**Anotações OmniStack**

Primeira Aula

Usaremos o React porque quando usamos apenas o html ou css puro, o backend que da conta de carregar tudo. Ao utilizar o react mantemos a parte do front no browser .

**SPA (Single Page applications)**

No SPA, as requisições fazem apenas dados como respostas e, com esses dados, o front-end pode preencher as informações em tela.

A página nunca recarrega, otimizando a performance e dando vida ao conceito de SPA. Retornando apenas JSON podemos ter quantos front-ends quisermos. Ao invés de retornar o html o back retorna uma estrutura de dados.

**Tradicional** Na abordagem tradicional, casa requisição**,** o servidor retorna o conteúdo completo da página, contendo todo HTML e CSS.

Essa abordagem limita o front-end para o browser jpa que o aplicativo mobile ou serviços externos não vão conseguir interpretar o HTML. Nessa abordagem não conseguimos ter várias aplicações, tipo para um mesmo site não conseguimos fazer um aplicativo, um login para o administrador uma página para controle de produto ou um serviço externo, porque o back-end ta retornando html, terá que mexer muito nele.

-- Javasprit core (Usado no react)   
- O curso irá utilizar o expo (framework de react)  
 Sem o Expo, precisamos instalar em nosso sistema tanto o Android Studio para obter a SDK de desenvolvimento Android e o Xcode (apenas no Mac) para obter a SDK do ios.

Nesse caso, nossa iniciação no desenvolvimento fica mais penos, já que essas SDK’s não são extremamente simples de instalar e livres de erros.  
  
 No nosso celular instalamos a EXPO, e dentro dele tudo o que precisamos para desenvolver no REACT Native já está instalado, como as API’S de mapas, Geolocalização.

Com isso, não precisamos nos preocupar em gerar o aplicativo pra Android e iOS já que o app do Expo instalado tem tudo o que precisamos e assim usamos apenas React

Segunda Aula

Utilizamos o ‘Insomnia’ para executar outros tipos de parâmetros além do get, uma vez que nossa interface não está criada.

Utilizamos o ‘Nodemon’ para não ficar executando toda vez o npm index.js, além disso instalamos com -D pois só será utilizado na hora do desenvolvimento  
 **‘npm’ =** instalar um pacote

**‘npx’ =** executar um pacote

/\*\*

 \* Rota / Recurso

 \*/

/\*\*

 \* Métodos HTTP:

 \*

 \* GET = Buscar uma informação do back-end

 \* POST = Criar uma informação no back-end

 \* PUT = Alterar uma informação no back-end

 \* DELETE = Deletar uma informação no back-end

 \*/

/\*\*

 \* Tipos de parâmetros:

 \*

 \* Query Params: Parâmetros nomeados enviados na rota após "?" (Filtros, paginação)

 \* Route Params: Parâmetros utilizados para identificar recursos

 \* Request Body: Corpo da requisição, utilizado para criar ou alterar recursos

 \*/

 /\*\*

  \* SQL: MySQL, SQLite, PostgreSQL, Oracle, Microsoft SQL Server

  \* NoSQL: MongoDB, CouchDB, etc

  \*/

  /\*\*

   \* Driver: Select \* FROM users

   \* Query Builder: table('users').select('\*').where()

   \*/

**Entidades**

- ONG  
- Casos

**Funcionalidades**- Login ONG  
- Logout ONG  
- Cadastro ONG  
- Cadastrar novos casos (Feito)  
- Deletar casos (Feito)  
- Listar casos específicos de uma ONG  
- Listar todos os casos  
- Entrar em contato via whats ou email