal abordagem.



Necessidades

- Para isso funcionar algumas coisas são necessárias:
 - Redes;
 - Protocolos.
 - Modelo Cliente-Servidor;
 - Infraestrutura;
 - Tecnologias.
- Para que dispositivos possam se comunicar, diversos protocolos são necessários:
 - HTTP;
 - FTP;
 - SMTP;
 - Entre outros.
- O funcionamento se dá através de requests (requisições) e responses (respostas).



Modelo de comunicação

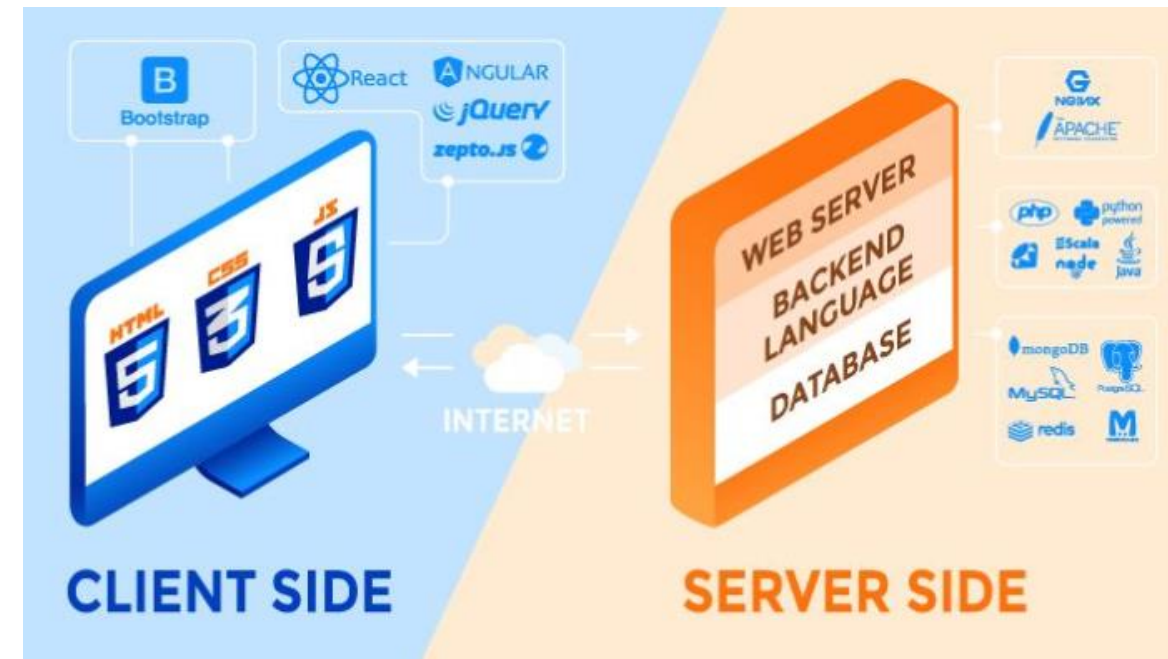
- Para uma requisição poder ser realizada, devemos saber para quem estamos solicitando:
 - Endereço conhecido.
- Clientes normalmente tem endereços mutáveis.
- Servidores normalmente tem endereços imutáveis.
- Com isso, diversos serviços podem ser oferecidos em uma rede:
 - Busca de arquivos;
 - Correio eletrônico;
 - Envio de arquivos de hipertexto;
 - Transmissão de áudio e vídeo;
 - Entre outros.
- Para tal, temos uma vasta lista de protocolos à nossa disposição.

Protocolo	Número da porta
HTTP	80
HTTPS	443
POP3	110
SMTP	25/587
IMAP3	220
IMAP4	143
FTP	20/21
TELNET	23
SSH	22
DNS	53
DHCP	67/68
IRC	194
SNMP	161/162

Desenvolvimento de Aplicações

Uma vez que entendemos o modelo Cliente-Servidor, vamos entender como criamos aplicações neste contexto.

- Client-Side:
 - Aplicações que interagem com o usuário (browser, mobile apps, etc.).
- Server-Side:
 - Aplicações que interagem com o servidor.



Perguntas pra debate

1. Atribuições do lado do cliente?
2. Atribuições do lado do servidor?



Desenvolvimento de Aplicações

- No lado do cliente, normalmente o foco é:
 - UX;
 - UI;
 - Device dependant;
 - Off-line running;
 - Consumir serviços de servidores;



Desenvolvimento de Aplicações

- No lado do servidor, normalmente o foco é:
 - Prover serviços;
 - Ligação com banco de dados;
 - Regras de negócio;
 - Processamento pesado;
 - Entre outros.



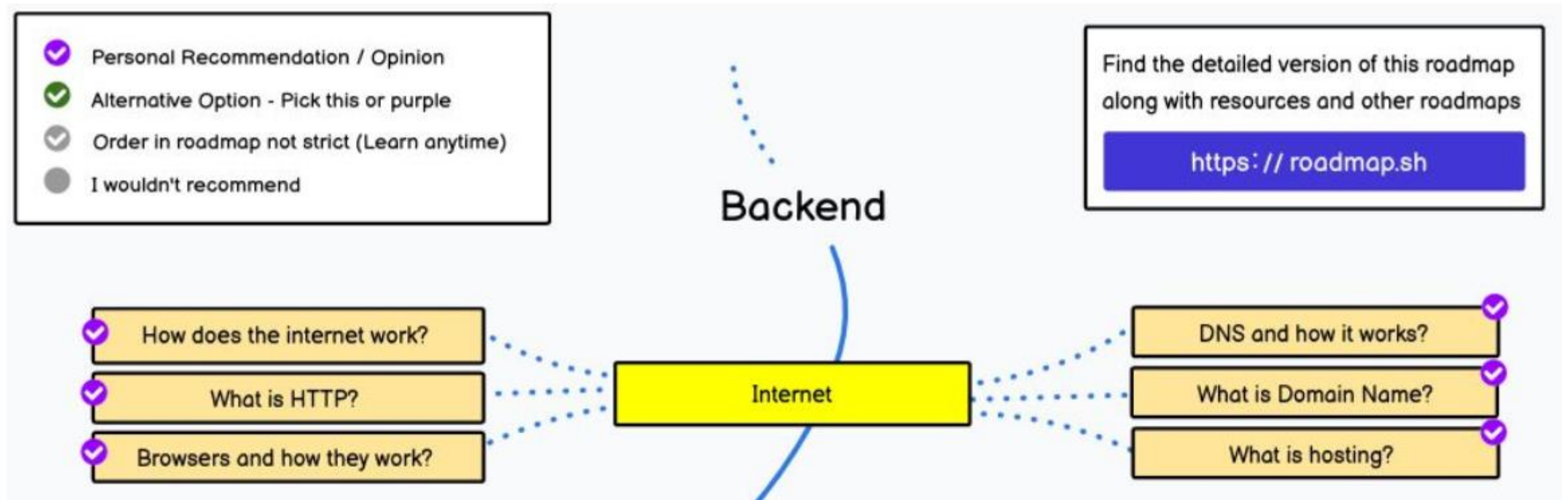
Algumas oportunidades de desenvolvedor

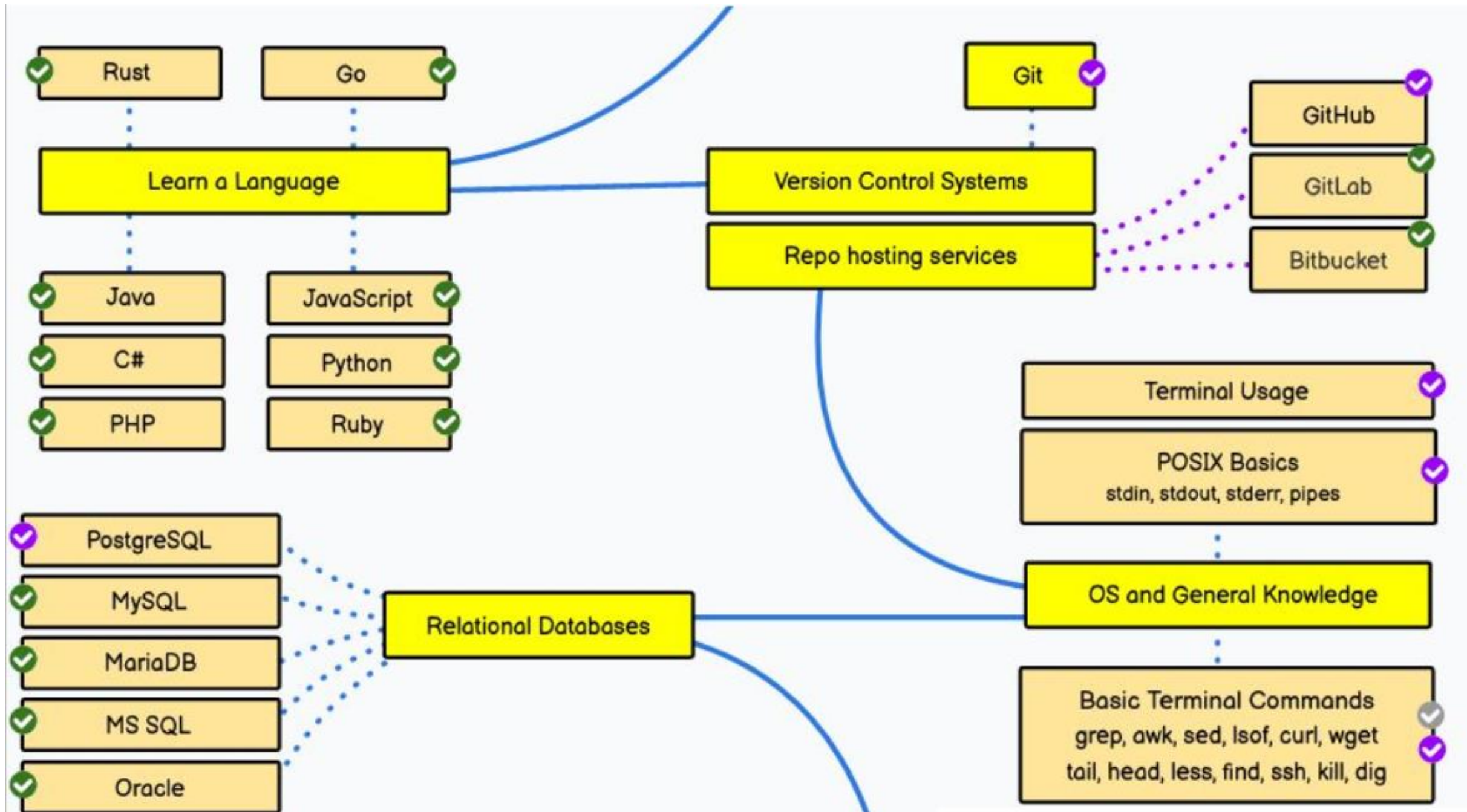
- Vamos ao LinkedIn...

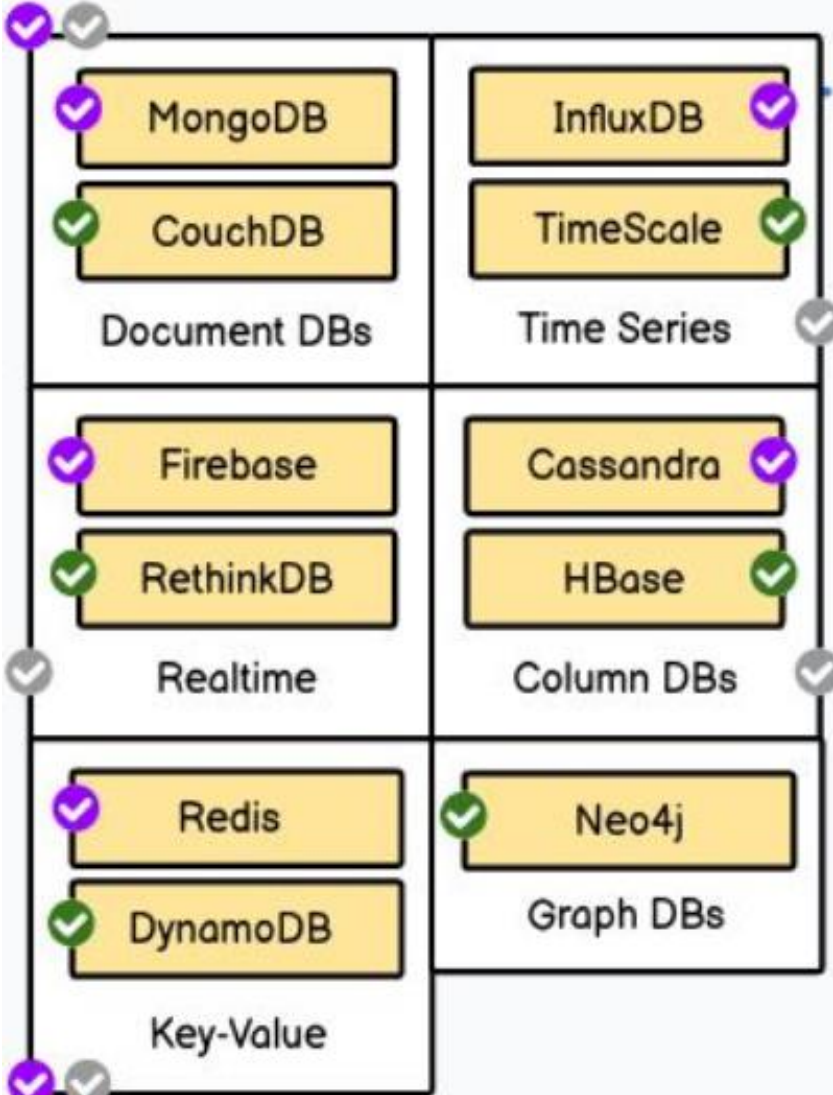
CI/CD

- CI/CD é a abreviação de *Continuous Integration/Continuous Delivery*, ou em português integração e entrega contínuas. É uma prática de desenvolvimento de software que utiliza builds e testes automatizados para tornar a integração de código mais eficiente.
 - Facilita a entrega de aplicações com mais frequência. Porém, necessita de regras de automação nas etapas de desenvolvimento de apps.
- Principais conceitos: integração, entrega e implantação contínuas.
 - Solução de problemas que a integração de novos códigos pode causar às equipes de operações e desenvolvimento, ou os problemas de implementação com os códigos/dados legados.

Roadmap desenvolvedor Back-End







NoSQL Databases

- ORMs
- ACID
- Transactions
- N+1 Problem
- Normalization
- Failure Modes
- Profiling Perfor.

