

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

**UnB Estágio: desenvolvimento de um aplicativo
mobile para divulgação de vagas de estágios**

Autor: Vitor Leal dos Santos

Orientador: Prof. Dr. John Lenon Cardoso Gardenghi

Brasília, DF

2022



Vitor Leal dos Santos

**UnB Estágio: desenvolvimento de um aplicativo *mobile*
para divulgação de vagas de estágios**

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. John Lenon Cardoso Gardenghi

Brasília, DF

2022

Vitor Leal dos Santos

UnB Estágio: desenvolvimento de um aplicativo *mobile* para divulgação de vagas de estágios/ Vitor Leal dos Santos. – Brasília, DF, 2022-
55 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. John Lenon Cardoso Gardenghi

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. Estágio. 2. React Native. I. Prof. Dr. John Lenon Cardoso Gardenghi.
II. Universidade de Brasília. III. Faculdade UnB Gama. IV. UnB Estágio:
desenvolvimento de um aplicativo *mobile* para divulgação de vagas de estágios

CDU 02:141:005.6

Vitor Leal dos Santos

UnB Estágio: desenvolvimento de um aplicativo *mobile* para divulgação de vagas de estágios

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 29 de Setembro de 2022:

**Prof. Dr. John Lenon Cardoso
Gardenghi**
Orientador

Prof. Dr. Tiago Alves da Fonseca
Convidado 1

**Prof.^a Dr.^a Thais Alves da Costa
Lamounier**
Convidado 2

Brasília, DF
2022

Dedico este trabalho a todos os que me apoiaram ao longo de minha jornada

Agradecimentos

Primeiramente gostaria de agradecer a meus pais, que sempre me incentivaram em todos os momentos.

A minha namorada, que me traz luz e me faz ter esperança em um mundo melhor.

Ao meu orientador, que foi peça fundamental para a entrega deste trabalho.

A todos os amigos que conheci durante essa jornada da graduação.

Resumo

Durante a graduação na Universidade de Brasília (UnB), os estudantes podem encontrar dificuldades para localizar vagas de estágio. Este trabalho tem a intenção de propor uma aplicação *mobile* (ou móvel) cujos objetivos são modernizar e difundir a divulgação das vagas de estágio. O desenvolvimento de uma tal aplicação engloba pesquisar e definir uma metodologia de desenvolvimento compatível com o software, principalmente em se tratando de aplicações móveis, em que também é necessário definir o tipo de desenvolvimento a ser seguido. A aplicação é um mínimo produto viável, já publicada na loja de aplicativos do sistema Android (*Play Store*), podendo ser usada livremente pela comunidade.

Palavras-chaves: estágio. react native. multiplataforma.

Abstract

During the graduation on University of Brasília (UnB), the students can face difficulties to find internship opportunities. This work is intended to propose an mobile application with the goal of modernize and advertise the opportunities for UnB students. The development of such application involves research and define a software development methodology compatible with the software. The application is an minimum viable product and it's already published in Android Play Store, being available to use by all the community.

Key-words: internship. react native. cross-plataform.

Lista de ilustrações

Figura 1 – Fluxo de Processos - TCC	31
Figura 2 – Arquitetura de serviços	39
Figura 3 – Diagrama de domínio	40
Figura 4 – Tela de Cadastro	42
Figura 5 – Tela Login	43
Figura 6 – Tela Home	44
Figura 7 – Tela Home - Administrador	45
Figura 8 – Tela Divulgação	46
Figura 9 – Tela Edição	47
Figura 10 – Tela Detalhes	48
Figura 11 – Tela login - <i>web</i>	49
Figura 12 – Tela home - <i>web</i>	50
Figura 13 – Tela de aprovação das vagas	50
Figura 14 – Tela lista de vagas - <i>web</i>	51

Lista de tabelas

Tabela 1 – Ferramentas Android e iOS	26
Tabela 2 – Product Backlog da Aplicação Mobile	37
Tabela 3 – Product Backlog da Aplicação Web	38

Lista de abreviaturas e siglas

IDE	Integrated Development Environment
SDK	Software Development Kit
TCC	Trabalho de conclusão de curso
TCE	Termo de Compromisso de Estágio
UnB	Universidade de Brasília

Sumário

1	INTRODUÇÃO	21
1.1	Contexto	21
1.2	Problema	22
1.3	Objetivos gerais	22
1.4	Objetivos específicos	22
2	REFERENCIAL TEÓRICO	25
2.1	Desenvolvimento Mobile	25
2.2	Android	25
2.3	iOS	26
2.4	Desenvolvimento nativo	26
2.5	Desenvolvimento multiplataforma	26
2.6	Metodologias de desenvolvimento	27
2.6.1	Kanban	27
2.6.2	Scrum	28
3	METODOLOGIA	31
3.1	Fluxo de processos	31
3.2	Ferramentas utilizadas para o Desenvolvimento de Software	32
3.2.1	JavaScript	32
3.2.2	React Native	33
3.2.3	ReactJS	33
3.2.4	Firebase	34
3.2.5	GitHub	34
3.3	Metodologias utilizadas para o Desenvolvimento de Software	35
4	PROPOSTA DO SISTEMA	37
4.1	<i>Product Backlog</i>	37
5	DESENVOLVIMENTO DAS APLICAÇÕES	39
5.1	Arquitetura do Sistema	39
5.2	Modelo de domínio	40
5.3	Publicação das aplicações	40
5.4	Aplicação <i>mobile</i>	42
5.5	Aplicação <i>web</i>	49
5.6	Teste de Usabilidade	51

6 CONSIDERAÇÕES FINAIS 53

REFERÊNCIAS 55

1 Introdução

Neste capítulo, serão descritos a contextualização, apresentando brevemente o tema, a justificativa, os porquês da elaboração do trabalho e os objetivos gerais e específicos.

1.1 Contexto

Através do Art. 1º da Lei nº 11.788, de 25 de setembro de 2008 ([BRASIL, 2008](#)), podemos afirmar que:

Estágio é ato educativo escolar supervisionado, desenvolvido no ambiente de trabalho, que visa à preparação para o trabalho produtivo de educandos que estejam frequentando 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.

Para que o estágio possa ser concedido, alguns requisitos obrigatórios devem ser observados, sendo eles ([BRASIL, 2008](#)):

1. Matrícula e frequência regular do educando público-alvo da lei;
2. Celebração do Termo de Compromisso de Estágio;
3. Compatibilidade entre as atividades desenvolvidas no estágio e as previstas no termo de compromisso;

O Termo de Compromisso de Estágio (TCE) é um acordo tripartite celebrado entre o educando, a parte concedente do estágio e a instituição de ensino, prevendo as condições de adequação do estágio à proposta pedagógica do curso, à etapa e modalidade da formação escolar do estudante e ao horário e calendário escolar. ([BRASIL, 2008](#)).

De forma geral, o estágio pode ser desenvolvido em instituições públicas ou privadas, desde que estejam com o devido registro regularizado em seus respectivos conselhos, podendo ser aproveitado de dois modos, sendo esses:

- Obrigatório
- Não obrigatório

O estágio obrigatório é definido como pré-requisito no projeto pedagógico do curso para aprovação e obtenção do diploma (BRASIL, 2008).

O estágio não obrigatório é uma atividade opcional, acrescida à carga horária regular e obrigatória do curso. (BRASIL, 2008).

1.2 Problema

Durante a graduação na UnB, alguns estudantes podem encontrar barreiras para terem conhecimento sobre as vagas de estágios existentes para suas respectivas áreas de estudo, isso acontece sobretudo, por uma falta de padronização de ferramenta para a divulgação dessas oportunidades.

1.3 Objetivos gerais

Este trabalho tem como objetivo geral a implementação de uma aplicação que auxilie no processo de padronização da ferramenta para divulgação de vagas de estágio.

1.4 Objetivos específicos

Para alcançar os objetivos gerais, foram definidos os seguintes objetivos específicos:

- Estabelecer metodologia de desenvolvimento de software
- Definir as funcionalidades do software
- Definir uma arquitetura para atender as necessidades do sistema
- Desenvolver a aplicação *mobile*
- Desenvolver a aplicação *web*
- Testar o funcionamento da aplicação

- Disponibilizar aplicação *web* para a utilização do público
- Disponibilizar aplicação *mobile* na loja de aplicativos do sistema Android para a utilização do público

2 Referencial Teórico

2.1 Desenvolvimento Mobile

Nos dias atuais, é extremamente comum vermos empresas oferecendo serviços ou produtos através de soluções tecnológicas. Normalmente os *websites* são utilizados para tal propósito, entretanto, para alguns tipos de sistemas que necessitam de maior praticidade e facilidade de acesso ou recursos específicos do próprio celular como a câmera e o *Bluetooth* por exemplo, para esse tipo de sistema, normalmente as aplicações móveis ou apps são utilizados (EL-KASSAS et al., 2017).

Um aspecto muito importante ao desenvolver aplicações móveis, é saber que as limitações como tamanho de tela, nível de bateria, uso de memória e conectividade têm de ser uma preocupação constante nos dispositivos móveis. Além disso, também temos uma variedade de sistemas operacionais para os dispositivos móveis, tendo como principais sistemas, os sistemas Android e iOS que são responsáveis por aproximadamente 99% dos dispositivos utilizados no mundo (ANGELOVA, 2019), criados e desenvolvidos pela Google e Apple respectivamente. Por serem os principais, atualmente o desenvolvimento de aplicativos móveis tem como foco abranger os usuários desses sistemas.

2.2 Android

O Android é um sistema operacional de código aberto que possui a Google LLC como o principal colaborador desse projeto, é baseado no sistema operacional Linux (CHINETHA; JOANN; SHALINI, 2015), amplamente conhecido e utilizado.

A primeira versão comercial do sistema foi lançada em Setembro de 2008 e, desde então vem recebendo melhorias até os dias atuais (tendo o Android 12 como a última versão lançada). Atualmente está presente em mais de 2,5 bilhões de dispositivos em todo o mundo (GOOGLE, 2022).

Para desenvolvermos nativamente para plataformas Android, utilizamos a linguagem Java ou Kotlin (sendo essa a mais nova e mais indicada atualmente) em conjunto com as ferramentas fornecidas pela Google (SDK e IDE). Também é possível utilizar as linguagens C/C++ no desenvolvimento, porém não é uma prática comum no mercado.

2.3 iOS

O iOS, ou o chamado iPhone OS 1 em seu lançamento, é um sistema operacional criado e desenvolvido pela Apple Inc, derivando-se a partir do sistema Mac OS X, sua distribuição é exclusiva para hardwares fabricados pela mesma (SHEIKH et al., 2013).

Sua primeira versão comercial foi lançada em Junho de 2007, coincidindo com o lançamento do primeiro iPhone. Anualmente o sistema recebe grandes atualizações, tendo como última grande versão lançada o iOS 15.

O kit de desenvolvimento para desenvolver aplicações nativamente para plataformas iOS é composto principalmente pela linguagem Swift, que vem substituindo a linguagem Objective-C nos últimos anos, em conjunto com as ferramentas fornecidas pela Apple (SDK e IDE).

2.4 Desenvolvimento nativo

O desenvolvimento nativo utiliza as ferramentas e linguagens específicas para cada plataforma, ou seja, tem como foco apenas uma única plataforma.

Para aplicações mobile Android e iOS, utilizamos as ferramentas e linguagens demonstradas na tabela abaixo.

	iOS	Android
Linguagem de programação	Swift	Kotlin
Loja de aplicativos	App Store	Play Store
IDE	XCode	Android Studio

Tabela 1 – Ferramentas Android e iOS

Um dos pontos positivos ao desenvolver aplicações nativas é obter um maior desempenho das aplicações, além do total controle sobre a aplicação na plataforma desejada. No entanto, caso seja necessário desenvolver a mesma aplicação para diferentes plataformas, é necessário desenvolver múltiplas vezes a depender do número de plataformas desejadas, dessa forma, um dos problemas de tal abordagem seria quanto ao custo e tempo de desenvolvimento para a disponibilização da aplicação nas diferentes plataformas desejadas (EL-KASSAS et al., 2017).

2.5 Desenvolvimento multiplataforma

De acordo com El-Kassas et al. (2017) a definição de desenvolvimento multiplataforma, ou do inglês *cross-platform/multi-platform*, é desenvolver uma aplicação para

ser executada em mais de uma plataforma com apenas um único código. Esse tipo de abordagem é o ideal para quando se deseja uma velocidade de desenvolvimento maior em mais de uma plataforma a um custo reduzido, havendo como principal ponto negativo o impacto no desempenho da aplicação, que a depender da tecnologia utilizada pode ser um impacto de desempenho maior ou menor.

2.6 Metodologias de desenvolvimento

2.6.1 Kanban

A ferramenta Kanban, é utilizada como organização e gerenciamento das tarefas de um projeto. Através dessa ferramenta, podemos visualizar por quantas etapas cada tarefa passa até sua conclusão, além da etapa atual em que a tarefa se encontra e seus devidos responsáveis.

O Kanban é uma ferramenta bastante conhecida, sua origem veio do sistema de produção da Toyota por volta da década de 1950. Os principais benefícios na utilização dessa ferramenta são:

- Permitir uma visualização ampla do projeto
- Flexibilidade
- Prevenir erros
- Melhoria contínua do processo
- Transparência de atividades

O Kanban utiliza-se de um quadro dividido em colunas, onde em cada coluna representa-se uma etapa em que a tarefa percorre durante o fluxo de trabalho. A configuração padrão de um quadro Kanban apresenta uma divisão em três colunas (ZAYAT; SENVAR, 2020), sendo elas:

- To do (Para fazer)
- Doing (Fazendo)
- Done (Feito)

O quadro Kanban pode ser personalizado em diversas colunas caso o projeto tenha tal necessidade. Sendo assim, é extremamente flexível e facilmente aplicável em diversos contextos e projetos.

2.6.2 Scrum

O Scrum é um *framework* que divide problemas diversos em várias tarefas de pequenos ciclos chamados de *sprints*. Sua abordagem de gestão de projetos ocorre de maneira cíclica e incremental (ZAYAT; SENVAR, 2020). Cada ciclo tem sua duração específica, dependendo muito da necessidade de cada projeto e equipe, na maioria dos projetos entretanto, as sprints tendem a ter duração entre 1 a 4 semanas. Um outro detalhe importante acerca das sprints é que o começo de um novo ciclo é exatamente no final do ciclo anterior, sendo assim, os ciclos ocorrem de maneira contínua e sem intervalos.

O Scrum divide a equipe dentre alguns papéis principais, sendo eles:

- Product Owner - O Product Owner, tem como principal função garantir que a cada iteração de sprints o máximo de valor possível é entregue ao produto, mantendo sua alta qualidade. Além disso, o Product Owner também atua na priorização e organização do *Backlog* da equipe (ZAYAT; SENVAR, 2020).
- Scrum Master - É o responsável em auxiliar a aplicação do Scrum conforme explicado no guia do Scrum (SCHWABER; SUTHERLAND, 2020). Consequentemente, seu papel é garantir que toda a equipe entenda as teorias, regras e valores do Scrum, com o foco de maximizar o valor que o time entrega ao projeto.
- Time de desenvolvimento ou *Developers* - A cada sprint um novo incremento de produto é entregue pelo time de desenvolvimento. Esse time é auto-organizável e composto por profissionais responsáveis em desenvolver o projeto de acordo com as especificações das tarefas listadas no *Product Backlog* e *Sprint Backlog* (SCHWABER; SUTHERLAND, 2020).

O Scrum realiza algumas cerimônias a cada *sprint*, sendo elas:

- Daily - É uma reunião diária, com foco em atualizar a equipe sobre o progresso das tarefas da sprint, evitando possíveis bloqueios e barreiras. Tem o objetivo de ser uma reunião curta, de no máximo 15 minutos. (SCHWABER; SUTHERLAND, 2020).
- Sprint Planning - É uma reunião de planejamento, realizada no início do ciclo, na qual se determina o que será entregue durante a sprint.
- Sprint Review - É uma reunião onde são observados entregas e resultados da sprint. Nessa reunião também ocorre a retrospectiva da sprint, onde são feitas sugestões de melhorias e adaptações no processo de trabalho da equipe.

Após a realização dessas cerimônias, dois principais artefatos são gerados ou atualizados:

- Product Backlog - É a lista de todas as tarefas que precisam ser feitas para que o projeto/produto chegue ao seu objetivo. (SCHWABER; SUTHERLAND, 2020).
- Sprint Backlog - É a lista de todas as tarefas que foram priorizadas pelo Product Owner para serem feitas durante a sprint.

3 Metodologia

Neste capítulo, serão apresentadas as etapas, ferramentas e tecnologias utilizadas necessárias para se alcançar os objetivos e resultados esperados para o presente trabalho.

3.1 Fluxo de processos

Para o fluxo de desenvolvimento deste trabalho foi utilizada a notação *Business Process Model and Notation (BPMN)*, com o foco em detalhar as atividades de cada parte do processo de desenvolvimento do trabalho.

Na Figura 1, estão detalhadas as atividades referentes ao TCC juntamente a figura que representa o fluxo de processos utilizando a notação BPMN:

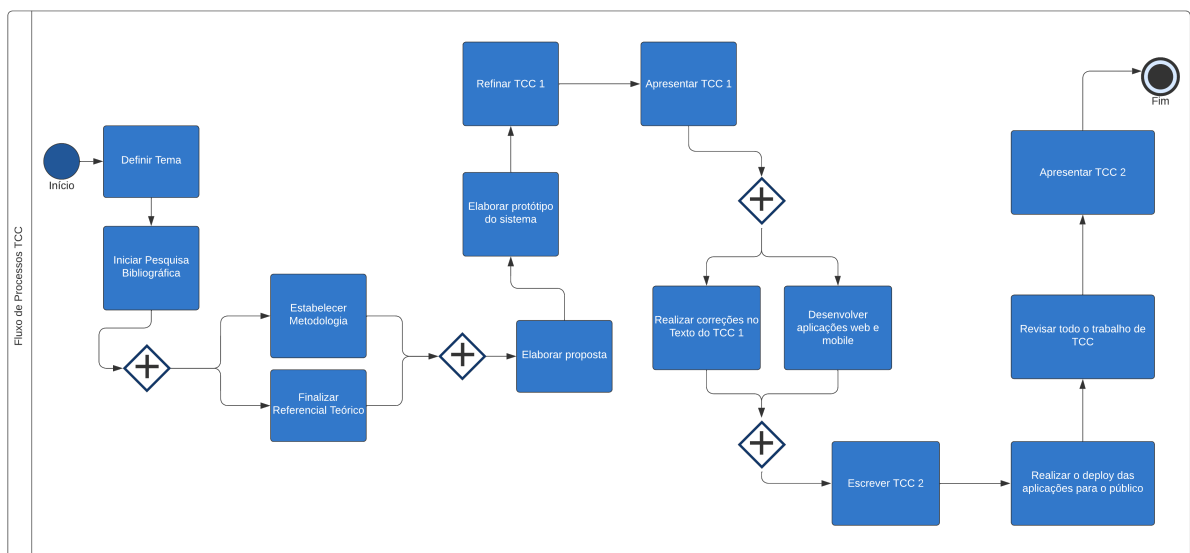


Figura 1 – Fluxo de Processos - TCC

- Definir tema: Foi pensado em como aplicar os conhecimentos de Engenharia de Software adquiridos ao longo do curso para resolver o problema já citado na Seção 1.2, com escopo delimitado;
- Iniciar pesquisa bibliográfica: Iniciou-se a pesquisa de bibliografias, com o objetivo de adquirir conhecimento para fundamentar e atingir os objetivos deste trabalho;
- Estabelecer metodologia: A metodologia foi estabelecida através da definição das tecnologias e ferramentas que serão utilizadas no desenvolvimento do software para alcançar os resultados esperados com o trabalho;

- Finalizar referencial teórico: Em paralelo com a tarefa de estabelecer a metodologia, a finalização do referencial teórico acontece através da base de pesquisa bibliográficas obtidas anteriormente;
- Elaborar proposta: Após a definição do referencial teórico e metodologias, é possível elaborar uma proposta de solução para o problema abordado neste trabalho;
- Elaborar protótipo do sistema: Essa atividade tem como foco elaborar um protótipo de alta fidelidade do aplicativo a ser desenvolvido no TCC 2;
- Refinar TCC 1: Essa atividade tem como foco realizar as verificações de erros ou inconsistências no trabalho e realizar as devidas correções;
- Apresentar TCC 1: Apresentar o TCC 1 para a banca;
- Realizar correções no texto do TCC 1: Após a apresentação para a banca, são feitas melhorias e correções no trabalho, baseados no *feedback* obtido;
- Desenvolver as aplicações *web* e *mobile*: Paralelo ao processo de correções, inicia-se o processo de desenvolvimento do projeto. Por se tratar de um processo longo, esse processo engloba a configuração e integração com os serviços do Firebase, além de todo o processo de desenvolvimento das aplicações;
- Escrever TCC 2: Após finalizado o processo de desenvolvimento da aplicação, ocorre a escrita de todo processo de desenvolvimento do TCC 2;
- Realizar o *deploy* das aplicações para o público: Após finalizada a escrita do TCC 2, ocorre a liberação das aplicações para o público;
- Revisar todo o trabalho de TCC: São realizadas revisões em todo o trabalho, em busca de erros ou inconsistências no texto ou nas aplicações, aplicando as devidas correções;
- Apresentar TCC 2: Apresentar o TCC 2 para a banca.

3.2 Ferramentas utilizadas para o Desenvolvimento de Software

3.2.1 JavaScript

O JavaScript, comumente abreviado como JS, é uma linguagem de programação interpretada de alto nível, amplamente utilizada em diversos sistemas existentes nos dias atuais. Seu principal uso é para sistemas web, porém, por ser uma linguagem extremamente versátil é possível desenvolver tanto sistemas web e *mobile* como sistemas *desktop* e *back-end*.

3.2.2 React Native

O React Native é um *framework* JavaScript de código aberto criado, desenvolvido e mantido pelo Facebook (atualmente conhecida como Meta) que permite o desenvolvimento de aplicações *mobile* focado nas plataformas Android e iOS. Sua criação teve como principal objetivo tornar mais simples o desenvolvimento de aplicações móveis, além de acelerar esse processo de desenvolvimento, permitindo o desenvolvimento de uma aplicação para multiplataformas com apenas um código fonte. Por ter como linguagem base o JavaScript, facilita o aprendizado por desenvolvedores que já tenham algum conhecimento em desenvolvimento web.

O React Native internamente possui uma camada de abstração chamada de *bridge*. Essa camada possibilita que exista uma conversão automática do código desenvolvido em JavaScript para a linguagem específica de cada plataforma móvel (iOS e Android), permitindo a renderização de interface reais e nativas através dos componentes nativos usados em cada uma dessas plataformas (EISENMAN, 2015).

A principal vantagem do React Native em relação aos seus concorrentes de desenvolvimento híbrido (Flutter por exemplo), é a renderização de elementos realmente nativos a cada plataforma, tendo um menor impacto no desempenho e uma interface mais próxima ao padrão seguido por ambas plataformas.

Por ser de código aberto, a comunidade que contribui com o desenvolvimento dessa tecnologia é extremamente grande. Além disso, é possível observar o seu uso dentro de grandes empresas como Tesla, Microsoft, Facebook e AirBnB. Existe também uma versão do React Native que possui suporte para aplicações Desktop, como MacOS e Windows, tal versão é mantida pela Microsoft.

Para o presente trabalho a escolha dessa tecnologia foi adotada com o objetivo de acelerar o processo de desenvolvimento, tornando viável a implementação para as plataformas Android e iOS, que são as plataformas suportadas pela aplicação que será desenvolvida.

3.2.3 ReactJS

O ReactJS é uma biblioteca JavaScript de código aberto também desenvolvido e mantido pelo Facebook, com foco para o desenvolvimento de aplicações *web*, permitindo a criação de interfaces de forma declarativa.

O uso dessa tecnologia para desenvolvimento da aplicação na parte *web* do projeto impacta positivamente neste trabalho, visto que a *stack* de desenvolvimento do projeto será dependente de somente uma linguagem de programação, no caso, o JavaScript. Facilitando assim, futuras manutenções no sistema como um todo caso seja necessário.

3.2.4 Firebase

O Firebase é uma plataforma desenvolvida pela Google e tem como um de seus principais serviços o funcionamento no modelo de *Back-end as a Service*.

Devido ao tempo para a criação e desenvolvimento deste projeto ser bastante curto e com somente um desenvolvedor, o Firebase permitiu tal criação ser acelerada, permitindo ter um foco maior nas plataformas que serão utilizadas pelos usuários finais.

Um dos principais serviços do Firebase que foram utilizados neste projeto foi o serviço chamado de *Firestore*, que é definido como um banco de dados não relacional em tempo real. Neste serviço, todo o armazenamento, leitura e escrita de dados acontece em tempo real. Também é possível definir regras de acesso para todos os dados inseridos no banco de dados, permitindo assim, uma base de dados segura e confiável.

Um outro serviço do Firebase que também foi utilizado, é o serviço chamado de *Firebase Authentication*, que pode ser definido como um sistema de autenticação que é gerenciado pelo Firebase, permitindo desde soluções de autenticação simples com email e senha até autenticações mais complexas como por exemplo, o login com as contas Google, Apple, Github e número de telefone, além de toda parte de armazenamento seguro dessas informações de login.

Através da SDK desenvolvida pelo Firebase para a comunicação com seus serviços, foi possível o desenvolvimento de uma aplicação com atualizações em tempo real e sem a preocupação com a escalabilidade e desempenho deste sistema, deixando este problema a cargo da nuvem de serviços do Firebase.

3.2.5 GitHub

O GitHub é uma plataforma extremamente conhecida e utilizada mundialmente para fazer hospedagem de código-fonte e arquivos através do método de controle de versão, nessa plataforma é possível utilizar a ferramenta Git para fazer o versionamento do código do projeto desejado.

Essa ferramenta permite dividir o código fonte em ramos, facilitando o trabalho em grupo de forma que no ramo principal só tenham modificações totalmente estáveis, enquanto o ramo de desenvolvimento terá as modificações que estarão sendo trabalhadas naquele momento.

O principal benefício que essa ferramenta traz a este trabalho é conseguir ter o controle de como cada versão do código evolui ao longo do tempo, além disso, facilitará a continuação do desenvolvimento dessa aplicação para trabalhos futuros. A divisão em diferentes ramos não terá benefícios para este trabalho dado que seu desenvolvimento será feito por apenas uma pessoa. Sendo assim, utilizado o modelo de *Trunk Based Develop-*

ment.

3.3 Metodologias utilizadas para o Desenvolvimento de Software

As metodologias utilizadas para o desenvolvimento deste trabalho serão Kanban e Scrum, conforme apresentadas nas Seções 2.6.1 e 2.6.2. Entretanto, como apenas uma pessoa está envolvida no desenvolvimento do projeto, não há necessidade de aplicar todos os elementos citados do Scrum, foi optado por apenas a utilização dos conceitos de *product backlog*, *sprint backlog*, *sprint* e *sprint planning*.

4 Proposta do sistema

Este capítulo tem como objetivo descrever os requisitos do sistema que foi desenvolvido para atingir os objetivos deste trabalho. O foco principal do sistema serão as aplicações mobile para iOS e Android, além de um sistema web para administração e aprovação das vagas.

Para esse sistema possuímos 3 tipos de usuário, sendo divididos em:

- Usuário comum: Esse tipo de usuário é o mais simples, pensado em atender exclusivamente os alunos, possui permissões no sistema apenas de leitura das vagas;
- Divulgador: Esse tipo de usuário foi pensado com o objetivo de permitir empresas ou terceiros em poder divulgar suas vagas, enviando-as para a aprovação do administrador;
- Administrador: Esse tipo de usuário é essencial para o sistema, permitindo a aprovação de vagas e possuindo acesso total ao sistema e todas as suas funcionalidades;

4.1 *Product Backlog*

Para o desenvolvimento das aplicações *mobile* e *web*, foi definido o *Product Backlog* demonstrado nas Tabelas 2 e 3, respectivamente:

Nome do Item	Descrição
PB1	Eu, como usuário, desejo me cadastrar no aplicativo.
PB2	Eu, como usuário, desejo logar no aplicativo.
PB3	Eu, como usuário, desejo poder visualizar uma lista com todas as vagas.
PB4	Eu, como usuário, desejo poder visualizar mais informações sobre uma vaga.
PB5	Eu, como usuário, desejo poder pesquisar por uma vaga de um curso específico.
PB6	Eu, como divulgador, desejo poder enviar uma vaga para aprovação de divulgação.
PB7	Eu, como divulgador, desejo poder editar as informações de uma vaga divulgada por mim.
PB8	Eu, como divulgador, desejo poder excluir a vaga divulgada por mim.
PB9	Eu, como administrador, desejo poder editar qualquer vaga aprovada.
PB10	Eu, como administrador, desejo poder excluir qualquer vaga aprovada.

Tabela 2 – Product Backlog da Aplicação Mobile

Nome do Item	Descrição
PB1	Eu, como divulgador, desejo logar na aplicação <i>web</i> .
PB2	Eu, como administrador, desejo logar na aplicação <i>web</i> .
PB3	Eu, como administrador, desejo poder visualizar uma lista com todas as vagas ativas.
PB4	Eu, como administrador, desejo poder visualizar uma lista com todas as vagas expiradas.
PB5	Eu, como administrador, desejo poder visualizar uma lista com todas as vagas pendentes de aprovação.
PB6	Eu, como administrador, desejo poder editar qualquer vaga.
PB7	Eu, como administrador, desejo poder excluir qualquer vaga.
PB8	Eu, como divulgador, desejo poder enviar uma vaga para aprovação de divulgação.
PB9	Eu, como divulgador, desejo poder editar as informações de uma vaga divulgada por mim.
PB10	Eu, como divulgador, desejo poder excluir a vaga divulgada por mim.

Tabela 3 – Product Backlog da Aplicação Web

5 Desenvolvimento das Aplicações

Neste capítulo serão demonstradas as etapas a realizadas no Desenvolvimento das Aplicações *web* e *mobile*:

5.1 Arquitetura do Sistema

Na Figura 2 é possível visualizar um diagrama de serviços que demonstra a comunicação das aplicações desenvolvidas neste trabalho com os serviços do Firebase, a conexão entre as pontas *web* e *mobile* foi feita através das SDKs nativas que são disponibilizadas pelo Firebase para diversas plataformas e linguagens.

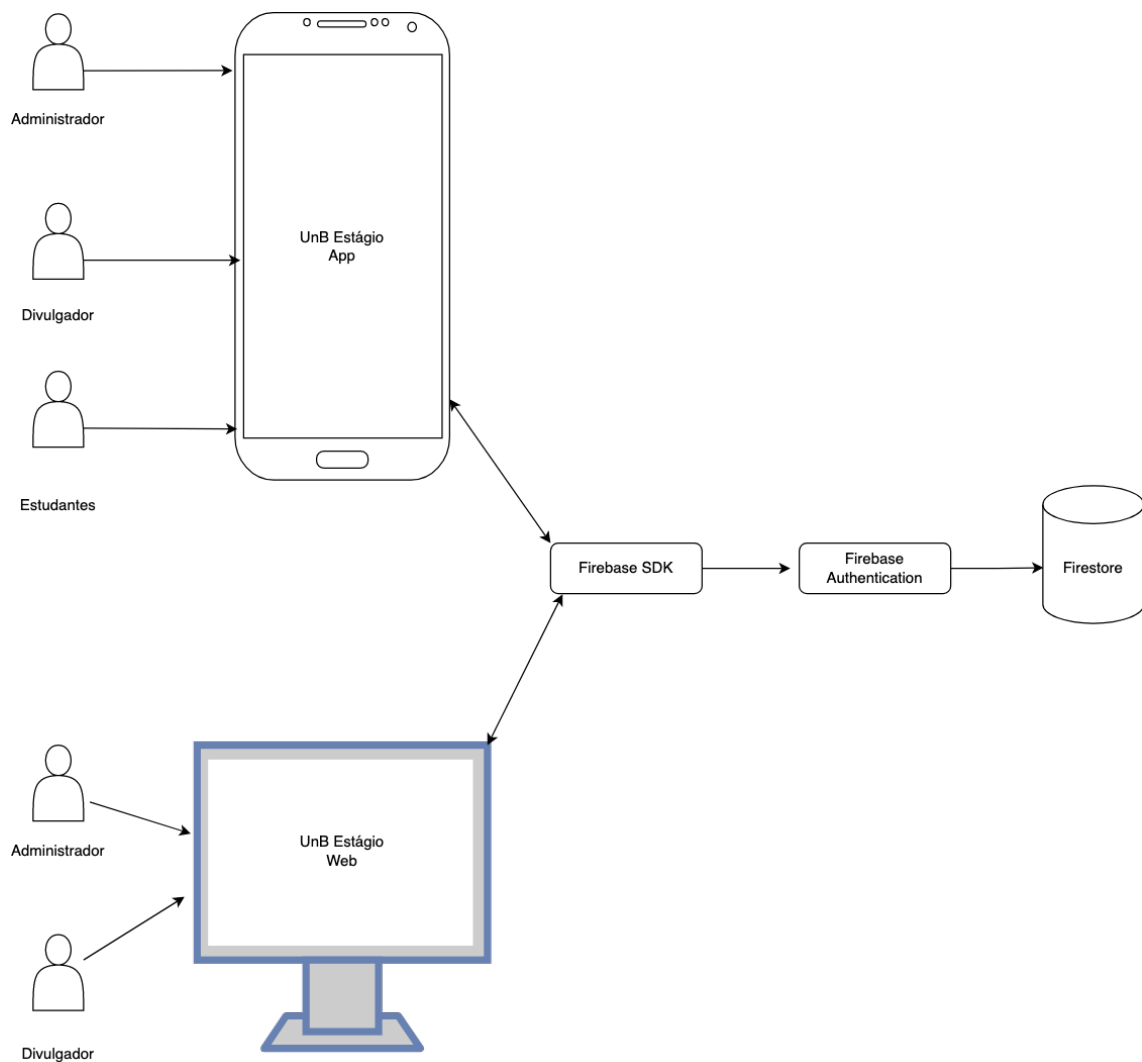


Figura 2 – Arquitetura de serviços

5.2 Modelo de domínio

Para atender os requisitos da aplicação, o modelo de domínio representado pela Figura 3 foi utilizado:



Figura 3 – Diagrama de domínio

O modelo de Usuários ou *Users* é composto por: Um identificador único de cada usuário, o email e nome deste usuário, além de sua *role* (Função) no sistema, podendo ser um dos valores representados por *AdminUserType*, *PromoteUserType*, *CommonUserType*.

Já o modelo de Empregos ou *Jobs* é composto por: Um identificador único de cada vaga, além dos campos de agência integradora, empresa, contato, lista de cursos, descrição, data de início e fim das inscrições, horas semanais, bolsa, título da vaga, e esperando por aprovação.

5.3 Publicação das aplicações

A publicação da aplicação *web* ocorreu na plataforma ¹[Netlify](#), que trata-se de uma plataforma de hospedagem de aplicações *web* de forma gratuita. O processo de publicação nessa plataforma foi simples, sendo necessário somente se cadastrar na plataforma e

¹ Link para acesso ao [Netlify](#)

fornecer o acesso da plataforma ao código fonte da aplicação. Para ter acesso a aplicação *web* é necessário acessar o link <<https://unbestagio.netlify.app/>>.

O aplicativo foi publicado na loja de aplicativos do sistema Android, conhecida como Play Store, e está disponível para *download* em todo Brasil através do link <<https://play.google.com/store/apps/details?id=com.unbestagio>> para dispositivos com o sistema operacional Android com versão igual ou superior a 6.0.

Para a publicação na loja de aplicativos, foi necessária a criação de uma conta de desenvolvedor Google, tal criação só é possível mediante o pagamento de uma taxa. Após a criação da conta, é necessário adicionar e preencher as informações sobre o aplicativo a ser enviado para a loja, dentre essas informações, podemos citar como as principais sendo:

- Nome do aplicativo;
- Indicação de aplicativo pago ou gratuito;
- Classificação indicativa;
- Categoria do aplicativo;
- Descrição;
- Ícone para a aplicação;
- *Screenshots* da aplicação;
- Países que podem fazer o *download*.

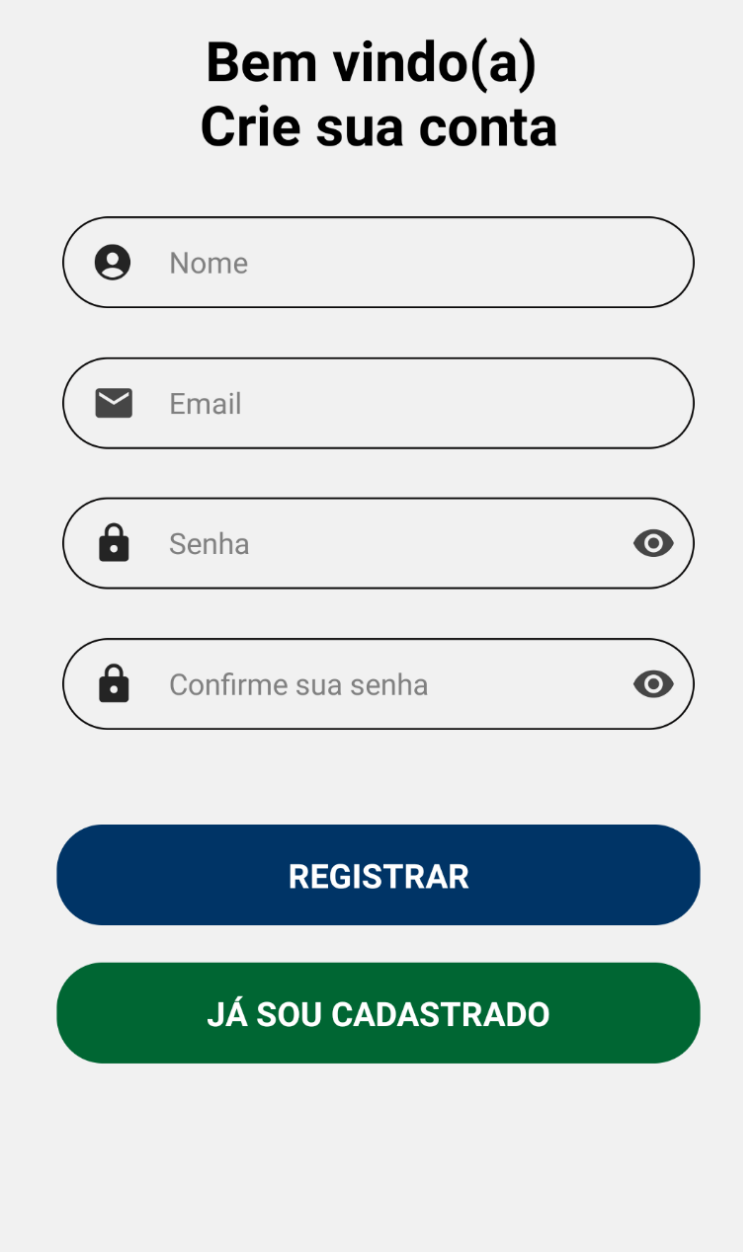
Após esse processo, é necessário fazer o *upload* do arquivo para a loja, e aguardar o processo de revisão ser finalizado, onde passará por uma série de análises automáticas para garantir que não há nenhum *malware* na aplicação. Após as análises automáticas serem finalizadas, uma análise manual é feita por algum funcionário da Google para garantir que o conteúdo e demais informações do aplicativo estão de acordo ao submetido para a loja.

O processo de revisão pela Google levou cerca de 6 dias para ser finalizado e aprovado. Todo o processo de geração dos arquivos para publicação foi feito de forma manual, podendo ser feito de forma automatizada em trabalhos futuros.

A publicação na loja de aplicativos da Apple, conhecida como App Store, não foi possível devido ao alto custo para se obter as contas de desenvolvedor, podendo também ser uma melhoria futura.

5.4 Aplicação *mobile*

Na Figura 4, podemos visualizar a página de cadastro do aplicativo, para isso devemos preencher os campos Nome, Email, Senha e Confirme sua senha. Após todos os campos preenchidos, devemos clicar no botão de Registrar para sermos redirecionados para a tela Home, representada pela Figura 6. Caso o usuário já possua uma conta cadastrada, é necessário pressionar o botão Já sou cadastrado para sermos redirecionados para a tela de Login, representada pela Figura 5.



A tela de cadastro apresenta o seguinte layout:

- Um cabeçalho com o texto "Bem vindo(a) Crie sua conta" em negrito.
- Quatro campos de entrada de texto, cada um com um ícone à esquerda e um ícone de olho à direita para alternar a visibilidade da senha:
 - Nome (ícone de pessoa)
 - Email (ícone de envelope)
 - Senha (ícone de cadeado)
 - Confirme sua senha (ícone de cadeado)
- Dois botões de ação no final da tela:
 - Um botão azul com o texto "REGISTRAR" em branco.
 - Um botão verde com o texto "JÁ SOU CADASTRADO" em branco.

Figura 4 – Tela de Cadastro



Figura 5 – Tela Login

Na Figura 5, podemos visualizar a página de login do aplicativo, para isso devemos preencher os campos Email, Senha. Após todos os campos preenchidos, devemos clicar no botão de Login para sermos redirecionados para a tela Home, representada pela Figura 6. Caso o usuário seja um usuário dos tipos Administrador ou Divulgador, a tela Home apresentada será a Figura 7.

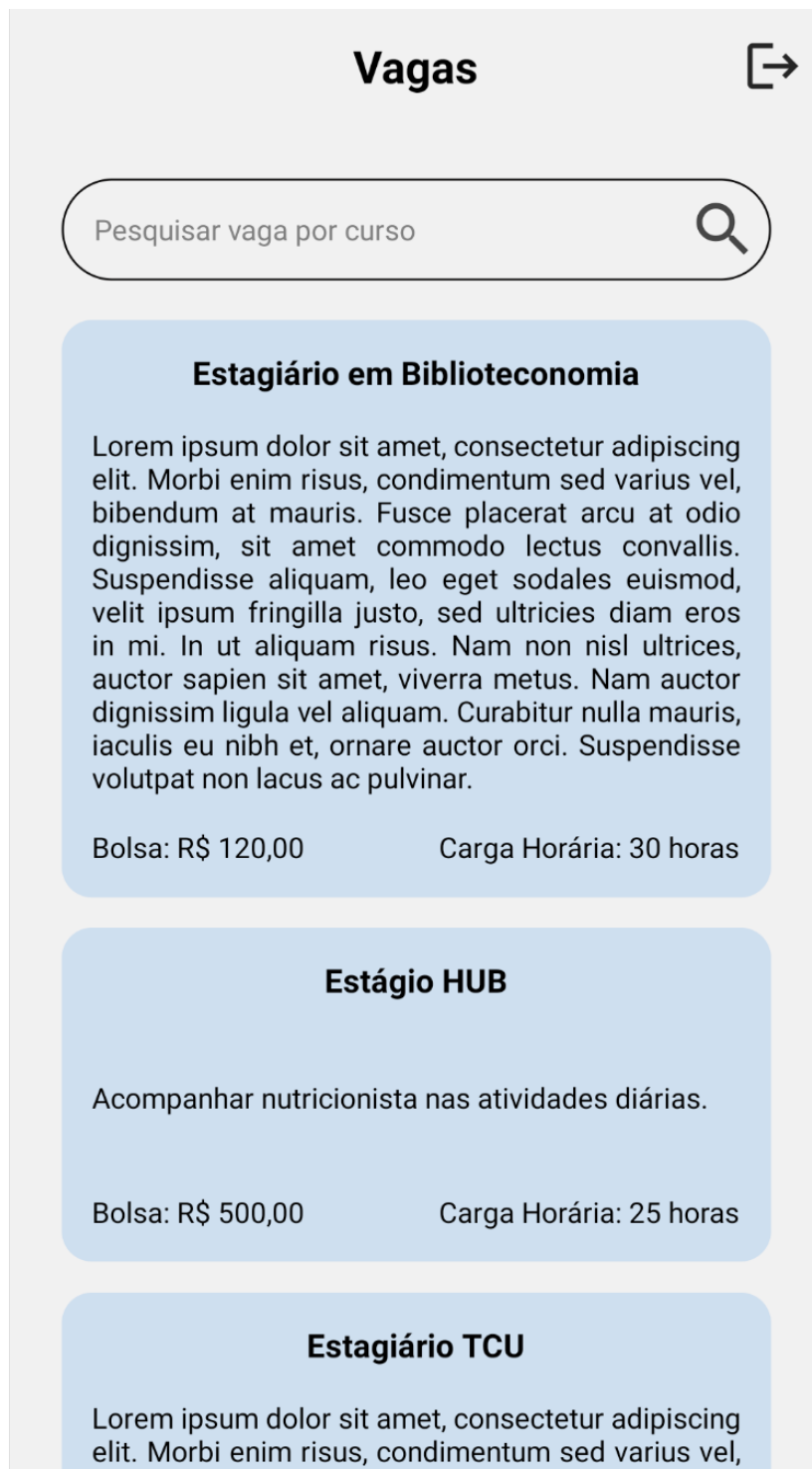


Figura 6 – Tela Home

Na Figura 6, podemos visualizar a página inicial do aplicativo contendo a lista de vagas e um botão para ativar o filtro para o curso desejado. Ao clicar sobre uma vaga desejada, devemos ser redirecionados para a tela de Detalhes, representada pela Figura 10.

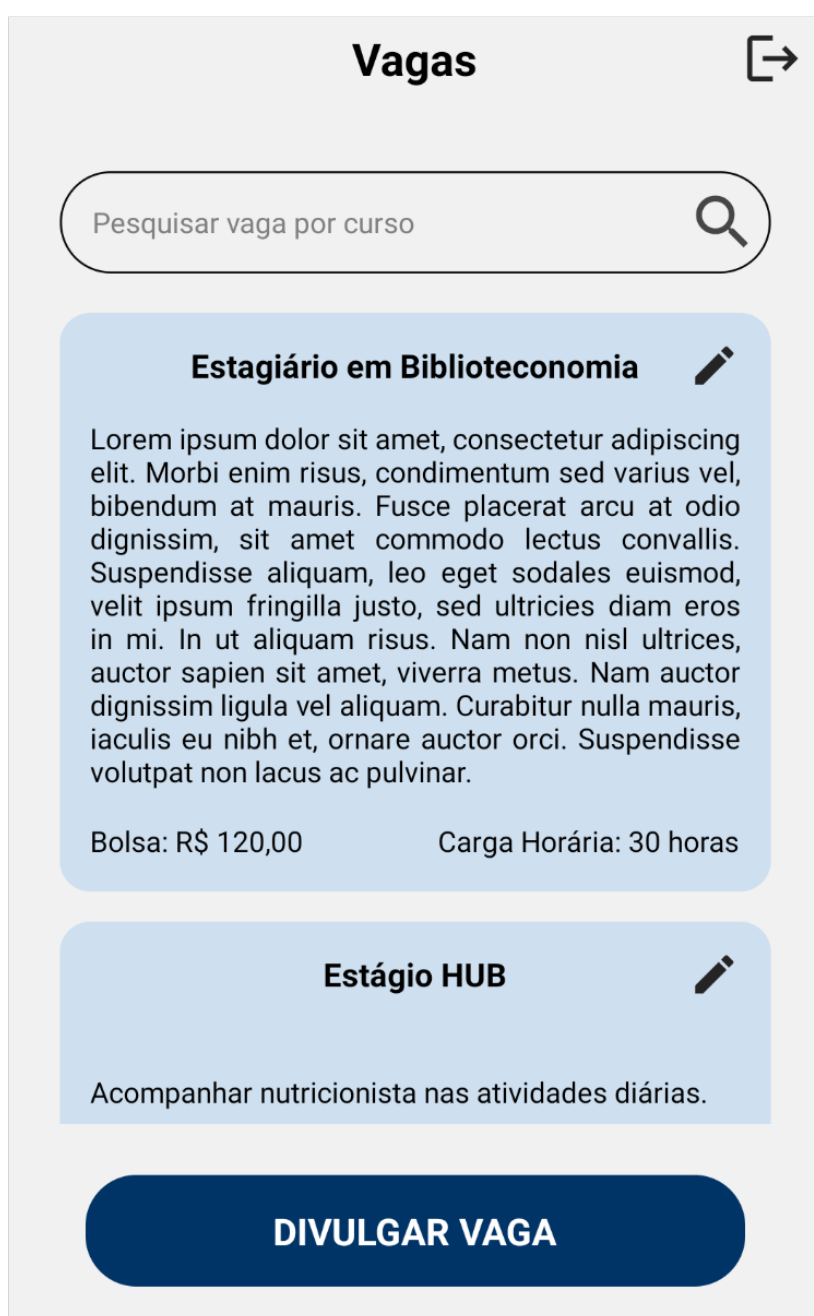


Figura 7 – Tela Home - Administrador

Na Figura 7, podemos visualizar tudo o que foi citado na Figura 6. Além disso, é possível clicar sobre o ícone de lápis em qualquer vaga para ser redirecionado para a tela de edição de vagas, representada pela Figura 9, para usuários do tipo Divulgador, esse ícone só aparecerá caso tenha sido uma vaga criada por este usuário. Caso o botão de Divulgar Vaga seja clicado, seremos redirecionados para a tela de Divulgação de Vaga, representada pela Figura 8.

← **Divulgar Vaga**

Contato (opcional)

Empresa

Agência integradora (opcional)

Início das inscrições

Fim das inscrições

Divulgar

Figura 8 – Tela Divulgação

Na Figura 8, podemos visualizar a página de criação de vagas, contendo os campos de Título, Descrição, Carga, Bolsa, Contato, Empresa, Agência integradora, Início das inscrições e Fim das inscrições. Após preenchidos os campos obrigatórios, basta pressionar o botão de Divulgar para enviar a vaga para aprovação do usuário Administrador.

← **Divulgar Vaga**

Contato (opcional)

UnB

Agência exemplo

08/09/2022

15/09/2022

Salvar Alterações

Apagar Vaga

Figura 9 – Tela Edição

Na Figura 9, podemos visualizar a página de edição de vaga, com todos os campos contidos na Figura 8 já devidamente preenchidos com as informações atuais. Caso desejemos editar as informações da vaga, basta preencher os campos a serem atualizados e confirmar pressionando o botão Salvar Alterações ao final da tela. Para deletar a vaga, é necessário pressionar o botão Apagar Vaga e confirmar.

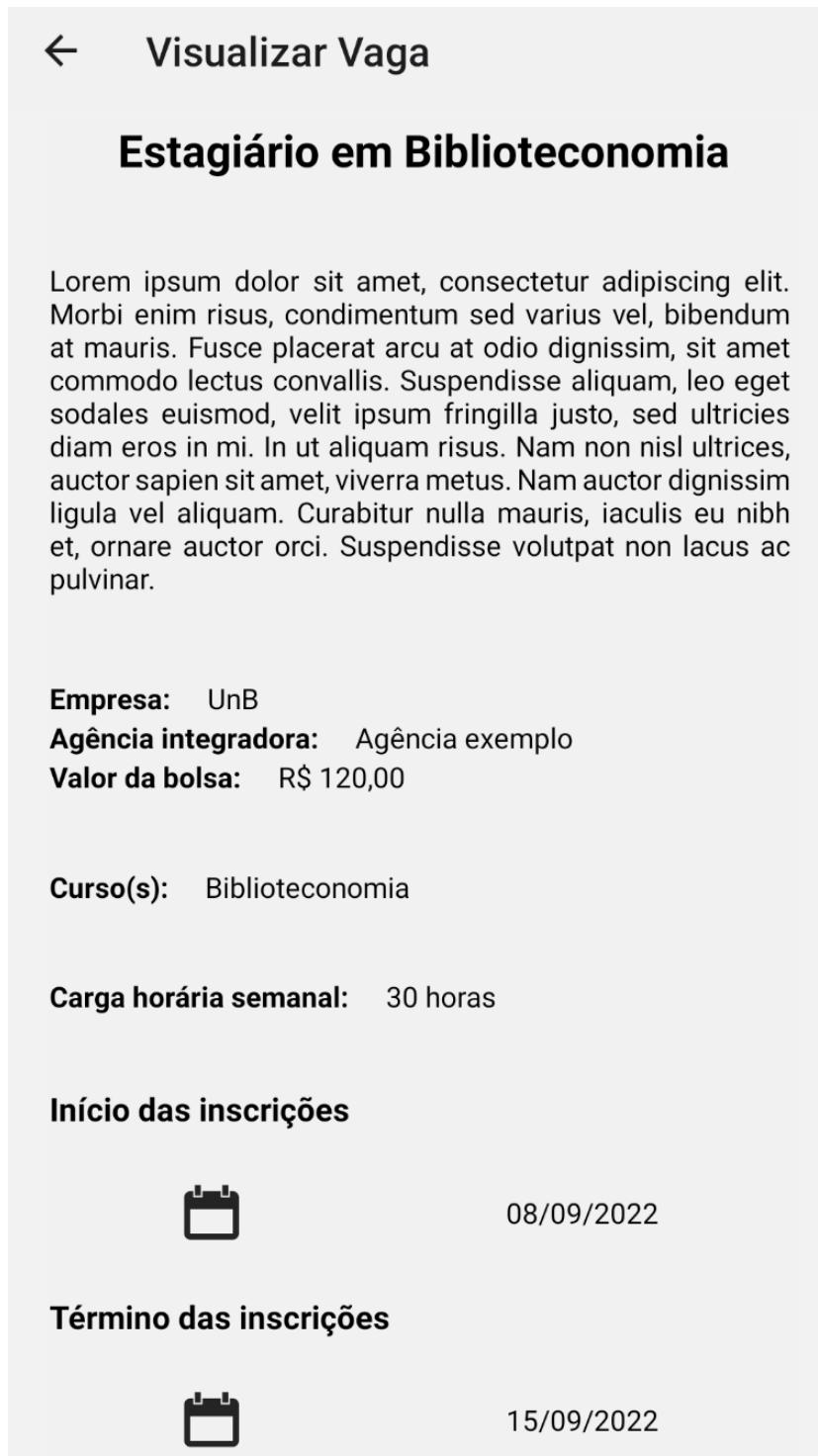


Figura 10 – Tela Detalhes

Na Figura 10, podemos visualizar todas as informações da vaga selecionada e um botão para retornarmos para a tela de lista de vagas.

5.5 Aplicação web



Figura 11 – Tela login - *web*

Na Figura 11, podemos visualizar a página de login da aplicação *web*, após preencher os campos Email, Senha devemos clicar no botão de Entrar para sermos redirecionados para a tela Home, representada pela Figura 6.

The screenshot shows a web browser window with the URL `unbestagio.netlify.app/home`. The page has a dark blue header with the text "UnB Estágio" on the left and navigation links "CRIAR VAGAS", "APROVAR VAGAS", "LISTA DE VAGAS", and "FAZER LOGOUT" on the right. The main content area is white and features the heading "Preencha os campos abaixo para criar uma nova vaga". Below the heading are several form fields: a text input for "Titulo *", a dropdown menu for "Selecione os cursos *", a text input for "Valor da bolsa *", a text input for "Carga Horária *" with a sub-label "Carga horária semanal", a large text area for "Descrição", and a text input for "Contato".

Figura 12 – Tela home - *web*

Na Figura 12, podemos visualizar a página de criação de vagas da aplicação *web*, possuindo os mesmos campos já demonstrados na Figura 8. Ao finalizar o preenchimento devemos clicar no botão de Criar Vaga, para enviar a vaga para aprovação.

The screenshot shows the "Aprovar vagas" page of the UnB Estágio application. The header is identical to the previous figure. The main content area is white and features the heading "Aprovar vagas". Below the heading is a light blue rounded rectangle containing the details of a "Vaga exemplo". The details include a placeholder text, "Empresa: empresa", "Agência Integradora: agência", "Bolsa: R\$ 1.000,00", "Cursos: Ciência da Computação (bacharelado), Ciências Sociais, Comunicação Social - Audiovisual", "Carga horária: 35 horas", "Início inscrições: 14/09/2022", and "Término inscrições: 19/09/2022". To the right of the rectangle is a green rounded button labeled "APROVAR VAGA".

Figura 13 – Tela de aprovação das vagas

Na Figura 13, podemos visualizar a página de aprovação de vagas da aplicação *web*, para aprovar uma vaga desejada, é necessário clicar sobre o botão de Aprovar Vaga localizado ao lado de cada vaga.



Figura 14 – Tela lista de vagas - *web*

Na Figura 14, podemos visualizar a página de lista de vagas da aplicação *web*, possuindo um funcionamento bastante similar a tela representada pela Figura 6, a maior diferença entre elas é a nova aba de Vagas expiradas, podendo assim ter o controle sobre as vagas já apresentadas anteriormente.

5.6 Teste de Usabilidade

Foi criado um formulário para a realização de um teste de usabilidade com usuários, o objetivo deste formulário é validar o sistema construído, além de obter possíveis pontos de melhoria para trabalhos futuros.

Os testes foram divulgados para um grupo de alunos e outro grupo de professores/coordenadores. O grupo de alunos somente realizou testes na plataforma *mobile*. O grupo de professores/coordenadores recebeu login de administrador no sistema. Após cada teste, perguntas de *feedback* sobre o teste foram feitas para ambos os grupos. Para o grupo de alunos os seguintes testes foram realizados:

- Fazer login no aplicativo através da criação de um novo usuário.

- Fazer login no aplicativo e procurar por uma vaga do curso de Engenharia de *Software*.

Já para o grupo de professores/coordenadores foram realizados testes nas aplicações *web* e *mobile*. Para a aplicação *web*, foram realizados os seguintes testes:

- Fazer login no *site* através da conta de administrador recebida juntamente da pesquisa.
- Enviar uma vaga para aprovação.
- Aprovar a vaga criada no item anterior.
- Editar as informações da vaga aprovada no item anterior e em seguida tente excluir essa vaga.

Para a aplicação *mobile*, foram realizados os seguintes testes:

- Fazer login no *aplicativo* através da conta de administrador recebida juntamente da pesquisa e em seguida enviar uma vaga para aprovação.
- Editar as informações de uma vaga qualquer.

6 Considerações Finais

Com relação aos testes realizados na Seção 5.6, foram obtidas 13 respostas no total e com isso recebidas algumas sugestões e *feedbacks* em relação ao sistema.

- 1 sugestão de mudança no processo de login - Adicionar opção de esqueci minha senha
- 1 sugestão de alteração no processo de pesquisa por vagas - Acrescentar pesquisas por empresas e possibilidade de inscrição nas vagas pelo próprio aplicativo
- 100% dos alunos consideram o aplicativo intuitivo
- 70% dos alunos consideram as cores utilizadas no aplicativo ótimas
- 81% considerou todos os testes fáceis e não foram relatadas dificuldades.
- 1 sugestão de mudança no processo de login - Incluir orientação sobre uso de *email* institucional caso seja necessário
- 1 sugestão de alteração no processo de enviar vaga para aprovação - Acrescentar a palavra *semanal* no campo de carga horária
- 1 sugestão de mudança no processo de edição e exclusão de vagas - Possibilidade de apresentar a proposta na tela após confirmação da edição
- Em relação a versão *mobile* foi dada uma sugestão de identificar os campos obrigatórios na área de edição de vagas

As aplicações desenvolvidas durante este trabalho representam um mínimo produto viável para o escopo deste trabalho, fornecendo funcionalidades básicas que permitem a visualização de vagas pelos alunos na plataforma *mobile* e criação dessas vagas através de ambas as plataformas, pelos usuários com as devidas permissões. Entretanto, há espaço para melhorias no sistema que podem ser implementadas em trabalhos futuros, como exemplo, podemos citar:

- Disponibilização da aplicação *mobile* para dispositivos iOS
- Implementar funcionalidade de adição de novos administradores pela aplicação *web*
- Implementar funcionalidade de adição de novos divulgadores pela aplicação *web*
- Exportação de vagas através de documentos estruturados

- Inscrição em vagas abertas pelo aplicativo
- Login com conta Google
- Acrescentar animações ou outras melhorias visuais na aplicação *mobile*
- Implementar seção informativa no aplicativo, contendo diversas informações sobre o processo de estágio.

Referências

- ANGELOVA, N. Mobile applications for business. *Trakia Journal of Sciences*, v. 17, p. 853–859, 07 2019. Citado na página 25.
- BRASIL. Lei n. 11.788, de 25 de setembro de 2008. *Diário Oficial da União*, 2008. Disponível em: <<https://www2.senado.leg.br/bdsf/bitstream/handle/id/385449/Lei%20do%20Estagio.pdf>>. Citado 2 vezes nas páginas 21 e 22.
- CHINETHA, K.; JOANN, J. D.; SHALINI, A. An evolution of android operating system and its version. *International Journal of Engineering and Applied Sciences*, v. 2, n. 2, 2015. Citado na página 25.
- EISENMAN, B. *Learning react native: Building native mobile apps with JavaScript*. [S.l.]: "O'Reilly Media, Inc.", 2015. Citado na página 33.
- EL-KASSAS, W. S. et al. Taxonomy of cross-platform mobile applications development approaches. *Ain Shams Engineering Journal*, v. 8, n. 2, p. 163–190, 2017. ISSN 2090-4479. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2090447915001276>>. Citado 2 vezes nas páginas 25 e 26.
- GOOGLE, L. *What is Android*. 2022. Disponível em: <https://www.android.com/intl/pt-BR_/what-is-android/>. Citado na página 25.
- SCHWABER, K.; SUTHERLAND, J. O guia do scrum: O guia definitivo para o scrum: As regras do jogo. 2020. Disponível em: <<https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-PortugueseBR-2.0.pdf>>. Citado 2 vezes nas páginas 28 e 29.
- SHEIKH, A. A. et al. Smartphone: Android vs ios. *The SIJ Transactions on Computer Science Engineering & its Applications (CSEA)*, v. 1, n. 4, p. 141–148, 2013. Citado na página 26.
- ZAYAT, W.; SENVAR, O. Framework study for agile software development via scrum and kanban. *International Journal of Innovation and Technology Management*, v. 17, n. 04, p. 2030002, 2020. Disponível em: <<https://doi.org/10.1142/S0219877020300025>>. Citado 2 vezes nas páginas 27 e 28.