Curso

Aplicações JAVA com SPRING BOOT

Prof. Msc. Antonio B. C. Sampaio Jr
ENGENHEIRO DE SOFTWARE & PROFESSOR

@abctreinamentos
@amazoncodebr

www.abctreinamentos.com.br
www.amazoncode.com.br

CONTEÚDO PROGRAMÁTICO

- UNIDADE 1 – INTRODUÇÃO
- UNIDADE 2 – FUNDAMENTOS DO SPRING BOOT
- UNIDADE 3 – PERSISTÊNCIA DE DADOS NO SPRING BOOT
- UNIDADE 4 – PROJETO WEB NO SPRING BOOT
- UNIDADE 5 – PROJETO REST API NO SPRING BOOT
- UNIDADE 6 – PROJETO REST API NO SPRING BOOT COM REACTJS

PROJETOS DO CURSO

- 1º Projeto Spring Boot – Impressão de Mensagens
- 2° Projeto Spring Boot – Impressão de Mensagens na WEB
- 3° Projeto Spring Boot – Aplicação Servidor Público
- 4° Projeto Spring Boot – Aplicação Servidor Público na WEB
- 5° Projeto Spring Boot – Aplicação Servidor Público no SGBD MYSQL
- 6° Projeto Spring Boot – Aplicação Servidor Público no MONGO DB

PROJETOS DO CURSO

- 7° Projeto Spring Boot – Aplicação Servidor Público WEB
- 8° Projeto Spring Boot – Aplicação Servidor Público REST API e MySQL
- 9° Projeto Spring Boot – Aplicação Servidor Público REST API com REACT
- 10° Projeto Spring Boot – Aplicação Servidor Público/Curso REST API – Monolítico
- 11° Projeto Spring Boot – Aplicação Servidor Público REST API - Microsserviços

UNIDADE 6

PROJETO REST API NO SPRING BOOT COM REACT

FRAMEWORKS FRONT-END

REACT

NODE.JS

JAVASCRIPT

AMBIENTE DE INTEGRAÇÃO

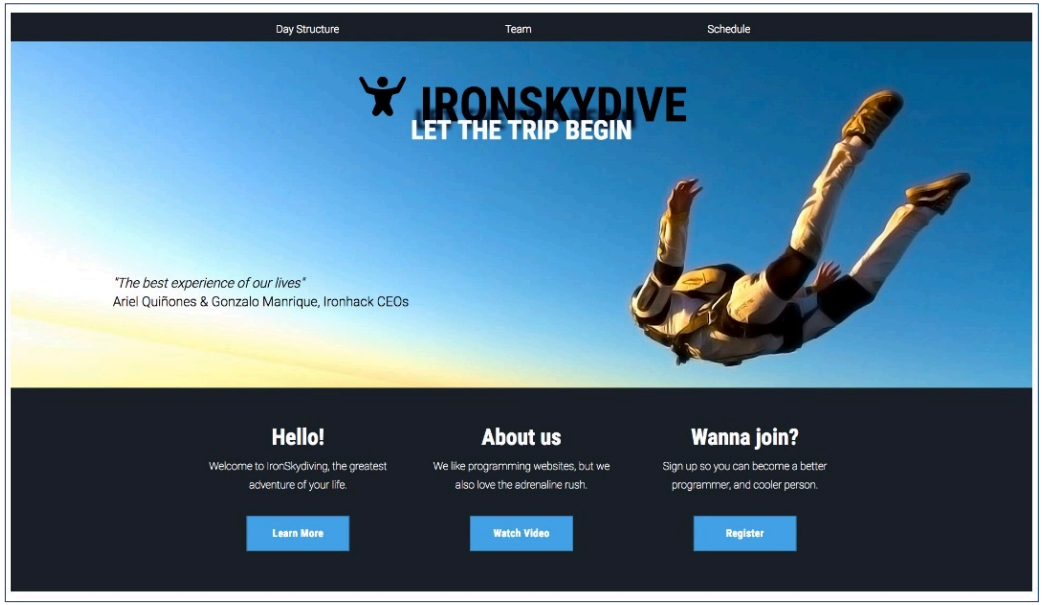
PROJETO PRÁTICO

9° Projeto Spring Boot – Aplicação Servidor Público REST API com REACT e MySQL

FRONT-END FRAMEWORKS

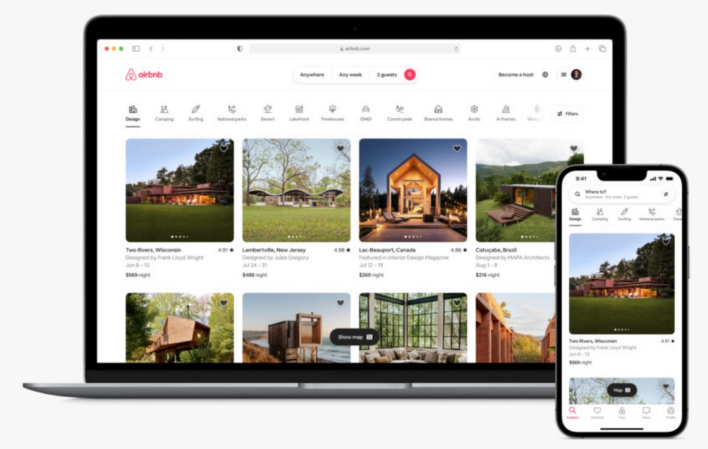
FRONT-END FRAMEWORKS

- **Definição**
  - A maneira mais simples de criar uma Aplicação Front-End é usando HTML, CSS e JavaScript. Esta é a maneira mais direta para criar sites web. Não é necessário o uso de ferramentas ou bibliotecas.



FRONT-END FRAMEWORKS

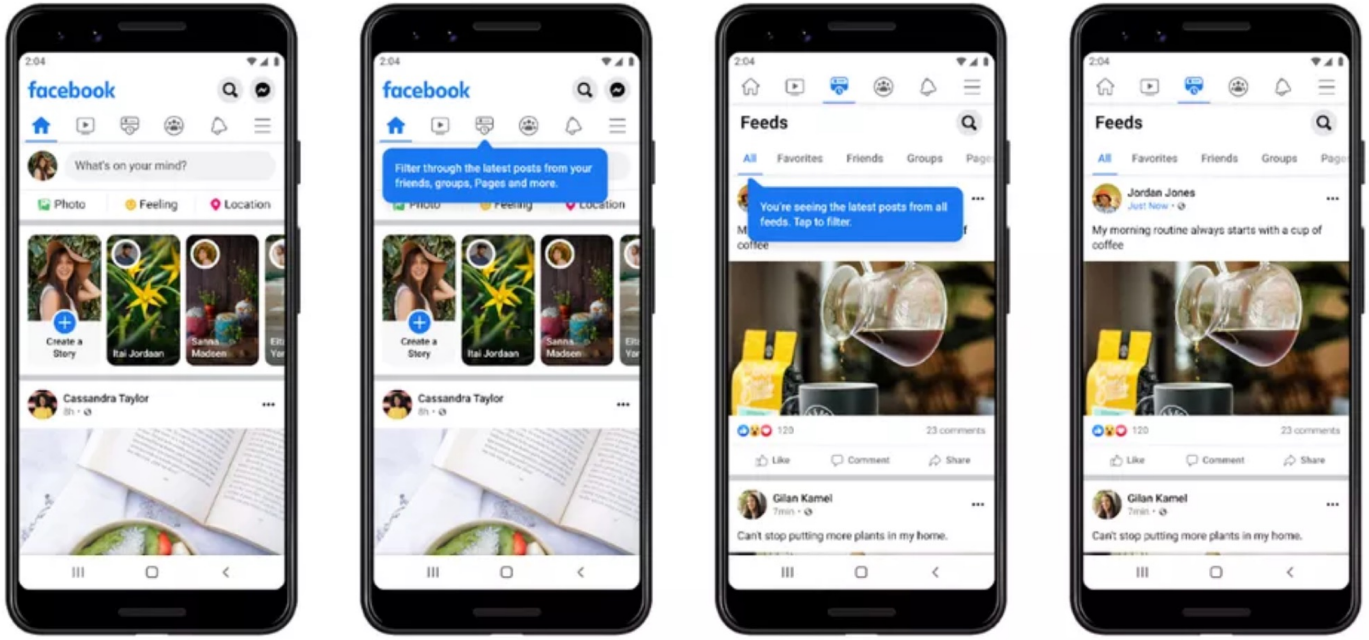
- **Definição**
  - No entanto, essa abordagem não é muito interessante ao se deparar como sites WEB grandes e complexos, pois, à medida que eles crescem, fica difícil de manter e escalar.



  - Por exemplo, pense em uma plataforma WEB como Facebook, Airbnb ou Instagram. Quantos arquivos HTML, CSS e JS são necessários para a criação dessas plataformas!? A resposta são milhares de arquivos.

FRONT-END FRAMEWORKS

- **Curiosidade** 😉
  - O Front-End do Facebook possui mais de 60 milhões de linhas de código!



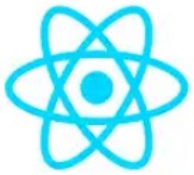
https://g1.globo.com/tecnologia/noticia/2022/07/21/facebook-muda-tela-inicial-para-mostrar-mais-conteudos-que-os-usuarios-nao-seguem.ghtml

FRONT-END FRAMEWORKS

- **Como Resolver esse Problema?**
  - Uma das alternativas é fazer uso dos Front-End Frameworks, pois elas auxiliam na criação de aplicações WEB com muitas funcionalidades, recursos visuais e agilidade.

  - Abaixo os mais utilizados pelo mercado:

REACT JS



ANGULAR

VUE JS


REACT

REACT

- **Definição**
  - O React é uma biblioteca JavaScript Front-End usada para criar aplicativos que são executados no navegador. Ele foi criado por Jordan Walke, um engenheiro de software que trabalha para o Facebook. O React foi implantado pela primeira vez no feed de notícias do Facebook em 2011 e no Instagram.com em 2012.

  - O React permite que os desenvolvedores criem aplicativos Web robustos que lidam com alterações dinâmicas nos dados sem recarregar a página. Além de aplicativos Web, também podemos usar o React para desenvolver aplicativos móveis Android e iOS. Isso é feito por meio da estrutura multiplataforma chamada React Native.

REACT

- **Vantagens**
  - Desenvolvimento baseado em Componentes – Aplicações feitas em React são baseadas em componentes que são partes menores reutilizáveis da interface do usuário.

  - Separação de preocupações - O React permite um design limpo e modular das aplicações. Os desenvolvedores podem trabalhar em diferentes partes do aplicativo, em um componente específico. Dessa forma, eles não interferem no trabalho de outras pessoas ou quebram o código existente.

  - Velocidade - O React permite manter uma estrutura de projeto clara que acelera o desenvolvimento.

REACT

- **Vantagens**
  - Impõe estrutura - O React impõe determinados padrões de design e estrutura de projeto. Se todos os desenvolvedores da equipe seguirem as mesmas práticas, a colaboração melhora de forma significativa.

REACT

- **Características Essenciais**
  - Baseado em Componentes – Os Componentes são blocos de construção de aplicativos React.

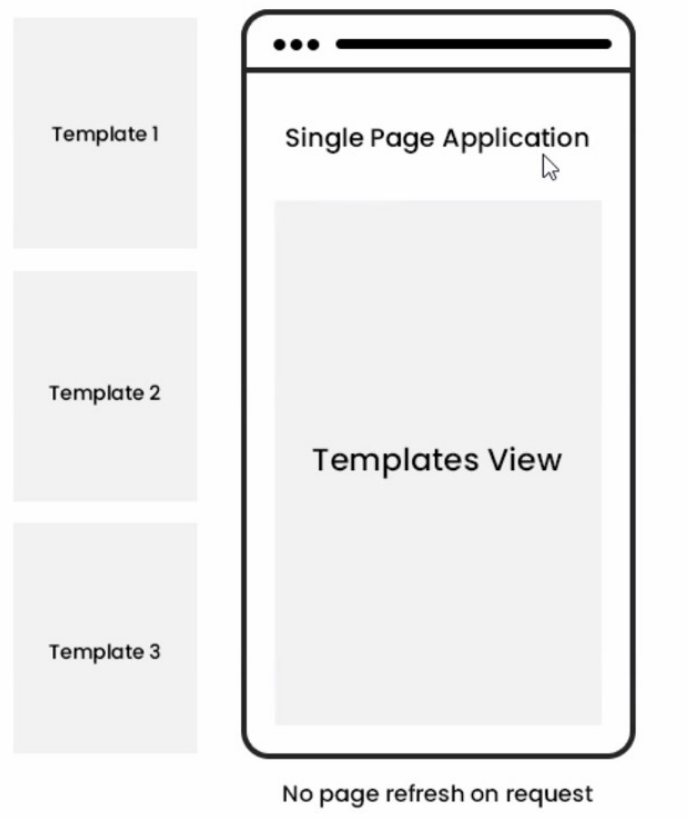
  - JSX – É uma extensão de sintaxe JavaScript usada para escrever HTML e JavaScript juntos. Em vez de separar JavaScript e HTML em arquivos diferentes, o React usa JSX para combinar HTML e JS juntos. O JSX acelera e simplifica a criação de aplicativos e componentes React.

  - Aprenda uma vez, escreva em qualquer lugar - O React pode ser usado para pré-renderizar páginas da Web no servidor com o Node. Além de aplicativos Web, podemos usar o React para criar aplicativos móveis com o React Native.

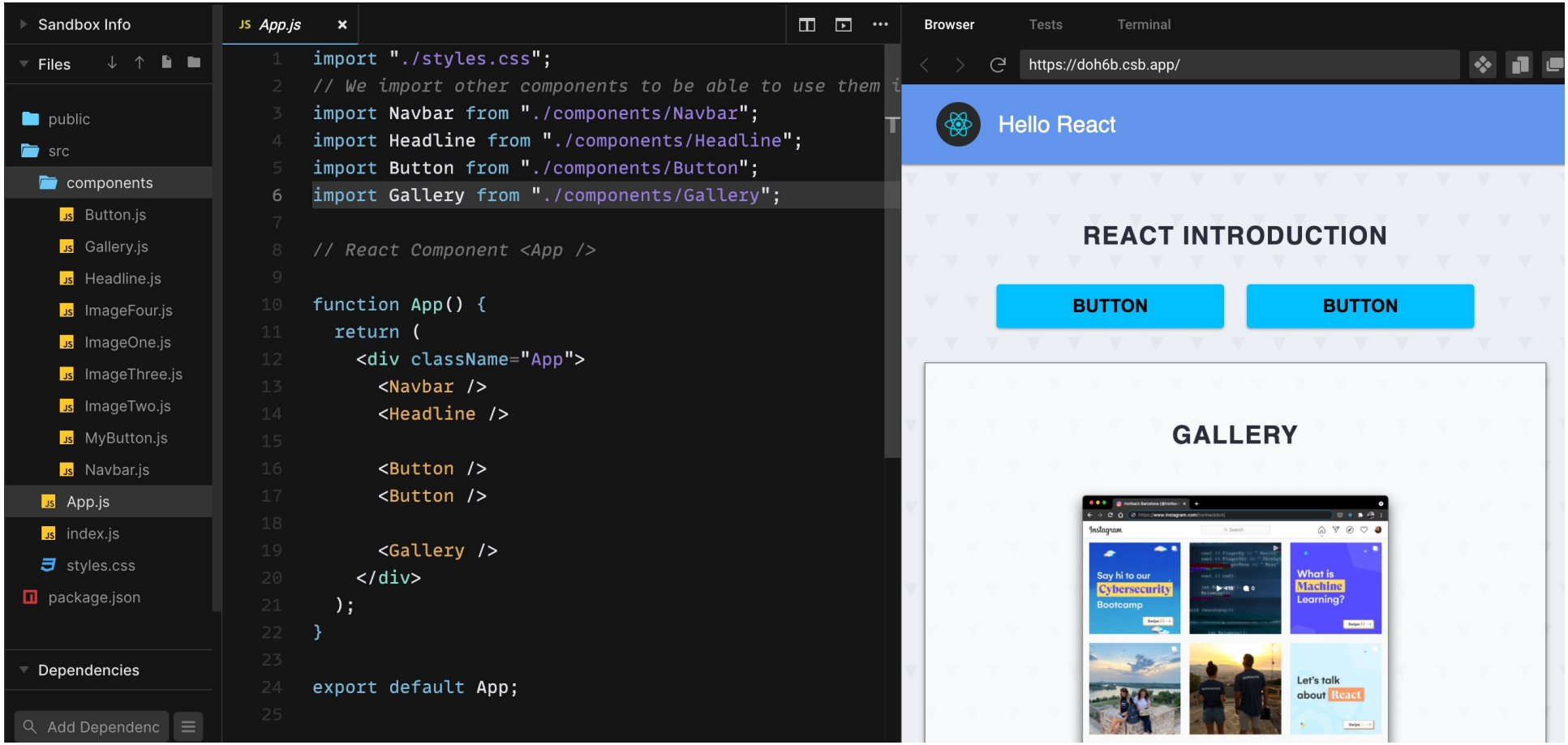
ESTRUTURA REACT

ESTRUTURA REACT

- **Principal Uso**
  - Criação de páginas SPA.



ESTRUTURA REACT



https://codesandbox.io/s/m3-d1-react-introduction-v2-doh6b?from-embed=&file=/src/App.js

ESTRUTURA REACT

- **Passo a Passo para a criação de Aplicativo REACT**
  - (1) Configuração do ambiente de desenvolvimento:
    - Instalar o Node.js
    - Usar o npm (gerenciador de pacotes do Node.js) para inicializar um novo projeto React.
    - Instalar o Create React App (CRA) globalmente: npm install -g create-react-app.
  - (2) Criação de um novo projeto React:
    - Executar o comando create-react-app nome-do-projeto para criar um novo projeto React com o CRA.
    - Navegar para o diretório do projeto: cd nome-do-projeto.

ESTRUTURA REACT

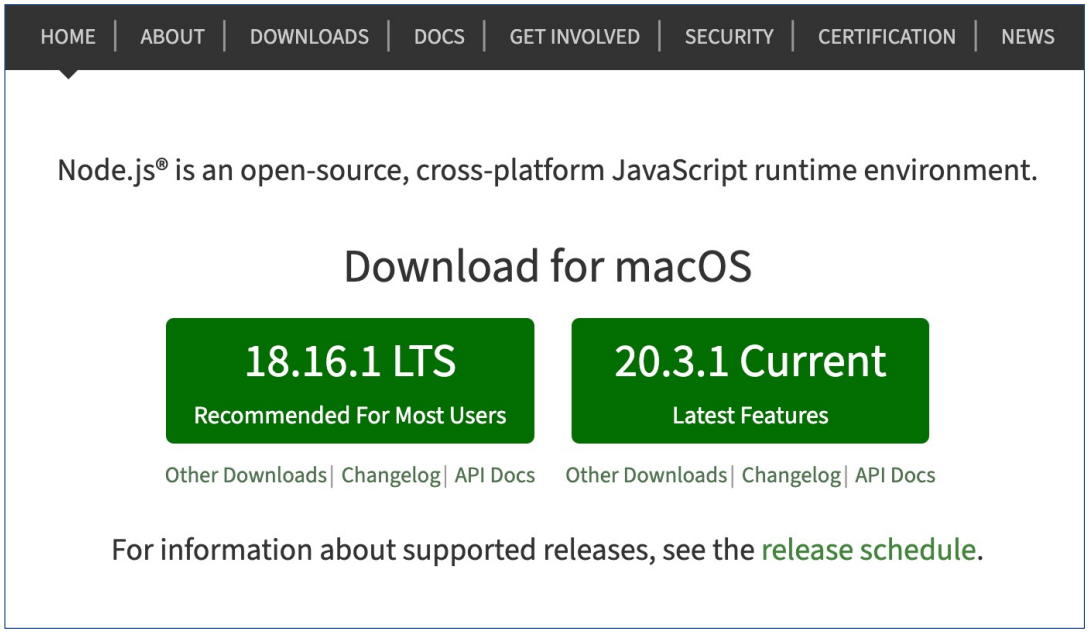
- **Passo a Passo para a criação de Aplicativo REACT**
  - (3) Definição de componentes:
    - Dentro da pasta do projeto, estará a estrutura de diretórios. Nela o componente principal é o arquivo src/App.js. Esse arquivo deve ser alterado para definir a estrutura básica da página SPA desejada.
    - Pode-se criar componentes adicionais em src/components para dividir o código e reutilizar partes da interface do usuário.
  - (4) Roteamento:
    - Para adicionar o roteamento na página SPA, deve-se usar a biblioteca React Router através da sua instalação (npm install react-router-dom).
    - As rotas são configuradas em src/App.js ou em um componente separado para lidar com a renderização de componentes específicos em diferentes URLs.

ESTRUTURA REACT

- **Passo a Passo para a criação de Aplicativo REACT**
  - (5) Estilização:
    - É possível estilizar os componentes REACT usando CSS ou bibliotecas de estilos como Bootstrap, Material-UI ou Tailwind CSS. Basta instalar as dependências necessárias e importar os estilos em seus componentes.
  - (6) Teste e execução:
    - Utilizar o comando npm start para iniciar o servidor de desenvolvimento e visualizar a página SPA no navegador.
    - Fazer ajustes nos componentes, adicionar estilos e/ou funcionalidades conforme necessário.

NODE.JS

- **Definição**
  - NodeJS (ou simplesmente Node) é um ambiente de tempo de execução JavaScript desenvolvido por Ryan Dahl (Google) que permite executar código JavaScript fora de um navegador.
  - Node permitiu executar código JavaScript em qualquer lugar, incluindo servidores web.



https://nodejs.org/en

- **Definição**
  - O Node não é um framework e muito menos uma linguagem de programação. Ele é um ambiente de tempo de execução de código aberto, baseado no motor JavaScript V8 do Chrome, que permite a execução de código JavaScript fora do navegador. Ele permite executar código JavaScript no lado do servidor, em vez de apenas no navegador, permitindo a construção de aplicativos de servidor escaláveis e de alto desempenho.

  - O Node possui NPM (*Node Package Manager*) que é o gerenciador de pacotes padrão. Ele é uma ferramenta que permite aos desenvolvedores instalar, gerenciar e compartilhar bibliotecas e pacotes de código JavaScript reutilizáveis.

  - Com o NPM, é fácil adicionar pacotes e dependências aos projetos Node.js.

- **NPM**
  - O NPM é amplamente utilizado na comunidade de desenvolvimento JavaScript e Node.js devido à sua facilidade de uso e ao ecossistema rico de pacotes disponíveis. Ele facilita a adição de funcionalidades aos seus projetos, economizando tempo e esforço ao aproveitar soluções já desenvolvidas e testadas pela comunidade.

  - Para utilizar o NPM, é necessário ter o Node.js instalado, pois o NPM é instalado automaticamente juntamente com o Node.js. Depois de instalar o Node.js, será possível usar o NPM no terminal ou prompt de comando para gerenciar as dependências dos projetos.

- **Principais Usos**
  - Ao contrário do JavaScript tradicional, que é executado no navegador e se concentra principalmente em interações com a interface do usuário, o Node permite a criação de aplicativos de servidor completos usando JavaScript. Ele fornece recursos e bibliotecas para lidar com operações de I/O de forma assíncrona e não bloqueante, o que o torna eficiente e adequado para aplicações em tempo real, manipulação de arquivos, conexões de rede e muito mais.

  - Node é amplamente utilizado para desenvolvimento de servidores web, APIs RESTful, aplicações em tempo real, microsserviços, aplicações de streaming, entre outros. Ele se tornou uma opção popular para desenvolvedores devido à sua eficiência, escalabilidade e facilidade de uso.

- **Exemplo**

```
// instalar o pacote “colors”
$ npm install colors
```

```
// rodar o código
$ node index.js
```

```
const myColors = require("colors/safe");


console.log(myColors.yellow('hello'));
console.log(myColors.red.underline('i like
cake and pies'))
console.log(myColors.inverse('inverse the
color'));
console.log(myColors.rainbow('OMG
Rainbows!'));
console.log(myColors.trap('Run the trap'));
```

index.js

```
~/Desktop/npm-getting-started » node index.js
hello
i like cake and pies
inverse the color
OMG Rainbows!
RUn tHE tRAP
```

JAVASCRIPT

JAVASCRIPT

- **Definição**
  - JavaScript é uma linguagem de programação interpretada que utiliza pouca memória durante a sua execução e que é baseada em funções. Linguagem interpretada é aquela que executa o programa traduzindo diretamente cada linha de código em código de máquina.
  - Foi criada em 1995 para ser incluída em páginas HTML para serem executadas pelos navegadores.
  - O JavaScript não deve ser confundido com a linguagem de programação Java, pois possuem filosofias de funcionamento bem diferentes.
  - Recentemente, a popularidade do JavaScript se expandiu ainda mais através do bem-sucedido Node.js, o interpretador multiplataforma mais popular de ambiente de execução JavaScript fora do navegador.

JAVASCRIPT

- **Baseada em Funções**
  - As funções são os principais "blocos de construção" de qualquer programa. Eles permitem que o código seja reutilizado muitas vezes sem repetição.
  - A declaração de função é o processo de criar uma função, mas não executá-la.

```
function nomeFuncao(parâmetros) {
  // ....
}
nomeFuncao(parâmetros);
```

  - O processo de execução chamando a função é conhecido como invocação de função.

JAVASCRIPT

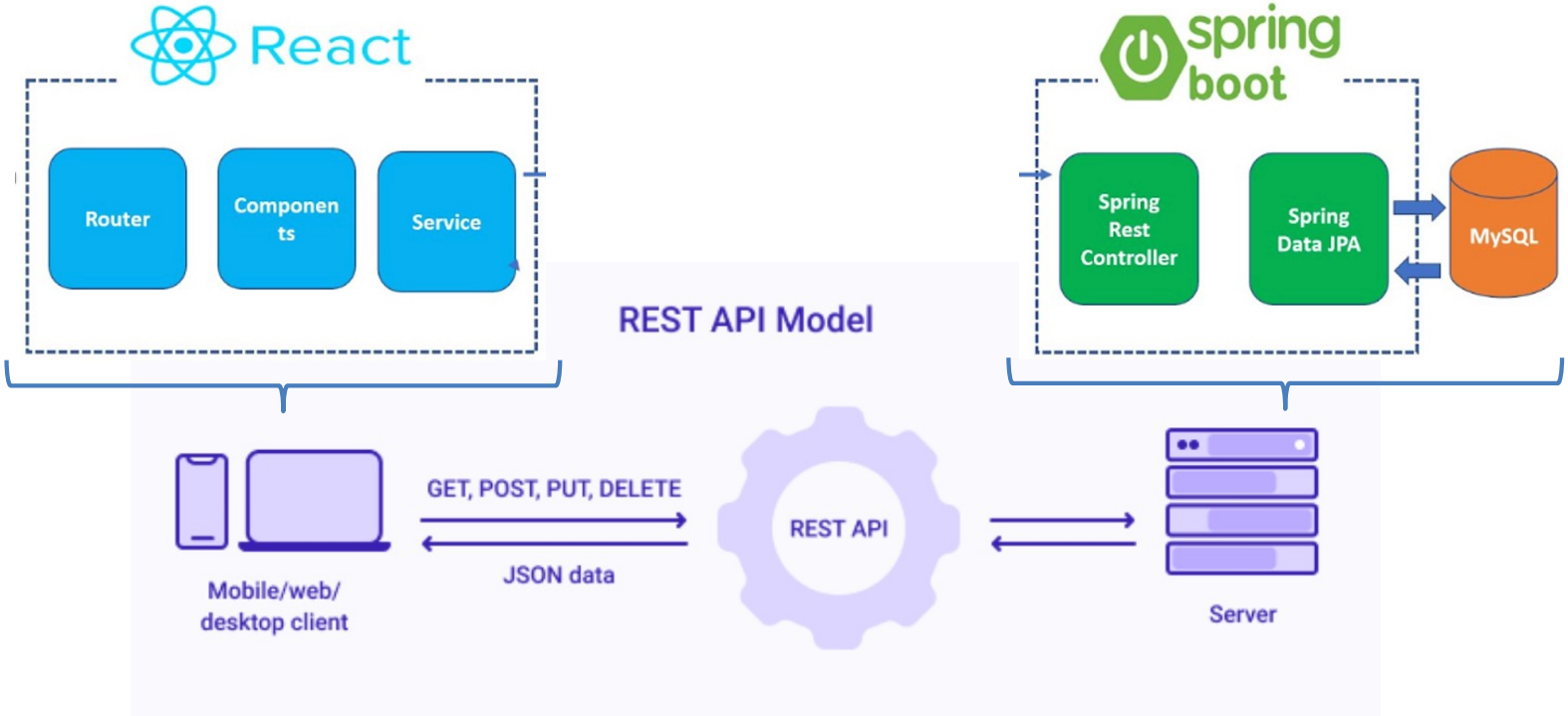
- **Exemplo**

```
// declaração da função
function digaAlo(aluno1, aluno2, aluno3) {
  console.log(`Alo ${aluno1}, ${aluno2} e ${aluno3}!`);
}

// invocação da função
digaAlo('João', 'José', 'Maria');
// saída: Alo João, José e Maria!
```

AMBIENTE DE INTEGRAÇÃO

INTEGRAÇÃO REACT e SPRING BOOT

- **Arquitetura**



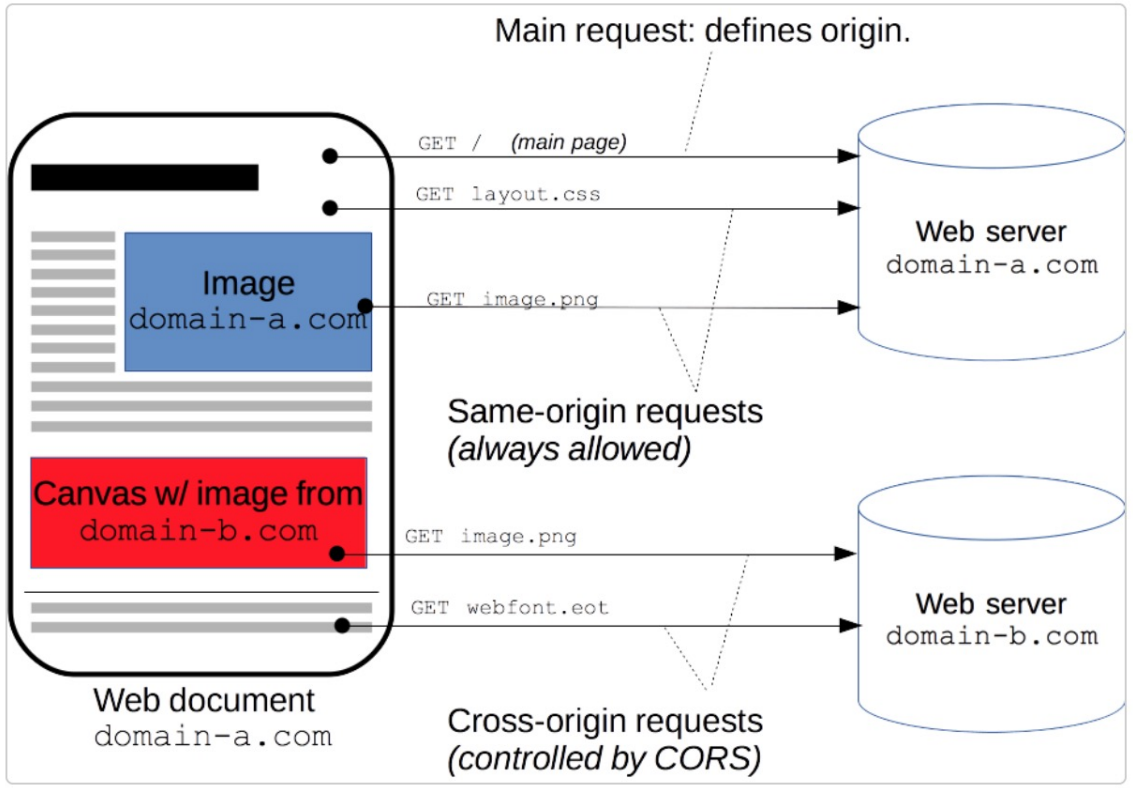
INTEGRAÇÃO REACT e SPRING BOOT

- **Especificação CORS**
  - O CORS (Cross-Origin Resource Sharing) é um mecanismo de segurança implementado pelos navegadores para controlar as solicitações HTTP feitas por um script em uma página web para outro domínio. Ele é usado para evitar ataques de scripts entre sites (*cross-site scripting*) e proteger a segurança e privacidade dos usuários.

  - Quando um navegador faz uma solicitação HTTP para um recurso em um domínio diferente do domínio da página atual, o CORS entra em ação. O servidor que hospeda o recurso precisa fornecer as configurações adequadas de cabeçalho de resposta para permitir ou negar a solicitação, dependendo da política de segurança definida.

INTEGRAÇÃO REACT e SPRING BOOT

- **Exemplo CORS**



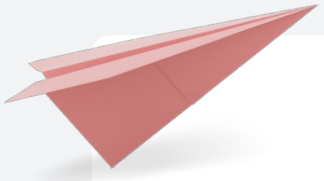
https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS

INTEGRAÇÃO REACT e SPRING BOOT

- **Código CORS no SPRING BOOT**

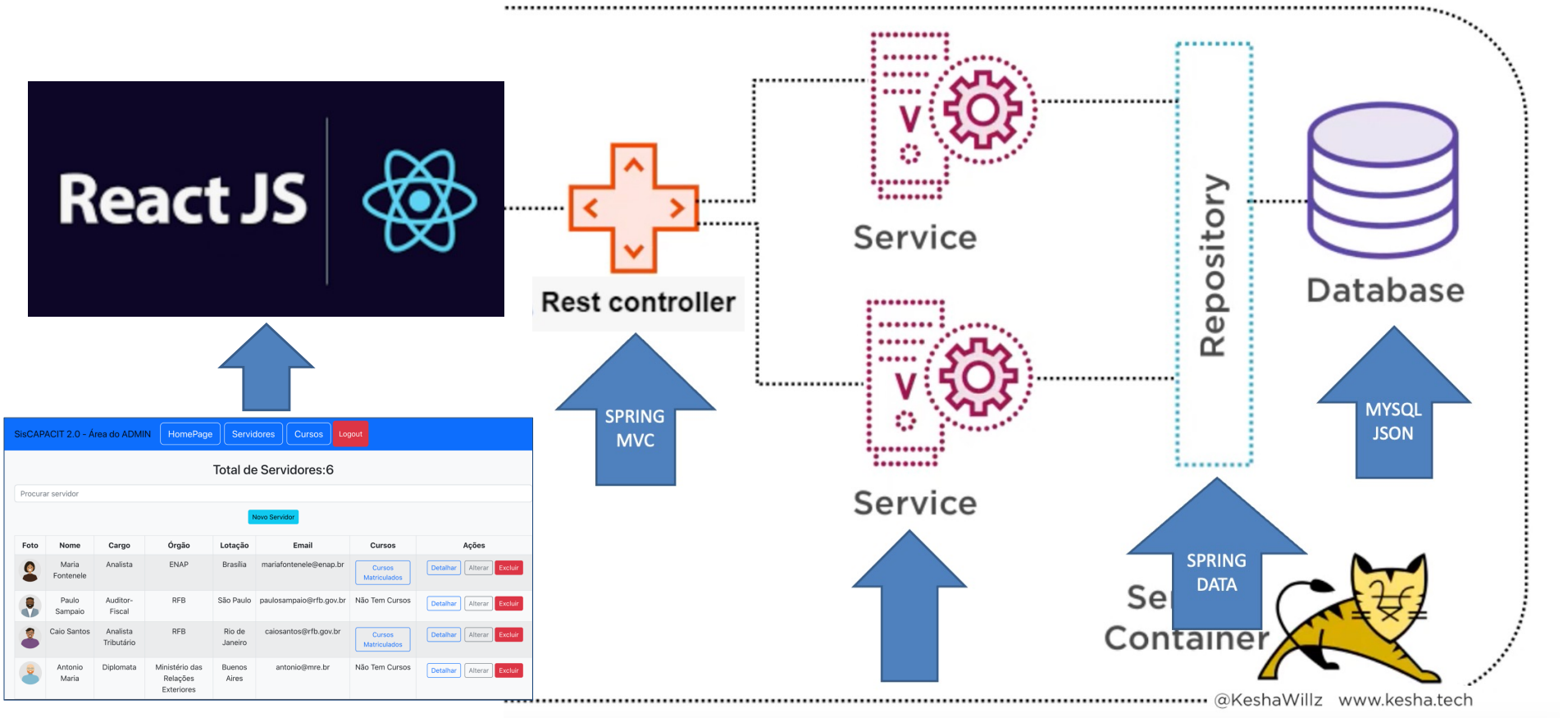
```
@Configuration
public class CorsConfig implements WebMvcConfigurer {

     @Override
     public void addCorsMappings(CorsRegistry registry) {
          registry.addMapping("/**")
          .allowedOrigins("http://localhost:3000")
          // Defina a origem permitida para o cliente ReactJS
          .allowedMethods("GET", "POST", "PUT", "DELETE")
          // Defina os métodos HTTP permitidos
          .allowedHeaders("*")
          // Defina os cabeçalhos permitidos
          .allowCredentials(true);
          // Permita o uso de cookies de autenticação (se aplicável)
     }
}
```

PROJETO PRÁTICO



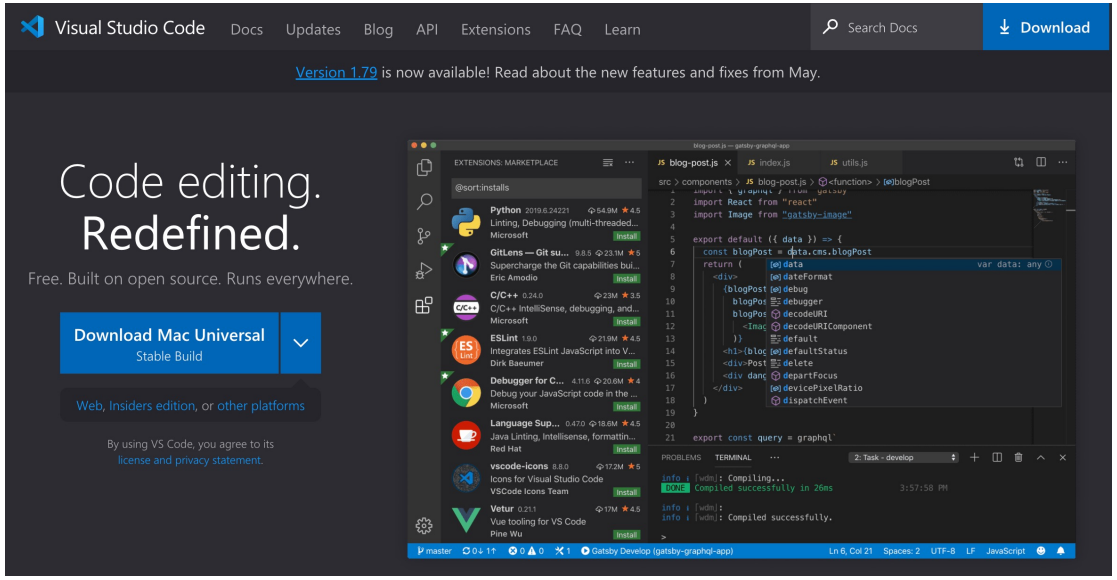
9° Projeto Spring Boot – Aplicação Servidor Público REST API com REACTJS e MySQL

ARQUITETURA REACTJS



Passos:

- Instalar o Node.js
- Instalar a IDE Visual Studio Code



https://code.visualstudio.com/

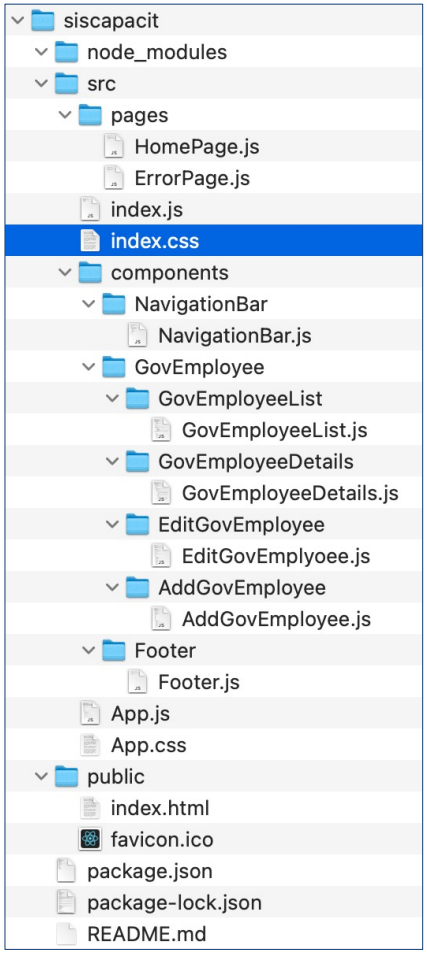
Passos:

- Copiar o projeto anterior (APIRest) e fazer os ajustes necessários (segurança CORS)

- Testar a APIRest com CORS no Postman

- Fazer o download da Aplicação REACTJS ‘Siscapacit’
(https://drive.google.com/drive/folders/1WWNe3Qrx7x1nfdJ-4cBmSgugOA0b87MF)
- Realizar a Integração REACTJS e SPRING BOOT

ARQUITETURA DO PROJETO - FRONTEND



ARQUITETURA DO PROJETO - BACKEND