

**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE SÃO  
PAULO - CAMPUS SALTO**

**ALUISIO JOSÉ GALVÃO DOS SANTOS  
IAGO SANTOS FERNANDES  
ROGÉRIO FRANCISCO CARVALHO VIANA**

# **JOB SEA**

**SALTO - SP 2020**

**INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE SÃO  
PAULO - CAMPUS SALTO**

**ALUISIO JOSÉ GALVÃO DOS SANTOS  
IAGO SANTOS FERNANDES  
ROGÉRIO FRANCISCO CARVALHO VIANA**

# **JOB SEA**

Relatório apresentado na disciplina de  
Desenvolvimento para Web II ao Prof. Cláudio Luís  
Roveri Vieira do curso de Análise e  
Desenvolvimento de Sistemas sob a orientação da  
Prof.<sup>a</sup> Ma Ana Paula da Rosa.

# DESENVOLVIMENTO DO PROJETO

## Introdução

No back-end foi realizado as etapas de cadastramento, atualização e desativação de usuário que se cadastram em nosso sistema e os projetos publicados pelos usuários. Falta ainda fazer um ambiente remoto de homologação e validações dos dados enviadas pelas as requisições HTTP feitas pelo front-end, também devem ser feitos refinamentos com filtros nas requisições de dados. As maiores dificuldades estão nas configurações do servidor remoto para hospedagem de uma versão de homologação do nosso sistema.

No front-end, ao qual está sendo utilizado o framework React com a linguagem Typescript, já foi implementada algumas telas como login, landing page, esqueceu a senha, cadastro do usuário.

Para estas foi realizada já a navegação entre telas e a validação de alguns formulários pelo front-end.

Também já foi feita toda as telas de web designer, como a homepage, cadastro, esqueceu a senha e etc, com o auxílio de ajuda externa e implementadas no projeto.

Dificuldade está sendo em deixar responsivo o site tanto para aparelhos móveis quanto desktop, alguns estilos acabam 'quebrando', mas estamos a caminho do refinamento, das telas concluídas, cerca de 80% delas já está responsiva para desktop.

## Referências

Documentos utilizados como fonte de informações para desenvolvimento do projeto.

- FlyWay Documentation - <https://flywaydb.org/documentation/> - acessado 15/12/2020
- Guide Spring Framework - <https://spring.io/guides> - acesso 15/12/2020
- Swagger - <https://swagger.io/tools/swagger-ui/> - acesso 15/12/2020

Documentos utilizados como fonte de informações para desenvolvimento do projeto.

- Typescript - <https://www.typescriptlang.org/> - acesso 15/12/2020 ;

## Organização e Estilo do Código

*O projeto esta utilizando as convenções Java de codificação, documentação da convenção pode ser acessadas no sítio de [www.oracle.com](http://www.oracle.com)*

*O projeto em seu front-end, segue o padrão de organização utilizado dentro da startup Rocketseat. Segue anexo de link sobre como é dispostos os itens para codificação, localização e entendimento de cada área: <https://youtu.be/X2RKRKdqqwM> ;*

## Comentários

*Java Doc de Classes*

*/\*\**

*\* <Descrição da funcionalidade da Classe>*

```

* @author: <Fulano da Silva>
*/
Java Doc de Métodos
/**
* <Descrição completa das funcionalidades do método>
* @return <tipo do retorno do método: String, int, etc>
* @param <parâmetro1> <Descrição do parâmetro>
* @param <parâmetro2> <Descrição do parâmetro>
* @exception <exceção retornada>
*/

```

#### *Trecho JSX*

```

{ /*
<TagJsx />
*/ }

```

#### *Trecho de funcionalidades TS*

```

//Função anônima que renderiza botões defaults
const renderizaBotoes=()=>>{
}

```

## **Nomeação**

### *Arquivos Java*

*Os arquivos Java devem usar a convenção “camel case” de nomes, começando com a primeira letra do nome maiúscula e as subseqüentes minúsculas exceto a primeira letra da próxima palavra, todas as palavras devem ser escritas juntas sem espaço de separação.*

*Exemplo:*

*NomeClasse.java*

### *Scripts SQL*

*Os scripts SQL devem seguir as seguintes regras:*

- *Prefixo com a letra “V” maiúscula*
- *Número da versão “1”*
- *Separador usando do traços “\_\_”*
- *Descrição com traço separando as palavras*
- *Sufixo “.sql”*

*Exemplo:*

*V1.0.0\_\_init.sql*

### *Componentes React*

*Os componentes React são dispostos dentro de uma pasta, que terá o nome do componente e deverá iniciar com letra maiúscula que é o padrão do próprio React. Os arquivos em si, não possuem esta necessidade, podendo eles iniciar com letras minúsculas.*

*Exemplo:*

*NomeComponente*

*//Demais arquivos*

*index.tsx*

## Declaração

1. *Padrão de indentação de declaração de métodos:*  
`someMethod(int anArg, Object anotherArg, String yetAnotherArg,  
 Object andStillAnother) {  
 ...  
 }`
2. *Padrão de indentação de declaração de componentes:*  
`const NomeComponente:React.FC = () =>{  
 return( );  
 }`

## Reutilização

*Esta seção fornece regras e diretrizes para facilitar a reutilização.*

- *O retorno dos dados devem ser sempre no formato de JSON*
- *Utilizar os conceitos de SOLID no projeto.*

*Esta seção fornece regras e diretrizes para facilitar a reutilização.*

- *O retorno dos dados devem ser sempre no formato de JSON*
- *Todo elemento passível de reuso, deve ser componentizado*
- *Nenhum elemento deve ficar fora de sua arquitetura pré definida.*

## Tratamento de Erros e de Exceções

- *Os controladores somente tem acesso as interfaces de serviços.*
- *Garantir que somente as classes de implementação de serviços faça o processamento de requisições vindas dos controladores.*
- *Somente classes de implentação tem acesso as classes repositórios.*
- *Qualquer erro será tratado na sessão de services destinada a tratativas junto ao banco de dados/backend.*

## Teste unitário

*Utilizado os testes manuais para validar o funcionamento do projeto.*

## Diretrizes Gerais

*Utilize sempre as diretrizes e recomendações de desenvolvimento de sistemas SOLID e . SPA's.*