

# 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

## **Website de Vagas de Estágio**

Desenho de aplicação para desenvolvimento  
na disciplina de Projeto Integrado I no 1º  
semestre de 2022.

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 ilustrações

Figura 1 – Arquitetura de Aplicação . . . . .	10
Figura 2 – Arquitetura Tecnológica . . . . .	11
Figura 3 – Arquitetura de Negócios . . . . .	11

# Lista de quadros

Quadro 1 – Comparação dos aplicativos concorrentes. . . . .	7
Quadro 2 – Requisitos funcionais . . . . .	12
Quadro 3 – Requisitos Não-funcionais . . . . .	12
Quadro 4 – Regras de negócios . . . . .	13

## Lista de abreviaturas e siglas

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>6</b>
<b>1.1</b>	<b>Justificativa</b>	<b>6</b>
<b>1.2</b>	<b>Proposta de solução</b>	<b>6</b>
<b>1.3</b>	<b>Objetivos</b>	<b>6</b>
<b>1.4</b>	<b>Análise de Concorrentes</b>	<b>7</b>
<b>2</b>	<b>REVISÃO DA LITERATURA</b>	<b>8</b>
<b>2.1</b>	<b>Estágio</b>	<b>8</b>
<b>3</b>	<b>DESENVOLVIMENTO DA APLICAÇÃO</b>	<b>9</b>
<b>3.1</b>	<b>Arquitetura</b>	<b>9</b>
3.1.1	Diagramas de arquitetura	9
<b>3.2</b>	<b>Escopo</b>	<b>12</b>
3.2.1	Requisitos	12
3.2.1.1	Requisitos Funcionais	12
3.2.1.2	Requisitos Não-funcionais	12
3.2.1.3	Regras de Negócio	13
<b>3.3</b>	<b>Tecnologias utilizadas</b>	<b>13</b>
<b>3.4</b>	<b>Modelagem e Definições Técnicas</b>	<b>13</b>
<b>3.5</b>	<b>Manutenibilidade</b>	<b>13</b>
3.5.1	Logs	14
3.5.2	Code Convention	14
3.5.3	Design Patterns	14
3.5.4	Integração continua	14
3.5.5	Análise estática	14
3.5.6	Testes	14
<b>3.6</b>	<b>Escalabilidade</b>	<b>14</b>
<b>3.7</b>	<b>Segurança, Privacidade e Legislação</b>	<b>14</b>
<b>3.8</b>	<b>Viabilidade Financeira</b>	<b>14</b>
3.8.1	Gerenciamento de custos	14
3.8.1.1	Desenvolvimento	14
3.8.2	Ambiente de produção	14
3.8.2.1	Frontend	14
3.8.2.2	Backend	15
3.8.2.3	Banco de dados	15
3.8.3	Monetização	15

3.8.4	Conclusão . . . . .	16
4	<b>PLANEJAMENTO E GERENCIAMENTO DO PROJETO . . . . .</b>	<b>17</b>
4.1	<b>Metodologia de Gestão e Desenvolvimento de Projeto . . . . .</b>	<b>17</b>

# 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

Existe, na contemporaneidade, uma grande dificuldade em adquirir experiência profissional através da prática de estágio, muitas vezes obrigatória no projeto pedagógico de curso das universidades. Tal problema se dá por meio das plataformas que disponibilizam tais vagas, porém com uma certa cobrança injusta em relação a habilidades que o candidato precisa possuir previamente. É também notável que existe uma certa dificuldade de conexão entre a empresa e o candidato, que muitas vezes não obtém o retorno sobre o processo de seleção da vaga.

## 1.2 Proposta de solução

Tendo em vista os problemas anteriormente descritos, o Portal de Vagas de Estágio é um sistema para aproximar novos estudantes e empresas com vagas de estágio 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:

- Ser um *website* de fácil usabilidade, onde os estudantes encontrem vagas sem passar por longos processos seletivos.

- Ser uma aplicação onde de fato os estudantes encontrem vagas que condizem com a realidade de um estagiário.
- Pensar sempre na experiência dos usuários, de modo que a aplicação seja simples e efetiva ao mesmo tempo.

## 1.4 Análise de Concorrentes

Para a elaboração da proposta, foram verificadas algumas soluções já existentes no mercado. A partir disso, as soluções que mais se assemelham com a proposta foram o *Companhia de Estágios*, *Cia de Talentos* e *Nube*. Com base neste levantamento, podemos observar algumas intersecções de funcionalidades oferecidas. O quadro 1 permite a melhor visualização deste levantamento.

Quadro 1 – Comparação dos aplicativos concorrentes.

Funcionalidades	Cia de Estágios	Cia de Talentos	Nube	Nosso Proj.
Login/Cadastro.	x	x	x	x
Aplicar em uma vaga.	x	x	x	x
Notificação a cada mudança do status no processo seletivo.			x	x
Recomendação de vagas e/ou empresas aos estudantes de acordo com as suas características.				x
Recomendação de estudantes mais compatíveis com as vagas registradas pelas empresas, de acordo com as características da vaga e da empresa.				x
Simplificação de contato via <i>WhatsApp</i> .				x
Denúncias de vagas incoerentes com a realidade.				x
<i>Feedback</i> de empresas pós-entrevista.				x

Fonte: Os Autores



## 2 Revisão da Literatura

### 2.1 Estágio

## 3 Desenvolvimento da Aplicação

### 3.1 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.

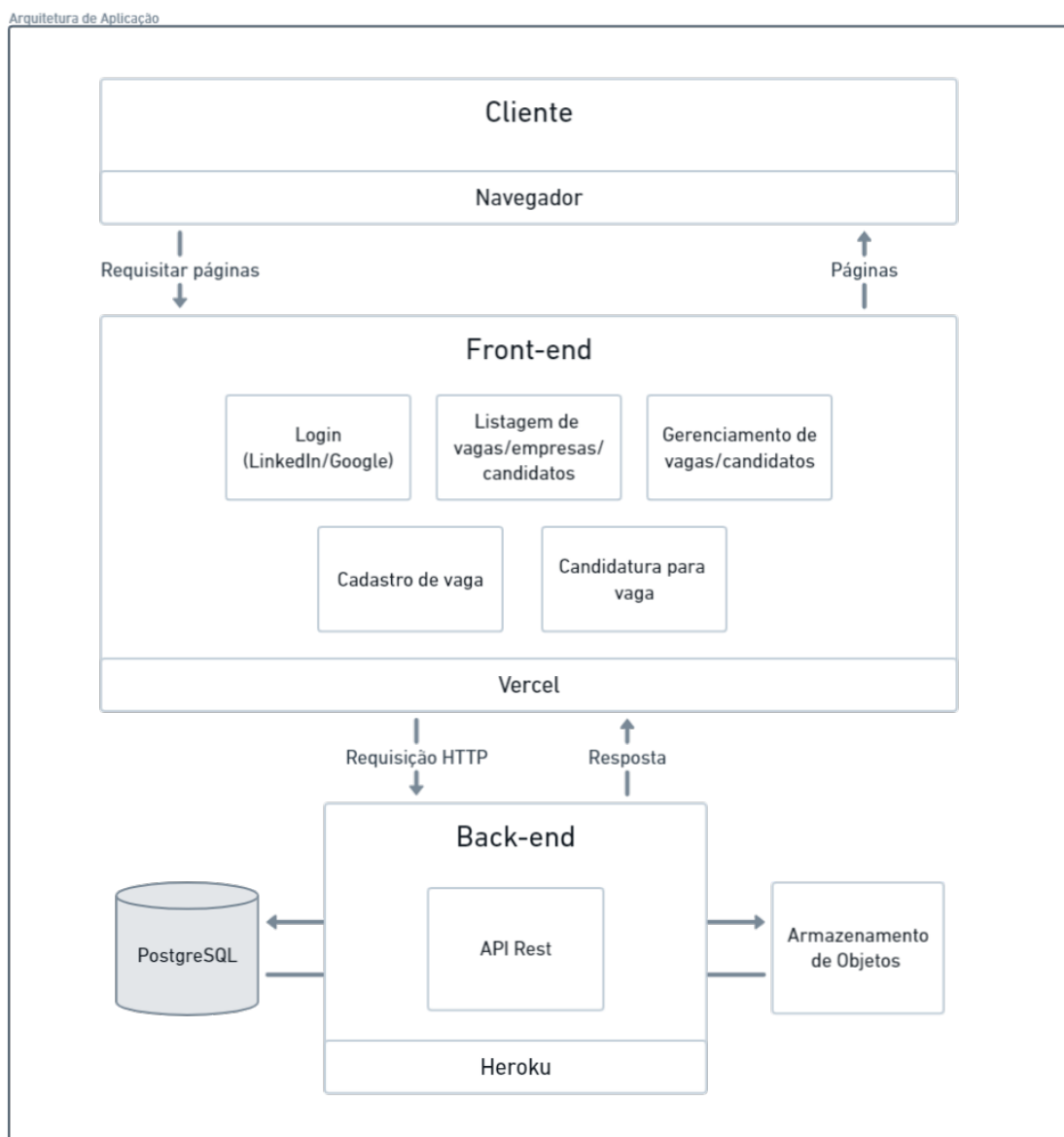
Em relação ao deploy das aplicações, o *front-end* será hospedado na plataforma Vercel, que é primariamente voltada para Javascript, proporcionando uma melhor agilidade de desenvolvimento, enquanto o *back-end* será hospedado no Heroku, que é uma plataforma como serviço de fácil manuseio e que nos permitirá 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.

#### 3.1.1 Diagramas de arquitetura

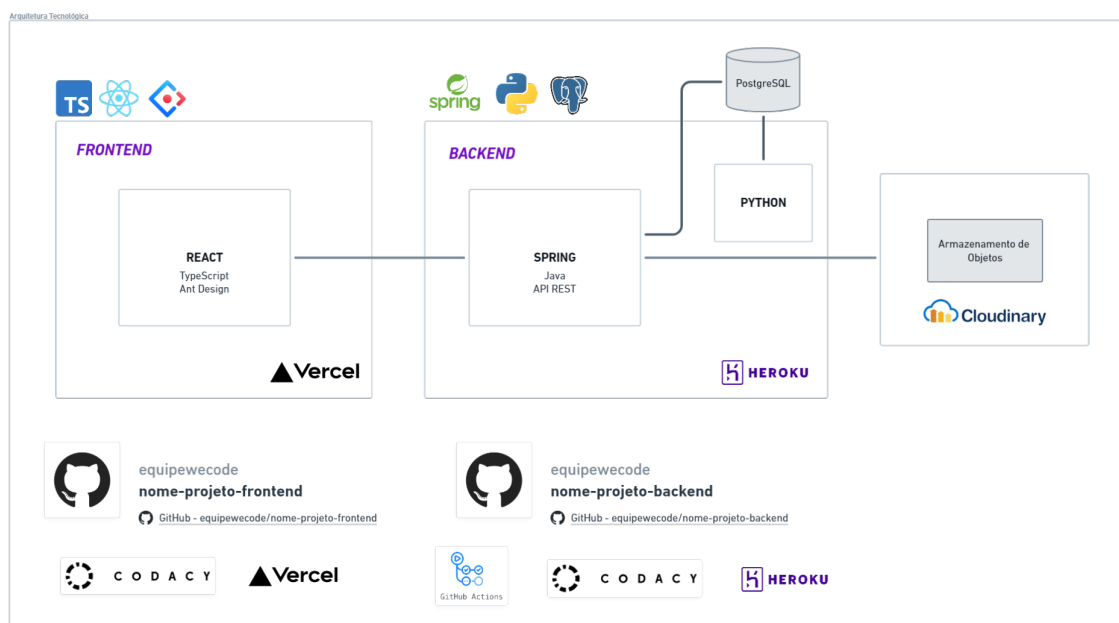
Os diagramas [Figura 1](#), [Figura 2](#) e [Figura 3](#) ilustram de modo geral a arquitetura pensada para a solução proposta, utilizando das tecnologias já citadas.

Figura 1 – Arquitetura de Aplicação



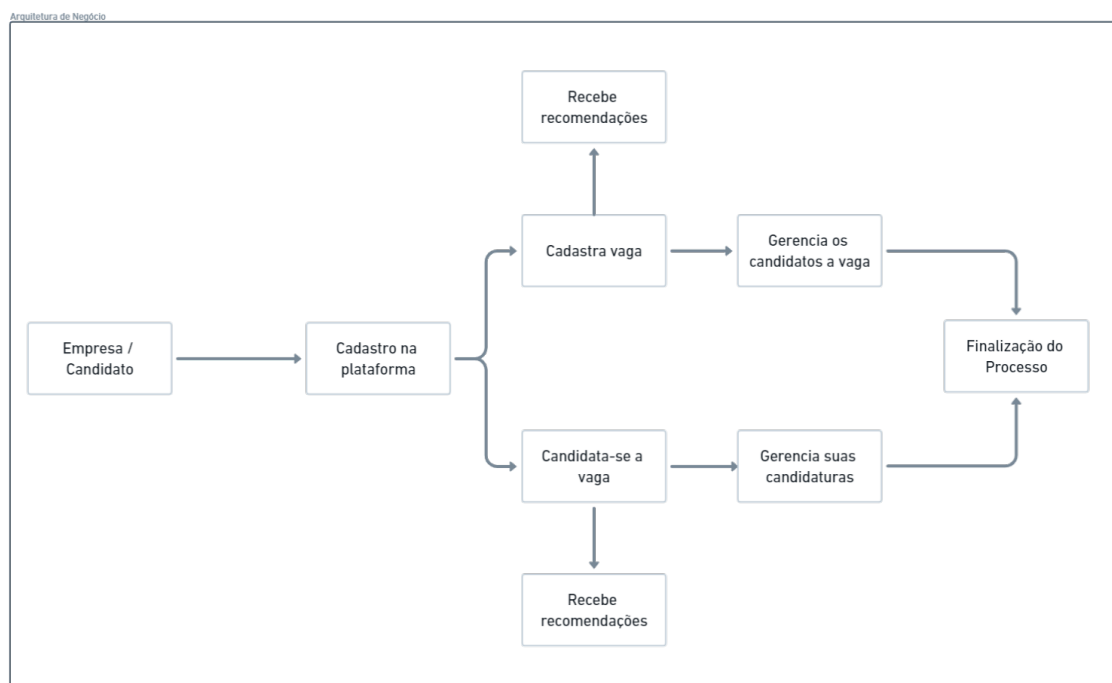
Fonte: Produzido pelos autores utilizando a ferramenta *Whimsical*

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*

## 3.2 Escopo

### 3.2.1 Requisitos

Para o futuro desenvolvimento da aplicação, 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*.

#### 3.2.1.1 Requisitos Funcionais

Durante nossa análise, decidimos que esses seriam os principais requisitos funcionais do nosso projeto:

Quadro 2 – Requisitos funcionais

Código	Descrição
RF-001	Realizar o gerenciamento de vagas entre os candidatos e as empresas de uma forma simplificada
RF-002	Recomendar vagas para estudantes, empresas para estudantes, estudantes para vagas/empresas
RF-003	Manter um histórico de vagas tanto para o candidato, quanto para a empresa
RF-004	Exibir uma linha do tempo do andamento da vaga
RF-005	Alertar os estudantes aplicados à vaga sobre cada mudança em seu processo
RF-006	Possibilitar que a empresa possa entrar em contato com os estudantes recomendados/aplicados à vaga
RF-007	Possibilitar que a empresa realize mudanças no status de andamento da vaga
RF-008	Possibilitar que o estudante realize um feedback da empresa pós-entrevista, que será visto por outros estudantes
RF-009	Não permitir o registro de vagas cujas horas de atividades ultrapassem a carga horária prevista por lei de acordo com a situação escolar de cada estudante

Fonte: Os Autores

#### 3.2.1.2 Requisitos Não-funcionais

Os requisitos não-funcionais do nosso projeto estão listados abaixo:

Quadro 3 – Requisitos Não-Funcionais

Código	Descrição
RNF-001	O sistema deve oferecer boa usabilidade (Ser fácil de aprender a usar)
RNF-002	O sistema deve estar disponível 24 horas por dia, 7 dias por semana
RNF-003	O sistema deve possuir possibilidade de escalabilidade
RNF-004	Tempo para o carregamento que satisfaça as expectativas do cliente
RNF-005	O sistema deve possuir uma taxa de ocorrência de falhas menor que 0.3%
RNF-006	O sistema deve estar de acordo com a Lei Geral de Proteção de Dados
RNF-007	O sistema deve estar de acordo com a lei Nº 11.788, de 25 de setembro de 2008, regulando a carga horária do estágio
RNF-008	O sistema deve ser responsivo aos diferentes dispositivos que os usuários podem utilizar para acessá-lo

Fonte: Os Autores

### 3.2.1.3 Regras de Negócio

As regras de negócio do nosso projeto estão listados abaixo:

Quadro 4 – Regras de negócios

Código	Descrição	Requisito Relacionado
RN-001	As vagas a serem cadastradas devem estar coerentes com o perfil buscado	RF-001
RN-002	Os históricos das vagas devem ser mantido por todo o período	RF-003
RN-003	A empresa é responsável pelo encaminhamento do status da vaga	RF-007
RN-004	Para o candidato enviar um feedback, ele deve ter pelo menos iniciado o processo seletivo	RF-008
RN-005	O feedback pode ser feito de forma anônima, mas o usuário deve estar logado e ter passado pelo processo seletivo	RF-008

Fonte: Os Autores

## 3.3 Tecnologias utilizadas

## 3.4 Modelagem e Definições Técnicas

## 3.5 Manutenibilidade

Para que a aplicação atinja um nível adequado de qualidade é fundamental que se estabeleça certos requisitos e parâmetros de manutenibilidade tal como ferramentas que facilitam esse processo. Através dos critérios estabelecidos, podemos medir o quanto o processo de desenvolvimento está de acordo com as boas práticas, e incentivar o uso das mesmas.

### 3.5.1 Logs

### 3.5.2 Code Convention

### 3.5.3 Design Patterns

### 3.5.4 Integração continua

### 3.5.5 Análise estática

### 3.5.6 Testes

## 3.6 Escalabilidade

## 3.7 Segurança, Privacidade e Legislação

## 3.8 Viabilidade Financeira

A análise de viabilidade financeira consiste em averiguar a viabilidade da manutenibilidade do projeto e da possibilidade de lucro do mesmo, a fim de fazer essa verificação será descrito cada processo.

### 3.8.1 Gerenciamento de custos

Serão abordados custos de desenvolvimento e aporte inicial do projeto

#### 3.8.1.1 Desenvolvimento

O projeto não possuirá nenhum custo de implementação, devido ao fato de ser um projeto educacional, todo o tempo de desenvolvimento da aplicação e documentação serao totalmente voluntários, sem custo adicional ao projeto.

### 3.8.2 Ambiente de produção

São apresentados os custos de manutenibilidade do projeto para os usuários. Onde será feita uma previsão anual de cada plataforma utilizada.

#### 3.8.2.1 Frontend

A camada cliente da aplicação será hospedada na plataforma Vercel, sendo o custo de processamento e requisições da aplicação baixo inicialmente, a hospedagem da camada cliente não apresentará custo adicional.

### 3.8.2.2 Backend

Inicialmente gratuito na plataforma Heroku

A partir do momento que for necessário grande porte, será indicado a migração para a amazon AWS ou Azure, visto que garante viabilidade econômica e estratégica (pois o preço é calculado a partir do uso).

Utilizando a calculadora da AWS e optando por um servidor Linux da instância t4g.micro com 1 vCPU e 1GiB, com armazenamento SSD de uso geral, será custeado o valor de 5,76 USD mensalmente para operar o mês inteiro.

Utilizando a calculadora da Microsoft Azure e optando por um servidor Linux da instância A1 v2 com 1 núcleo e 2GB de ram, com 10GB de armazenamento temporário, será custeado o valor de 57,10 USD mensalmente para operar o mês inteiro.

### 3.8.2.3 Banco de dados

Inicialmente gratuito na plataforma Heroku

Caso a aplicação fique com um porte maior, será indicado a migração para a amazon AWS RDS, que suporta o serviço de banco de dados, cujo o custo é calculado em relação ao uso

Utilizando a calculadora da AWS e optando por um servidor da instância t3.micro de modelo Single-AZ OnDemand, com armazenamento SSD para cada instância, será custeado o valor de 27,36 USD mensalmente para operar o mês inteiro.

## 3.8.3 Monetização

Afim de gerar receita para a plataforma, são consideradas duas possibilidades de monetização

- Propagandas: Será utilizado mediador de anúncio Google AdSense, onde o valor varia por visualizações de anúncios e cliques nos anúncios, quanto maior a quantidade de conversão de cliques por visualização, maior será a sua renda.
- Contratos: Empresas interessadas em impulsionar as suas vagas para atingir um número maior de visualizações ou oferecer ferramentas de análises mais precisas e um melhor suporte, feito por intermédio da realização de contratos com a plataforma e que consequentemente gerará renda.

Com a estimativa de 100 a 250 visitantes por dia, considerando que pelo menos 2 páginas são visualizadas por visitantes, sendo a taxa de cliques em anúncios 1% e o custo do clique 0.20 USD, o valor mensal será de aproximadamente 10.5 USD. A monetização



por propaganda seria a forma de renda mais rápida para o projeto e os contratos seriam feitos a médio/longo prazo.

#### **3.8.4 Conclusão**

Utilizando inicialmente os servidores de baixo porte detalhados acima, não haverá custo adicional a priori. Contudo, o valor calculado para 250 visitantes com os parâmetros detalhados arrecadará 10.5 USD.

Caso o engajamento da aplicação aumente, a medida que o número de usuários aumenta incrementando proporcionalmente o rendimento com o AdSense, poderá ser revisto os planos dos servidores para atender maiores níveis de requisições e buscar contratos com empresas para aumentar a rentabilidade da plataforma.

## 4 Planejamento e Gerenciamento do Projeto

### 4.1 Metodologia de Gestão e Desenvolvimento de Projeto