



ARQUITETURA SERVER-SIDE

Miguel Gomes Xavier - Aula 03 - Aula 01

Professores

ADRIANA CÁSSIA REIS DOS SANTOS

Professora Convidada

Adriana Cássia iniciou sua carreira na área de TI aos 14 anos, em seu primeiro estágio do Ensino Médio como Suporte Técnico, onde permaneceu por seis anos. Após se graduar em Engenharia da Computação, passou a atuar como Analista de Sistemas. Realizou cursos e formações voltadas para programação, entre eles certificação Java (SCJP), migrando para a área de Programação, onde tem atuado nos últimos anos.

MIGUEL GOMES XAVIER

Professor PUCRS

Possui mestrado em Ciência da Computação pela Pontifícia Universidade Católica do Rio Grande do Sul e está cursando doutorado em Ciência da Computação na mesma instituição, atuando principalmente nas áreas de alto desempenho, sistemas distribuídos, virtualização e cloud computing. Atualmente participa de projetos de pesquisa em cooperação com diferentes universidades envolvendo gerência de recursos em arquiteturas de alto desempenho. Tem participado de projetos de análise de dados (BigData), realizando contribuições científicas em prol do avanço da área na indústria e na academia.

Ementa da disciplina

Estudo sobre Arquitetura cliente-servidor para aplicações web. Introdução aos frameworks MVC server-side: Node.js, Express, Nestjs. Estudo de programação assíncrona e programação reativa. Desenvolvimento de aplicações web com o conceito de uso de serviços.

PUCRS online

Desenvolvimento Full Stack

Arquitetura cliente-servidor para aplicações web

Miguel Xavier

miguel.xavier@pucrs.br

Áreas de Interesse e Pesquisa

- **Formação e capacitação**

- Mestrado e Doutorado (PPGCC/PUCRS)

- Projetos de **pesquisa, desenvolvimento e inovação**

- Cloud Computing, Internet of Things (IoT), Big data e HPC, Virtualização e Redes de Computadores



ESCOLA
POLITÉCNICA



<https://www.linkedin.com/in/miguel-xavier/>

PUCRS online

Agenda

- Introdução aos frameworks MVC server-side
 - Node.js, Express, Nestjs
- Estudo de programação assíncrona e programação reativa
- Desenvolvimento de aplicações web com o conceito de uso de serviços

Introdução

Introdução

- Não há dúvida de que a experiência do usuário é impactada pelo tempo de carregamento percebido.
- Com os front-ends mais pesados de hoje em dia, a renderização do lado do cliente não parece ser uma boa opção.

Introdução

- A renderização do lado do servidor está ganhando cada vez mais tração.
- A pré-renderização também é uma estratégia muito boa. Qual é a diferença entre essas soluções e um aplicativo totalmente renderizado pelo cliente?

Renderização do lado do cliente (SSR)

- Existem frameworks como Angular e React que permitem renderização do lado do cliente (browser).
- Com uma solução de renderização do lado do cliente, a solicitação é redirecionada para um único arquivo HTML sem conteúdo (ou com uma tela de carregamento) até que os dados sejam carregados na sequência.
- O browser compila tudo antes de renderizar o conteúdo.
- Com uma conexão de internet boa e confiável, é bem rápido mais rápido.

Renderização do lado do servidor (SSR) - 1

- Uma solução SSR é algo que costumávamos fazer muito, muitos anos atrás.
- Com as antigas soluções de renderização do lado do servidor, era possível criar uma página da Web (com PHP, por exemplo) onde o servidor compilava tudo, incluía os dados e entregava uma página HTML totalmente preenchida ao cliente.

Renderização do lado do servidor (SSR) - 1

- Mas... toda vez que usuário navegava para uma página diferente, o servidor tinha que fazer todo o trabalho de novo: obter o arquivo PHP, compilá-lo e entregar o HTML, com todo o CSS e JS atrasando o carregamento da página para algumas centenas de ms ou mesmo segundos inteiros.

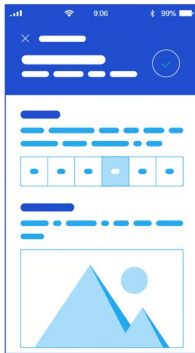
Renderização do lado do servidor (SSR) - 2

- E se você pudesse carregar a primeira página com a solução SSR e depois usar um framework para fazer roteamento dinâmico com AJAX, buscando apenas os dados necessários?
- É por isso que o SSR está ganhando cada vez mais força na comunidade
- React, Angular, etc.

Renderização do lado do servidor (SSR) - 3

- De qualquer forma, a vantagem dessa solução é poder desenvolver um app server-side e client-side com o mesmo código e entregar uma experiência muito rápida ao usuário com dados customizados.
- O SSR é usado para buscar dados e pré-preencher uma página com conteúdo personalizado, aproveitando a conexão de internet confiável do servidor. Ou seja, a própria conexão de internet do servidor é melhor do que a de um usuário), portanto, é possível pré-buscar dados antes de entregá-los ao usuário.

Pré-renderização



Pré-renderização

- Há outro problema: para que essa técnica funcione, você ainda precisa ter um proxy ou algo que redirecione o usuário para o arquivo correto.
- Por que?

Pré-renderização

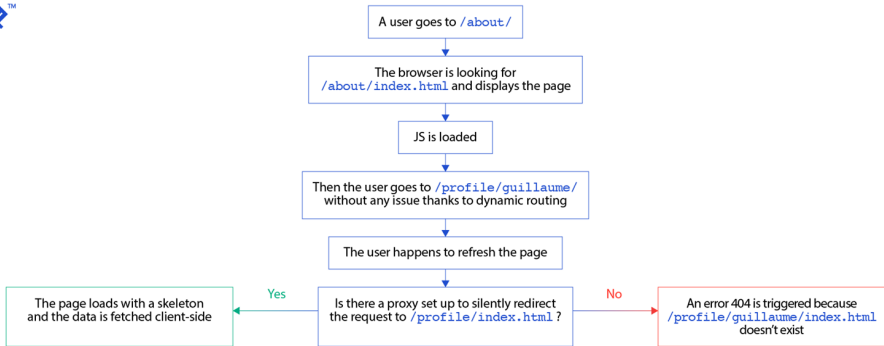
- Com um aplicativo de página única, é preciso redirecionar todas as solicitações para o arquivo raiz e, em seguida, a estrutura redireciona o usuário com seu sistema de roteamento integrado.
- Portanto, o carregamento da primeira página é sempre o mesmo arquivo raiz.

Pré-renderização

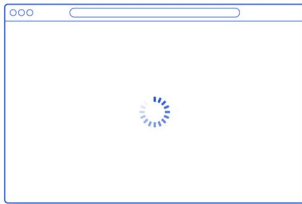
- A solução é informar ao seu proxy que cada uma dessas quatro rotas precisa de um arquivo específico:
 - `https://my-website.com` → Redirecionar para o `index.html` arquivo raiz
 - `https://my-website.com/about` → Redirecionar para o `/about/index.html` arquivo
 - `https://my-website.com/jobs` → Redirecionar para o `/jobs/index.html` arquivo
 - `https://my-website.com/blog` → Redirecionar para o `/blog/index.html` arquivo

Esta solução pode ser útil para pequenos aplicativos - você pode ver como seria doloroso se tivesse algumas centenas de páginas.

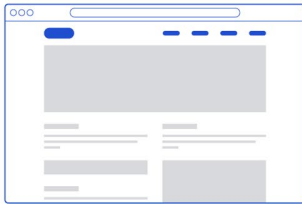
Pré-renderização



Pré-renderização



Loading screen



Skeleton



Fully-rendered page

A renderização somente do cliente geralmente não é suficiente

- Dependendo da estratégia, às vezes usamos todas essas três exibições e, às vezes, saltamos direto para uma página totalmente renderizada. Apenas em um caso de uso somos forçados a usar uma abordagem diferente:

Um amplo ecossistema...



PUCRS online

Desenvolvimento Full Stack

Web Services

Miguel Xavier

miguel.xavier@pucrs.br

PUCRS online  **uol** edtech.