

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

EstagiEI
Sistema de vagas de estágio

São Paulo - SP - Brasil

2022

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

EstagiEI
Sistema de vagas de estágio

Documentação de Mínimo Produto Viável
para aprovação na disciplina de Projeto Inte-
grado 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 – Roteiro Geral de PI1A5	15
Figura 2 – Roteiro Geral de PI1A5 - Detalhe Inicial	15
Figura 3 – Roteiro Geral de PI1A5 - Detalhe Final	16
Figura 4 – <i>Epic</i> da Empresa	18
Figura 5 – <i>Epic</i> do Estudante	19
Figura 6 – Roteiro Geral para PI2A6	19
Figura 7 – Arquitetura de Aplicação	22
Figura 8 – Arquitetura Tecnológica	22
Figura 9 – Arquitetura de Negócios	23
Figura 10 – Caso de Uso - EstagiEI	29
Figura 11 – URL de <i>Actions</i> do <i>back-end</i>	36
Figura 12 – URL de <i>Actions</i> do <i>front-end</i>	36
Figura 13 – Modelagem Entidade Relacionamento	40
Figura 14 – Diagrama Entidade Relacionamento	41
Figura 15 – Linhas de código - SVN	47
Figura 16 – Atividade por hora do dia - SVN	48
Figura 17 – Atividade por dia da semana - SVN	48
Figura 18 – Visão geral - Projeto <i>front-end</i>	49
Figura 19 – URL do repositório <i>front-end</i>	49
Figura 20 – Visão geral - Projeto <i>back-end</i>	50
Figura 21 – URL do repositório <i>back-end</i>	50
Figura 22 – Visão geral - Projeto Documentos	51
Figura 23 – URL do repositório de documentos \LaTeX	51
Figura 24 – Linhas de código - Projeto <i>front-end</i>	52
Figura 25 – Linhas de código - Projeto <i>back-end</i>	52
Figura 26 – Extensão de arquivos - Projeto <i>front-end</i>	53
Figura 27 – Dias da semana - Projeto <i>front-end</i>	53
Figura 28 – Lista de autores - Projeto <i>back-end</i>	53
Figura 29 – Lista de autores - Projeto Documentos	54
Figura 30 – Dias da semana - Projeto <i>front-end</i>	54
Figura 31 – Dias da semana - Projeto <i>back-end</i>	55
Figura 32 – Validação dos <i>headers</i>	55
Figura 33 – URL da documentação dos nossos <i>endpoints</i> (<i>Swagger UI</i>)	56
Figura 34 – Teste de <i>Transport Layer Security</i> (TLS)	56
Figura 35 – URL do <i>front-end</i> da nossa aplicação	57
Figura 36 – Teste de desempenho do <i>front-end</i>	57

Figura 37 – Análise de código do <i>front-end</i>	58
Figura 38 – Análise de código do <i>back-end</i>	59
Figura 39 – Validação do <i>Hypertext Markup Language</i> (HTML)	60
Figura 40 – URL do blog da equipe L ^A T _E X	65

Lista de quadros

Quadro 1 – Comparação dos aplicativos concorrentes	10
Quadro 2 – Divisão de responsabilidades da equipe.	14
Quadro 3 – Cronograma de Sprints - 1º semestre	17
Quadro 4 – Cronograma de Sprints - 2º semestre	20
Quadro 5 – Requisitos funcionais	24
Quadro 6 – Requisitos não funcionais	25
Quadro 7 – Regras de negócio	25
Quadro 8 – Histórias de usuário - Empresa	26
Quadro 9 – Histórias de usuário - Estudante	27
Quadro 10 – Histórias de usuário - Administrador	28
Quadro 11 – Casos de uso	29
Quadro 12 – <i>SOLID Principles</i>	34
Quadro 13 – <i>12 Factor App parameters</i>	35
Quadro 14 – Legenda	42
Quadro 15 – Campos de Usuário	42
Quadro 16 – Campos de Pessoa	42
Quadro 17 – Campos de Vaga	43
Quadro 18 – Campos de Curso	43
Quadro 19 – Campos de Estudante	43
Quadro 20 – Campos de Empresa	44
Quadro 21 – Campos de Representante RH	44
Quadro 22 – <i>Endpoints</i> de Estudante	44
Quadro 23 – <i>Endpoints</i> de Vaga	45
Quadro 24 – <i>Endpoints</i> de Empresa	45
Quadro 25 – <i>Endpoints</i> de Competência	45

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i> - Interface de Programação de Aplicativos - Citado em 31 , 37 , 55 , 61
CLT	Consolidação das Leis do Trabalho - Citado em 11
CNPJ	Cadastro Nacional da Pessoa Jurídica - Citado em 37
HTML	<i>Hypertext Markup Language</i> - Linguagem de Marcação de Hipertexto - Citado em 3 , 7 , 30 , 31 , 59 , 60
HTTP	<i>Hypertext Transfer Protocol</i> - Protocolo de transferência de hipertexto - Citado em 21 , 44
HTTPS	<i>Hypertext Transfer Protocol Secure</i> - Protocolo seguro de transferência de hipertexto - Citado em 37 , 56
JSON	<i>JavaScript Object Notation</i> - Notação de Objeto JavaScript - Citado em 21 , 52
LGPD	Lei Geral de Proteção de Dados - Citado em 25 , 37
LIBRAS	Língua Brasileira de Sinais - Citado em 31 , 62
MVP	<i>Minimum Viable Product</i> - Produto Mínimo Viável - Citado em 15 , 17 , 30
POC	<i>Prove of Concept</i> - Prova de Conceito - Citado em 17 , 30
RH	Recursos Humanos - Citado em 30
SPA	<i>Single Page Application</i> - Aplicação de Página Única - Citado em 21
SSD	<i>Solid-State Drive</i> - Unidade de Estado Sólido - Citado em 38
SSO	<i>Single Sign-On</i> - Login único - Citado em 30 , 31
TLS	<i>Transport Layer Security</i> - Segurança da Camada de Transporte - Citado em 2 , 7 , 56 , 61
URL	<i>Universal Resource Locator</i> - Localizador universal de recurso - Citado em 61
USD	<i>United States Dollar</i> - Dólares Americanos - Citado em 38 , 39

Sumário

1	INTRODUÇÃO	9
1.1	Justificativa	9
1.2	Proposta de solução	9
1.3	Objetivo	9
1.4	Análise de Concorrentes	10
2	REVISÃO DA LITERATURA	11
2.1	Estágio	11
2.1.1	Definição	11
2.1.2	Tipos de estágio	11
2.1.3	Carga horária	11
2.2	Competências	12
3	PLANEJAMENTO E GERENCIAMENTO DO PROJETO	14
3.1	Gestão e Desenvolvimento do Projeto	14
3.2	Organização da equipe	14
3.3	Cronograma	15
4	DESENVOLVIMENTO DA APLICAÇÃO	21
4.1	Arquitetura	21
4.1.1	Diagramas de arquitetura	21
4.2	Escopo	23
4.2.1	Requisitos	23
4.2.1.1	Requisitos Funcionais	23
4.2.1.2	Requisitos Não-funcionais	24
4.2.1.3	Regras de Negócio	25
4.2.2	Histórias de usuário	25
4.2.3	Casos de uso	28
4.2.4	Fases de entrega	30
4.2.4.1	Prova de Conceito (POC)	30
4.2.4.2	Produto Mínimo Viável (MVP)	30
4.2.4.3	Entrega Final	30
4.3	Integrações	31
4.3.1	Login com o Google e LinkedIn	31
4.3.2	Entrar em contato via <i>WhatsApp</i>	31
4.3.3	Acessibilidade com VLibras	31

4.3.4	API dos Correios	31
4.4	Manutenibilidade	31
4.4.1	<i>Logs</i>	32
4.4.2	<i>Code Convention</i>	32
4.4.2.1	Codificação geral	32
4.4.2.2	<i>Commits</i>	33
4.4.3	<i>Design Patterns</i> e boas práticas	33
4.4.3.1	<i>Clean Code</i>	33
4.4.3.2	<i>SOLID</i>	34
4.4.3.3	<i>12 Factor App</i>	34
4.4.4	Integração contínua	35
4.4.4.1	<i>Back-end</i>	35
4.4.4.2	<i>Front-end</i>	36
4.5	Segurança, Privacidade e Legislação	36
4.6	Viabilidade Financeira	37
4.6.1	Gerenciamento de custos	37
4.6.1.1	Desenvolvimento	38
4.6.2	Ambiente de produção	38
4.6.2.1	<i>Front-end</i>	38
4.6.2.2	<i>Back-end</i>	38
4.6.2.3	Banco de dados	38
4.6.3	Monetização	39
4.6.4	Conclusão	39
4.7	Modelagem e definições técnicas	39
4.7.1	Modelo Entidade Relacionamento	40
4.7.2	Diagrama Entidade-Relacionamento	41
4.7.3	Dicionário de Dados	41
4.7.4	<i>Endpoints</i> da API	44
4.7.5	Listagem das Competências	45
5	ESTATÍSTICAS E VALIDAÇÕES	47
5.1	SVN	47
5.2	GitHub	49
5.3	Teste dos <i>headers</i> da API	55
5.4	Teste de TLS do <i>front-end</i>	56
5.5	Teste de desempenho do <i>front-end</i>	57
5.6	Análise de código	58
5.7	Validador HTML	59
	GLOSSÁRIO	60

APÊNDICES	64
APÊNDICE A – PUBLICAÇÕES DO BLOG	65
APÊNDICE B – POC OVERVIEW	66
ANEXOS	68
ANEXO A – NOTA DOS HEADERS	69

1 Introdução

Nesse capítulo são 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 nos projetos pedagógicos de cursos profissionalizantes, técnicos, universitários e de outras modalidades de ensino. O estágio

[...] é uma etapa fundamental no processo de desenvolvimento e aprendizagem do aluno, porque promove oportunidades de vivenciar na prática conteúdos acadêmicos, propiciando a aquisição de conhecimentos e atitudes relacionadas com a profissão escolhida pelo estagiário. (??) conteúdo...

No entanto, ainda que os grandes objetivos do estágio sejam de aprendizado, aquisição de experiências e um primeiro contato com o mercado de trabalho, existem vagas de estágio com requisitos de experiência prévia, não condizentes com a situação do estudante, além disso, também nota-se que existe certa dificuldade de conexão entre a empresa e o candidato, que muitas vezes não obtém o retorno sobre o processo de seleção da vaga. Tais dificuldades ficam mais perceptíveis nas plataformas que disponibilizam vagas e não possuem um filtro para não hospedar vagas incoerentes com a condição de estágio nem uma prática de *feedback* ou alerta para os candidatos, que ficam sem saber sobre a situação da vaga nem a sua própria dentro do processo seletivo.

1.2 Proposta de solução

Tendo em vista os problemas anteriormente descritos, *EstagiEI* 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 Objetivo

Com nossa solução buscamos 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 competências e perfil sejam condizentes com o que é procurado.

Podemos definir nosso objetivo principal como:

- Construir uma aplicação onde de fato os estudantes encontrem vagas que condizem com a realidade de um estagiário.

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 são *Companhia de Estágios*, *Cia de Talentos*, *Nube* e *CIEE*. Com base neste levantamento, podemos observar algumas intersecções de funcionalidades oferecidas. O [Quadro 1](#) permite uma melhor visualização deste levantamento.

Quadro 1 – Comparação dos aplicativos concorrentes

Funcionalidades	Cia de Estágios	Cia de Talentos	Nube	CIEE	EstagiEI.
Login/Cadastro.	x	x	x	x	x
Aplicar em uma vaga.	x	x	x	x	x
Notificação a cada mudança do status no processo seletivo.			x	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	x
Denúncias de vagas incoerentes com a realidade.					x
<i>Feedback</i> de empresas pós-entrevista.					x

Fonte: Os Autores

Como citado em [seção 1.1](#), as plataformas não possuem um sistema de avaliação da vagas quanto ao cumprimento do que se espera de uma oportunidade de estágio. *Nube* e *CIEE* possuem sistemas de notificação sobre o *status* da vaga, mas apenas o *CIEE* oferece um contato mais direto com o candidato.

2 Revisão da Literatura

Neste capítulo buscamos explicitar conceitos e informações relevantes para o desenvolvimento da nossa proposta de solução *EstagiEI*, um sistema de vagas de estágio.

2.1 Estágio

Nesta seção são apresentados os principais elementos do estágio: sua definição, tipos e carga horária seguindo o estabelecido na legislação brasileira.

2.1.1 Definição

De acordo com a lei nº 11.788, de 25 de setembro de 2008, define-se estágio da seguinte forma:

Art. 1º Estágio é ato educativo escolar supervisionado, desenvolvido no ambiente de trabalho, que visa à preparação para o trabalho produtivo de educandos que estejam freqüentando o ensino regular em instituições de educação superior, de educação profissional, de ensino médio, da educação especial e dos anos finais do ensino fundamental, na modalidade profissional da educação de jovens e adultos. (??)

Portanto, o estágio se trata de uma fase intermediária entre os estudos e a entrada no mercado de trabalho, paralelamente ao primeiro, de acordo com as características da modalidade de ensino, projeto pedagógico e situação do estudante.

2.1.2 Tipos de estágio

Os estágios podem ser obrigatórios ou não-obrigatórios, dependendo do que foi previsto no projeto pedagógico do curso no qual o estudante está matriculado. O estágio do tipo obrigatório se caracteriza pelo requisito de cumprimento de uma determinada quantidade de horas estágio, juntamente com a aprovação nas disciplinas do curso, para a obtenção de diploma. O estágio não-obrigatório é opcional e as horas cumpridas são acrescidas à carga horária obrigatória do curso. (??)

2.1.3 Carga horária

O estágio não é regido pela [Consolidação das Leis do Trabalho \(CLT\)](#), assim possui sua própria especificação de jornada e carga horária. De acordo com o Art. 10 (??), a

jornada do estágio é definida em um acordo entre a escola e a empresa, ressaltando que não pode ultrapassar:

I – 4 (quatro) horas diárias e 20 (vinte) horas semanais, no caso de estudantes de educação especial e dos anos finais do ensino fundamental, na modalidade profissional de educação de jovens e adultos;

II – 6 (seis) horas diárias e 30 (trinta) horas semanais, no caso de estudantes do ensino superior, da educação profissional de nível médio e do ensino médio regular.

§ 1º O estágio relativo a cursos que alternam teoria e prática, nos períodos em que não estão programadas aulas presenciais, poderá ter jornada de até 40 (quarenta) horas semanais, desde que isso esteja previsto no projeto pedagógico do curso e da instituição de ensino.

§ 2º Se a instituição de ensino adotar verificações de aprendizagem periódicas ou finais, nos períodos de avaliação, a carga horária do estágio será reduzida pelo menos à metade, segundo estipulado no termo de compromisso, para garantir o bom desempenho do estudante.(??)

Nota-se que a carga horária dos estágios tentam ser de tal forma que o estudante tenha condições mínimas de realizar o estágio e ainda conseguir frequentar as aulas de modo apropriado.

2.2 Competências

As habilidades que possuímos podem ser classificadas como técnicas e comportamentais, nos termos em inglês, *hard skills* e *soft skills*, respectivamente.

O que chamamos de competências se referem às *soft skills*, ou seja, o conjunto de características comportamentais da pessoa. Essas características podem ser determinantes na busca por estágio e emprego, assim se faz necessário darmos destaque ao entendimento do que são tais habilidades.

"São capacidades subjetivas, que atuam no espectro comportamental e social do ser humano e não dependem de diplomas ou certificados."(??) As habilidades técnicas são mais fáceis de serem registradas, detectadas, mapeadas e relacionadas, porém o caráter subjetivo das competências faz com que a mesma habilidade possa ser descrita de modos distintos, ainda que similares, como 'Trabalho em equipe' pode ser dito como 'Saber trabalhar em equipe' ou 'Trabalhar bem em equipe', além de comumente serem importantes na obtenção e manutenção de uma vaga de estágio/emprego.

As vagas existentes de diversas áreas buscam por determinados perfis, os quais comumente se referem ao tipo de atitude procurada para preencher a vaga além dos conhecimentos técnicos. Conhecer o seu próprio perfil e aprimorar certas características dará maiores chances de encontrar vagas compatíveis consigo.

Na [subseção 4.7.5](#) estão listadas as competências parametrizadas para a associação das vagas com os estudantes no sistema *EstagiEI*.

3 Planejamento e Gerenciamento do Projeto

Neste capítulo são abordados a metodologia e ferramenta da gestão da equipe e do projeto, os papéis dos integrantes da equipe e informações a cerca do cronograma seguido no desenvolvimento do projeto e sua documentação.

3.1 Gestão e Desenvolvimento do Projeto

A equipe decidiu por utilizar a metodologia ágil *Scrum*, juntamente com a ferramenta de gerenciamento *Jira Software*. O *Scrum* possui três fases, uma inicial de planejamento geral, uma intermediária de produção e uma final de encerramento. A fase intermediária se trata de uma série de iterações, onde em cada iteração são desenvolvidas atividades/funcionalidades a serem entregues/incrementadas. Estas iterações são chamadas de *Sprints* (??), cuja duração é fixa e a equipe decidiu por durar uma semana (7 dias) no primeiro semestre e duas semanas (14 dias) no segundo semestre. Todas as atividades, elementos e artefatos que precisaram ser produzidos foram organizados, monitorados e atribuídos aos membros da equipe via *Jira Software*, onde foi possível verificar o status da atividade, assim como marcar prazos.

3.2 Organização da equipe

Após avaliarmos as principais competências de cada integrante da equipe, resolvemos separar as tarefas de cada um como indicado no *Quadro 2*.

Quadro 2 – Divisão de responsabilidades da equipe.

Responsabilidade	Bruna	Daniel	Igor	Leonardo	Lucas	Marcelo
<i>Back-end</i> .			X	X		X
<i>Front-end</i> .	X	X		X	X	
Banco de Dados.		X	X			
Blog.	X	X	X	X	X	X
Documentação.	X	X	X	X	X	X
Design.	X				X	
Gestão.	X					

Fonte: Os Autores

Considerando os papéis inerentes ao *Scrum* e as responsabilidades expostas no *Quadro 2*, o papel do *Scrum Master* será desempenhado pela integrante Bruna da Silva

Pires, já a equipe de desenvolvimento será composta por todos os integrantes da equipe, sem exceção.

3.3 Cronograma

No início do projeto tínhamos uma organização dos macro itens (*Epics*) que precisavam ser desenvolvidos com base nas entregas da disciplina de PI1A5 e dentro dos *Epics* estipulamos as tarefas a serem feitas. Por meio do [Jira Software](#) podemos ter uma visão geral do andamento dos *Epics* e os prazos, além das *Sprints* planejadas, realizadas e aquela que está em andamento.

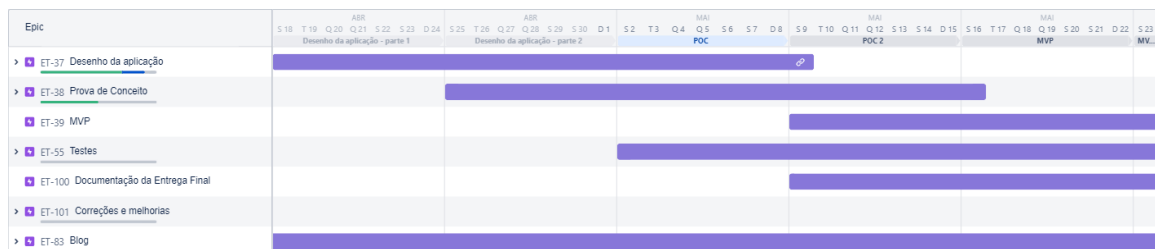
Figura 1 – Roteiro Geral de PI1A5



Fonte: Os Autores

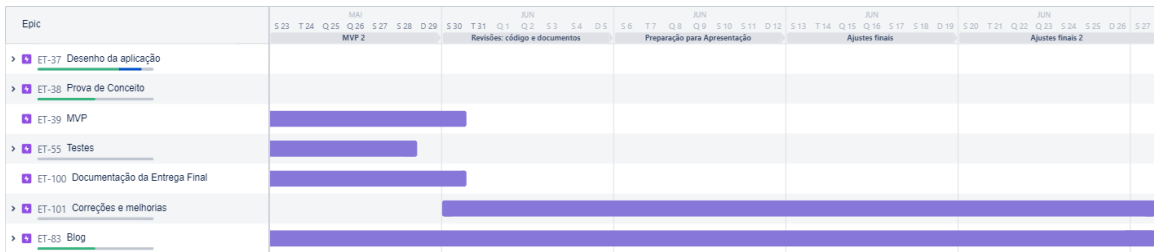
A [Figura 1](#) mostra à esquerda a lista dos *Epics* considerados para a construção do sistema *EstagiEI* no início do projeto, começando pelo Desenho da Aplicação, então Prova de Conceito, *Minimum Viable Product* (MVP), Testes, Documentação da Entrega Final, Correções e Melhorias e Blog. Na parte superior estão as datas e o período englobado por cada *Sprint*. As marcações em azul mostram o período de duração de cada *Epic*. A seguir, [Figura 2](#) e [Figura 3](#) mostram de modo mais próximo a figura anterior para fins de melhor visualização.

Figura 2 – Roteiro Geral de PI1A5 - Detalhe Inicial



Fonte: Os Autores

Figura 3 – Roteiro Geral de PI1A5 - Detalhe Final



Fonte: Os Autores

Apresentamos em [Quadro 3](#) as *Sprints* e algumas informações expostas em [Figura 1](#), [Figura 2](#) e [Figura 3](#) do que foi planejado e realizado em PI1A5.

Quadro 3 – Cronograma de Sprints - 1º semestre

Sprint	Data Inicial	Data Final	Descrição	Status
Desenho da aplicação 1	18/04/22	25/04/22	Elaboração da documentação do Desenho da Aplicação.	Concluída
Desenho da aplicação 2	25/04/22	02/05/22	Continuação da elaboração do Desenho da Aplicação. Planejamento para a <i>Prove of Concept (POC)</i> .	Concluída
POC	02/05/22	09/05/22	Finalização do Desenho da Aplicação. Início do desenvolvimento dos itens da POC	Concluída
POC 2	09/05/22	16/05/22	Continuação do desenvolvimento dos itens da POC.	Concluída
MVP	16/05/22	23/05/22	Aproveitamento do que foi desenvolvido para a POC com melhorias e ampliação conforme possível para o MVP.	Concluída
MVP 2	23/05/22	30/05/22	Continuação do trabalho no desenvolvimento do MVP.	Concluída
Revisões: código e documentos	30/05/22	06/06/22	Finalização e revisão tanto do desenvolvimento quanto da documentação.	Concluída
Preparação para a Apresentação	06/06/22	13/06/22	Organização e planejamento da apresentação do projeto e sua documentação.	Concluída
Ajustes finais	13/06/22	20/06/22	Ajustes a serem feitos para correção e/ou melhoria do projeto apresentado.	Concluída
Ajustes finais 2	20/06/22	27/06/22	Continuação de correções e ajustes para a entrega do projeto no semestre.	Concluída
Ajustes finais 3	27/06/22	04/07/22	Finalização dos ajustes finais e correções para a entrega definitiva do projeto no semestre.	Concluída

Fonte: Os Autores

Para a continuação do projeto, percebemos que uma mudança de planejamento seria necessária. Assim, modificamos nossos *Epics* a fim de estarem mais alinhados com o projeto em desenvolvimento, possuindo subitens de Funcionalidades que se referem a partes menores do produto em si, as quais por sua vez contém as Histórias de Usuário, que descrevem as ações/funções de cada usuário dentro do sistema.

Como o [Jira Software](#) não possui uma divisão intermediária entre *Epics* e Histórias de Usuário, a seguir apresentamos alguns esquemas da visão que temos das divisões do projeto:

Figura 4 – *Epic* da Empresa



Fonte: Os Autores

Em [Figura 4](#) está a Área da Empresa, que seria uma grande fatia do projeto, contendo as Funcionalidades pertinentes a entidade Empresa e o que se relaciona com ela, como o Gerenciamento de Vagas, o qual se abre em diversas ações que são as Histórias de Usuário dessa funcionalidade.

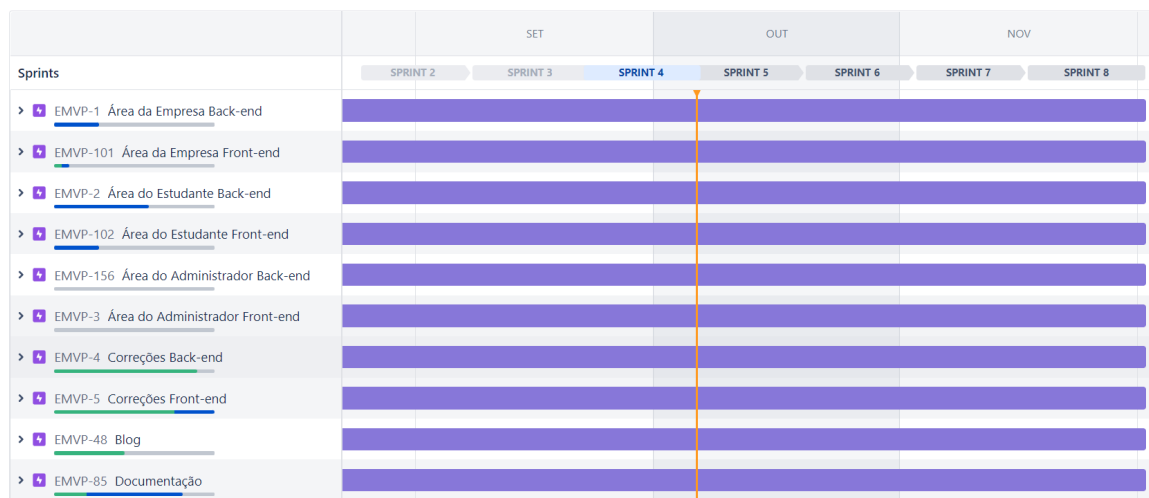
Figura 5 – *Epic* do Estudante

Fonte: Os Autores

Do mesmo modo como o *Epic* da Empresa, o *Epic* do Estudante também possui Funcionalidades pertinentes a entidade Estudante e as ações que um usuário deste tipo precisa ter no sistema, como o Gerenciamento de Candidaturas.

Além de reestruturarmos o modo como vemos e trabalhamos as etapas do projeto, também adaptamos as *Sprints* para serem de duas semanas ao invés de uma, devido à situação de tempo dos membros da equipe. O Figura 6 ilustra a nova organização de *Epics* divididas entre *back-end* e *front-end* e as *Sprints* de suas semanas.

Figura 6 – Roteiro Geral para PI2A6



Fonte: Os Autores

Considerando tudo o que já foi apontado, nosso Cronograma (a seguir) se tornou mais genérico, pois a cada *Sprint* definimos o que seria feito, portanto apenas há algo

definido explicitamente no início e no fim da segunda etapa de desenvolvimento, como a entrega final.

Quadro 4 – Cronograma de Sprints - 2º semestre

Sprint	Data Inicial	Data Final	Descrição	Status
<i>Sprint 1</i>	11/08/22	25/08/22	Reorganização da equipe, reestruturação do código e do projeto de modo geral	Concluída
<i>Sprint 2</i>	25/08/22	08/09/22	Refinamento, ajustes e adaptações do que já foi produzido; adequação da documentação existente às alterações	Concluída
<i>Sprint 3</i>	08/09/22	22/09/22	Desenvolvimento e testes de novas funcionalidades do sistema	Concluída
<i>Sprint 4</i>	22/09/22	06/10/22	Desenvolvimento e testes de novas funcionalidades do sistema	Concluída
<i>Sprint 5</i>	06/10/22	20/10/22	Desenvolvimento e testes de novas funcionalidades do sistema	Concluída
<i>Sprint 6</i>	20/10/22	03/11/22	Desenvolvimento e testes de novas funcionalidades do sistema	Concluída
<i>Sprint 7</i>	03/11/22	17/11/22	Ajustes finais, Entrega e Apresentação da aplicação	Em Progresso
<i>Sprint 8</i>	17/11/22	01/12/22	Correções e ajustes; Entrega final	Não Iniciada

Fonte: Os Autores

4 Desenvolvimento da Aplicação

Neste capítulo apresentamos a arquitetura do *EstagiEI*, seu escopo, integrações, questões de segurança, privacidade e legislação, assim como itens de manutenibilidade e viabilidade financeira.

4.1 Arquitetura

Para o desenvolvimento do projeto, e tendo em vista que seria construída uma aplicação *web* de página única, utilizamos de ferramentas que cerceiam o ecossistema de *Single Page Application* (SPA). Para isso, dividimos o projeto em *front-end* e *back-end* de modo que eles se comuniquem via protocolo *Hypertext Transfer Protocol* (HTTP) com requisições e respostas no formato *JavaScript Object Notation* (JSON). Para o desenvolvimento do *front-end* utilizamos *TypeScript* por meio da biblioteca *React*; o *back-end* foi desenvolvido utilizando Java com o micro *framework* *Spring Boot*.

Em relação ao *deploy* das aplicações, o *front-end* está hospedado na plataforma *Netlify*, que hospeda e mantém um site com implantação contínua e HTTPS, proporcionando uma melhor agilidade de desenvolvimento, enquanto o *back-end* está hospedado no *Heroku*, que é uma plataforma como serviço de fácil manuseio e que permite a equipe ter um maior foco no desenvolvimento do projeto. Através do *Heroku* podemos também fazer a utilização do *PostgreSQL* por meio do serviço de apoio *Heroku Postgres*.

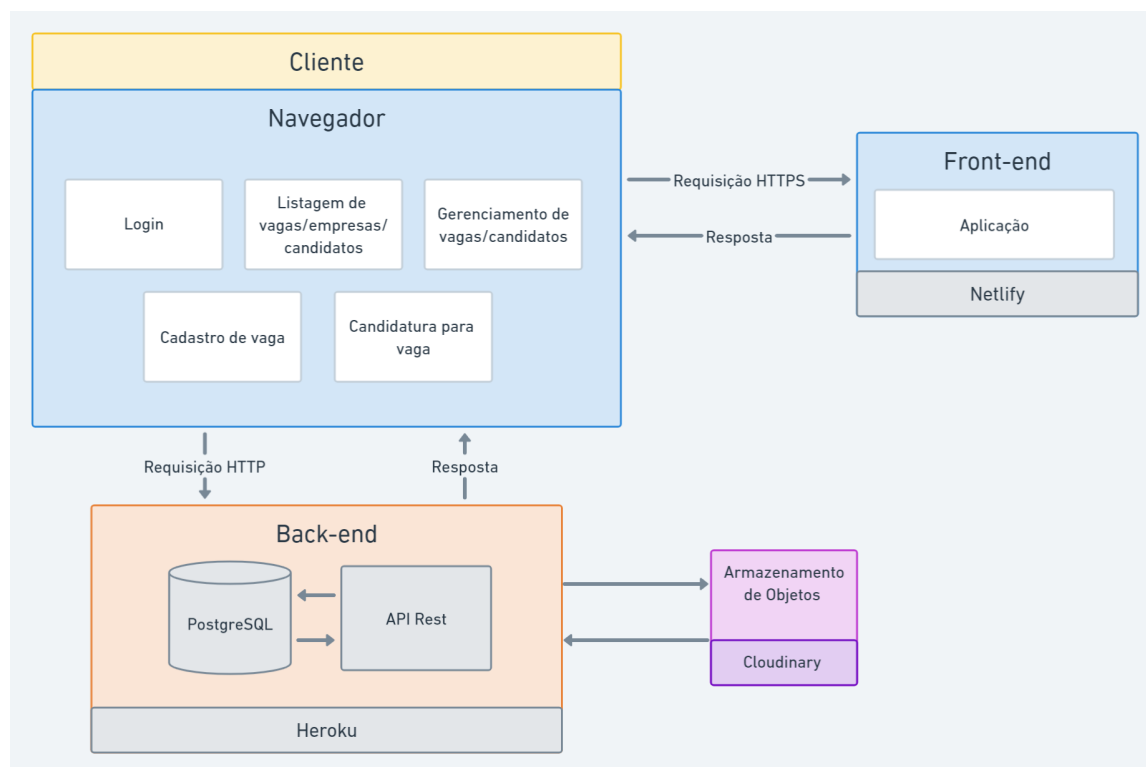
Ademais, para o armazenamento de objetos como arquivos ou imagens, utilizamos a plataforma *Cloudinary*, principalmente por sua fácil integração com a linguagem de programação Java através de bibliotecas.

4.1.1 Diagramas de arquitetura

Os diagramas *Figura 7*, *Figura 8* e *Figura 9* ilustram de modo geral a arquitetura planejada e implementada para a solução proposta, utilizando das tecnologias já citadas.

A *Figura 7* ilustra a aplicação construída seguindo o ecossistema *SPA*, onde o navegador carrega toda a aplicação hospedada no *Netlify*, então realiza requisições para o *back-end* hospedado no *Heroku*, recebendo respostas no formato *JSON*, as quais são manipuladas e exibidas de acordo com a necessidade pelo *TypeScript* da aplicação no navegador.

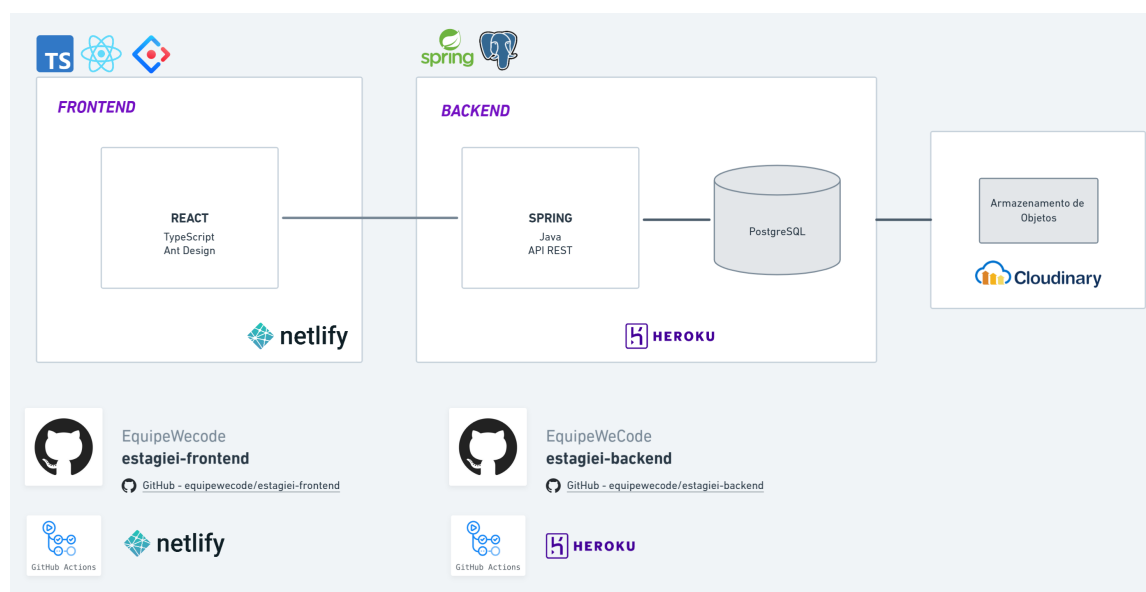
Figura 7 – Arquitetura de Aplicação



Fonte: Os Autores

A Figura 8 ilustra a mesma arquitetura, porém dando destaque para as tecnologias utilizadas tanto para hospedagem quanto desenvolvimento e versionamento.

Figura 8 – Arquitetura Tecnológica

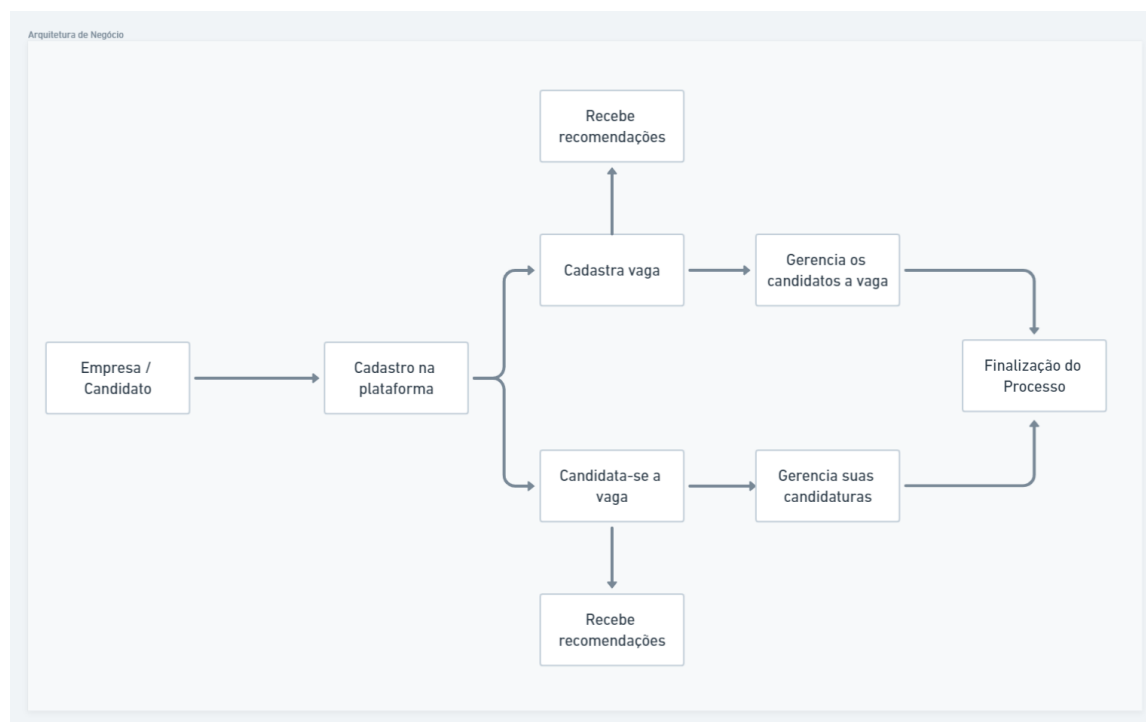


Fonte: Os Autores

A Figura 9 por sua vez, resume a ideia de negócio por trás do sistema *EstagiEI*

com relação ao cadastramento de empresas e estudantes (Candidato), suas ações com as vagas (cadastramento ou candidatura), o gerenciamento de suas relações com as vagas e o recebimento de recomendações, sendo para empresa de candidatos e os estudantes recebem vagas de acordo com seu perfil.

Figura 9 – Arquitetura de Negócios



Fonte: Os Autores

4.2 Escopo

Neste tópico são abordados os casos de uso da aplicação (forma de descrever uma funcionalidade do sistema); diagrama de requisitos (identificação das funcionalidades a serem implementadas); histórias de usuário (descrição das necessidades do usuário); e definição de entregas (quais funcionalidades estarão disponíveis nas principais entregas).

4.2.1 Requisitos

Para o desenvolvimento da aplicação *EstagiEI*, são expostos os requisitos funcionais, não-funcionais e regras de negócio que a aplicação terá, tais requisitos foram formados a partir de estudos de como irão funcionar os processos do sistema em construção.

4.2.1.1 Requisitos Funcionais

Os requisitos funcionais dizem respeito às principais funcionalidades que o sistema deve empenhar (??). Durante nossa análise, foram decididos os principais requisitos

funcionais da aplicação como descrito no [Quadro 5](#):

Quadro 5 – Requisitos funcionais

Código	Descrição
RF-001	Permitir a busca de vagas por filtros
RF-002	Recomendar vagas para estudantes
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 entre em contato com os estudantes recomendados e/ou 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 <i>feedback</i> 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
RF-010	Permitir o cadastro de vagas por parte da empresa, seguindo as regras estabelecidas
RF-011	Recomendar estudantes para vagas
RF-012	Manter um histórico de vagas para a empresa
RF-013	Possibilitar que a Empresa gerencie suas Vagas
RF-014	Possibilitar que a Empresa gerencie seus Representantes
RF-015	Possibilitar que a empresa faça um pré-cadastro para ter acesso ao sistema
RF-016	Permitir que estudantes se cadastrem no sistema
RF-017	Possibilitar que estudantes gerenciem o seu perfil, adicionando, alterando e/ou retirando informações.
RF-018	Possibilitar que os estudantes possam se candidatarem à uma vaga.
RF-019	Possibilitar que os estudantes possam retirar suas candidaturas às vagas.
RF-020	Possibilitar que a empresa visualize facilmente as informações das suas vagas e candidaturas aplicadas à elas
RF-021	Possibilitar que o Administrador do sistema possa gerenciar as empresas
RF-022	Possibilitar que o Administrador entre em contato com o estudante
RF-023	Possibilitar que o Administrador entre em contato com a empresa

Fonte: Os Autores

4.2.1.2 Requisitos Não-funcionais

Ao contrário dos requisitos funcionais, os requisitos não-funcionais não estão ligados às principais funcionalidades de um sistema, mas sim com seus fatores de restrições e especificações. É a partir deles que são observados aspectos como desempenho, usabilidade, segurança e outros aspectos não-funcionais que tangem o sistema (??). Tendo isto em mente, no [Quadro 6](#) são elencados os principais requisitos não-funcionais.

Quadro 6 – 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 (LGPD)
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

4.2.1.3 Regras de Negócio

As regras de negócio, que estão ligadas aos requisitos funcionais previamente descritos, do nosso projeto estão listados no [Quadro 7](#).

Quadro 7 – Regras de negócio

Código	Descrição	Requisito Relacionado
RN-001	As vagas a serem cadastradas devem estar coerentes com o perfil buscado	RF-010
RN-002	Os históricos das vagas devem ser mantidos 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 <i>feedback</i> , ele deve ter pelo menos iniciado o processo seletivo	RF-008
RN-005	O <i>feedback</i> pode ser feito de forma anônima, mas o usuário deve estar logado e ter passado pelo processo seletivo	RF-008
RN-006	A recomendação de vagas deve ocorrer para estudantes devidamente cadastrados que possuam ao menos uma competência em seu perfil.	RF-002

Fonte: Os Autores

4.2.2 Histórias de usuário

A seguir apresentamos as histórias de usuário da aplicação, divididas entre os *Epics*. Primeiro as histórias relacionadas ao *Epic* da Empresa em [Quadro 8](#), posteriormente o do Estudante em [Quadro 9](#) e por fim do Administrador em [Quadro 10](#).

Quadro 8 – Histórias de usuário - Empresa

Código	Nome	Descrição
H001	Cadastrar Vagas	Como empresa, eu quero poder gerenciar (cadastrar, editar, visualizar detalhes, listar) vagas dentro do sistema para poder deixá-las visíveis e acessíveis para possíveis candidatos.
H002	Detalhes da Vaga	(Idem H001)
H003	Abrir Candidaturas	Como empresa, eu quero ter a possibilidade de abrir as candidaturas para uma vaga para que estudantes possam se candidatar.
H004	Fechar Candidaturas	Como empresa, eu quero ter a possibilidade de fechar as candidaturas para uma vaga para que estudantes não possam mais se candidatar.
H005	Editar Vagas	(Idem H001)
H006	Excluir Vagas	Como empresa, eu quero poder excluir uma vaga para retirá-la de visualização completamente.
H007	Listar Vagas	(Idem H001)
H008	Receber Recomendações	Como empresa, eu quero receber recomendações de estudantes para as vagas para que a busca por candidatos seja facilitada.
H009	Histórico de Candidaturas	Como empresa, eu quero um histórico dos estudantes que se candidataram às vagas para que eu possa contactá-los se necessário para aquela vaga ou outras ainda em aberto.
H010	Dashboard de Visualização	Como empresa, eu quero uma forma rápida e fácil de visualizar as informações pertinentes às minhas vagas para gerenciá-las.
H011	Cadastrar Representantes	Como empresa, eu quero gerenciar (cadastrar, editar, ver detalhes, listar) representantes para que eu tenha melhor controle e conhecimento daqueles que podem gerenciar minhas vagas por mim.
H012	Detalhes do Representante	(Idem H011)
H013	Editar Representantes	(Idem H011)
H014	Excluir Representante	Como empresa, eu quero excluir um representante para que este não possa mais acessar o sistema em meu nome nem gerenciar minhas vagas.
H015	Listar Representantes	(Idem H011)
H016	Login Representantes	Como empresa, eu quero autorizar que um representante entre no sistema em meu nome para que este representante gere minhas vagas.
H017	Solicitar Cadastro no site	Como empresa, eu quero poder solicitar meu cadastro no site para ter acesso ao sistema.
H018	Contato com Estudante	Como empresa, eu quero me comunicar de forma fácil com os estudantes candidatos para que o processo seja mais ágil.
H019	Solicitar exclusão da Conta	Como empresa, eu quero poder solicitar a exclusão da minha conta e todos os meus dados para que minha propriedade sobre eles seja respeitada, de acordo com a LGPD.

Fonte: Os autores

Quadro 9 – Histórias de usuário - Estudante

Código	Nome	Descrição
H020	Buscar Vagas	Como estudante, eu quero buscar vagas podendo usar de filtros para facilitar minha pesquisa e escolha de vaga.
H021	Receber Recomendações	Como estudante, eu quero receber recomendações de vaga para que a minha pesquisa seja facilitada.
H022	Candidatar-se à Vaga	Como estudante, eu quero poder me candidatar à uma vaga para ter a chance de ser selecionado para um estágio.
H023	Retirar Candidatura	Como estudante, eu quero poder retirar minha candidatura para uma vaga para que eu não possa ser selecionado para ela.
H024	Listar Candidaturas	Como estudante, eu quero um histórico de todas as minhas vagas já aplicadas para poder gerenciá-las melhor.
H025	Acompanhar Etapas	Como estudante, eu quero uma linha do tempo com os principais passos do processo para que eu possa acompanhá-lo de forma fácil e rápida.
H026	Notificação de Mudança na Vaga	Como estudante, eu quero ser alertado sobre as mudanças no status da vaga para que possa saber de forma rápida sua situação.
H027	Denunciar Vagas	Como estudante, eu quero poder denunciar vagas que possuam algum tipo de irregularidade para que elas sejam retiradas do sistema.
H028	Buscar Empresas	Como estudante, eu quero poder buscar as empresas cadastradas no site para facilitar minha busca pelas vagas pertencentes àquela empresa.
H029	Detalhes de uma Empresa	Como estudante, eu quero poder ver os detalhes de uma empresa para poder decidir se suas vagas podem vir a me interessar ou não.
H030	Vagas de uma empresa	Como estudante, eu quero poder ver as vagas de uma empresa específica para facilitar a minha busca.
H031	Dar Feedback	Como estudante, eu quero poder dar um feedback sobre o processo seletivo/empresa da qual participei da seleção/estágio para que outros utilizadores do site tenham mais informações sobre aquela empresa.
H032	Detalhes do Perfil	Como estudante, eu quero ver os detalhes do meu perfil para verificar as informações contidas ali.
H033	Editar Perfil	Como estudante, eu quero editar o meu perfil para melhorar/atualizar/corrigir as informações apresentadas.
H034	Adicionar Competências	Como estudante, eu quero poder gerenciar (adicionar, retirar, ver todas) as competências do meu perfil para mantê-lo atualizado e/ou direcionar as recomendações de vagas.
H035	Retirar Competências	(Idem H034)
H036	Listar Competências	(Idem H034)
H037	Solicitar exclusão da Conta	Como estudante, eu quero poder solicitar a exclusão da minha conta e todos os meus dados para que minha propriedade sobre eles seja respeitada, de acordo com a LGPD.
H038	Cadastro no site	Como estudante, eu quero poder me cadastrar no site para poder ter acesso ao sistema.
H039	Login	Como estudante, eu quero poder fazer login no site para ter acesso às funcionalidades do sistema.

Fonte: Os autores

Quadro 10 – Histórias de usuário - Administrador

Código	Nome	Descrição
H040	Cadastrar Empresa	Como Admin, eu quero poder gerenciar (cadastrar, editar, ver detalhes) as empresas do site para ter um melhor controle de quem está oferecendo vagas de estágio.
H041	Editar Empresa	(Idem H040)
H042	Página da Empresa	(Idem H040)
H043	Excluir Empresa	Como Admin, eu quero poder excluir uma conta de empresa e todos os seus dados para cumprir com a LGPD quando houver uma solicitação condizente.
H044	Receber Denúncias	Como Admin, eu quero receber as denúncias feitas por estudantes para facilitar a busca por vagas inadequadas no sistema.
H045	Excluir Vagas Inadequadas	Como Admin, eu quero poder excluir vagas inadequadas do sistema para deixá-lo o mais condizente com a lei e com as necessidades dos estudantes.
H046	Contato com Estudante	Como Admin, eu quero ter contato fácil com os estudantes para transmitir informações pertinentes sobre o sistema e as vagas.
H047	Contato com a Empresa	Como Admin, eu quero ter contato fácil com as empresas para transmitir informações pertinentes sobre o sistema e as vagas.
H048	Receber solicitações de exclusão de Contas	Como Admin, eu quero receber solicitações de exclusão de contas dos usuários para poder cumprir com as determinações da LGPD..
H049	Excluir Perfil/Conta do Estudante	Como Admin, eu quero poder excluir uma conta de estudante e todos os seus dados para cumprir com a LGPD quando houver uma solicitação condizente.

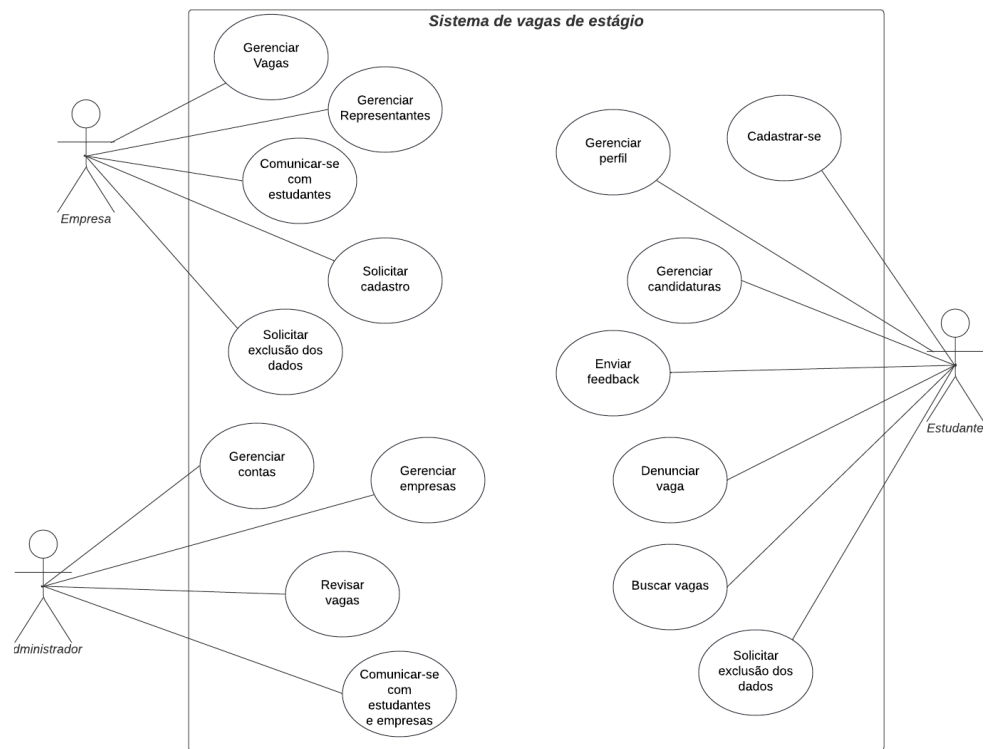
Fonte: Os autores

Cada história de usuário está relacionada com um caso de uso, os quais serão explicados melhor a seguir, e com algum requisito e/ou regra de negócio. As relações são indicadas de modo mais detalhado no <AUTO-REFERENCIAR APÊNDICE>.

4.2.3 Casos de uso

A [Figura 10](#) mostra os casos de uso que são pertinentes a aplicação, demonstrando os principais atores e suas funcionalidades dentro do sistema.

Figura 10 – Caso de Uso - EstagiEI



Fonte: Os Autores

O [Quadro 11](#) mostra os mesmos casos de uso outra forma, associando cada caso com um código que será usado em <AUTO-REFERENCIAR APÊNDICE>, relacionando com as histórias de usuário, requisitos e regras de negócio.

Quadro 11 – Casos de uso

Código	Ator	Descrição
UC001	Empresa	Gerenciar Representantes
UC002	Empresa	Cadastrar a Vaga
UC003	Empresa	Gerenciar a Vaga
UC004	Empresa	Mudar Status da Vaga
UC005	Estudante	Buscar Vagas
UC006	Estudante	Gerenciar Candidaturas
UC007	Estudante	Denunciar Vaga
UC008	Estudante	Enviar Feedbak sobre a empresa
UC009	Admin	Moderar as Vagas oferecidas no sistema
UC010	Admin	Gerenciar Contas
UC011	Admin	Gerenciar Empresas
UC012	Empresa	Solicitar cadastro no sistema
UC013	Estudante	Cadastrar-se no sistema
UC014	Empresa/Estudante	Solicitar exclusão dos dados/conta do sistema
UC015	Estudante	Gerenciar Perfil
UC016	Admin	Comunicar-se com Empresas e/ou Estudantes
UC017	Empresa	Comunicar-se com os Estudantes

Fonte: Os autores

4.2.4 Fases de entrega

Nessa seção são expostas quais funcionalidades do sistema foram e serão desenvolvidas, tendo em vista as principais fases de entrega da disciplina, sendo elas a [POC](#), o [MVP](#) e a Entrega Final.

4.2.4.1 Prova de Conceito (POC)

Na fase de [POC](#), foram entregues as funcionalidades mais básicas do nosso software. Dentre elas, o cadastro de estudantes via *Single Sign-On (SSO)* da *Google*, onde é explicado o processo no site possibilitando a criação de uma conta com informações básicas, necessárias apenas para o funcionamento padrão do sistema, e o cadastro de empresas, que será feito no próprio sistema *EstagiEI*, onde a empresa preenche as informações e passa por uma aprovação da equipe. Além disso, o software permite o *login* desses usuários já cadastrados, onde poderão consultar suas informações básicas.

Ao se cadastrar no sistema, a empresa também poderá registrar uma pessoa do [Recursos Humanos \(RH\)](#), que será responsável por gerenciar as vagas daquela organização, e essa pessoa poderá criar novas vagas com informações básicas, apenas para serem visíveis na tela de consulta de vagas. Na parte do estudante, será possível para ele(a), consultar as vagas que existem no sistema através de filtros básicos e internacionalização de linguagem.

4.2.4.2 Produto Mínimo Viável (MVP)

Na entrega do [MVP](#), foi incrementado o que já foi desenvolvido durante a [POC](#) com funcionalidades importantes ao nosso sistema, como a recomendação de vagas ao estudante e edição de perfil. Além disso, foram feitos os testes unitários e validações de segurança, [HTML](#), testes de interface, etc. A fim de garantir que a aplicação esteja em conformidade com os requisitos solicitados.

4.2.4.3 Entrega Final

Na entrega final, serão acrescentados no nosso projeto o que já foi feito antes com o restante das funcionalidades, tais como a possibilidade do estudante denunciar uma vaga por não ser coerente com a proposta da nossa aplicação, que é ser um sistema que possua vagas de estágio coerentes com a realidade de um estagiário; recomendação de candidatos para empresas; opção de contato com o candidato via *Whatsapp*; *dashboard* de vagas para a empresa; histórico de vagas para os estudantes; mudança de *status* das vagas por parte da empresa; *feedback* de empresas após o processo seletivo; e acessibilidade com o [VLibras](#).

4.3 Integrações

Nessa seção apresentamos as integrações elencadas como sendo possíveis e/ou desejáveis para a aplicação, que foram pensadas com base em outras aplicações do mercado.

4.3.1 Login com o Google e LinkedIn

Levando em conta a experiência de usuário, consideramos colocar a opção do estudante se logar através do [SSO](#) dessas empresas. Dessa forma, não seria necessário digitar a senha toda vez que o usuário for usar o sistema, precisando apenas clicar em um botão e fazer o *login* com uma dessas alternativas. Contudo, como as empresas e seus representantes usam um sistema de cadastro e login próprio da nossa aplicação, tornou-se complexo manter mais de uma forma de acesso para o sistema, assim, optamos por não dar continuidade com essa integração.

4.3.2 Entrar em contato via *WhatsApp*

A aplicação teria uma forma da empresa contatar o estudante via *WhatsApp*. Essa integração seria feita via [Application Programming Interface \(API\)](#) disponibilizada pela própria empresa que mantém o aplicativo. Dessa forma, com apenas um clique, seria possível enviar uma mensagem diretamente ao estudante. No entanto, a [API](#) do *WhatsApp* é exclusiva para parceiros, assim esta integração só poderá ser implementada no futuro.

4.3.3 Acessibilidade com VLibras

A Lei Brasileira de Inclusão, Art. 63, estipula que os sites devem ser acessíveis de modo a garantir o acesso às informações disponíveis (??), assim, realizaremos a integração com a aplicação [VLibras](#), que é um tradutor de texto escrito em Português para [Língua Brasileira de Sinais \(LIBRAS\)](#). De acordo com o manual do [VLibras](#), esta integração pode ser realizada com a inclusão de um trecho de código na página [HTML](#) da aplicação (??).

4.3.4 API dos Correios

Realizaremos uma integração do a [API](#) dos Correios a fim de resgatar as informações de endereço dos usuários cadastrados a partir do CEP informado.

4.4 Manutenibilidade

Para que a aplicação atinja um nível adequado de qualidade é fundamental que se estabeleça certos requisitos e parâmetros de manutenibilidade, tais como ferramentas

que facilitam esse processo. Através dos critérios estabelecidos, podemos medir o quanto o processo de desenvolvimento concorda com as boas práticas e incentivar o uso das mesmas.

4.4.1 *Logs*

Para o monitoramento da aplicação em tempo de execução, essencialmente na camada de servidor, os *logs* serão usados para monitorar o estado dos objetos. A ferramenta a ser utilizada será a implementação de textitlogs do *Spring Boot* que utiliza a implementação *Logback*. A ferramenta permite diversos registros, como:

- *debug*
- *info*
- *warn*
- *error*

Assim, a cada bloco de falha da aplicação um *log* será colocado para que os problemas sejam identificados, analisados e resolvidos.

4.4.2 *Code Convention*

Visando facilitar o entendimento mútuo entre a equipe, são feitas as convenções de código com o propósito de padronizar como os integrantes da equipe produzem seus respectivos códigos, de modo que o estilo de programação seja independente de seus autores. As convenções de código estabelecem estilos para a organização do código textualmente, ou seja, como os comentários são posicionados, nome de variáveis escolhidas.

4.4.2.1 Codificação geral

As convenções adotadas são baseadas na especificação da *SUN MICROSYSTEMS*, de 1996. É comumente usada no desenvolvimento na linguagem *Java*, e relativamente próxima do padrão adotado no *JavaScript*, podendo destacar os seguintes pontos:

- Minimização do uso de variáveis, funções e objetos globais.
- Declarações globais estarão de forma preferencial no início do arquivo.
- Declaração de variáveis próximo do ponto onde são inicializadas.
- Indentação de 4 espaços no *back-end* e 2 espaços no *front-end*.
- Classes e interfaces em **CamelCase** e substantivos.

- Métodos em **camelCase** e verbos.
- Constantes em **UPPER_CASE**.

No *back-end* os pacotes são bem divididos, tendo o pacote *entity* para as entidades mapeadas do banco de dados, *controller* para os *controllers* e *endpoints* e *service*, além de outros pacotes para fins de separação de código.

4.4.2.2 Commits

Para os *commits* dos repositórios de *front-end* e *back-end* estamos utilizando a convenção de usar prefixos que melhor identificam do que se trata aquela *commit*.

São eles:

- **fix**: correção de erros no código;
- **feat**: introdução de uma nova funcionalidade;

Além disso também realizamos a prática de realizar *Pull Requests* ao invés de mandar as alterações diretamente no ramo principal do repositório.

4.4.3 Design Patterns e boas práticas

Para padrões de projetos, serão essencialmente utilizados 3 padrões muito utilizados pela comunidade de desenvolvimento: *Factory Method*, *Builder* e *Facade*, além da possibilidade de usarmos outros conforme a necessidade.

4.4.3.1 Clean Code

O *Clean Code* é um conjunto de boas práticas de programação que visam melhorar o entendimento do código, facilitando a leitura do mesmo. Algumas das principais boas práticas são listadas a seguir:

- Nomes significativos para as variáveis, classes, métodos, atributos e objetos.
- Utilização de constantes e *enums* para evitar números mágicos.
- Evitar comentários que são redundantes e podem ser convertidos em códigos.
- Utilização de funções pequenas, com uma única responsabilidade abstrata.
- Evitar booleanos de forma explícita.
- Diminuir a redundância e a repetição de código (*Don't Repeat Yourself*).

- Aumentar a ortogonalidade do código: diminuindo as dependências; aumentando o desacoplamento e a independência entre os módulos de modo a deixá-los mais fáceis de serem modificados (*Easy To Change*).

4.4.3.2 SOLID

O *SOLID* é um acrônimo para 5 (cinco) princípios da programação orientada a objetos, tais princípios são fundamentais para o desenvolvimento e manutenção de software, visto que trazem facilidade e flexibilidade no código em se adequar à mudanças, algo frequente no desenvolvimento. Os princípios do *SOLID* estão expostos no [Quadro 12](#).

Quadro 12 – *SOLID Principles*

Princípio	Descrição
<i>Single Responsibility Principle</i>	Uma classe deve ter apenas um motivo para mudar.
<i>Open-Closed Principle</i>	Uma classe deve estar aberta para extensão e fechada para modificação, recomendando sempre utilizar a herança e não modificar o código-fonte original.
<i>Liskov Substitution Principle</i>	Uma classe derivada deve ser substituível por sua classe base.
<i>Interface Segregation Principle</i>	Utilizar muitas interfaces específicas é melhor que uma interface genérica.
<i>Dependency Inversion Principle</i>	Dependa de abstrações e não de implementações.

Fonte: (??)

4.4.3.3 12 Factor App

A aplicação doze-fatores é uma metodologia para construir softwares como serviço que seguem os parâmetros expostos no [Quadro 13](#).

Quadro 13 – 12 Factor App parameters

Parâmetros	Descrição
Base de Código	Uma base de código com rastreamento utilizando controle de revisão, muitos <i>deploys</i> .
Dependências	Declare e isole as dependências.
Configurações	Armazene as configurações no ambiente.
Serviços de Apoio	Trate os serviços de apoio, como recursos ligados.
Construa, lance, execute	Separe estritamente os <i>builds</i> e execute em estágios.
Processos	Execute a aplicação como um ou mais processos que não armazenam estado.
Vínculo de porta	Exporte serviços por ligação de porta.
Concorrência	Dimensione por um modelo de processo.
Descartabilidade	Maximizar a robustez com inicialização e desligamento rápido.
<i>Dev/prod</i> semelhantes	Mantenha o desenvolvimento, teste, produção o mais semelhante possível.
<i>Logs</i>	Trate logs como fluxo de eventos.
Processos de <i>Admin</i>	Executar tarefas de administração/gerenciamento como processos pontuais.

Fonte: (??)

4.4.4 Integração contínua

Para manter o serviço sempre atualizado para o usuário, a ferramenta de integração contínua do [Heroku CI](#) foi selecionada para a implantação da aplicação no *back-end* em produção. O mesmo foi feito com o [Netlify](#) em relação ao *front-end*.

4.4.4.1 Back-end

1. Uma mudança é feita no código do *back-end* e enviada ao repositório no [GitHub](#);
2. Após a mudança do código no repositório, uma instância da [Heroku CI](#), identifica automaticamente a linguagem de programação usada;
3. O processo do *deploy* é iniciado e a [Heroku CI](#) constrói o código em uma aplicação temporária;
4. A aplicação temporária passa por testes paralelos, cujos resultados são mostrados ao usuário através de uma interface;
5. Caso a aplicação temporária construída passe pelos testes com sucesso, é feito o *deploy* da aplicação na plataforma de hospedagem, no caso sendo o [Heroku](#).

A implantação dos itens acima se deu por meio de *Actions* do [GitHub](#).

Figura 11 – URL de *Actions* do *back-end*

<<https://github.com/EquipeWeCode/estagiei-backend/blob/develop/.github/workflows/deploy-develop.yml>>

Fonte: Os Autores.

4.4.4.2 *Front-end*

1. Uma mudança é feita no código do *front-end* e enviada ao repositório no *GitHub*;
2. Após uma mudança do código no *GitHub*, uma instância da *Netlify*, que tem acesso ao código do *GitHub*, identifica automaticamente essa mudança;
3. O novo código passa por testes, cujos resultados são mostrados ao usuário através de uma interface;
4. Após passar nos testes com sucesso, é feito o *deploy* da aplicação na plataforma de hospedagem, no caso sendo o *Netlify*.

A implantação dos itens acima se deu por meio de *Actions* do *GitHub*.

Figura 12 – URL de *Actions* do *front-end*

<<https://github.com/EquipeWeCode/estagiei-frontend/blob/develop/.github/workflows/main.yml>>

Fonte: Os Autores.

4.5 Segurança, Privacidade e Legislação

Para o desenvolvimento da aplicação, foi levado em consideração alguns aspectos de segurança, privacidade e legislação. A lei brasileira que diz respeito a como lidar com

dados de pessoas em plataformas digitais (sobretudo em aplicações disponíveis na internet) é a Nº 13.709 (??), que está em vigor desde 2020, a [LGPD](#).

De acordo com o estabelecido na [LGPD](#), a aplicação irá, se necessário, recuperar o mínimo de dados possíveis do usuário para prosseguir com a sua utilização, como *e-mail*, nome e informações sobre a instituição de ensino do usuário por parte do candidato e o [Cadastro Nacional da Pessoa Jurídica \(CNPJ\)](#) da empresa por parte da empresa que irá cadastrar as vagas. Sempre que for necessário a obtenção de tais informações por parte do sistema, o usuário será alertado de tal ocorrência.

Também foram levadas em consideração algumas outras questões fundamentais de segurança durante o desenvolvimento da aplicação, visto que utilizamos no *back-end* uma [API](#) para a transferência de dados e comunicação com o *front-end*:

- Autenticação e Autorização: As requisições apenas são aceitas se o usuário estiver autenticado no sistema e os *endpoints* funcionam de acordo com a autorização baseada em papéis;
- Criptografia: Seguimos o protocolo e padrão [Hypertext Transfer Protocol Secure \(HTTPS\)](#) para a transferência de mensagens entre *back-end* e *front-end*, de modo que as mensagens fiquem encriptadas e garantir maior segurança na aplicação;
- Não exposição de dados sensíveis à aplicação: Durante o desenvolvimento da aplicação, senhas para comunicação com serviços externos e outras ferramentas não ficam expostas em código, e sim, são passadas através de variáveis de ambiente de modo a não expor chaves e/ou senhas importantes.
- Política de senhas: as senhas dos usuários nunca serão armazenadas diretamente no banco de dados; um algoritmo gera um *hash* e faz a sua comparação no momento da autenticação.

4.6 Viabilidade Financeira

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

4.6.1 Gerenciamento de custos

Aqui serão abordados os custos de desenvolvimento e o porte inicial do projeto.

4.6.1.1 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 serão totalmente voluntários, sem custo adicional ao projeto.

4.6.2 Ambiente de produção

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

4.6.2.1 *Front-end*

A camada cliente da aplicação é hospedada na plataforma [Netlify](#), sendo o custo de processamento e requisições da aplicação baixo inicialmente, a hospedagem da camada cliente não apresentará custo adicional.

4.6.2.2 *Back-end*

Inicialmente gratuito na plataforma [Heroku](#).

A partir do momento que for necessário grande porte, será indicado a migração para a [AWS](#), [Heroku](#) com o seu plano pago 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.

4.6.2.3 Banco de dados

Inicialmente gratuito na plataforma [Heroku](#) através do serviço de apoio [Heroku Postgres](#).

Caso a aplicação fique com um porte maior, será indicado a migração para a [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.

4.6.3 Monetização

A fim de gerar receita para a plataforma, são consideradas duas possibilidades de monetização:

- Propagandas, através do 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 renda e
- Contratos, através de 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, consequentemente gerando 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.

4.6.4 Conclusão

Utilizando inicialmente os servidores de baixo porte detalhados acima, não haverá custo adicional a priori. Contudo, com o advento do encerramento dos planos gratuitos do *Heroku*, ocasionando no futuro desligamento do *back-end*, a análise de viabilidade e custos de provisionamento de infraestrutura para a hospedagem do projeto, citada anteriormente, é um grande auxílio para se pensar no futuro mantimento da aplicação em funcionamento. Com isso em mente, foi calculado que para 250 visitantes diários, utilizando os parâmetros detalhados anteriormente, teremos uma arrecadação de 10.5 USD mensais.

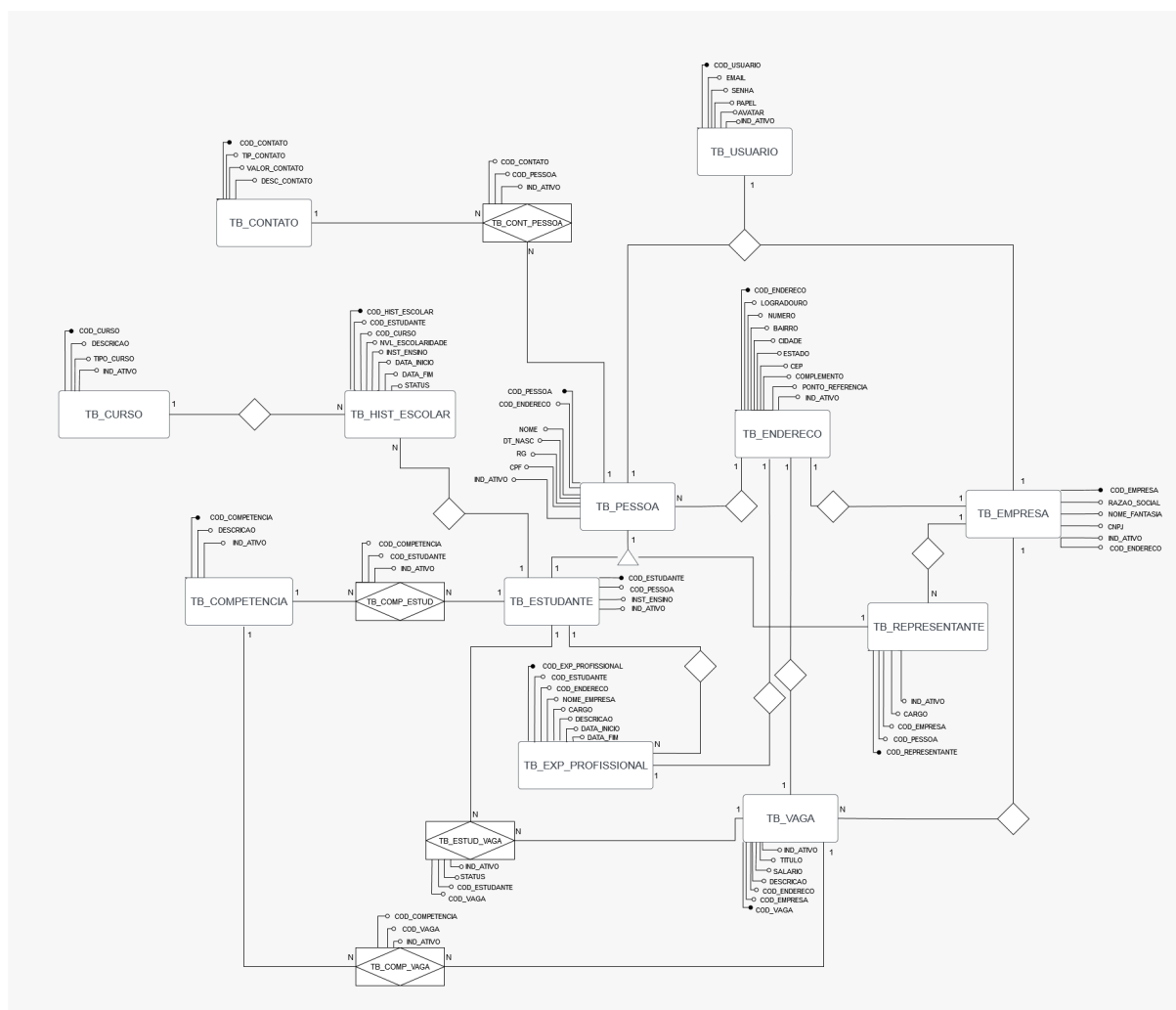
Caso o engajamento da aplicação aumente, a medida que o número de usuários aumenta, incrementando proporcionalmente o rendimento com o *Google Adsense*, poderá ser revisto os planos dos servidores para atender maiores níveis de requisições e buscar contratos com empresas para aumentar a rentabilidade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

4.7 Modelagem e definições técnicas

Esta seção tem por objetivo demonstrar as modelagens e padronizações utilizadas no desenvolvimento da aplicação.

4.7.1 Modelo Entidade Relacionamento

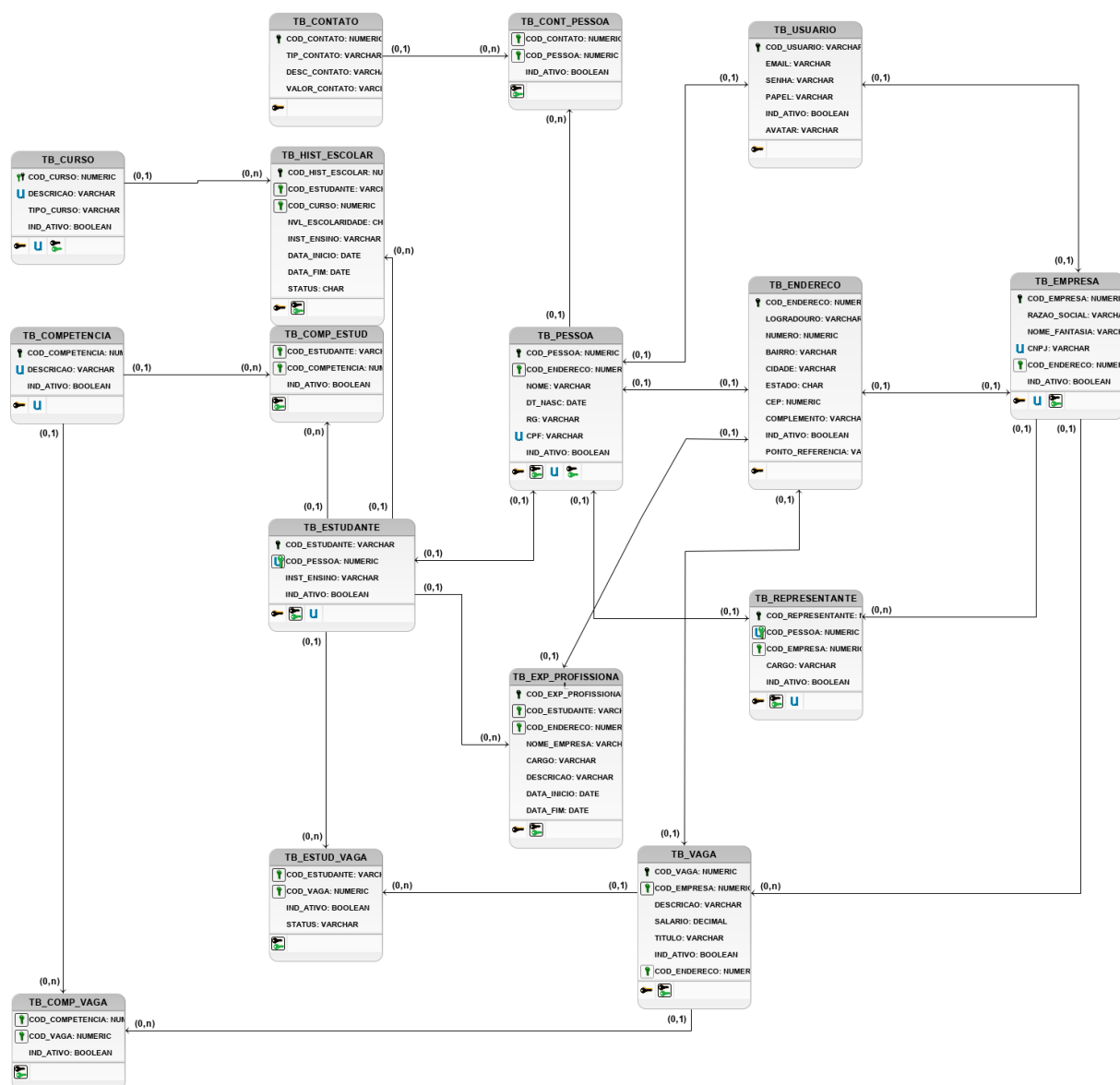
Figura 13 – Modelagem Entidade Relacionamento



Fonte: Os Autores

4.7.2 Diagrama Entidade-Relacionamento

Figura 14 – Diagrama Entidade Relacionamento



Fonte: Os Autores

4.7.3 Dicionário de Dados

A seguir mostramos as tabelas do dicionário de dados.

Quadro 14 – Legenda

Sigla	Descrição
PK	<i>Primary Key</i>
FK	<i>Foreign Key</i>
NN	<i>Not Null</i>
UQ	<i>Unique</i>
CK	<i>Check</i>
DEFAULT	<i>Default</i>

Fonte: Os Autores

Quadro 15 – Campos de Usuário

Campo	Tipo	Restrição	Descrição
Cod_usuario	VARCHAR	PK	Identificador único para usuário
Senha	VARCHAR(50)		Senha de acesso
Papel	VARCHAR(25)	DEFAULT	Papel de acesso do usuário no sistema
E-mail	VARCHAR(50)		E-mail de acesso
Avatar	VARCHAR(100)		Armazenamento de imagem do perfil Google
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado

Fonte: Os Autores

Quadro 16 – Campos de Pessoa

Campo	Tipo	Restrição	Descrição
Cod_Pessoa	SERIAL	PK	Identificador único para pessoa
Cod_Endereco	SERIAL	FK, NN	Chave estrangeira vinda de tb_endereco
Nome	VARCHAR(50)	NN	Nome
Dt_Nasc	DATE	NN	Data de nascimento
RG	VARCHAR(11)	NN	Registro Geral
CPF	VARCHAR(13)	NN, UQ	Cadastro de Pessoa Física
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado

Fonte: Os Autores

Quadro 17 – Campos de Vaga

Campo	Tipo	Restrição	Descrição
Cod_Vaga	SERIAL	PK	Indicador único
Cod_Empresa	SERIAL	FK NN	Chave estrangeira de tb_empresa
Descricao	TEXT		Descrição da vaga
Salario	FLOAT(5)		Remuneração da vaga
Titulo	VARCHAR(30)		Titulo da vaga
Cod_endereco	SERIAL	FK NN	Chave estrangeira de tb_endereco
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado da vaga

Fonte: Os Autores

Quadro 18 – Campos de Curso

Campo	Tipo	Restrição	Descrição
Cod_Curso	SERIAL	PK	Identificador único
Descricao	VARCHAR(50)	NN, UQ	Descrição do curso
Tipo_Curso	CHAR(1)	NN	Tipo do Curso)
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado de tb_curso

Fonte: Os Autores

Quadro 19 – Campos de Estudante

Campo	Tipo	Restrição	Descrição
Cod_Estudante	VARCHAR	PK	Identificador único
Cod_Pessoa	SERIAL	FK, NN, UQ	Chave estrangeira de tb_pessoa
Cod_Hist_Escolar	SERIAL	FK, NN	Chave estrangeira de tb_hist_escolar
Cod_Exp_Profissional	SERIAL	FK	Chave estrangeira de tb_exp_profissional
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado do estudante

Fonte: Os Autores

Quadro 20 – Campos de Empresa

Campo	Tipo	Restrição	Descrição
Cod_Empresa	SERIAL	PK	Identificador único para a empresa
Razao_Social	VARCHAR(50)	NN	Razão social da empresa
Nome_Fantasia	VARCHAR(50)		Nome fantasia da empresa
CNPJ	VARCHAR(20)	NN, UQ	Cadastro Nacional de Pessoas Jurídicas
Cod_Endereco	SERIAL	FK, NN	Chave estrangeira vinda de tb_endereco
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado da empresa

Fonte: Os Autores

Quadro 21 – Campos de Representante RH

Campo	Tipo	Restrição	Descrição
Cod_Representante	SERIAL	PK	Identificador único
Cod_Pessoa	SERIAL	FK, NN	Chave estrangeira de tb_pessoa
Cod_Empresa	SERIAL	FK	Chave estrangeira de tb_empresa
Cargo	VARCHAR(50)	NN	Descrição do cargo do representante
Ind_Ativo	BOOLEAN	DEFAULT	Indicador de estado do representante

Fonte: Os Autores

4.7.4 Endpoints da API

A seguir são apresentados os *endpoints* mapeados até o momento, com seus respectivos métodos de requisição [HTTP](#).

Quadro 22 – *Endpoints* de Estudante

Método	Endpoint
GET	/api/estudante/{codEstudante}
PUT	/api/estudante/{codEstudante}
POST	/api/loginEstudante
GET	/api/estudante
GET	/api/estudante/{codEstudante}/recomendacao

Fonte: Os Autores

Quadro 23 – *Endpoints* de Vaga

Método	<i>Endpoint</i>
GET	/api/vaga
POST	/api/vaga
GET	/api/estudante/{codEstudante}/recomendacao

Fonte: Os Autores

Quadro 24 – *Endpoints* de Empresa

Método	<i>Endpoint</i>
GET	/api/empresa
POST	/api/empresa
GET	/api/empresa/{codEmpresa}

Fonte: Os Autores

Quadro 25 – *Endpoints* de Competência

Método	<i>Endpoint</i>
GET	/api/competencia

Fonte: Os Autores

4.7.5 Listagem das Competências

A seguir são apresentadas as competências parametrizadas a fim de realizar as recomendações de vagas para os estudantes de acordo com seu perfil.

- Adaptação
- Atitude positiva
- Autoconfiança
- Autogestão
- Boa escrita
- Capacidade de resolver problemas
- Capacidade de tomar decisões
- *Coaching*
- Colaboração

- Comunicação
- Conhecimento político e cultural
- Criatividade
- Desenvolvimento da equipe
- Desenvolvimento pessoal
- Empatia
- Estabelecimento de confiança
- Ética no trabalho
- Flexibilidade
- Gerenciamento de conflitos
- Honestidade
- Influência
- Inteligência emocional
- Interesse em aprender
- Liderança
- Motivação
- Organização
- Pensamento crítico
- Poder de negociação
- Proatividade
- Relacionamento interpessoal
- Resiliência
- Trabalho em Equipe
- Trabalho sob pressão

5 Estatísticas e Validações

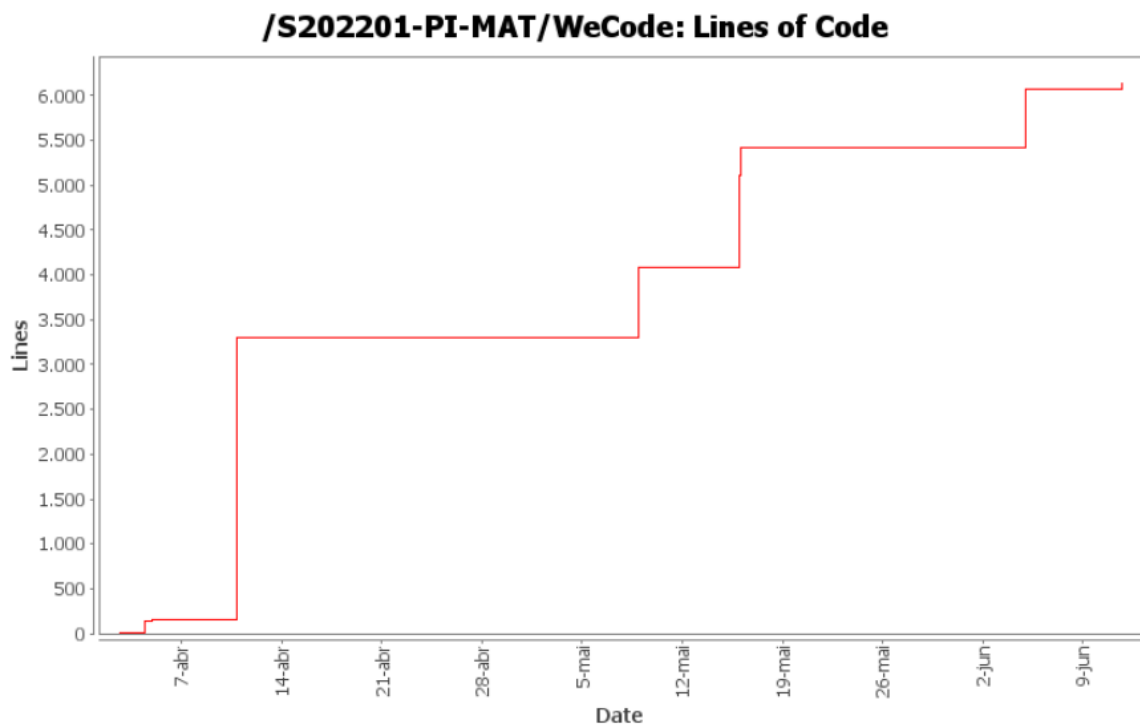
Neste capítulo são apresentadas as estatísticas de cada versionador de código, com detalhes da atuação de cada integrante da equipe e dos *commits* feitos durante o desenvolvimento, além de algumas validações de segurança, interface e código que fizemos na nossa aplicação e a justificativa delas.

5.1 SVN

Estatísticas sobre o repositório no [SVN](#) foram geradas através do [StatSVN](#) apesar de ter sido usado apenas para atualização recorrente do repositório, então não foram considerados dados estatísticos sobre atividade de cada membro da equipe.

A [Figura 15](#) mostra a evolução de linhas de código conforme o tempo.

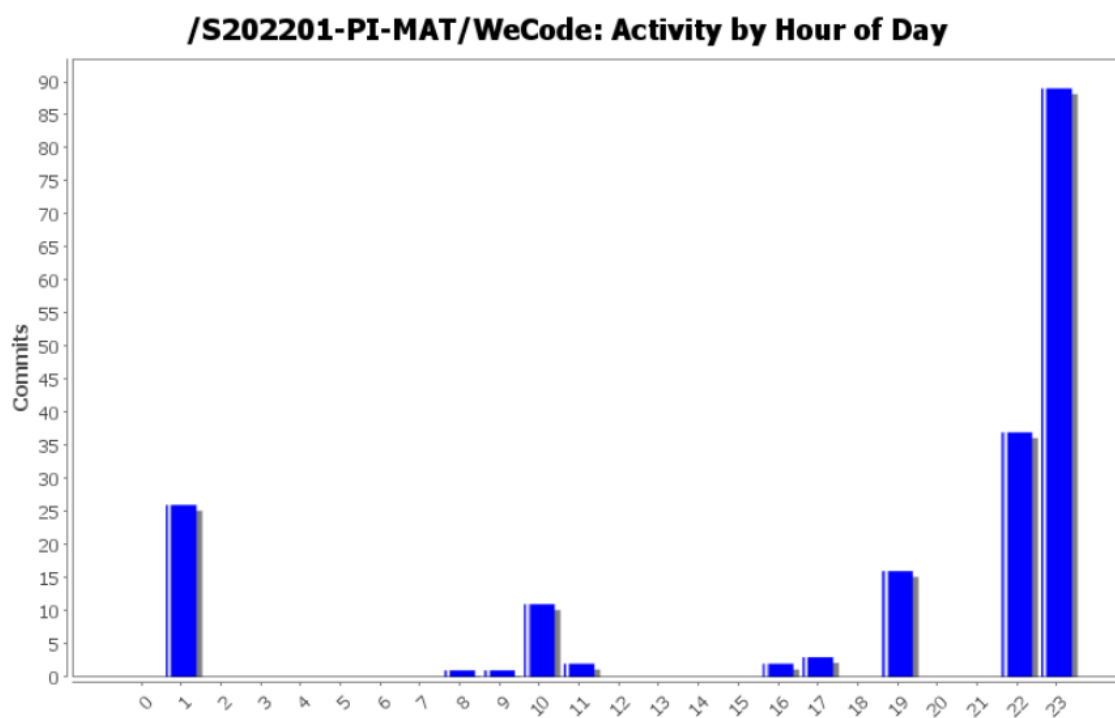
Figura 15 – Linhas de código - SVN



Fonte: Os Autores.

A [Figura 16](#) mostra os horários onde mais foram feitos *commits* em todos os dias.

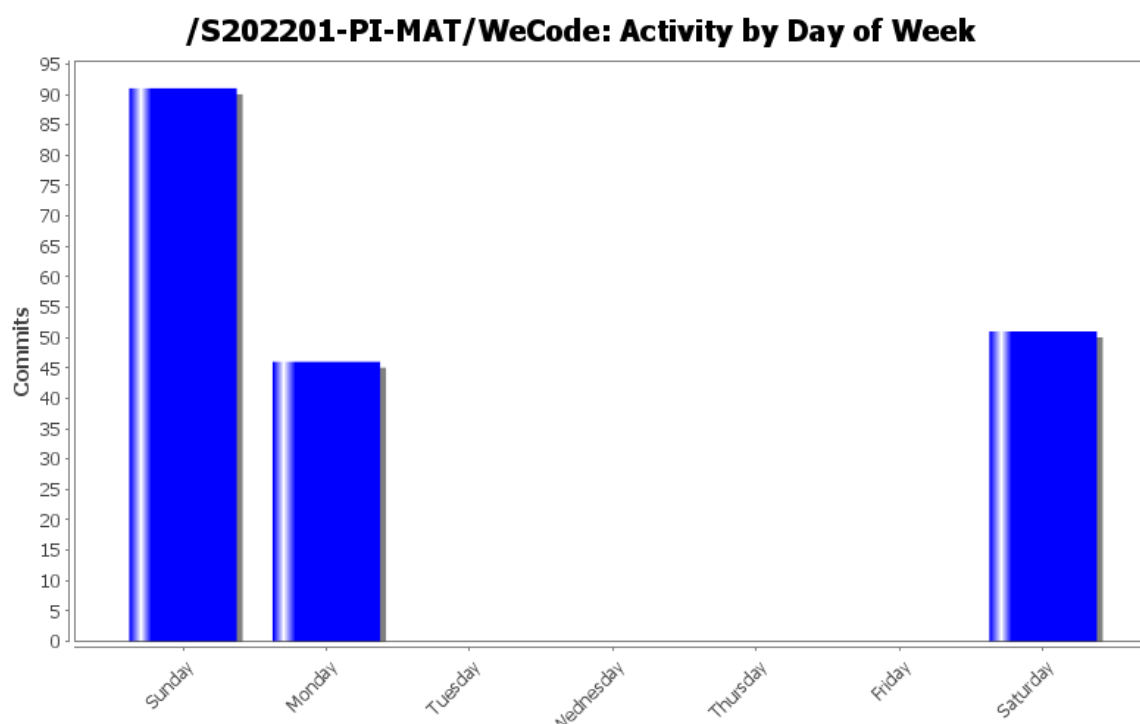
Figura 16 – Atividade por hora do dia - SVN



Fonte: Os Autores.

A Figura 17 mostra a quantidade de *commits* por dia da semana, é notório a alta quantidade de atividade aos domingos.

Figura 17 – Atividade por dia da semana - SVN



Fonte: Os Autores.

5.2 GitHub

A partir dos *commits* feitos nos repositórios, foram levantadas estatísticas sobre o projeto através do [GitStats](#), que servem de base para se ter uma ideia de como foi o andamento do projeto e o que cada integrante da equipe fez durante o período. O [Git](#) foi usado como nosso versionador principal, então os dados abaixo estão de acordo com a atividade de cada integrante.

A [Figura 18](#) demonstra uma visão geral sobre alguns dados do repositório onde hospedamos o nosso *front-end*.

Figura 18 – Visão geral - Projeto *front-end*

```
Project name:
  estagiei-frontend
Generated:
  2022-07-02 11:53:15 (in 0 seconds)
Generator:
  GitStats (version 55c5c28), git version 2.34.1, gnuplot 5.4 patchlevel 2
Report Period:
  2022-04-04 21:37:37 to 2022-06-19 20:23:59
Age:
  76 days, 13 active days (17.11%)
Total Files:
  87
Total Lines of Code:
  5287 (6662 added, 1375 removed)
Total Commits:
  80 (average 6.2 commits per active day, 1.1 per all days)
Authors:
  2 (average 40.0 commits per author)
```

Fonte: Os Autores

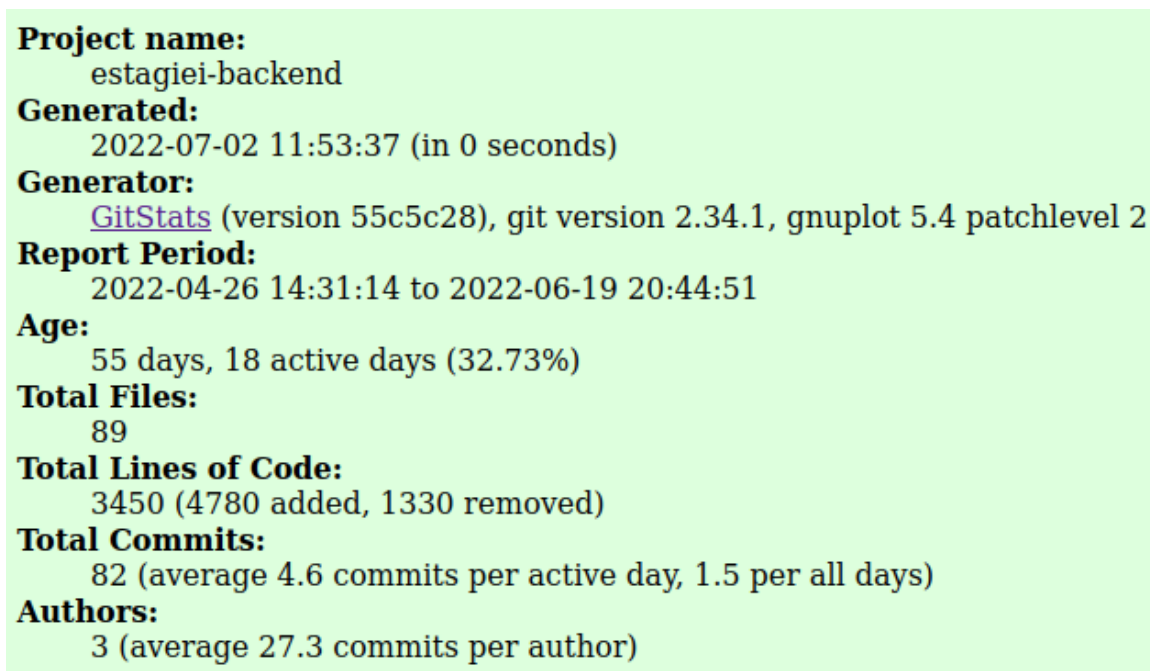
A [Figura 20](#) demonstra uma visão geral sobre alguns dados do repositório onde foi hospedado o *back-end*. É possível notar algumas diferenças entre ele e o *front-end*.

Figura 19 – URL do repositório *front-end*



<<https://github.com/EquipeWeCode/estagiei-frontend/>>

Fonte: Os Autores.

Figura 20 – Visão geral - Projeto *back-end*

Fonte: Os Autores.

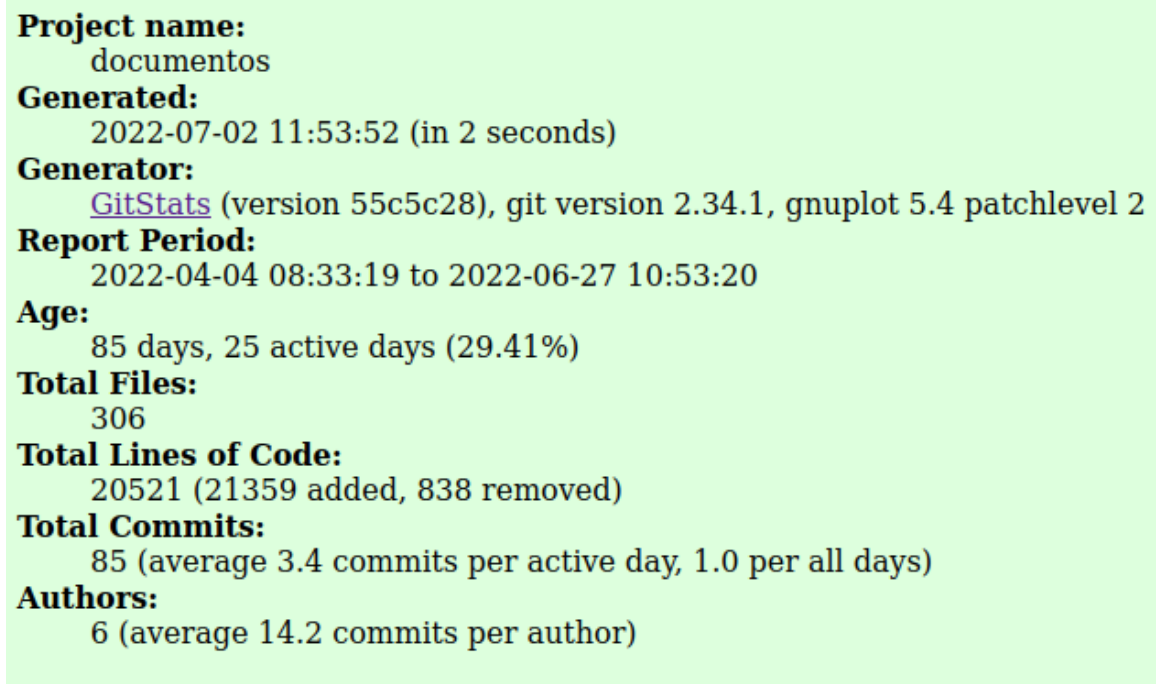
Figura 21 – URL do repositório *back-end*

<<https://github.com/EquipeWeCode/estagiei-backend>>

Fonte: Os Autores.

A [Figura 22](#) demonstra uma visão geral sobre alguns dados principais do repositório onde foram versionados os documentos \LaTeX do projeto.

Figura 22 – Visão geral - Projeto Documentos



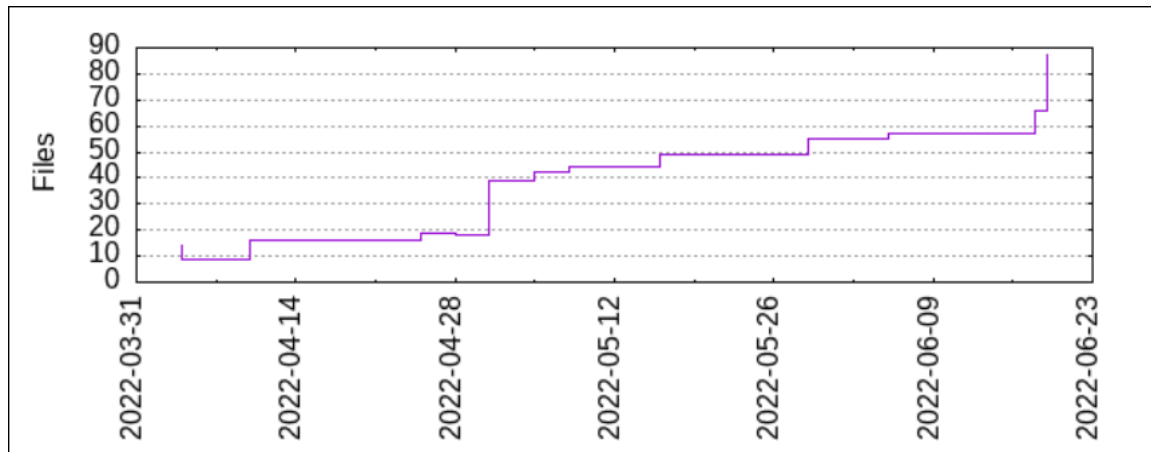
Fonte: Os Autores.

Figura 23 – URL do repositório de documentos \LaTeX 

<<https://github.com/EquipeWeCode/documentos>>

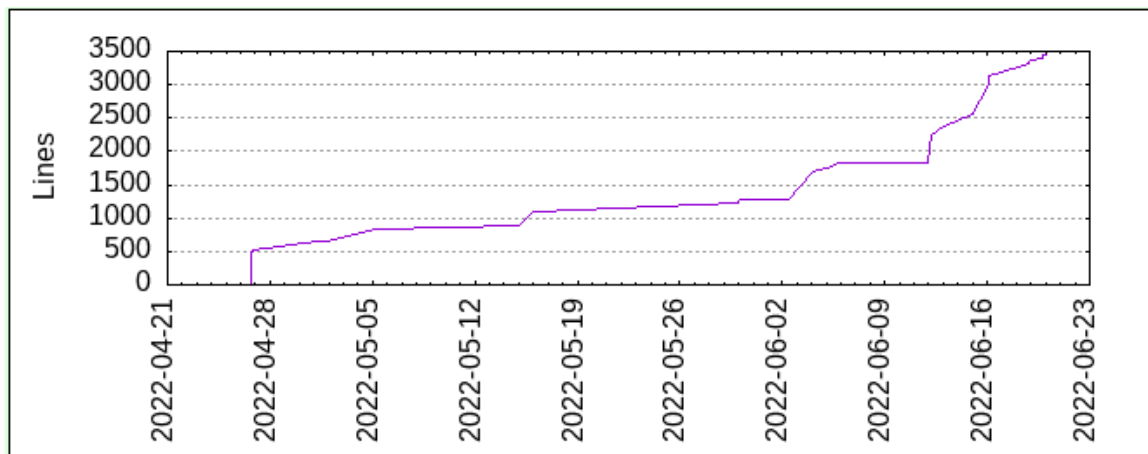
Fonte: Os Autores.

A [Figura 24](#) demonstra a evolução em número de linhas do repositório onde foi hospedado o *front-end*.

Figura 24 – Linhas de código - Projeto *front-end*

Fonte: Os Autores.

A Figura 25 demonstra a evolução em número de linhas do repositório onde foi hospedado o *back-end*.

Figura 25 – Linhas de código - Projeto *back-end*

Fonte: Os Autores.

Cabe ressaltar que o número de linhas não necessariamente diz que foi desenvolvido muito código, como demonstra a Figura 26, grande parte das linhas do nosso *front-end* são de arquivos de extensão *JSON*, principalmente do *package-lock.json*, que é gerado automaticamente, documentando e versionando as dependências do projeto.

Figura 26 – Extensão de arquivos - Projeto *front-end*

Extension	Files (%)	Lines (%)	Lines/file
	3 (3.45%)	58 (1.10%)	19
css	12 (13.79%)	267 (5.05%)	22
example	1 (1.15%)	1 (0.02%)	1
html	1 (1.15%)	20 (0.38%)	20
json	6 (6.90%)	3421 (64.71%)	570
md	1 (1.15%)	14 (0.26%)	14
png	1 (1.15%)	41 (0.78%)	41
svg	5 (5.75%)	137 (2.59%)	27
toml	1 (1.15%)	9 (0.17%)	9
ts	23 (26.44%)	415 (7.85%)	18
tsx	32 (36.78%)	881 (16.66%)	27
yml	1 (1.15%)	21 (0.40%)	21

Fonte: Os Autores.

A Figura 27 lista os autores que contribuíram no repositório *front-end*.Figura 27 – Dias da semana - Projeto *front-end*

Author	Commits (%)	+ lines	- lines	First commit	Last commit	Age	Active days	# by commits
Leonardo Marques	75 (93.75%)	6557	1375	2022-04-04	2022-06-19	75 days, 22:46:22	13	1
Lucas Lima	5 (6.25%)	211	106	2022-05-29	2022-06-19	20 days, 21:49:54	2	2

Fonte: Os Autores.

A Figura 28 lista os autores que contribuíram no repositório *back-end*.Figura 28 – Lista de autores - Projeto *back-end*

Author	Commits (%)	+ lines	- lines	First commit	Last commit	Age	Active days	# by commits
Leonardo Marques	66 (80.49%)	3300	1279	2022-05-01	2022-06-19	49 days, 19:46:30	16	1
Marcelo Junior	9 (10.98%)	541	11	2022-04-26	2022-05-05	8 days, 17:26:36	2	2
Igor Nathan	7 (8.54%)	936	56	2022-05-05	2022-06-14	40 days, 19:02:13	5	3

Fonte: Os Autores.

A Figura 29 lista os autores que contribuíram no repositório de documentos, no qual todos os membros da equipe participaram.

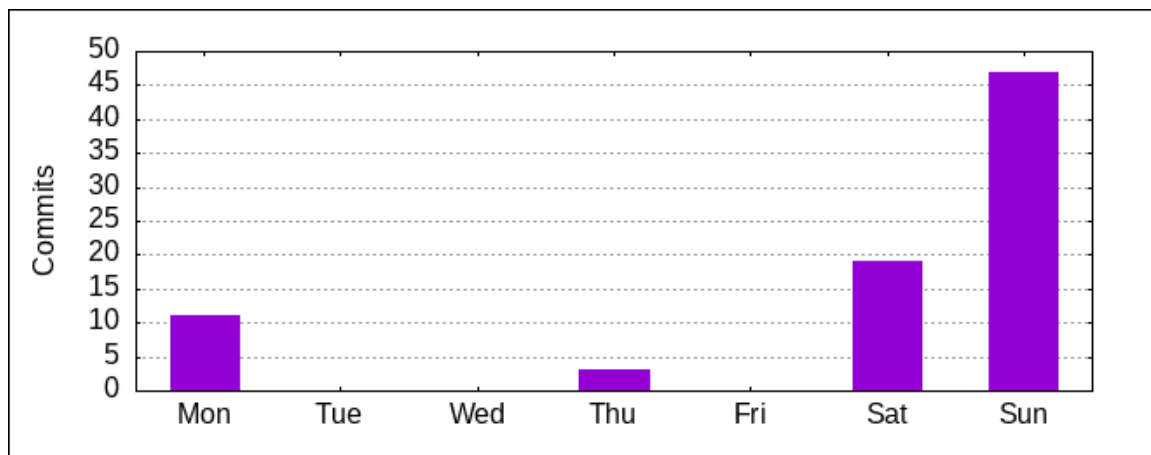
Figura 29 – Lista de autores - Projeto Documentos

Author	Commits (%)	+ lines	- lines	First commit	Last commit	Age	Active days	# by commits
Bruna S. Pires	40 (47.06%)	17697	427	2022-04-04	2022-06-13	70 days, 2:35:18	16	1
Leonardo Marques	29 (34.12%)	733	286	2022-04-09	2022-06-13	64 days, 16:46:57	9	2
Igor Nathan	6 (7.06%)	128	89	2022-04-13	2022-06-27	74 days, 13:14:58	3	3
Marcelo Junior	5 (5.88%)	2525	34	2022-04-10	2022-04-25	15 days, 4:43:36	4	4
Daniel Roberto	3 (3.53%)	17	4	2022-06-12	2022-06-27	14 days, 11:25:22	3	5
Lucas Lima	2 (2.35%)	253	10	2022-04-22	2022-04-27	5 days, 2:55:01	2	6

Fonte: Os Autores.

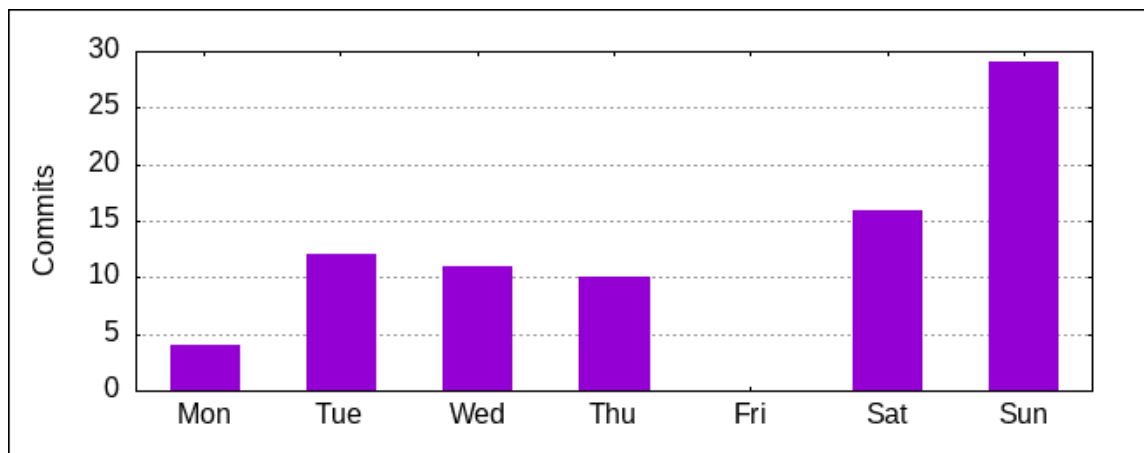
A Figura 30 mostra os dias da semana em que foram feitos mais *commits* no repositório *front-end*, é notável que aos fins de semana, principalmente de domingo, a quantidade de *commits* é maior do que no resto da semana.

Figura 30 – Dias da semana - Projeto *front-end*



Fonte: Os Autores.

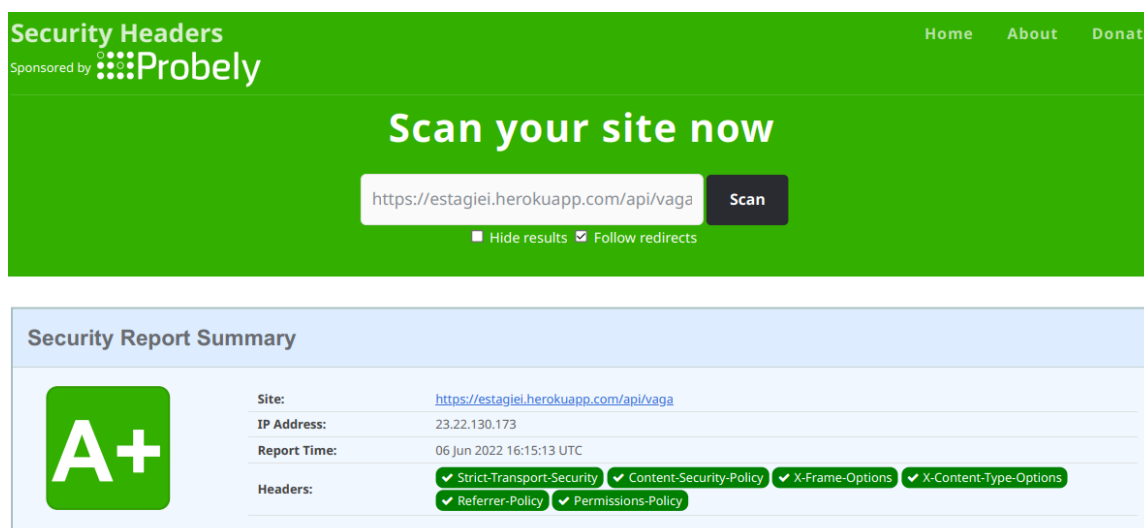
A Figura 31 mostra os dias da semana em que foram feitos mais *commits* no repositório *back-end*, pode-se notar uma alta taxa de *commits* durante o domingo, enquanto o resto da semana permanece de forma quase igual, com exceção da sexta-feira e segunda-feira que possuem um percentual de atividade menor que os demais dias da semana.

Figura 31 – Dias da semana - Projeto *back-end*

Fonte: Os Autores.

5.3 Teste dos *headers* da API

Para os testes de *headers* de segurança, foi usado o site *Security Headers*, onde é possível verificar em qual nota se enquadrava a API do projeto. Foram feitos testes de um dos nossos *endpoints* (O resultado é o mesmo para todos), e durante o primeiro teste, o site indicou que nota F, pela resposta da aplicação não conter nenhum *header* de segurança, então com base nesse resultado, foram adicionadas as dependências que faltavam e assim, foi possível subir a nota, como demonstra a Figura 32.

Figura 32 – Validação dos *headers*

Fonte: Os Autores.

Figura 33 – URL da documentação dos nossos *endpoints* (*Swagger UI*)


<<https://estagiei.herokuapp.com/api/swagger-ui/>>

Fonte: Os Autores.

5.4 Teste de TLS do *front-end*

O sistema *EstagiEI* também foi testado com relação ao certificado *TLS*, que indica se um site utiliza o protocolo *HTTPS* ou não. Devido a aplicação estar hospedada no *Netlify*, ele automaticamente já providencia um certificado com o *Let's Encrypt* quando é criado um domínio personalizado. Assim como mostra a *Figura 34*, a aplicação possui um certificado *TLS* ativo.

Figura 34 – Teste de TLS

Qualys. SSL Labs

[Home](#)[Projects](#)[Qualys Free Trial](#)[Contact](#)

You are here: [Home](#) > [Projects](#) > [SSL Server Test](#) > estagiei.herokuapp.com

SSL Report: estagiei.herokuapp.com

Assessed on: Mon, 06 Jun 2022 16:28:35 UTC | [Hide](#) | [Clear cache](#)

[Scan Another >>](#)

	Server	Test time	Grade
1	54.243.238.66 ec2-54-243-238-66.compute-1.amazonaws.com Ready	Mon, 06 Jun 2022 16:22:36 UTC Duration: 89.306 sec	A+
2	107.22.57.98 ec2-107-22-57-98.compute-1.amazonaws.com Ready	Mon, 06 Jun 2022 16:24:05 UTC Duration: 89.85 sec	A+
3	3.209.172.72 ec2-3-209-172-72.compute-1.amazonaws.com Ready	Mon, 06 Jun 2022 16:25:34 UTC Duration: 91.364 sec	A+
4	23.22.130.173 ec2-23-22-130-173.compute-1.amazonaws.com Ready	Mon, 06 Jun 2022 16:27:06 UTC Duration: 88.833 sec	A+

SSL Report v2.1.10

Copyright © 2009-2022 [Qualys, Inc.](#) All Rights Reserved.

[Try Qualys for free!](#) Experience the award-winning [Qualys Cloud Platform](#) and the entire collection of [Qualys Cloud Apps](#), including [certificate security](#) solutions.

[Terms and Conditions](#)

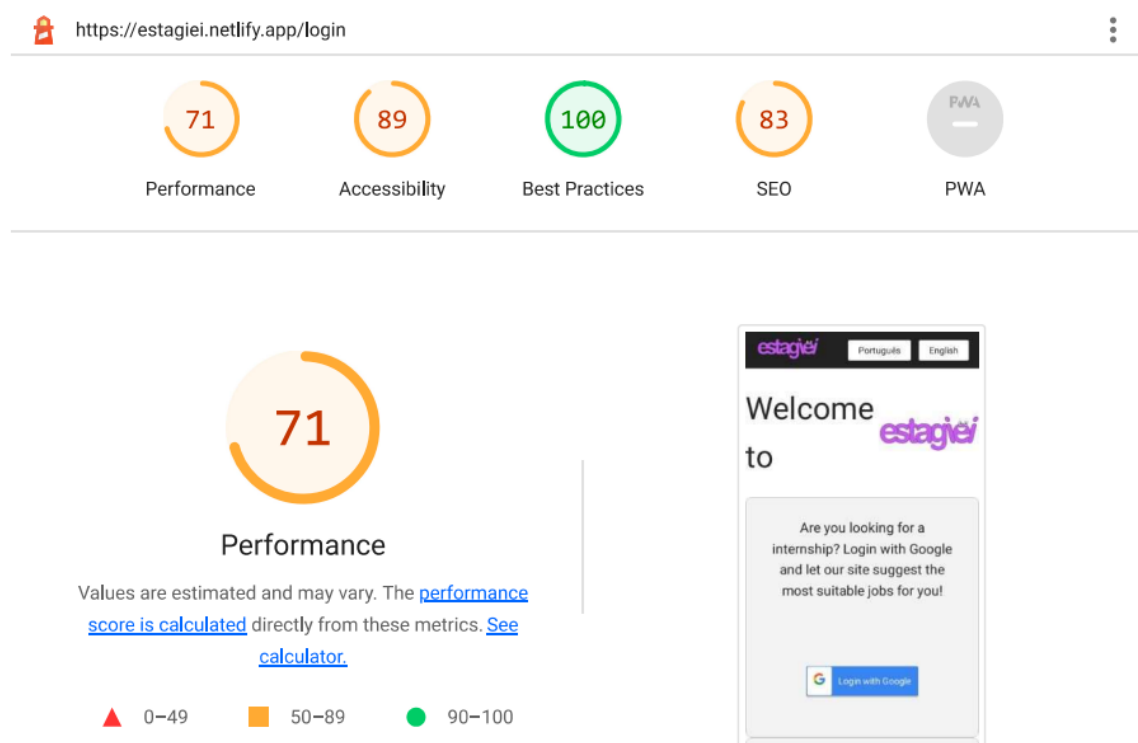
Fonte: Os Autores

Figura 35 – URL do *front-end* da nossa aplicação<<https://estagiei.netlify.app/>>

Fonte: Os Autores.

5.5 Teste de desempenho do *front-end*

Através da extensão *Lighthouse*, foi verificado como estava o desempenho, acessibilidade, etc. Da tela de login, e foi percebido pontos a serem melhorados, principalmente na questão do desempenho. Pontos esses que serão ajustados no próximo semestre. A Figura 36 ilustra o resultado.

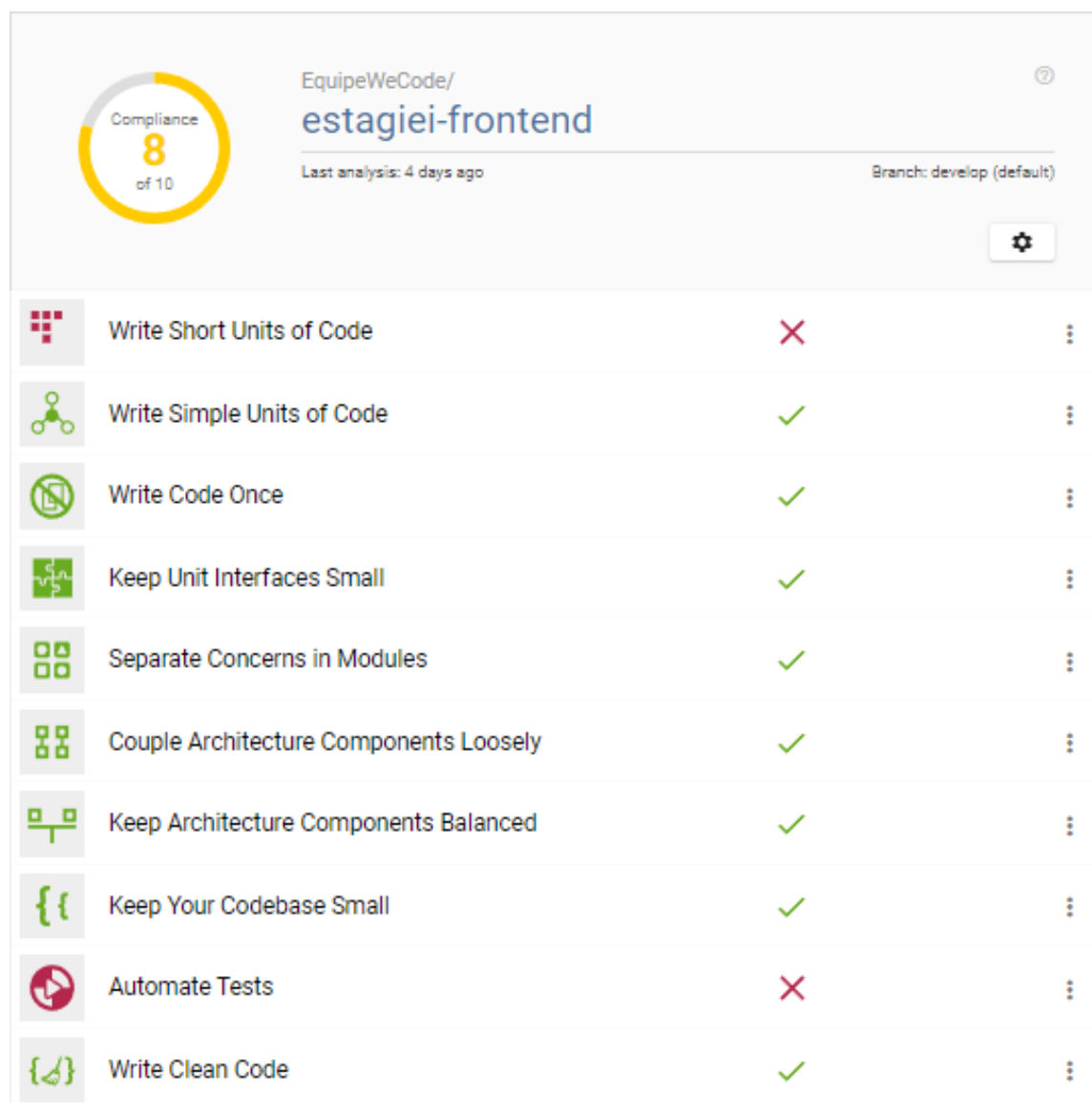
Figura 36 – Teste de desempenho do *front-end*

Fonte: Os Autores.

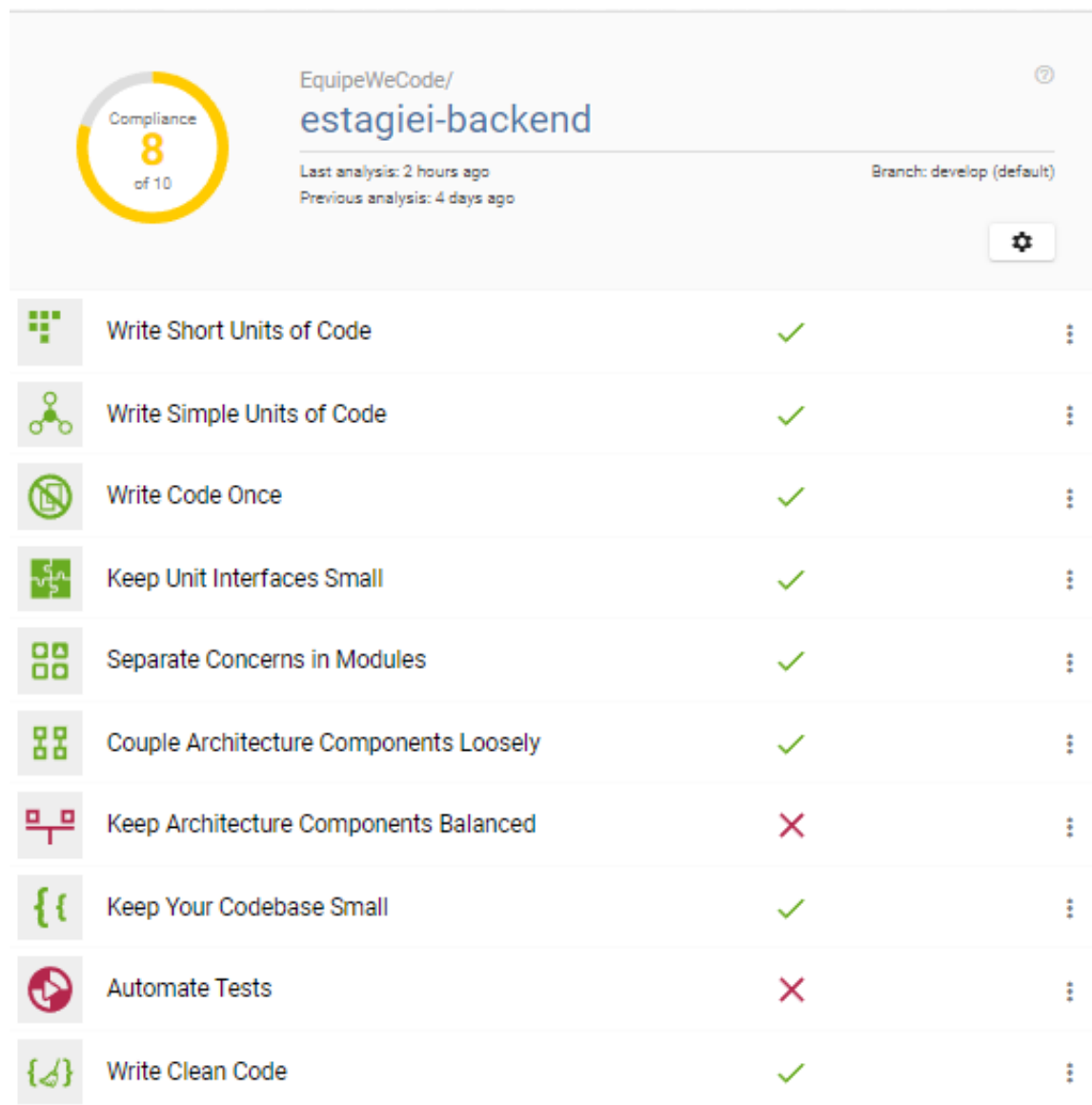
5.6 Análise de código

Fizemos também uma verificação no nosso código, tanto no *front-end* como no *back-end*, e o resultado foi relativamente satisfatório, faltando apenas alguns pontos que serão melhorados ao longo do tempo. As figuras [Figura 37](#) e [Figura 38](#) demonstram os resultados do *front-end* e *back-end* respectivamente.

Figura 37 – Análise de código do *front-end*



Fonte: Os Autores.

Figura 38 – Análise de código do *back-end*

Fonte: Os Autores.

5.7 Validador HTML

Foi feito também uma verificação no [HTML](#) do *front-end*. A [Figura 39](#) demonstra o resultado.

Figura 39 – Validação do HTML

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://estagiei.netlify.app/>

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by address

<https://estagiei.netlify.app/>

[Check](#)

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 89 milliseconds.

Fonte: Os Autores.

Glossário

AWS	Amazon Web Services - Plataforma em nuvem <i>on-demand</i> que disponibiliza diversos serviços web. - Citado em 38 , 62
back-end	Camada do sistema da aplicação que não é acessado diretamente pelo usuário, responsável pelo processamento de dados e a implementação de funcionalidades que satisfazem uma ou mais regras de negócios da aplicação. - Citado em 2 , 3 , 14 , 19 , 21 , 32 , 33 , 35 , 36 , 37 , 49 , 50 , 52 , 53 , 54 , 55 , 58 , 59
deploy	Refere-se ao processo de configuração de um computador ou sistema até o ponto em que esteja pronto para o processamento em ambiente de produção. - Citado em 21 , 35 , 36
endpoint	Localização digital onde uma API recebe requisições sobre um recurso específico em seu servidor. Os endpoints comumente são uma <i>Universal Resource Locator</i> (URL), indicando uma ponta da conexão para a recuperação do recurso através da API . - Citado em 2 , 4 , 33 , 37 , 44 , 45 , 55 , 56
framework	Estrutura base para desenvolvimento de um sistema e/ou projeto com um conjunto de elementos e conexões pré-estabelecidas e/ou indicadas. - Citado em 21 , 62
front-end	Camada do sistema da aplicação que é responsável pela integração do usuário com o sistema, oferecendo uma interface que se comunica com o usuário e com o sistema. - Citado em 2 , 3 , 7 , 14 , 19 , 21 , 32 , 33 , 35 , 36 , 37 , 49 , 51 , 52 , 53 , 54 , 56 , 57 , 58 , 59
Git	Sistema de controle de versão de arquivos. - Citado em 49 , 61
GitHub	Provedor de hospedagem na internet para desenvolvimento de software e controle de versionamento usando Git . - Citado em 35 , 36
GitStats	Ferramenta que gera estatísticas de repositórios Git - Citado em 49
Heroku	Plataforma em nuvem como um serviço que suporta diversas linguagens de programação. - Citado em 21 , 35 , 38 , 61
Heroku CI	Instância da Heroku responsável pela integração contínua. - Citado em 35
Jira Software	Ferramenta de gerenciamento que permite o monitoramento de tarefas e acompanhamento de projetos. - Citado em 14 , 15 , 18
Let's Encrypt	Gerador de certificado TLS gratuito para <i>websites</i> - Citado em 56
Lighthouse	Ferramenta que verifica o desempenho de um site e aponta melhorias - Citado em 57

Linux	Kernel open-source usado em diversos sistemas operacionais. - Citado em 38
Logback	Logback é uma estrutura de log para aplicações java, criada como sucessora do popular projeto log4j. - Citado em 32
Netlify	Plataforma em nuvem que faz o <i>host</i> de páginas web. - Citado em 21 , 35 , 36 , 38 , 56
PostgreSQL	Sistema de Gerenciamento de Banco de Dados Relacional, gratuito e open-source. - Citado em 21
RDS	Amazon Relational Databases - Serviço de banco de dados da AWS . - Citado em 38
React	Biblioteca JavaScript gratuita e open-source para a construção de interfaces baseadas em componentes. - Citado em 21
Scrum	Metodologia ágil de software concebida por Jeff Sutherland e sua equipe de desenvolvimento no início dos anos 90. - Citado em 14 , 62
Scrum Master	Papel de gerência e coordenação na metodologia Scrum . O Scrum Master é o intermediário entre a equipe de desenvolvimento e os clientes. - Citado em 14
Security Headers	Site que verifica a resposta de um <i>endpoint</i> e atribui uma nota de acordo com a quantidade de <i>headers</i> de segurança que esse <i>endpoint</i> retorna - Citado em 55
Spring Boot	Spring Boot é um framework baseado em Java de código aberto usado para criação de micro serviços e aplicações web no geral. - Citado em 21 , 32
Sprint	Uma iteração de desenvolvimento, unidade de planejamento do Scrum na qual se verifica o trabalho (funcionalidade) a ser entregue, os recursos necessários e ocorre o desenvolvimento do software de fato. - Citado em 4 , 14 , 15 , 16 , 17 , 19 , 20
StatSVN	Ferramenta que gera estatísticas de repositórios SVN - Citado em 47
SUN MICROSYSTEMS	I. Java coding conventions, 1996. - Citado em 32
SVN	Subversion - Sistema de controle de versão de arquivos. - Citado em 47
TypeScript	Linguagem de programação fortemente tipada sobre JavaScript. - Citado em 21
VLibras	Conjunto de ferramentas para a tradução de texto em Português para LIBRAS gratuitas e de código aberto, mais informações disponíveis no endereço https://www.gov.br/governodigital/pt-br/vlibras . - Citado em 30 , 31

Apêndices

APÊNDICE A – Publicações do Blog

Figura 40 – URL do blog da equipe \LaTeX



`<https://wecodeifsp.blogspot.com/>`

Fonte: Os Autores.

APÊNDICE B – POC Overview

Proof Of Concept (POC) Overview **(Visão Geral da Prova de Conceito)**

Neste documento buscamos relatar de modo mais direto os itens compondo a POC apresentada em aula no dia 16/05/2022.

A equipe se propôs a demonstrar a hospedagem da aplicação seguindo a arquitetura planejada e integrando com as tecnologias e ferramentas escolhidas, possibilitar o Login via uma conta Google e recuperar uma listagem de vagas do banco de dados.

PLANEJAMENTO:

Ferramenta de tarefas: Jira Software;

Metodologia: SCRUM;

Controle de versão: Git com GitHub, sendo os códigos e documentos alocados em uma organização no GitHub;

Comunicação: Commits seguindo um padrão de prefixo que identifica o que foi feito, que tipo de modificações foram realizadas .

BANCO DE DADOS

Tipo: Relacional;

Codificação: SQL;

Hospedagem: Heroku Postgres.

BACKEND

Codificação: Java com framework Spring;

Hospedagem: Heroku.

FRONTEND

Codificação: TypeScript com o framework React;

Hospedagem: Inicialmente na Vercel, posteriormente migrado para a Netlify a fim de podermos prosseguir com o desenvolvimento em uma conta gratuita com uma organização no GitHub.

INTEGRAÇÃO:

Conexão dos repositórios do GitHub com as plataformas de hospedagem, permitindo assim integração contínua, pois a cada atualização no repositório, é feito um deploy nas correspondentes plataformas.

Foram realizadas as configurações de ambientes específicas de cada camada, posteriormente, quando já possuíamos clareza o suficiente sobre o projeto, o MER foi desenhado e as tabelas elaboradas.

Para a POC apenas as tabelas essenciais foram adicionadas, assim como apenas os endpoints mínimos foram construídos e mapeados.

Anexos

ANEXO A – Nota dos headers

Scan your site now

Scan

☐ Hide results ☒ Follow redirects

Security Report Summary



Site: <https://estagiei.herokuapp.com/api/vaga>

IP Address: 23.22.130.173

Report Time: 06 Jun 2022 16:15:13 UTC

Headers: ✔ Strict-Transport-Security ✔ Content-Security-Policy ✔ X-Frame-Options ✔ X-Content-Type-Options ✔ Referrer-Policy ✔ Permissions-Policy

Supported By

Probely

Wow, amazing grade! Perform a deeper security analysis of your website and APIs:

[Try Now](#)

Raw Headers

HTTP/1.1	200
Server	Cowboy
Connection	keep-alive
Accept	application/json
Strict-Transport-Security	max-age=63072000; includeSubDomains; preload
Content-Security-Policy	default-src 'self' https://estagiei.herokuapp.com
X-Frame-Options	DENY
X-Content-Type-Options	nosniff
Referrer-Policy	same-origin
Permissions-Policy	microphone=none; geolocation=none; camera=none
Vary	Origin
Vary	Access-Control-Request-Method
Vary	Access-Control-Request-Headers
Content-Type	application/json
Transfer-Encoding	chunked
Date	Mon, 06 Jun 2022 16:15:13 GMT
Via	1.1 vegur

Upcoming Headers

Expect-CT	Expect-CT allows a site to determine if they are ready for the upcoming Chrome requirements and/or enforce their CT policy.
Cross-Origin-Embedder-Policy	Cross-Origin Embedder Policy allows a site to prevent assets being loaded that do not grant permission to load them via CORS or CORP.
Cross-Origin-Opener-Policy	Cross-Origin Opener Policy allows a site to opt-in to Cross-Origin Isolation in the browser.
Cross-Origin-Resource-Policy	Cross-Origin Resource Policy allows a resource owner to specify who can load the resource.

Additional Information

Server	Server value has been changed. Typically you will see values like "Microsoft-IIS/8.0" or "nginx 1.7.2".
Strict-Transport-Security	HTTP Strict Transport Security is an excellent feature to support on your site and strengthens your implementation of TLS by getting the User Agent to enforce the use of HTTPS.
Content-Security-Policy	Content Security Policy is an effective measure to protect your site from XSS attacks. By whitelisting sources of approved content, you can prevent the browser from loading malicious assets. Analyse this policy in more detail. You can sign up for a free account on Report URI to collect reports about problems on your site.
X-Frame-Options	X-Frame-Options tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site you can defend against attacks like clickjacking.
X-Content-Type-Options	X-Content-Type-Options stops a browser from trying to MIME-sniff the content type and forces it to stick with the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".
Referrer-Policy	Referrer Policy is a new header that allows a site to control how much information the browser includes with navigations away from a document and should be set by all sites.
Permissions-Policy	Permissions Policy is a new header that allows a site to control which features and APIs can be used in the browser.