PROGRAMAÇÃO WEBII

1

Apresentação da disciplina

Competências e Habilidades

Competências

1. Desenvolver sistemas para internet utilizando persistência em banco de dados, interface com o usuário e programação em lado servidor

Habilidades

1.1 Codificar software em linguagem para web.

1.2 Utilizar banco de dados relacionais para persistência dos dados.

1.3 Utilizar interface baseada em navegador para interação com usuário.

Bases Tecnológicas

1. Introdução a scripts lado servidor Geração dinâmicas de páginas; Arquitetura de aplicações Web em camadas (Cliente/Navegador, Servidor Web, Aplicação); Conjunto de tecnologias (Marcação, Estilo, Scripts lado cliente, Scripts lado servidor).

2. Variáveis e tipos de dados Decisão e laços; Funções e procedimentos.
3. Comunicação entre navegador e aplicação URL e QueryString; Métodos HTTP (POST, GET); Formulários; Sessões; Cookies.

4. Persistência em banco de dados Conexões Execução de comandos SQL Operações CRUD Consultas parametrizadas Sanitização e prevenção de SQL Injection e XSS (cross-site scripting).

5. Modularização e organização dos programas Paradigma orientado a objetos Classes e objetos Atributos e métodos Separação em camadas Classes do domínio Do negócio Classes com regras de negócios (business objects) Classes de acesso a dados (data access objects).

Na prática...

- Introdução
- Componentes e Templates
- Data binding
- Diretivas
- Serviços
- Formulários
- Roteamento
- Integração com Servidor
- CRUD

Avaliações

- Exercícios práticos
- Avaliação teórica
- Projeto

Menções:

MB – B – R - I

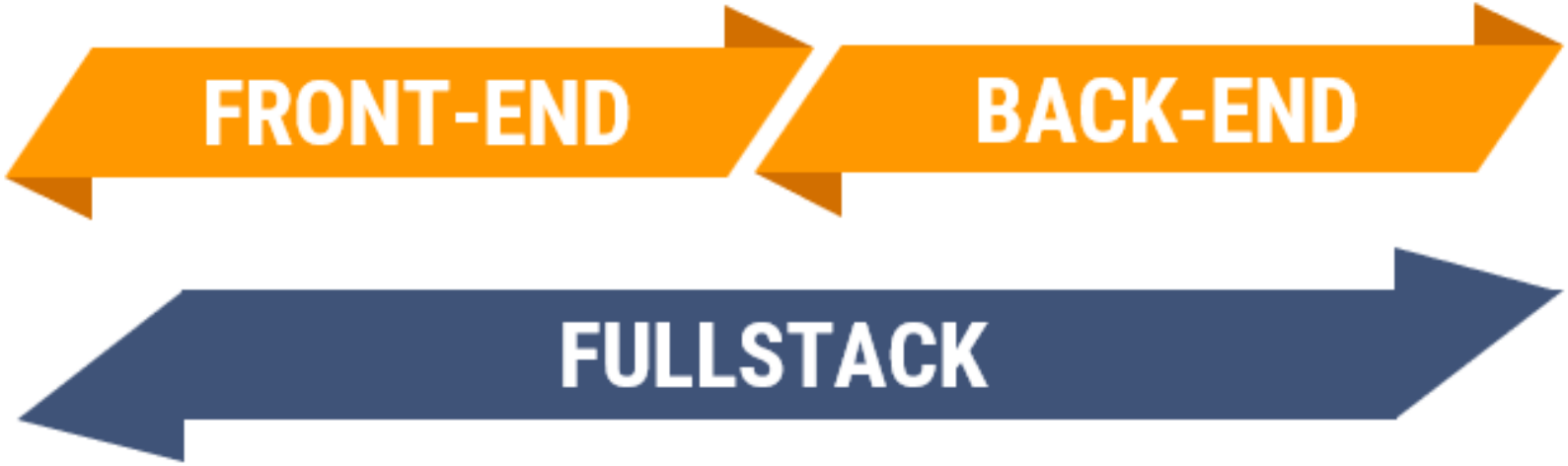
2

Introdução

Aplicações WEB

Mundo do Desenvolvimento

No mundo de desenvolvimento de software, existem três termos muito comuns que provavelmente já nos deparamos:



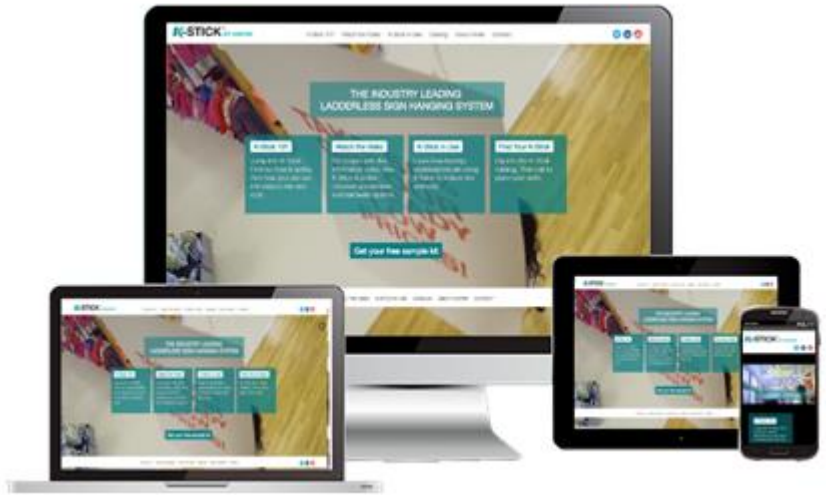
Front End

- Front-End: Para desenvolver essas interfaces, os programadores usam três ferramentas principais:
  - HTML, CSS e JavaScript.
- Outros frameworks foram criados dentro do JavaScript, como o React e o Angular, que facilitam o trabalho de um programador e que permitem a criação de elementos que são posteriormente replicados em outras áreas da plataforma em construção.

Front End

- Front-End abrange a configuração e o design de tudo o que nós vemos quando usamos um site ou um app.

- A tarefa desses programadores é criar interfaces bonitas e intuitivas, tornando a experiência do usuário muito fluida e intuitiva.



Front End


Geralmente, trabalha com linguagens como HTML, CSS e JavaScript
HTML
Não conhecem nada ou conhecem muito pouco de back-end
Não lidam diretamente com banco de dados ou servidores de aplicação complexos
Criam telas com boa usabilidade e carregamento rápido, garantindo o funcionamento nos diferentes browsers

Back-End

- Back-end são aqueles que preferem configurar bancos de dados, descobrir como otimizar o desempenho do servidor e assim poder lidar adequadamente com a carga de trabalho; e são, ainda, aqueles que sabem tirar proveito dos recursos que as APIs de terceiros podem fornecer para alcançar funcionalidades excelentes

Back-End


Geralmente, trabalha com linguagens como Java, C#, PHP, Python, Ruby
Garante todas as regras de negócio
Fornece o acesso a banco de dados, segurança e escalabilidade
Não são muito bons em deixar páginas bonitas e com boa usabilidade

Front-End e Back-End

Full-Stack Developer

- Você não seria o primeiro. Se você se interessa tanto pela parte de design de front-end, quanto pela parte do desenvolvimento back-end, é possível direcionar a sua carreira para se tornar um Full-stack Developer.
- Embora ainda exista algum ceticismo em torno desse perfil, considerando que é melhor se especializar em uma área ou em outra, a verdade é que as empresas estão investindo cada vez mais em Full-stack Developers que tanto são capazes de desenhar as interfaces, como conseguem dominar toda a infraestrutura que existe por detrás.

3

Aplicações Web

Básica

Aplicações Web

Aplicações Web - MPA

- **Aplicações WEB comuns ou MPA (Multi Page Applications)**

  Trabalham com mecanismos de template de frameworks back-end ou simplesmente misturam dados e layout para apresentar as páginas. Camada de dados e apresentação juntas.

- HTML, CSS e JavaScript juntas compõem o Front-End e são interpretadas por qualquer navegador. Os Servidores são a parte que conhecemos por Back-End.

- O modelo de Aplicações Web comum sobrecarrega o servidor (Back-End) com responsabilidades que deveriam ser exclusivas do navegador (Front-End), solicitando que o mesmo processe toda a parte visual juntamente com os pacotes de resposta.

Aplicações Web SPA

Antes disso, no principio tudo era GIFS:

Experiência completa: https://www.cameronsworld.net/

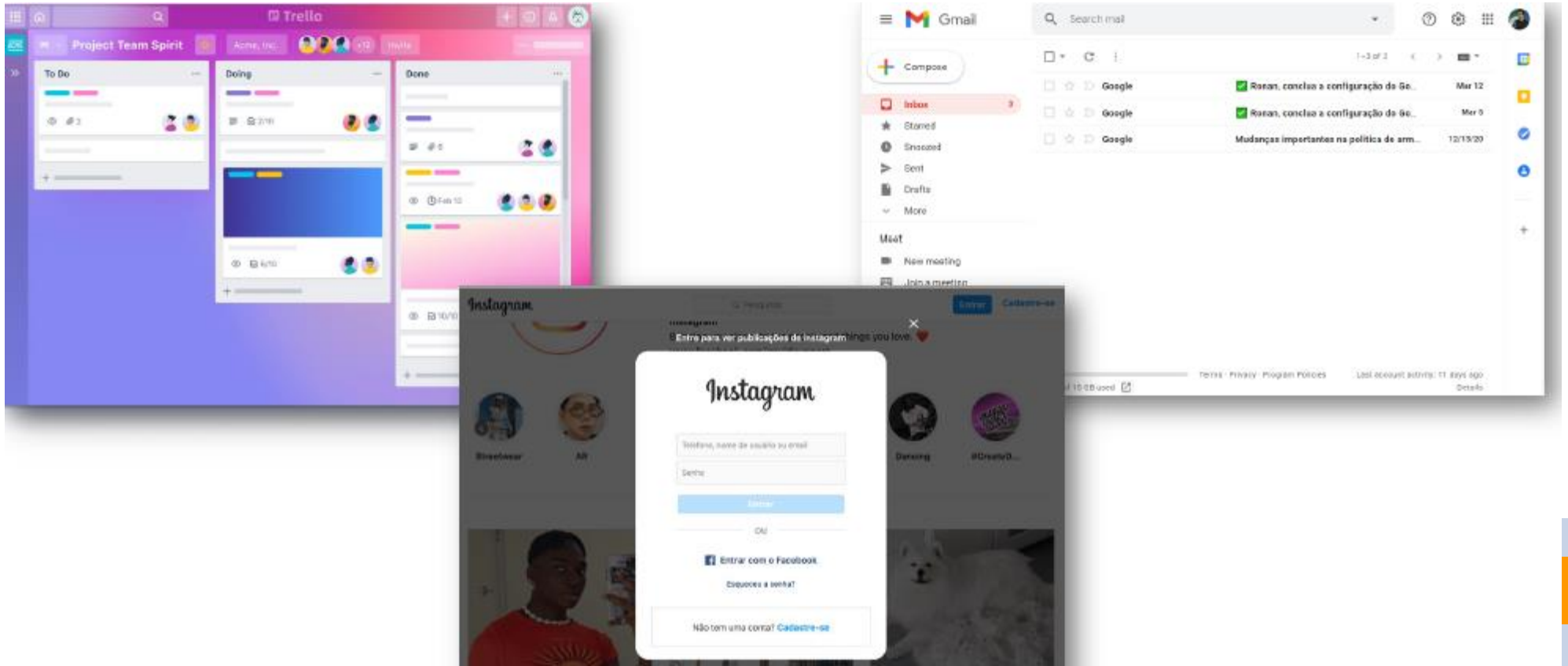
Aplicações Web SPA

- **SPA – Single Page Apllications – Aplicativo de Página Única**

- Com a evolução da linguagem Javascript e a chegada do HTML5, tivemos um grande salto na qualidade e interatividade das páginas de internet e aumento das aplicações web. Aplicações Web são sistemas completos, feitos para serem executados diretamente nos navegadores.
- As ***Single Page Applications*(SPA)** são aplicações web que carregam uma única página HTML e atualiza seu conteúdo dinamicamente, sem a necessidade de recarregar toda a página.
- Com SPA, as aplicações web se comportam semelhantemente a programas nativos desktop ou mobile.
- ExemplosdeSPAs:Gmail,Facebook,Twitter,Instagram,Trello,entreoutros...

Aplicações Web SPA

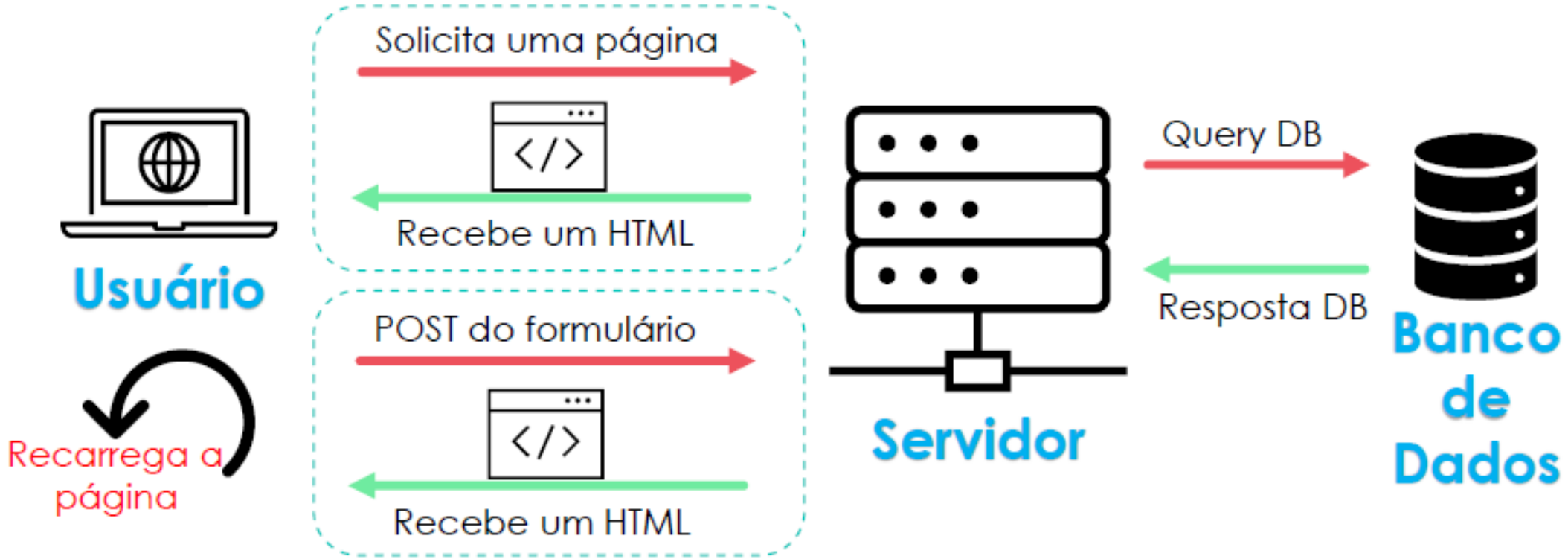
- **Exemplos de SPAs:**
  - Gmail, Facebook, Twitter, Instagram, Trello, entre outros...



Funcionamento - MPA

- As aplicações web tradicionais funcionam no sistema Multi Page Applications(MPA).
- Quando a aplicação precisa de novos dados ou interação com um banco de dados, essa solicitação é enviada ao servidor, que retorna uma nova página HTML completa ao usuário.
- É perceptível ao usuário que uma nova página foi carregada.

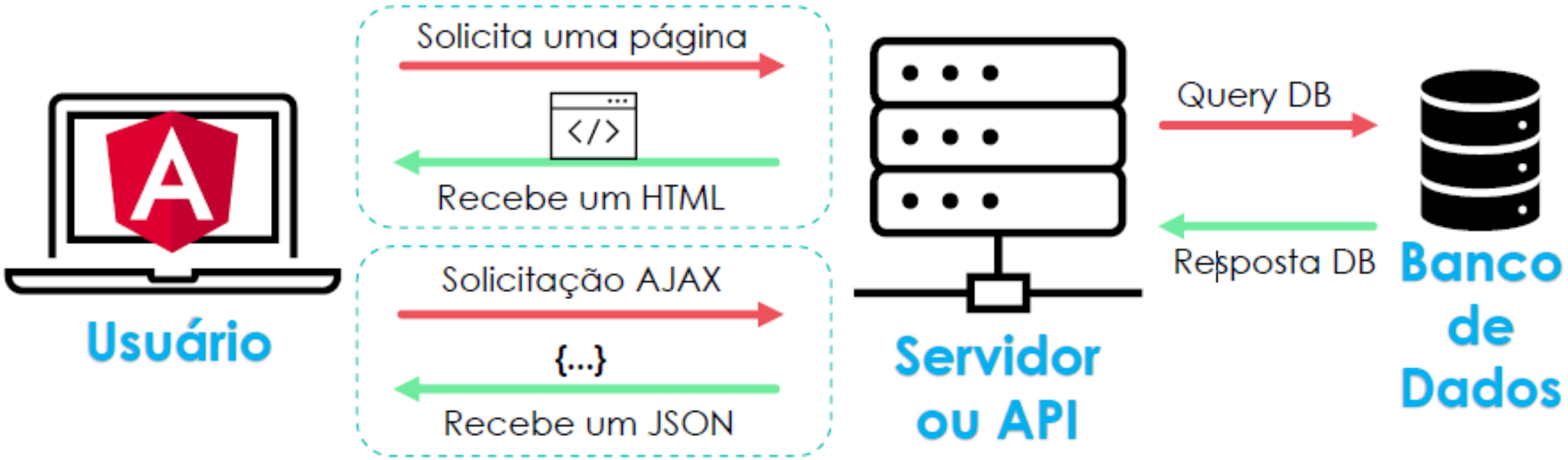
Funcionamento - MPA



Funcionamento - SPA

- Uma única página é carregada com todos os códigos HTML, CSS e Javascript(JS) que compõe a aplicação completa.
- Embora a URL mude, somente os dados e imagens são carregados dinamicamente, geralmente recebidos no formato JSON através de AJAX.
- Todo o processamento ocorre no lado do cliente, utilizando APIs para comunicação com o bancos de dados.

Funcionamento - SPA



Funcionamento - SPA

Nessa imagem podemos ver que a página é formada por vários componentes: menu, barra de navegação, lista de tarefas e cada tarefa são elementos construídos separadamente para funcionar em conjunto.

Figura 1. Conceito de componentes

Vantagens

- **Frameworks facilitam seu desenvolvimento**:
  - Angular, React, Vue.js, Ember.js, entre outros.
- **Experiência do usuário:** O usuário tem uma experiência parecida com programas desktop e mobile, sem precisar de *refresh* na página.
- **Performance**: Carregamento completo inicial. Os demais dados são requisitados de forma assíncrona, conforme a necessidade e utilização do usuário.
- **Manutenção**: Com SPA, o *front-end* é bem separado do *back-end*, permitindo a adoção de times separados e manutenção ou ajustes, sem que a outra parte seja afetada.

Desvantagens

- **Aprendizado**: A curva de aprendizado pode ser complexa no início.
- ***Search Engine Optimization* (SEO ou Otimização para motores de busca):** Apesar dos buscadores atuais já conseguir em indexar retornos AJAX, há constantes dúvidas sobre sua eficácia e nem todos os frameworks possuem otimização para facilitar essa funcionalidade.
- **Performance**: Pode ser um problema caso seja mal otimizado.

Conclusão

- Aplicações Web SPA são melhores do que Aplicações Web comuns nos quesitos principais de aplicações web:
  - Separação de responsabilidades
  - Escalabilidade
  - Manutenção
  - Performance
  - Segurança

SPA ou MPA???

- Acesse o site:
  - www.bbc.com
- Acesse o site:
  - www.linkedin.com.br

Repare na suavidade ao navegar entre as páginas do LinkedIn, o mesmo não ocorre no site da BBC.

SPA ou MPA???

- O primeiro ponto para perceber a diferença entre uma SPA e uma aplicação convencional, é como é feito o carregamento de suas páginas ao clicar em um link interno.
- Em uma SPA, esse carregamento é muito suave, simulando um app de celular.
- Em uma MPA, existe uma tela branca, isso ocorre pois ao clicar no link, o seu navegador faz uma nova requisição para o servidor, o servidor por sua vez retorna essa requisição com uma nova página web completa.
- O mesmo não ocorre na SPA, pois o conteúdo é carregado progressivamente (por meio das famosas API's).

SPA ou MPA???

- Já o esqueleto das páginas, pode ocorrer de duas maneiras:

1. É carregado todo o esqueleto logo na primeira requisição (este é o caso mais comum). O problema desse caso é que dependendo do tamanho da sua aplicação, o arquivo que é baixado é muito grande, o que acaba deixando a aplicação um pouco lenta.
2. O esqueleto é carregado progressivamente (mais complexo). Nesse caso o problema anterior não existe, porém ele é um pouco mais avançado de ser executado e para quem está começando não vai ver isso sendo aplicado logo de cara.

SPA ou MPA???

- Já o esqueleto das páginas, pode ocorrer de duas maneiras:

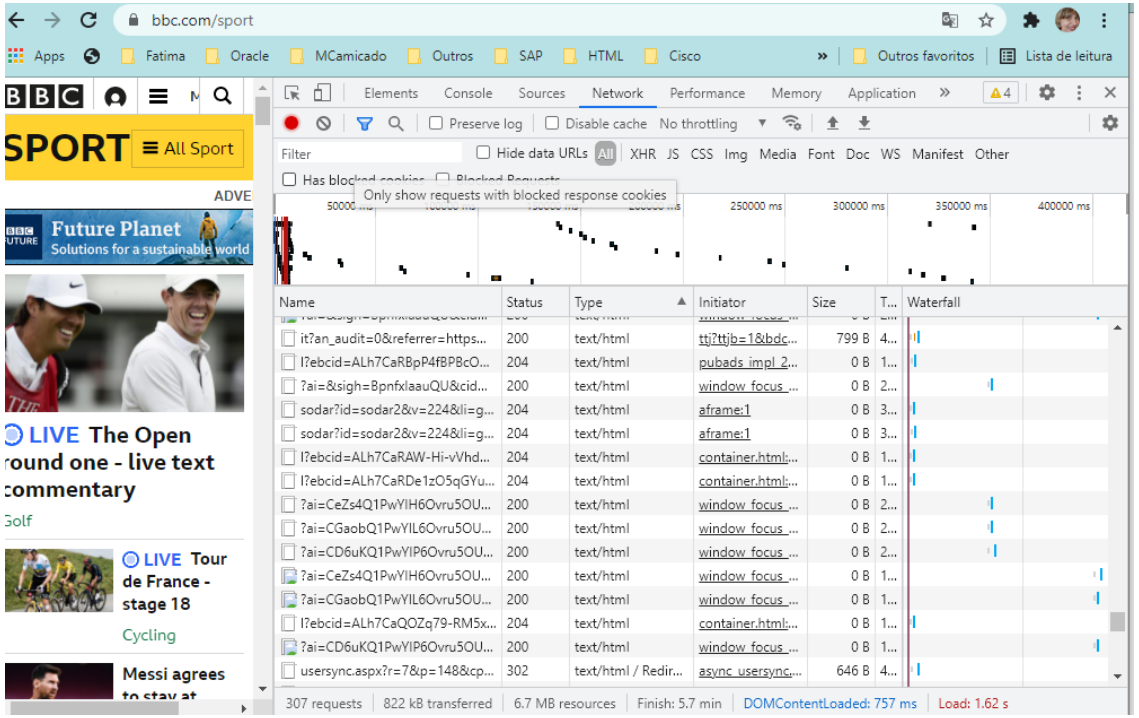
1. É carregado todo o esqueleto logo na primeira requisição (este é o caso mais comum). O problema desse caso é que dependendo do tamanho da sua aplicação, o arquivo que é baixado é muito grande, o que acaba deixando a aplicação um pouco lenta.
2. O esqueleto é carregado progressivamente (mais complexo). Nesse caso o problema anterior não existe, porém ele é um pouco mais avançado de ser executado e para quem está começando não vai ver isso sendo aplicado logo de cara.

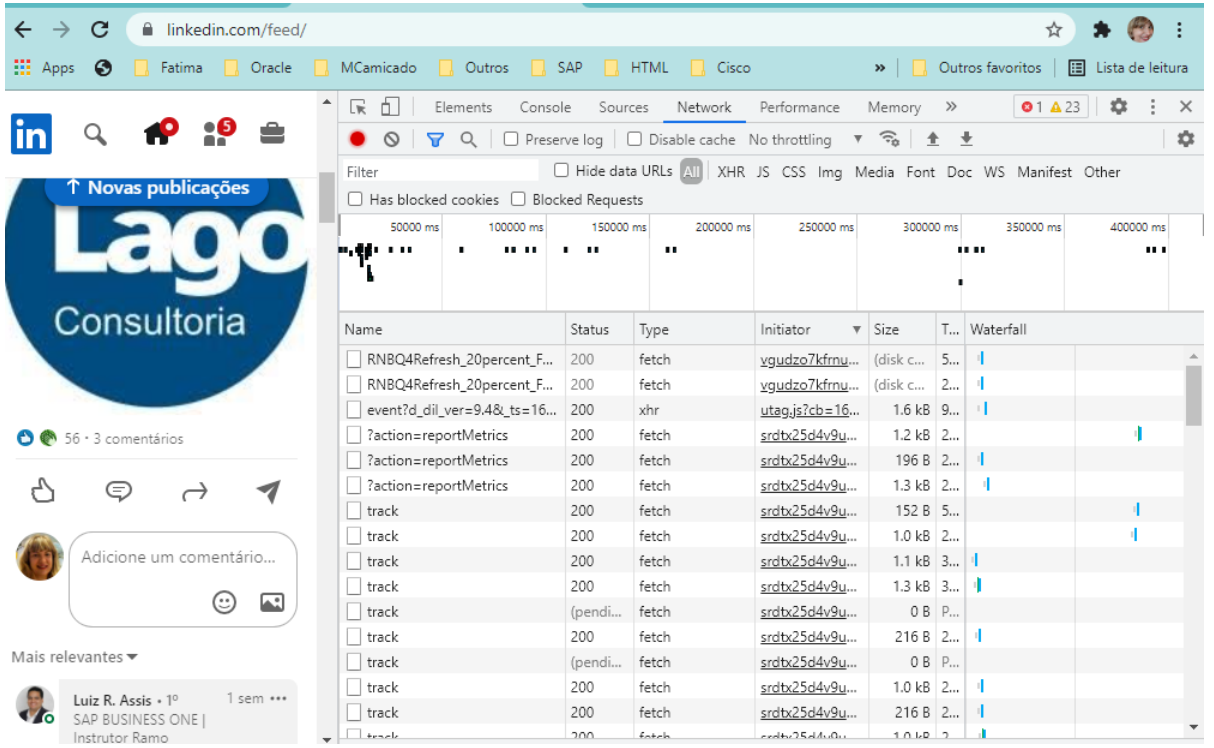
SPA ou MPA???

- Uma outra técnica um pouco mais simples é observar na guia *Network* do console do navegador.
  - Ferramentas de desenvolvedor / Network

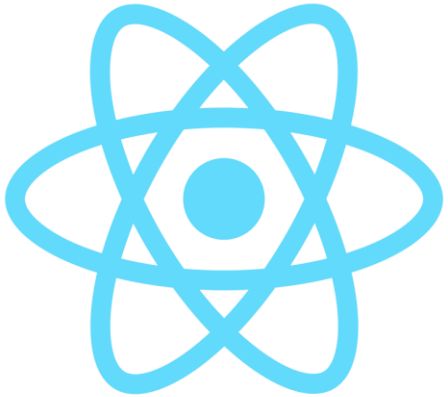
- Se a cada troca de página houver uma requisição para um HTML então não é SPA.

SPA ou MPA???


bbc.com/sport
SPORT
All Sport
LIVE The Open round one - live text commentary
Golf
LIVE Tour de France - stage 18
Cycling
Messi agrees to stay at
Elements Console Sources Network Performance Memory Application
Only show requests with blocked response cookies
307 requests | 822 kB transferred | 6.7 MB resources | Finish: 5.7 min | DOMContentLoaded: 757 ms | Load: 1.62 s


linkedin.com/feed/
↑ Novas publicações
Lago Consultoria
56 • 3 comentários
Adicione um comentário...
Mais relevantes
Luiz R. Assis • 1º
SAP BUSINESS ONE | Instrutor Ramo
Elements Console Sources Network Performance Memory

Principais frameworks JavaScript



Os principais frameworks JavaScript para desenvolvimento front-end em 2020:

https://blog.trainning.com.br/sem-categoria/os-principais-frameworks-javascript-para-desenvolvimento-front-end-em-2020/

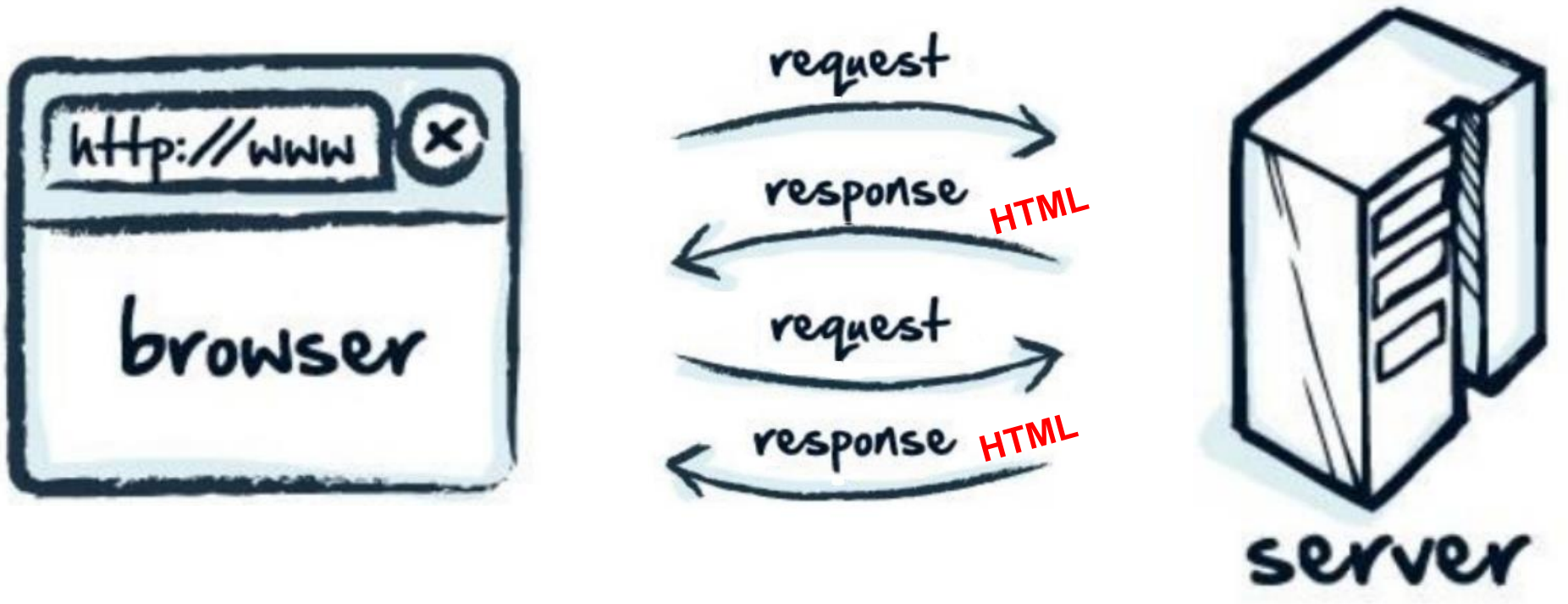
- Hoje, o Angular tornou-se muito avançado e modular para uso no desenvolvimento de front-end.
- A flexibilidade da Angular é louvável. É por isso que as versões 1.x do Angular ainda estão em demanda. No entanto, muitos desenvolvedores atualmente confiam no Angular 2+ devido à sua arquitetura MVC, que mudou substancialmente para uma arquitetura baseada em componentes.
- Angular tem alguns desafios adicionais. Você é quase obrigado a usar o TypeScript para garantir a segurança do tipo nos aplicativos Angular.

4

Arquitetura

Básica

ARQUITETURA TRADICIONAL



ARQUITETURA MODERNA