

IFSP - Instituto Federal de Educação, Ciência e Tecnologia  
Câmpus São Paulo

Bruna da Silva Pires	SP3056651
Daniel Roberto Pereira	SP3046702
Igor Nathan de Oliveira Rocha	SP305263X
Leonardo Marques da Silva	SP3052591
Lucas Lima de Santana	SP3046559
Marcelo Carlos Olimpio Junior	SP3046583

**"Portal de vagas de estágio"**

Proposta de projeto para disciplina PI1A5

Professor: Carlos Henrique Veríssimo Pereira

IFSP - Instituto Federal de Educação, Ciência e Tecnologia  
Câmpus São Paulo

Tecnologia em Análise e Desenvolvimento de Sistemas

PI1A5 - Projeto Integrado I

São Paulo - SP - Brasil

2022

# Lista de abreviaturas e siglas

- API *Application Programming Interface* - Interface de Programação de Aplicativos - Citado em [9](#)
- SSO *Single Sign-On* - Login único - Citado em [9](#)

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>4</b>
1.1	Justificativa	4
1.2	Proposta de solução	4
1.3	Objetivos	4
<b>2</b>	<b>REQUISITOS</b>	<b>6</b>
2.1	Requisitos Funcionais	6
2.2	Requisitos Não-funcionais	6
2.3	Regras de Negócio	6
<b>3</b>	<b>PROCESSOS MODELADOS</b>	<b>7</b>
3.1	Recomendação de vagas para o estudante	7
3.1.1	Descrição	7
3.1.2	Diagrama BPMN	7
3.2	Recomendação de estudantes para vagas cadastradas	7
3.2.1	Descrição	7
3.2.2	Diagrama BPMN	7
3.3	Recomendação de empresas para o estudante	7
3.3.1	Descrição	7
3.3.2	Diagrama BPMN	7
<b>4</b>	<b>TECNOLOGIAS</b>	<b>8</b>
4.1	Arquitetura	8
4.1.1	Diagramas de arquitetura	8
4.2	Integrações	8
4.2.1	Login com o Google e LinkedIn	9
4.2.2	Entrar em contato via <i>Whatsapp</i>	9

# 1 Introdução

Nesse capítulo serão mostrados os principais pontos do nosso projeto, os objetivos e quais os problemas que queremos solucionar com nossa aplicação.

## 1.1 Justificativa

## 1.2 Proposta de solução

O Portal de vagas de estágio é um sistema para aproximar novos profissionais/estudantes da área de TI e empresas com vagas de estágio/trainee disponíveis, de modo que os candidatos possam receber indicações de vagas condizentes com seu perfil e empresas recebam recomendações de candidatos possivelmente adequados às vagas anunciadas.

## 1.3 Objetivos

O objetivo principal da nossa solução é promover um meio de conexão mais direto entre os estudantes em busca de estágio e empresas que buscam interessados em suas vagas de estágio alinhados com o perfil buscado. Através do sistema de recomendações, tanto os estudantes quanto as empresas têm papel ativo no processo de encontrar um(a) estudante/vaga ideal, cujas as competências e perfil sejam condizentes com o que é procurado.

A partir do nosso objetivo principal, podemos listar alguns objetivos mais práticos da nossa solução:

- Realizar o gerenciamento de vagas entre os candidatos e as empresas de uma forma simplificada;
- Recomendar vagas para estudantes, empresas para estudantes, estudantes para vagas/empresas;
- Manter um histórico de vagas aplicadas pelo estudante;
- Manter um histórico de candidatos aplicados a vaga;
- Exibir uma linha do tempo da situação da vaga;
- Alertar os estudantes aplicados à vaga sobre cada mudança em seu status;
- Possibilitar o gerenciamento da vaga pela empresa que a registrou/publicou;

- 
- Possibilitar que a empresa possa acionar (entrar em contato) com os estudantes recomendados/aplicados à vaga;
  - Possibilitar que a empresa realize mudanças no status da vaga;

## 2 Requisitos

Nesse capítulo serão expostos os requisitos funcionais, não-funcionais e regras de negócio que nossa aplicação terá, tais requisitos foram formados a partir de estudos de como irá funcionar os processos de nosso *website*.

### 2.1 Requisitos Funcionais

### 2.2 Requisitos Não-funcionais

### 2.3 Regras de Negócio

## 3 Processos modelados

### 3.1 Recomendação de vagas para o estudante

#### 3.1.1 Descrição

#### 3.1.2 Diagrama BPMN

### 3.2 Recomendação de estudantes para vagas cadastradas

#### 3.2.1 Descrição

#### 3.2.2 Diagrama BPMN

### 3.3 Recomendação de empresas para o estudante

#### 3.3.1 Descrição

#### 3.3.2 Diagrama BPMN

## 4 Tecnologias

Nesse capítulo serão citadas a arquitetura do nosso projeto com ilustrações demonstrando de forma mais lúdica, as possíveis integrações que nossa aplicação terá com sistemas externos.

### 4.1 Arquitetura

Para o desenvolvimento do projeto, e tendo em vista que será construída uma aplicação web de página única, utilizaremos de ferramentas que cerceiam o ecossistema de *Single Page Applications*. Para isso, teremos a divisão do projeto em *front-end* e *back-end* de modo que eles se comuniquem via protocolo HTTP com requisições e respostas no formato JSON. Para o desenvolvimento do *front-end* utilizaremos Typescript por meio da biblioteca React; o *back-end* será desenvolvido utilizando Java com o micro *framework* Spring Boot. Um módulo de apoio no lado do servidor poderá ser possível, e para ele utilizaremos Python.

Em relação ao deploy das aplicações, o *front-end* será hospedado na plataforma Vercel, que é primariamente voltada para Javascript, proporcionando uma melhor agilidade de desenvolvimento, enquanto o *back-end* será hospedado no Heroku, que é uma plataforma como serviço de fácil manuseio e que nos permitirá ter um maior foco no desenvolvimento do projeto. Através do Heroku podemos também fazer a utilização do banco de dados PostgreSQL por meio do serviço de apoio Heroku Postgres.

Ademais, se for necessário o armazenamento de objetos como arquivos ou imagens, utilizaremos a plataforma Cloudinary principalmente por sua fácil integração com a linguagem de programação Java através de bibliotecas.

#### 4.1.1 Diagramas de arquitetura

Os diagramas [Figura 1](#), [Figura 2](#) e [Figura 3](#) ilustram de modo geral a arquitetura pensada para a solução proposta, utilizando das tecnologias já citadas.

### 4.2 Integrações

Nessa seção serão citadas as possíveis integrações que nossa aplicação terá, que foram decididas baseadas em outras aplicações do mercado.



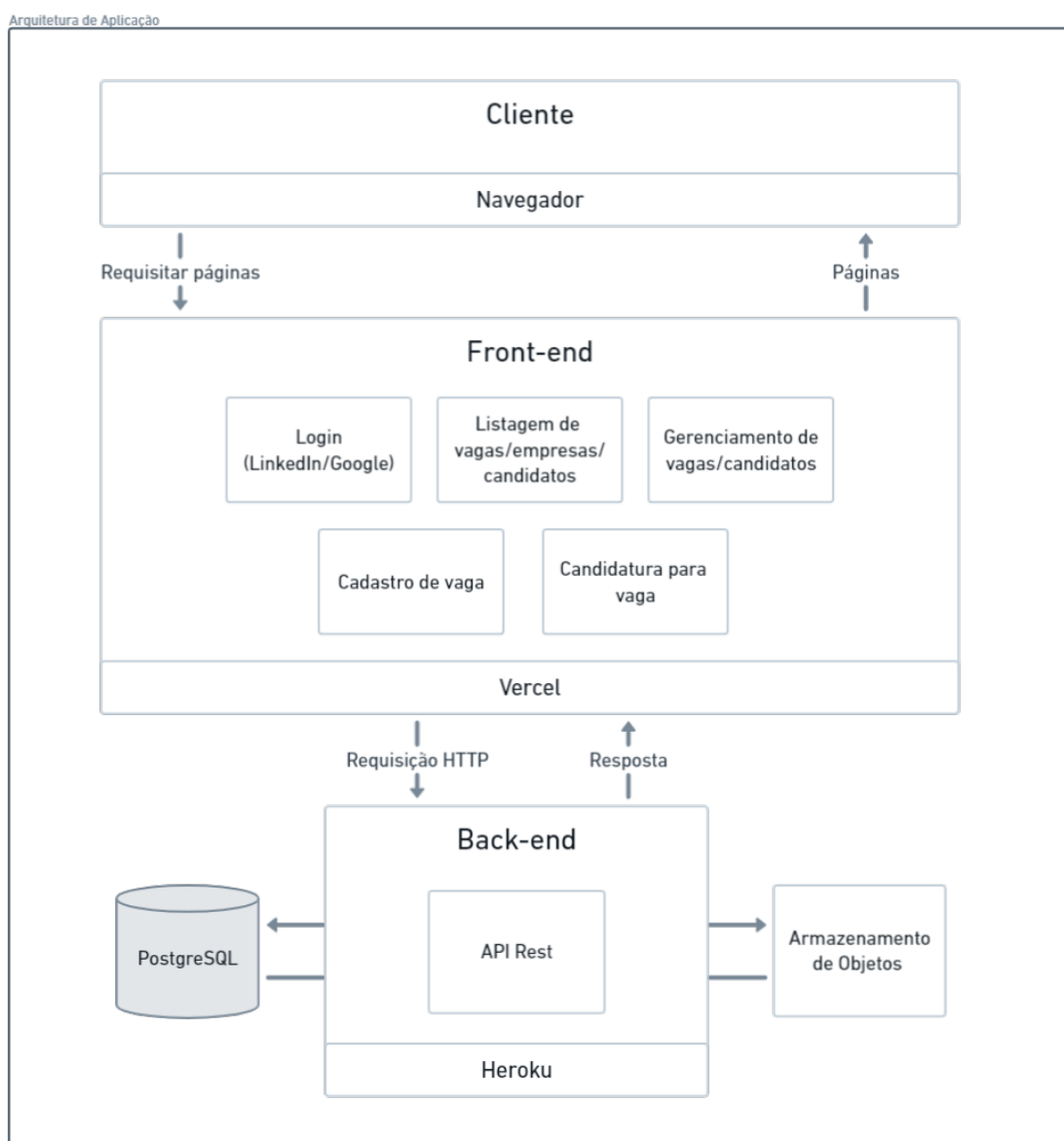
### 4.2.1 Login com o Google e LinkedIn

Pensando na experiência de usuário, nossa aplicação terá a opção do estudante se logar através do *Single Sign-On (SSO)* dessas empresas. Dessa forma, não será necessário digitar a senha toda vez que o usuário for usar nosso *website*, precisando apenas clicar um botão e fazer o login em uma dessas alternativas.

### 4.2.2 Entrar em contato via *Whatsapp*

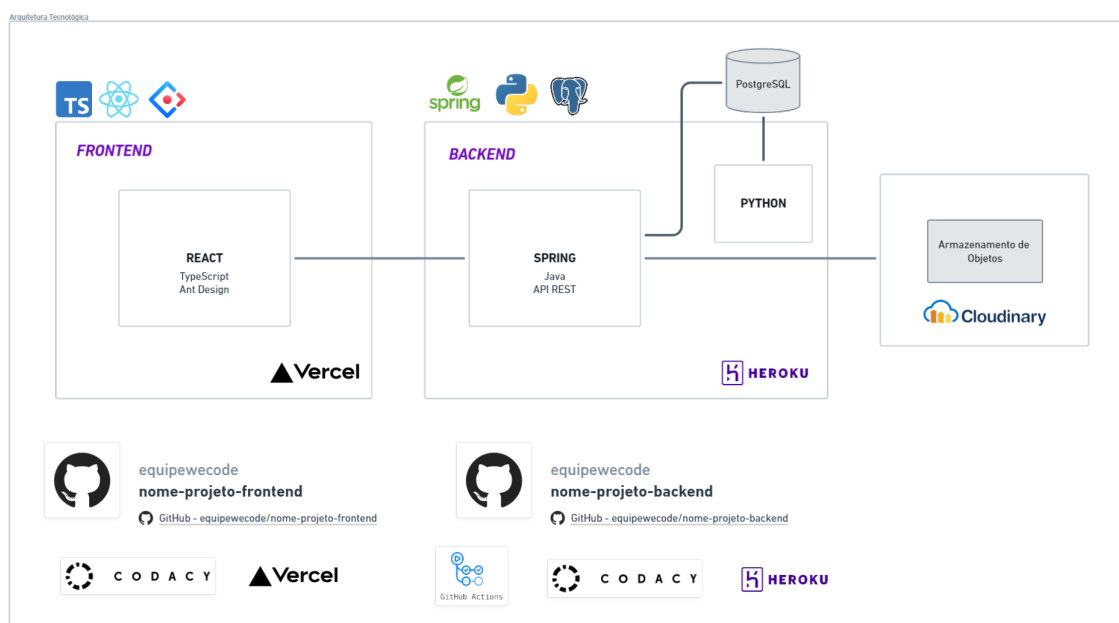
Nossa aplicação terá, também, uma forma da empresa contatar o estudante via *Whatsapp*. Essa integração será feita via *Application Programming Interface (API)* disponibilizada pela própria empresa que mantém o aplicativo (Meta). Dessa forma, com apenas um clique, será possível enviar uma mensagem diretamente ao estudante.

Figura 1 – Arquitetura de Aplicação



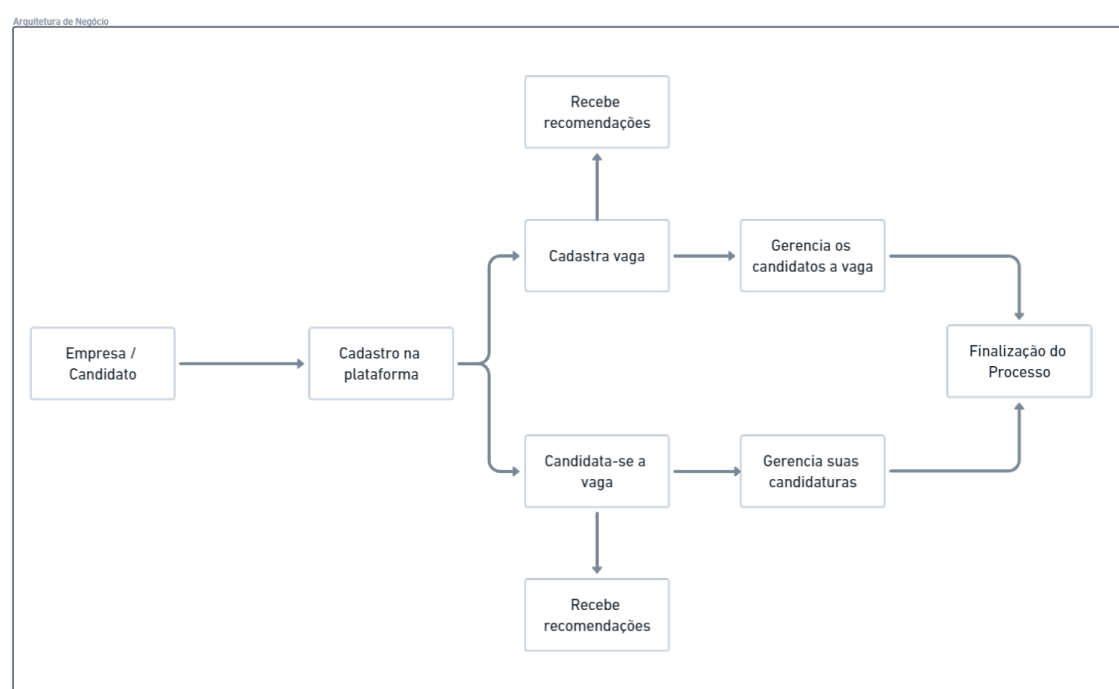
Fonte: Produzido pelos autores utilizando a ferramenta *Whimsical*

Figura 2 – Arquitetura Tecnológica



Fonte: Produzido pelos autores utilizando a ferramenta *Whimsical*

Figura 3 – Arquitetura de Negócios



Fonte: Produzido pelos autores utilizando a ferramenta *Whimsical*