

UNIVERSIDADE FEDERAL DO PARANÁ

ANDRÉ KLINGENFUS ANTUNES

BRUNO FERNANDES

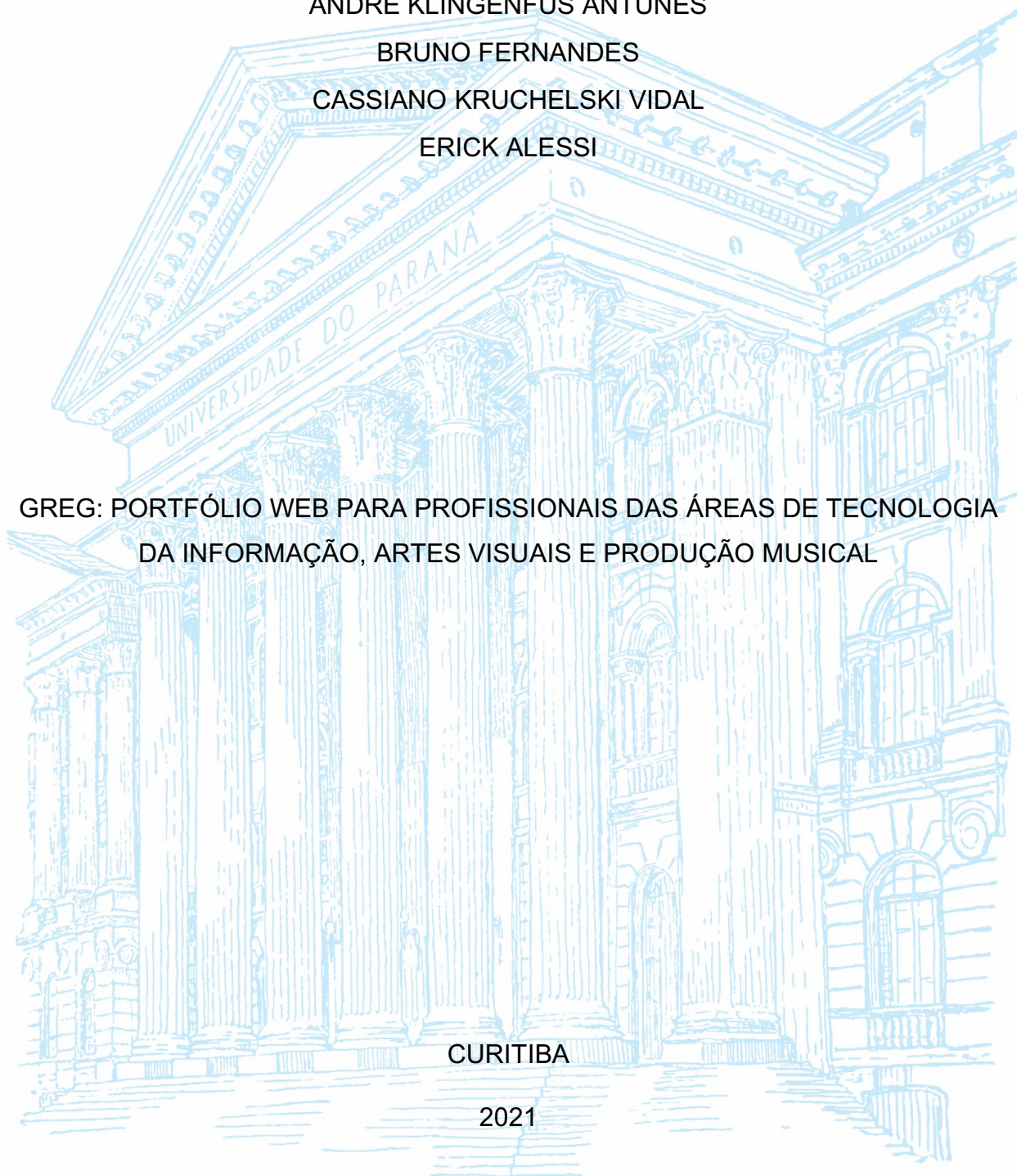
CASSIANO KRUCHELSKI VIDAL

ERICK ALESSI

GREG: PORTFÓLIO WEB PARA PROFISSIONAIS DAS ÁREAS DE TECNOLOGIA
DA INFORMAÇÃO, ARTES VISUAIS E PRODUÇÃO MUSICAL

CURITIBA

2021



ANDRÉ KLINGENFUS ANTUNES
BRUNO FERNANDES
CASSIANO KRUCHELSKI VIDAL
ERICK ALESSI

GREG: PORTFÓLIO WEB PARA PROFISSIONAIS DAS ÁREAS DE TECNOLOGIA
DA INFORMAÇÃO, ARTES VISUAIS E PRODUÇÃO MUSICAL

Trabalho de Conclusão de Curso apresentado ao Curso de Tecnologia em Análise e Desenvolvimento de Sistemas da Universidade Federal do Paraná como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Dr. Alexander Robert Kutzke

CURITIBA

2021



UNIVERSIDADE FEDERAL DO PARANÁ

ATA DE REUNIÃO

TERMO DE APROVAÇÃO

ANDRÉ KLINGENFUS ANTUNES

BRUNO FERNANDES

CASSIANO KRUCHELSKI VIDAL

ERICK ALESSI

GREG: PORTFÓLIO WEB PARA PROFISSIONAIS DAS ÁREAS DE TECNOLOGIA DA INFORMAÇÃO, ARTES VISUAIS E PRODUÇÃO MUSICAL

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Dr. Alexander Robert Kutzke (Orientador)

Orientador - SEPT/UFPR

Prof. Dra. Rafaela Mantovani Fontana

SEPT/UFPR

Prof. Dra. Sandramara Scandelari Kusano de Paula Soares

SEPT/UFPR

Curitiba, 29 de março de 2021.



Documento assinado eletronicamente por **ALEXANDER ROBERT KUTZKE, PROFESSOR DO MAGISTERIO SUPERIOR**, em 29/03/2021, às 20:29, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **RAFAELA MANTOVANI FONTANA, PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 29/03/2021, às 20:30, conforme art. 1º, III, "b", da Lei 11.419/2006.



Documento assinado eletronicamente por **SANDRAMARA SCANDELARI KUSANO DE PAULA SOARES, PROFESSOR ENSINO BAS/TEC/TECNOL**, em 29/03/2021, às 20:30, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **3407979** e o código CRC **35211D35**.

RESUMO

Este projeto tem como proposta a produção de um portfólio virtual para os profissionais dos setores de tecnologia da informação, artes visuais e produção musical. Para tanto, o portfólio consiste em um sistema *web* integrado com plataformas de divulgação de trabalhos nessas áreas, que possibilita a um profissional apresentar seus trabalhos de maneira personalizada e compartilhar seu portfólio como carta de apresentação de serviços. Além disso, permite que interessados em contratar esses profissionais realizem buscas baseadas em *Tags*, que especificam os tipos de serviços fornecidos pelo profissional, fazendo com que os interessados encontrem mais facilmente o serviço que mais se adequa às suas necessidades. O desenvolvimento do *back-end* foi realizado em linguagem Java, em conjunto com o banco de dados relacional PostgreSQL para guardar informações de usuários, o banco de dados em grafo Neo4j para gerenciar o sistema de *Tags* e o *framework* Spring Boot para configuração e publicação da aplicação. Quanto ao *front-end* foi utilizado o *framework* Angular em conjunto com as linguagens Html, Css e Javascript, além da biblioteca Bootstrap.

Palavras-Chave: Portfólio Website. Tecnologia da Informação. Artes visuais. Produção musical. Desenvolvimento de Sistemas Web.

ABSTRACT

The present report proposes to produce a virtual portfolio for professionals in the sectors of information technology, visual arts and music production. To this end, the portfolio consists of a web application system integrated with work exposition platforms of these areas, allowing professionals to present their work in a customized way, sharing their portfolio as a service presentation letter. In addition, it allows those who are interested in hiring these professionals to perform searches based on tags, which specify the types of services provided by the professional, making it easier to find the service that best suits their needs. The development of the back-end was performed in Java language, in conjunction with the PostgreSQL relational database to store user information, the Neo4j graph database platform to manage the tagging system and the Spring Boot for configuration and deploy of the application. As for the front-end, the Angular framework was used together with Html, Css and Javascript languages, along with the Bootstrap library.

Keywords: Portfolio Website. Information Technology. Visual Arts. Music production. Development of Web Systems.

LISTA DE QUADROS

QUADRO 1 - CARACTERÍSTICAS DE PLATAFORMAS DE SERVIÇOS GERAIS	20
QUADRO 2 - CARACTERÍSTICAS DE PLATAFORMAS DE TI.....	23
QUADRO 3 - CARACTERÍSTICAS DE PLATAFORMAS DE ARTES VISUAIS	25
QUADRO 4 - CARACTERÍSTICAS DE PLATAFORMAS DE PRODUÇÃO MUSICAL.....	27
QUADRO 5 - REQUISITOS FUNCIONAIS	34
QUADRO 6 - REQUISITOS NÃO FUNCIONAIS.....	35
QUADRO 7 - CRONOGRAMA DE ATIVIDADES	36

LISTA DE ILUSTRAÇÕES

FIGURA 1 - PÁGINA DE REPOSITÓRIOS DO GITHUB	22
FIGURA 2 - PÁGINA DE ALBÚNS DA PLATAFORMA FLICKR	24
FIGURA 3 - PÁGINA DE PERFIL DA PLATAFORMA FREESOUND.....	26
FIGURA 4 - DIAGRAMA DE CASO DE USO	30
FIGURA 5 - DIAGRAMA DE CLASSES	31
FIGURA 6 - DIAGRAMA FÍSICO DO BANCO DE DADOS RELACIONAL	32
FIGURA 7 - DIAGRAMA SEQUÊNCIA 001 – PESQUISAR PORTFÓLIOS	33
FIGURA 8 - ARQUITETURA DA APLICAÇÃO.....	43
FIGURA 9 - DIAGRAMA EXEMPLO ESTRUTURA EM GRAFOS.....	46
FIGURA 10 - GRAFOS GERADOS PELO NEO4J.....	47
FIGURA 11 - DATA VIEW 01: TELA DE PESQUISAR PORTFÓLIOS.....	52
FIGURA 12 - DATA VIEW 02: TELA DE APRESENTAR RESULTADOS	53
FIGURA 13 - DATA VIEW 03: TELA DE REALIZAR CADASTRO	54
FIGURA 14 - DATA VIEW 04: TELA DE REALIZAR LOGIN.....	54
FIGURA 15 - DATA VIEW 01: TELA DE PESQUISAR PORTFÓLIOS (USUÁRIO LOGADO) ..	55
FIGURA 16 - DATA VIEW 06: TELA DE MANTER USUÁRIO	55
FIGURA 17 - DATA VIEW 08: TELA DE MANTER PÁGINA.....	56
FIGURA 18 - DATA VIEW 12: TELA DE MANTER BIOGRAFIA DO USUÁRIO.....	57

SUMÁRIO

1	INTRODUÇÃO	10
1.1	TEMA E CONTEXTO	10
1.2	PROBLEMA.....	11
1.3	JUSTIFICATIVA	12
1.4	OBJETIVO GERAL.....	12
1.5	OBJETIVOS ESPECÍFICOS	12
2	REVISÃO LITERÁRIA	14
2.1	CARTA DE APRESENTAÇÃO	14
2.2	PORTFÓLIO.....	14
2.3	SISTEMA WEB.....	15
2.4	MECANISMOS DE BUSCA.....	15
2.5	GRAFOS E BANCO DE DADOS.....	16
2.6	TRABALHOS RELACIONADOS	18
2.7	ANÁLISE COMPARATIVA DE SERVIÇOS EXISTENTES.....	18
2.7.1	Plataformas de serviços gerais.....	19
2.7.2	Plataformas de tecnologia da informação.....	21
2.7.3	Plataformas de artes visuais.....	23
2.7.4	Plataformas de produção musical	25
2.7.5	Resumo da comparação com serviços existentes.....	28
3	METODOLOGIA	29
3.1	SOLUÇÃO PROPOSTA	29
3.2	FUNCIONAMENTO DO SISTEMA.....	30
3.3	LEVANTAMENTO DOS REQUISITOS FUNCIONAIS.....	34

3.4	LEVANTAMENTO DOS REQUISITOS NÃO FUNCIONAIS.....	35
3.5	PROCESSO DE TRABALHO	35
3.6	MODELO DE NEGÓCIO	37
3.7	TECNOLOGIAS UTILIZADAS	38
3.7.1	<i>VSCode (Visual Studio Code)</i>	38
3.7.2	<i>Angular</i>	38
3.7.3	<i>Java</i>	39
3.7.4	<i>HTML (HyperText Markup Language)</i>	39
3.7.5	<i>CSS (Cascading StyleSheets)</i>	39
3.7.6	<i>JavaScript</i>	40
3.7.7	<i>Bootstrap</i>	40
3.7.8	<i>Spring Boot</i>	40
3.7.9	<i>Hibernate</i>	41
3.7.10	<i>PostgreSQL</i>	41
3.7.11	<i>Neo4j</i>	41
3.7.12	<i>GitHub</i>	41
3.7.13	<i>TypeScript</i>	42
4	IMPLEMENTAÇÃO	43
4.1	BACK-END	43
4.1.1	Sistema de Usuários.....	44
4.1.2	Sistema de Login.....	44
4.1.3	Sistema de Gerenciamento de Componentes	45
4.1.4	Banco de Dados Relacional	45
4.1.5	Banco de Dados em Grafos	45
4.1.6	Geração do grafo de <i>Tags</i>	46

4.1.7	Motor de Busca por <i>Tags</i>	47
4.2	FRONT-END	48
4.2.1	Tela de apresentação.....	48
4.2.2	Tela de personalização de portfólios	49
4.2.3	Tela de <i>preview</i> de portfólio.....	51
5	APRESENTAÇÃO DO SISTEMA	52
6	CONSIDERAÇÕES FINAIS	58
	REFERÊNCIAS	59
	APÊNDICE A – DIAGRAMA DE ENTIDADE RELACIONAMENTO	61
	APÊNDICE B – DIAGRAMA DE SEQUÊNCIA	62
	APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO	79
	APÊNDICE D – PROTÓTIPOS DAS TELAS	103
	APÊNDICE E – ENTRY POINTS DA API.....	108
	APÊNDICE F – DIAGRAMA DE GANTT	119

1 INTRODUÇÃO

Neste capítulo é apresentada a introdução ao projeto, contemplando o contexto em que o sistema está inserido, abordando o problema a ser solucionado, a justificativa e os objetivos do presente trabalho.

1.1 TEMA E CONTEXTO

Há várias plataformas disponíveis no mercado que fornecem a seus usuários uma carta de apresentação profissional, por meio da exposição de valores como conhecimentos, habilidades e experiências. Dessa forma, os profissionais podem divulgar um currículo online com suas competências e os interessados têm condições de contatá-los.

Nos últimos anos, percebe-se o crescimento da popularidade dessas plataformas, na forma de redes sociais por meio das quais os usuários podem criar perfis, conhecer novas pessoas, compartilhar conteúdos e expor suas opiniões. Os serviços de redes sociais permitem a criação de conteúdo aberto para comentários e avaliações, além disso, são responsáveis por grandes movimentações nas áreas de vendas, marketing e publicidade (CHANG, LIU e SHEN, 2016).

Uma rede social que foi criada com objetivo de conectar profissionais em todo o mundo, permitindo que usuários construam páginas de apresentação profissional e que vagas de empregos sejam divulgadas é a *LinkedIn*. Ela foi criada em 2003 e hoje é a maior rede social profissional contando com mais de 675 milhões de usuários (LinkedIn, 2020).

A rede social *Fiverr* tem um público-alvo diferente, focando principalmente no mercado de serviços criativos. Dessa maneira, ela busca atrair profissionais de áreas como: design gráfico, escrita e tradução, animação de vídeos, produção sonora, programação e tecnologia da informação, marketing e propaganda. Ela foi lançada em 2010 e logo foi adotada por trabalhadores autônomos, e obteve no ano de 2019 receita igual a 107,1 milhões de dólares (Fiverr, 2020).

Na área de tecnologia da informação, pode-se citar a rede GitHub que cria um ambiente cooperativo onde os usuários podem aprender e compartilhar conhecimentos para o desenvolvimento de aplicações. Em seu último balanço de agosto de 2019, a rede apresentava 50 milhões de usuários e 100 milhões de repositórios (GitHub, 2020).

Seguindo a proposta do GitHub, outra rede voltada para a Tecnologia da Informação é a GitLab, criada com o intuito de promover a cooperação entre público e organizações no desenvolvimento de projetos. Atualmente, ela conta com 40 milhões de usuários, é utilizada por 100 mil organizações e tem mais de 3 mil colaboradores no desenvolvimento de código (GitLab, 2020).

Na área de design visual, destaca-se a rede DevianArt que possibilita a livre expressão de artistas e a criação e compartilhamento de cultura. A plataforma, que tem mais de 48 milhões de membros e registra o tráfego de mais de 45 milhões de visitantes por mês, permite aos membros publicar trabalhos como: pinturas, esculturas, artes digitais, filmes e animes (DevianArt, 2020).

Já na área de produção musical, pode-se destacar a rede SoundCloud, que é voltada para artistas que buscam compartilhar e comercializar suas músicas. A rede teve suas atividades iniciadas em 2008, e hoje é a maior plataforma de áudio e música do mundo (SoundCloud, 2020).

1.2 PROBLEMA

O problema do projeto aborda a capacidade das plataformas mencionadas ter como público-alvo profissionais de apenas uma área específica, além de proporcionar baixa personalização de página.

O projeto busca abordar essa problemática, facilitando a divulgação de experiências profissionais e trabalhos dos profissionais das áreas de tecnologia da informação, artes visuais e produção musical com a integração a essas plataformas disponíveis, além de proporcionar um ambiente com maiores possibilidades de personalização de páginas ao usuário.

1.3 JUSTIFICATIVA

Por meio da análise das plataformas de apresentação profissional disponíveis no mercado, percebe-se que não há uma rede que seja focada em portfólio e que permita ampla personalização da página, contato com interessados e forneça ao usuário a busca pelo serviço mais adequado às suas necessidades.

O projeto permite ao usuário a criação de uma carta de apresentação de maneira simples, rápida e intuitiva, possibilitando a exposição de seu portfólio e que seja encontrado por interessados. Além disso, a plataforma viabiliza a integração de profissionais das áreas de tecnologia da informação, produção visual e musical, em um ambiente favorável à personalização de páginas de apresentação.

1.4 OBJETIVO GERAL

Desenvolver uma plataforma de apresentação profissional que possibilite a criação de uma carta de apresentação focada em portfólio com possibilidade de ampla personalização da página, e que permita que um interessado realize buscas pelo serviço mais adequado às suas necessidades.

1.5 OBJETIVOS ESPECÍFICOS

1. Realizar o levantamento dos requisitos dos usuários, identificando os casos de uso essenciais para elaboração da estrutura do sistema.
2. Realizar pesquisa das plataformas de apresentação profissional que estão disponíveis no mercado, identificando características, pontos fracos e fortes.
3. Realizar a modelagem do sistema, esclarecendo os detalhes da estrutura e do comportamento do sistema por meio de diagramas.
4. Revisar o modelo proposto, verificando se todos os requisitos de usuários foram contemplados.

5. Desenvolver a plataforma de apresentação profissional, aplicando técnicas e linguagens de programação.

2 REVISÃO LITERÁRIA

Neste capítulo são apresentados os referenciais teóricos utilizados como base para estruturar o projeto, abordando trabalhos relacionados ao assunto e uma pesquisa comparativa acerca das plataformas de portfólio profissional disponíveis no mercado.

2.1 CARTA DE APRESENTAÇÃO

Uma carta de apresentação tem como objetivo apresentar o perfil de um profissional de maneira sucinta, destacando qualidades profissionais e pessoais para chamar a atenção de clientes ou recrutadores (WATTHIER, 2016).

Uma carta se constitui em um gênero textual com a função de propiciar a interação entre duas ou mais pessoas que se encontram em espaços diferentes. Uma carta pode ser escrita com diferentes intenções, valendo ressaltar as comerciais, voltadas para anúncios e vagas de empregos, e as publicitárias, relacionadas a apresentação de trabalhos e publicações (WATTHIER, 2016).

Por meio das cartas de apresentação, um candidato fornece suas informações, comumente, em resposta a um anúncio de emprego ou oportunidade. No âmbito de seleção empresarial de candidatos, a carta de apresentação é o documento pelo qual um candidato busca “vender sua imagem” a um selecionador (CONTO, 2009).

2.2 PORTFÓLIO

O conceito de portfólio, segundo Alvarenga e Araújo (2006, apud WATERMAN, 1991) consiste em um documento que contém uma coleção de trabalhos, que evidenciam o desenvolvimento, as competências e as habilidades do autor do trabalho.

Além disso, define-se que as informações que compõe um portfólio devem representar os esforços do indivíduo numa área de estudo determinada, demonstrando sua integração e aplicação no desenvolvimento de trabalhos. Da mesma forma, os

portfólios apresentam a evolução do indivíduo em um determinado assunto, servindo de base para a avaliação contínua de seu progresso (ALVARENGA e ARAÚJO, 2006).

2.3 SISTEMA WEB

O desenvolvimento de sistemas Web pode ser dividido em três etapas: identificação do objetivo geral da aplicação, criação de um modelo de análise detalhando os requisitos e a construção de páginas Web, utilizando tecnologias como HTML, CSS e Javascript (MILETTO e BERTAGNOLLI, 2014).

Uma linguagem presente no ambiente Web é o Javascript, que é responsável por especificar o comportamento dos elementos em uma página Web. É uma linguagem de alto nível, dinâmica, interpretada e não tipada, amplamente utilizada no desenvolvimento de sites e conveniente para estilos de programação orientados a objetos e funcionais (FLANAGAN, 2013).

O desenvolvimento das páginas Web foi realizado utilizando Angular, um framework Javascript, que facilita o desenvolvimento e implementação de páginas web, garantindo design e estrutura da página e da aplicação em acordo com os padrões MVC (Model View Controller) e MVVM (Model View ViewModel).

O Angular controla como os elementos são dispostos no navegador do cliente, além disso, traz benefícios como: mecanismo de escopo que permite fácil acoplagem de dados para HTML, extensibilidade que permite a customização conforme necessidade, promove o desenvolvimento de uma codificação lógica e limpa, além de favorecer a reutilização de código (DAYLEY et al., 2018).

2.4 MECANISMOS DE BUSCA

Mecanismos de busca ou pesquisadores são sites ou ferramentas especializadas em localizar informações. Seu funcionamento se dá inicialmente quando um usuário digita um termo desejado e solicita que a busca seja realizada, na sequência esses

mecanismos buscam este termo em seus bancos de dados e apresentam os resultados encontrados que contém os termos buscados (BRANSKI, 2000).

De acordo com Branski (2000), uma das categorias em que um mecanismo de busca pode ser enquadrado é a de catálogo por assunto. Nessa categoria, as informações são organizadas e classificadas em categorias temáticas com a ajuda de pessoas. Assim, o catálogo limita-se a verificar a ocorrência do termo pesquisado na descrição enviada pelo autor.

Um conceito que vai de encontro a organização por assunto é o de *Tag*, ou etiqueta. As *Tags* permitem categorizar documentos por assunto, especificando a qual tema esse documento mais se encaixa. Por meio delas, o mecanismo de busca pode ter seu desempenho melhorado já que buscará por temas especificados dentro das *Tags* registradas (BISCHOFF, FIRAN, NEJDL, PAIU, 2008).

As *Tags* podem especificar diversas características de um documento, porém é importante se atentar ao processo de categorização de acordo com a necessidade do negócio, por exemplo, em uma plataforma voltada para música a *Tag* de “Gênero Musical” seria a mais relevante. Assim, pode-se perceber que é necessário um gerenciamento das *Tags* que estarão disponíveis para que não sejam criadas etiquetas subjetivas, que sejam baseadas em opinião ou que não tenham relevância para a finalidade do sistema (BISCHOFF, FIRAN, NEJDL, PAIU, 2008). Assim, pode-se utilizar as *Tags* para classificar e organizar, por exemplo, profissionais por áreas de atuação ou por conhecimentos adquiridos, e, além disso, favorecer os mecanismos de busca com a redução de custos computacionais. E para que o mecanismo realize as buscas por essas *Tags*, optou-se por implementar um sistema baseado na teoria de grafos e na teoria do menor caminho.

2.5 GRAFOS E BANCO DE DADOS

Um grafo é composto por um conjunto de vértices, um conjunto de arestas e uma função de incidência que associa a cada aresta um par não ordenado de vértices, chamados de extremos. Os grafos podem ser representados por diagramas, onde cada

vértice é um ponto e uma aresta é uma linha ligando os pontos, que representam os seus extremos (LUCCHESI, 1979).

Segundo Davis Jr. (1997), a teoria do menor caminho se pauta no algoritmo que determina qual é o menor caminho entre dois vértices dentro de uma rede. Assim, o problema reside em encontrar o percurso de menor custo entre dois nós da rede, considerando a soma dos custos associados às arestas percorridas.

Por outro lado, esse mecanismo de busca deve estar associado a um banco de dados que seja compatível a estrutura de grafos. Segundo Elmasri e Navathe (2005), um banco de dados é uma coleção de dados relacionados de maneira lógica e coerente que é projetada para atender um propósito específico. Esses dados representam fatos, possuem um significado e podem ser gravados, como, por exemplo: nomes e números, e assim que uma coleção de dados passa a representar informações sobre um tema específico é que se tem um banco de dados.

Como suporte aos sistemas que dão importância a interconectividade de dados, surgiu o banco de dados em grafos como uma alternativa ao tradicional banco de dados relacional. Em um sistema de banco de dados em grafos, os dados são modelados por meio de vértices e arestas, facilitando a modelagem de contextos complexos e definindo as relações existentes entre as entidades de uma base (PENTEADO et al., 2014).

O Sistema Gerenciador de Banco de Dados em Grafo é o meio pelo qual a coleção de dados modelada logicamente e mapeada para uma forma de armazenamento físico é manipulada. A modelagem lógica é feita pelo modelo de grafos, no qual a topologia de um grafo é uma função de um conjunto de vértices e de um conjunto de arestas (PENTEADO et al., 2014).

Segundo Penteado (2014), os sistemas que utilizam listas de adjacências para armazenar grafos e executam algoritmos de exploração de grafos em suas consultas são considerados sistemas de gerenciamento de banco de dados em grafos nativos. Um destes sistemas é o Neo4j, lançado em fevereiro de 2010 pela empresa *Neo Technology* e com implementação feita em Java, que é transacional e oferece dois tipos de arquitetura, a centralizada e a distribuída com suporte à replicação.

Além disso, Penteado (2014) expõe que o Neo4j permite que o armazenamento físico possa se dar tanto em memória quanto em disco, seu modelo físico é baseado em

repositórios chave-valor, e, também, possui uma linguagem de consulta proprietária para manipulação (carregamento e consulta) dos dados, a Cypher.

2.6 TRABALHOS RELACIONADOS

Primeiramente, destaca-se o sucesso das redes sociais na internet e sua capacidade de ser uma ferramenta para contato e relacionamento interpessoal, como expõe Ciribeli e Paiva (2011). Além disso, essas ferramentas permitem a exposição de opiniões, sentimentos e a liberdade de expressão dos conteúdos postados.

Ainda segundo Ciribeli e Paiva (2011), a ligação entre uma rede social e o mercado de trabalho é cada vez mais explorada por recrutadores e empresas, que buscam conhecer todos os detalhes do profissional. Uma rede especializada no meio comercial é o LinkedIn, que permite a criação de currículo detalhado, recomendação de antigos empregadores e indicações de colegas.

Redes sociais que não são voltadas completamente para o lado profissional, como o Facebook por exemplo, podem ser utilizadas como meios de divulgação de trabalhos. Além disso, de acordo com Ciribeli e Paiva (2011, apud COSTA, 2008), essas ferramentas podem ser utilizadas para ofertar vagas de emprego, e as redes especializadas dão maior confiabilidade às informações do profissional e permitem fazer buscas por conhecimentos específicos.

Segundo Fernández-Pérez et al. (2015), tanto as redes sociais pessoais quanto as profissionais influenciam os universitários a atuar de maneira empreendedora. Por meio dessas redes, universitários que são especialistas em algum assunto, porém apresentam dificuldades ou negligenciam o contexto social, são capazes de desenvolver atitudes de relacionamento social para aproveitar oportunidades de negócios.

2.7 ANÁLISE COMPARATIVA DE SERVIÇOS EXISTENTES

Para conhecer as tecnologias disponíveis no mercado, foi elaborada uma pesquisa para identificar as características das principais plataformas de portfólio

profissional e exposição de trabalhos, de acordo com sua área de atuação. O levantamento identificou as principais plataformas que oferecem esse serviço aos profissionais e suas principais características.

Pesquisou-se questões relativas a preço e custos, existência de anúncios ou propagandas, limite de publicações por perfil, a possibilidade de realizar transações comerciais, a integração com redes sociais, a opção de customização da página de apresentação e a presença de Chats que permitem a interação do usuário com o profissional. Além disso, realizou-se uma avaliação da interface apresentada ao usuário em uma escala de 1 a 10, na qual 1 se refere a uma interface nada amigável ao usuário e 10 a uma interface totalmente amigável e intuitiva.

2.7.1 Plataformas de serviços gerais

As plataformas nessa categoria permitem aos profissionais de qualquer área oferecer serviços, por meio da criação de uma página de apresentação profissional na qual podem descrever seu perfil, expondo características, conhecimentos e habilidades, listar certificados e compartilhar experiências anteriores. Nessa seção foram analisadas as seguintes plataformas: Fivver, Monster, LinkedIn, GetNinjas e PortfolioBox.

A plataforma Fivver proporciona um ambiente em que um profissional pode criar um perfil listando suas habilidades em *Tags* e oferecendo serviços, os quais podem ser avaliados pelos clientes. O modelo de negócio é voltado para profissionais autônomos, ou *Freelancer*, e pode ser acessado pelo endereço: <https://br.fiverr.com/>.

A rede Monster é mais voltada para a construção de carreira de um profissional, fornecendo ferramentas que auxiliam na colocação no mercado, como fornecendo modelos de currículos, cartas de referência e guias sobre como procurar por empregos. A plataforma conta com áreas separadas para profissionais e para recrutadores, e pode ser acessado em: <https://www.monster.com/>.

A LinkedIn é uma rede social voltada para negócios, que permite com que um profissional crie um currículo e o divulgue para diversas empresas cadastradas na rede. A rede permite com que um profissional divulgue suas produções, experiências e habilidades, e pode ser acessada em: <https://www.linkedin.com/>.

A GetNinjas é uma plataforma que tem como modelo de negócios a prestação de serviços, na qual os usuários podem realizar buscas por serviços e como resposta são sugeridos profissionais cadastrados na rede, avaliados em nível de satisfação pelos usuários. A rede permite com que os usuários exponham sua opinião acerca dos serviços prestados na forma de comentários, e pode ser acessada pelo endereço: <https://www.getninjas.com.br/>.

A PorfolioBox é uma rede mais focada em Portfólio, na qual um usuário pode expor seus trabalhos por meio de imagens e permite maior personalização de página. A plataforma conta com modelos de páginas de currículo, galerias de imagens e produção musical, podendo ser acessada em: <https://www.portfoliobox.net/>.

Além disso, os usuários podem entrar em contato com o profissional para combinar detalhes do serviço, avaliar, recomendar e expor suas opiniões sobre o profissional, o que auxilia a tomada de decisão de outros usuários na escolha do prestador. O Quadro 1 apresenta as características que foram levantadas sobre as principais plataformas de serviços gerais.

QUADRO 1 - CARACTERÍSTICAS DE PLATAFORMAS DE SERVIÇOS GERAIS

Característica	Serviços Gerais				
	Fiverr	Monster	Linkedin	GetNinjas	PortfolioBox
Preço / Custo:	20 % das vendas	Gratuito	Gratuito (assinatura opcional)	Conforme demanda	Gratuito
Contém anúncios ou propaganda	Não	Não	Sim	Não	Não
Limite de publicações free	Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Limite de publicações pago	Ilimitado	Ilimitado	Ilimitado	Ilimitado	Ilimitado
Permite transações comerciais	Sim	Não	Sim	Sim	Sim
Possui integração com rede social	Não	Não	Sim	Não	Não
Permite customização da página pessoal	Baixa	Nenhuma	Baixa	Baixa	Alta
Permite customização do portfólio	Baixa	Nenhuma	Nenhuma	Baixa	Alta

FONTE: OS AUTORES (2021).

QUADRO 1 - CARACTERÍSTICAS DE PLATAFORMAS DE SERVIÇOS GERAIS

(Conclusão)

Possui Chat com o usuário	Sim	Não	Sim	Sim	Sim
UI/UIX - Avaliação	9	6	7	7	9

FONTE: OS AUTORES (2021).

2.7.2 Plataformas de tecnologia da informação

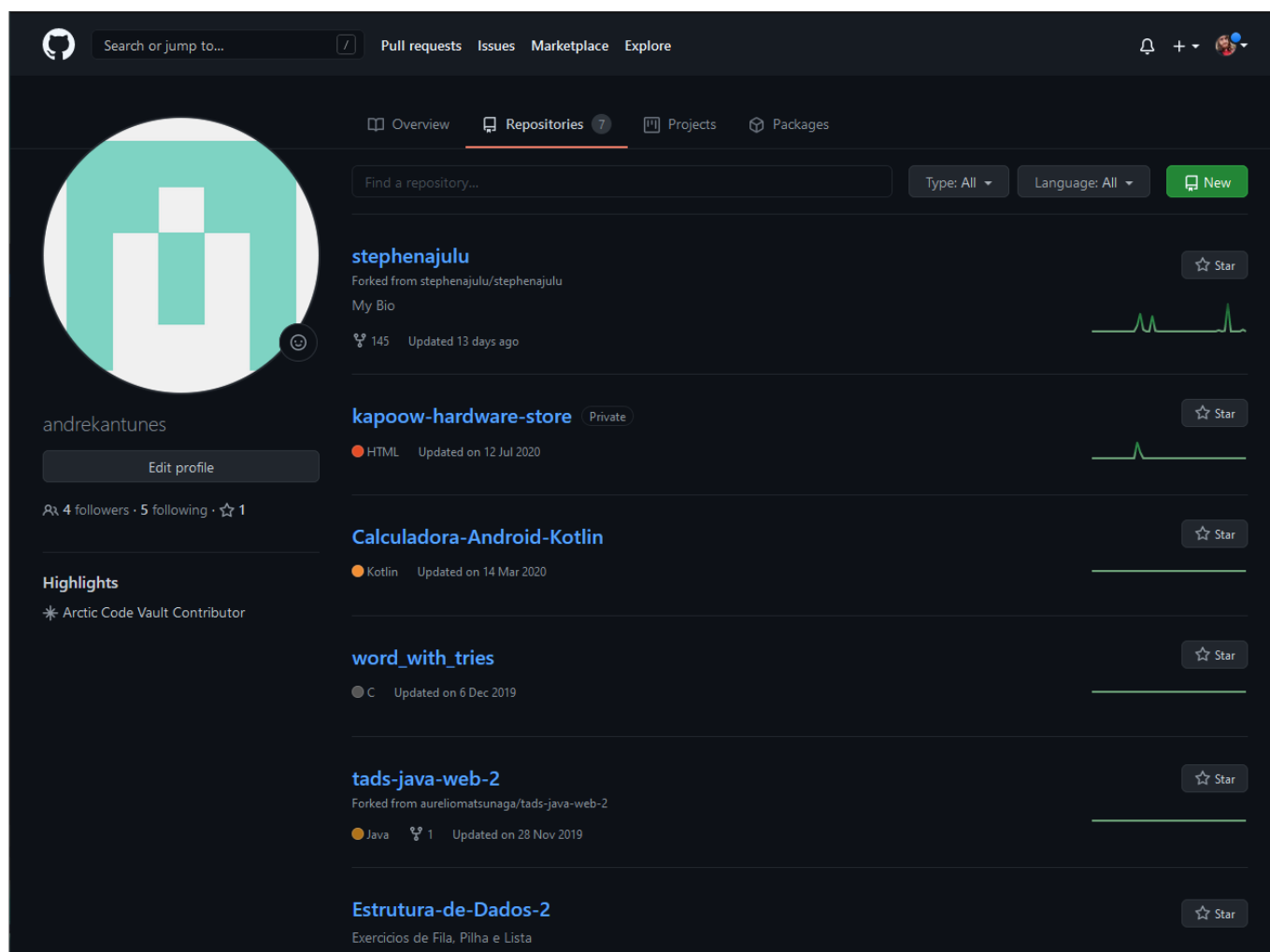
Por meio dessas plataformas, um profissional da área de tecnologia da informação pode, além de construir uma página de apresentação com exposição de seu currículo, expor trabalhos na forma de códigos. Dessa maneira, pode-se identificar o nível de conhecimento de um programador em algoritmos e o domínio de uma linguagem de programação. Na Figura 1, é apresentada a tela de repositórios da plataforma GitHub, na qual um profissional dessa área pode expor e compartilhar seus trabalhos. Nessa seção foram analisadas as seguintes plataformas: GitHub, GitLab e Stackblitz.

A plataforma GitHub permite com que profissionais criem repositórios de seus trabalhos, que podem ser compartilhados entre outros usuários, permitindo com que visualizem o conteúdo ou atuem como colaboradores do projeto. A rede permite com que o desenvolvimento de sistemas seja compartilhado por uma equipe, configurando um sistema de controle de versionamento, e pode ser acessado no seguinte endereço: <https://github.com/>.

A rede GitLab, assim como a GitHub, é uma plataforma que permite a hospedagem de código-fonte. De mesma forma, permite o controle de versão, métricas para acompanhamento de qualidade de código, testes de usabilidade e pode ser acessada em: <https://gitlab.com>.

Por fim, a Stackblitz é uma rede que permite o desenvolvimento de sistemas desde sua criação, fornecendo um ambiente de criação, edição e implantação online. A plataforma tem como foco o desenvolvimento web e pode ser encontrada no seguinte endereço: <https://stackblitz.com/>.

FIGURA 1 - PÁGINA DE REPOSITÓRIOS DO GITHUB



FONTE: GITHUB (www.github.com, 2021).

As plataformas dessa área permitem, além do contato com o profissional, o compartilhamento de arquivos, de trechos de códigos e de soluções em diversas linguagens. Nela, um prestador de serviço pode ser avaliado pelo número de interações com os usuários, número de vezes que seus códigos foram referenciados e quantidade de projetos que atuou como colaborador. O Quadro 2 apresenta as características que foram levantadas sobre as principais plataformas de TI.

QUADRO 2 - CARACTERÍSTICAS DE PLATAFORMAS DE TI

Característica	Tecnologia da Informação		
	GitHub	GitLab	Stackblitz
Preço / Custo:	Gratuito	Gratuito	Gratuito
Contém anúncios ou propaganda	Não	Não	Não
Limite de publicações free	Ilimitado	Ilimitado	Ilimitado
Limite de publicações pago	Ilimitado	Ilimitado	Ilimitado
Permite transações comerciais	Não	Não	Não
Possui integração com rede social	Sim	Sim	Sim
Permite customização da página pessoal	Baixa	Baixa	Baixa
Permite customização do portfólio	Baixa	Baixa	Baixa
Possui Chat com o usuário	Não	Não	Não
UI/UIX - Avaliação	7	8	6

FONTE: OS AUTORES (2021).

2.7.3 Plataformas de artes visuais

As plataformas voltadas para artes visuais permitem aos profissionais apresentar brevemente uma carta de apresentação e expor seus trabalhos na forma de uma coleção de imagens. Assim, pode-se identificar a afinidade do profissional com estilos de desenho, arranjo de cores e habilidades com ferramentas de criação visual. Na Figura 2, é apresentada a tela de álbuns da plataforma Flickr, na qual imagens ou fotografias podem ser expostas e compartilhadas. Nessa seção foram analisadas as seguintes plataformas: Behance, DevianArt e Pinterest.

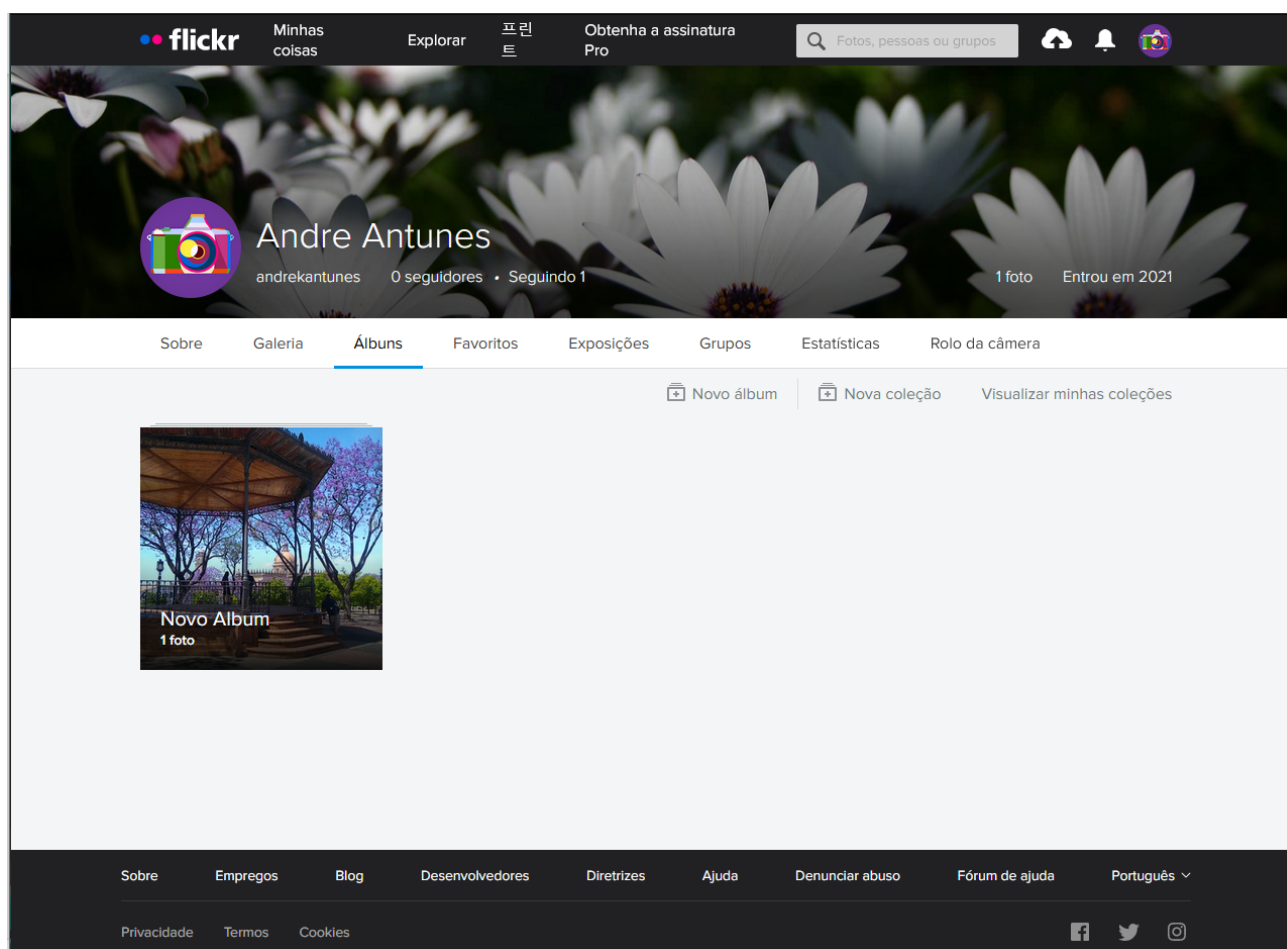
A Behance permite a um profissional que divulgue seus trabalhos por meio de um Portfólio, focado na divulgação de imagens e vídeos, permitindo a categorização de

serviços por *Tags*. Além disso, a rede permite com que cada um dos trabalhos expostos em um perfil seja avaliado, e pode ser encontrada em: <https://www.behance.net/>.

A DevianArt permite a um usuário a criação de uma página de perfil contendo uma breve biografia, uma seção de comentários sobre o profissional e seus trabalhos e uma galeria na qual podem ser expostas suas produções visuais. Além disso, permite com que suas produções sejam comercializadas no próprio site, podendo ser acessada em: <https://www.deviantart.com/>.

A Pinterest é uma rede voltada para o compartilhamento de imagens, na qual usuários podem classificar suas imagens em categorias, arquivando-as em forma de uma galeria. A plataforma pode ser acessada em: <https://br.pinterest.com/>.

FIGURA 2 - PÁGINA DE ALBÚNS DA PLATAFORMA FLICKR



FONTE: FLICKR (www.flickr.com, 2021).

Além de permitir o contato entre profissionais e usuários por meio de uma ferramenta de chat, algumas plataformas permitem a comercialização de seus produtos por meio de transações comerciais no próprio Website.

O Quadro 3 apresenta as características que foram levantadas sobre as principais plataformas de artes visuais.

QUADRO 3 - CARACTERÍSTICAS DE PLATAFORMAS DE ARTES VISUAIS

Característica	Visual		
	Behance	DevianArt	Pinterest
Preço / Custo:	Gratuito	Gratuito	Gratuito
Contém anúncios ou propaganda	Não	Não	Não
Limite de publicações free	Ilimitado	Ilimitado	2.000 Boards e 200.000 Pins
Limite de publicações pago	Ilimitado	Ilimitado	2.000 Boards e 200.000 Pins
Permite transações comerciais	Sim	Sim	Não
Possui integração com rede social	Sim	Sim	Sim
Permite customização da página pessoal	Baixa	Média	Baixa
Permite customização do portfólio	Média	Média	Média
Possui Chat com o usuário	Sim	Sim	Sim
UI/UIX - Avaliação	8	8	5

FONTE: OS AUTORES (2021).

2.7.4 Plataformas de produção musical

As plataformas voltadas para os profissionais de produção musical permitem com que projetos sejam expostos em forma de arquivos de áudio, como músicas, trilhas ou efeitos sonoros. Na Figura 3, é apresentada a tela de perfil da plataforma FreeSound, na

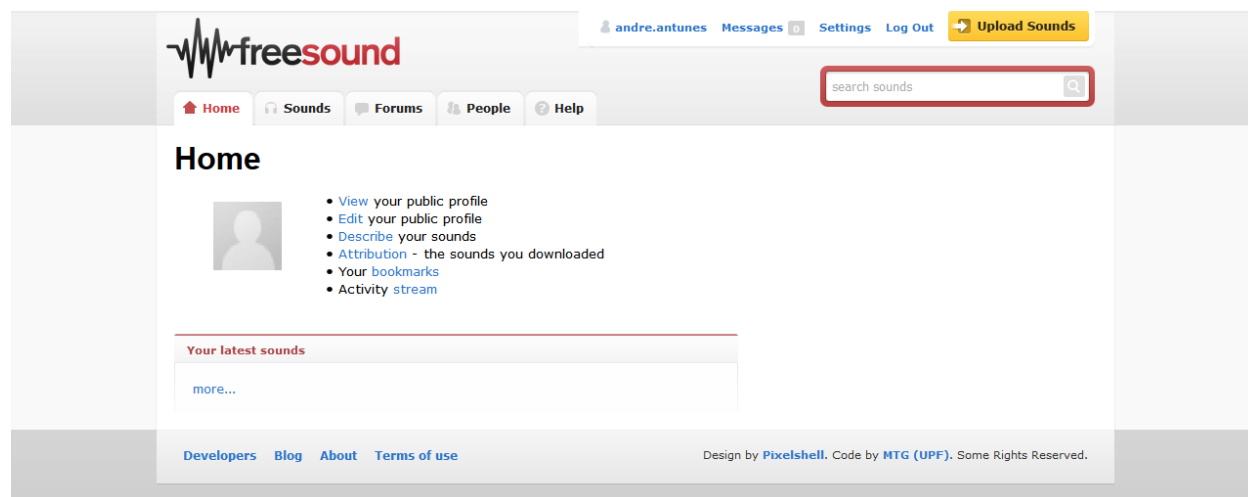
qual ficam listadas as produções sonoras do usuário. Nessa seção foram analisadas as seguintes plataformas: BandCamp, FreeSound e SoundCloud.

A BandCamp é uma Plataforma que permite com que profissionais compartilhem e comercializem suas produções musicais. A rede permite categorização das produções sonoras por estilo musical e formato de mídia, podendo ser acessada no seguinte endereço: <https://bandcamp.com/>.

A FreeSound é uma rede mais focada em produções de efeitos sonoros, que são elencadas em forma de lista em uma página de perfil de um usuário. A plataforma também permite na página de perfil, uma breve descrição do profissional mídia e uma seção de comentários com opiniões dos interessados, podendo ser acessada em: <https://freesound.org/>.

A SoundCloud, assim como a BandCamp, é uma plataforma voltada para o compartilhamento de produções musicais. A rede permite com que um usuário crie um perfil, escolha *Tags* que definam seu estilo musical e receba comentários de outros usuários. A plataforma pode ser acessada em: <https://soundcloud.com/>.

FIGURA 3 - PÁGINA DE PERFIL DA PLATAFORMA FREESOUND



FONTE: FreeSound (www.freesound.org, 2021).

Elas também permitem o contato com o profissional por meio de Chat no próprio Website, e, além disso, algumas permitem que um cliente possa ouvir um trecho de áudio

da produção e se desejar efetuar uma transação comercial no próprio Website. O Quadro 4 apresenta as características que foram levantadas sobre as principais plataformas de serviços relacionados a produção e de conteúdo musical e/ou sonoro.

QUADRO 4 - CARACTERÍSTICAS DE PLATAFORMAS DE PRODUÇÃO MUSICAL

Característica	Aúdio		
	Soundcloud	Bandcamp	Freesound
Preço / Custo:	Opcional	Opcional	Gratuito
Contém anúncios ou propaganda	Não	Não	Não
Limite de publicações free	3 horas	Ilimitado.	Ilimitado
Limite de publicações pago	Ilimitado	Ilimitado	Ilimitado
Permite transações comerciais	Não	Sim	Não
Possui integração com rede social	Sim	Parcial	Parcial
Permite customização da página pessoal	Baixa	Média	Baixa
Permite customização do portfólio	Alta	Média	Média
Possui Chat com o usuário	Sim	Sim	Sim
UI/UIX - Avaliação	7	8	6

FONTE: OS AUTORES (2021).

Através dos quadros apresentados, percebe-se que nos diversos segmentos de plataformas de portfólio de apresentação profissional, as páginas pessoais e as de exposição de portfólio permitem de baixa a média personalização, além de não apresentar funcionalidades intuitivas conforme avaliação da experiência de usuário (UI/UIX).

2.7.5 Resumo da comparação com serviços existentes

A partir dos quadros de recém apresentados, segue uma sumarização do resultado da análise comparativa da seguinte forma:

Categorias:

- Serviços Gerais;
- Tecnologia da Informação;
- Artes Visuais;
- Produção Musical;

Aspectos:

- Custos e limites de publicação;
- Comunicação com usuário (chat);
- Personalização de página e portfólio;
- Avaliação da interface e experiência de usuário

Pontos Fracos:

- Interfaces complexas;
- Pouca ênfase em portfólio;
- Baixo nível de personalização;
- Custos para utilização;
- Limitação de postagens;

3 METODOLOGIA

Neste capítulo é apresentada a solução proposta ao problema, assim como o levantamento dos requisitos, o modelo de negócio e as tecnologias utilizadas.

3.1 SOLUÇÃO PROPOSTA

O projeto tem o intuito de criar uma aplicação web que permita a um profissional dos setores de tecnologia da informação, artes visuais ou produção musical criar uma carta de apresentação, na qual possa apresentar brevemente suas competências, e um portfólio para expor seus projetos de maior relevância.

A aplicação Web foi desenvolvida utilizando o framework Angular que permite manipulação fácil de elementos dentro de uma aplicação em uma página, o que busca reduzir a complexidade da criação dinâmica de elementos quando o usuário desejar adicionar ou remover componentes de sua página.

Para isso, foram utilizadas a linguagem de marcação HTML (Hyper Text Markup Language) na versão 5 para criar a estrutura da página e a linguagem CSS (Cascading Style Sheets) na versão 3 para adicionar estilos a estrutura de uma página. Ambas as linguagens são intrínsecas à versão do Framework Angular utilizado para o desenvolvimento.

A aplicação tem integração com plataformas de portfólio de tecnologia da informação, artes visuais e produção musical, o que permite ao usuário criar componentes que expõem o seu portfólio de outras plataformas. Assim, por exemplo, um profissional poderá incluir em sua página seus trabalhos de artes visuais da plataforma Flickr, de tecnologia de informação da plataforma GitHub e de produção musical da plataforma Freesound. Por meio dessa integração, um profissional terá condições de unir seus trabalhos em uma só página de apresentação de portfólio.

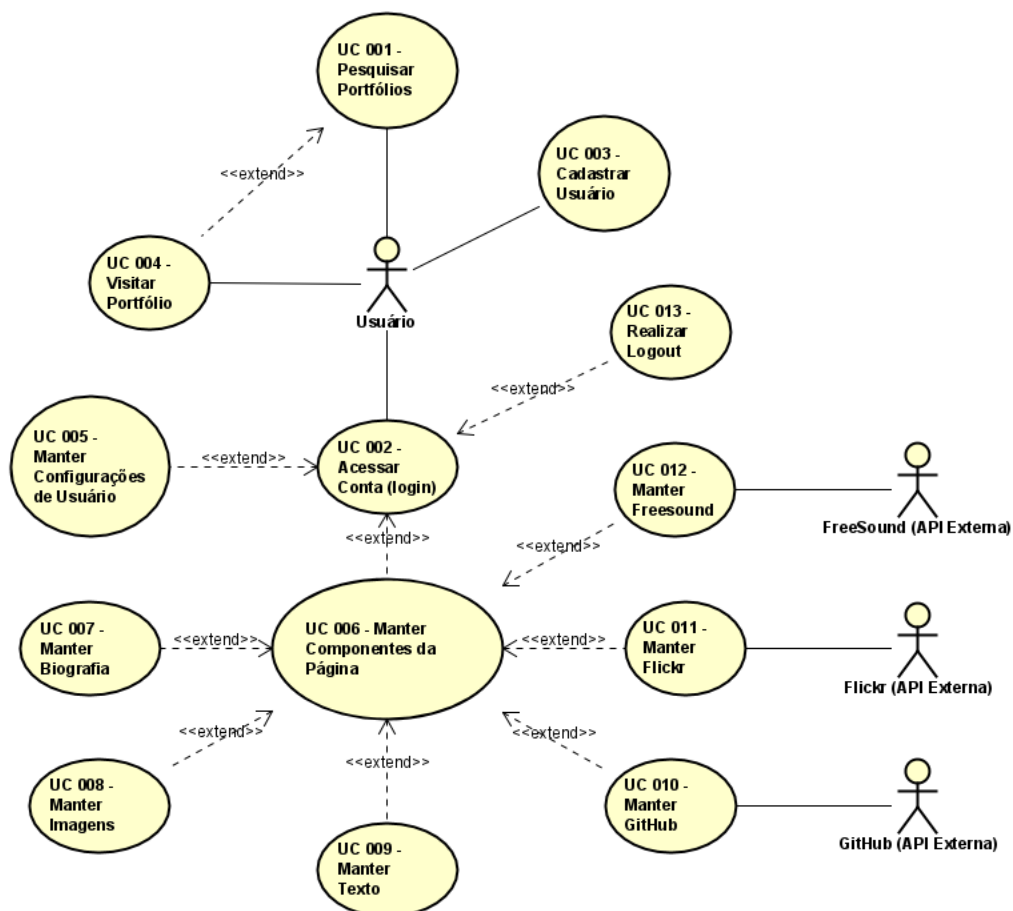
Ao criar uma página de apresentação, o profissional tem que escolher *Tags* que categorizam os seus serviços, buscando identificar quais são os tipos de serviços prestados. Essa categorização de serviço será armazenada em um banco de dados em grafos, que permite ao mecanismo de busca a busca por *Tags* que mais se aproximam

do termo de busca desejado pelo usuário, e não apenas aquelas que são sintaticamente similares, ampliando assim as possibilidades de resultados da busca.

3.2 FUNCIONAMENTO DO SISTEMA

A aplicação permite a criação de uma página de apresentação a um usuário cadastrado, que pode personalizá-la por meio da adição de componentes de sua preferência e possibilita que qualquer usuário (logado ou não) realize uma pesquisa por profissionais de uma área de atuação específica, por meio das *Tags*. Os casos de uso do sistema ficam apresentados conforme mostrado na Figura 4:

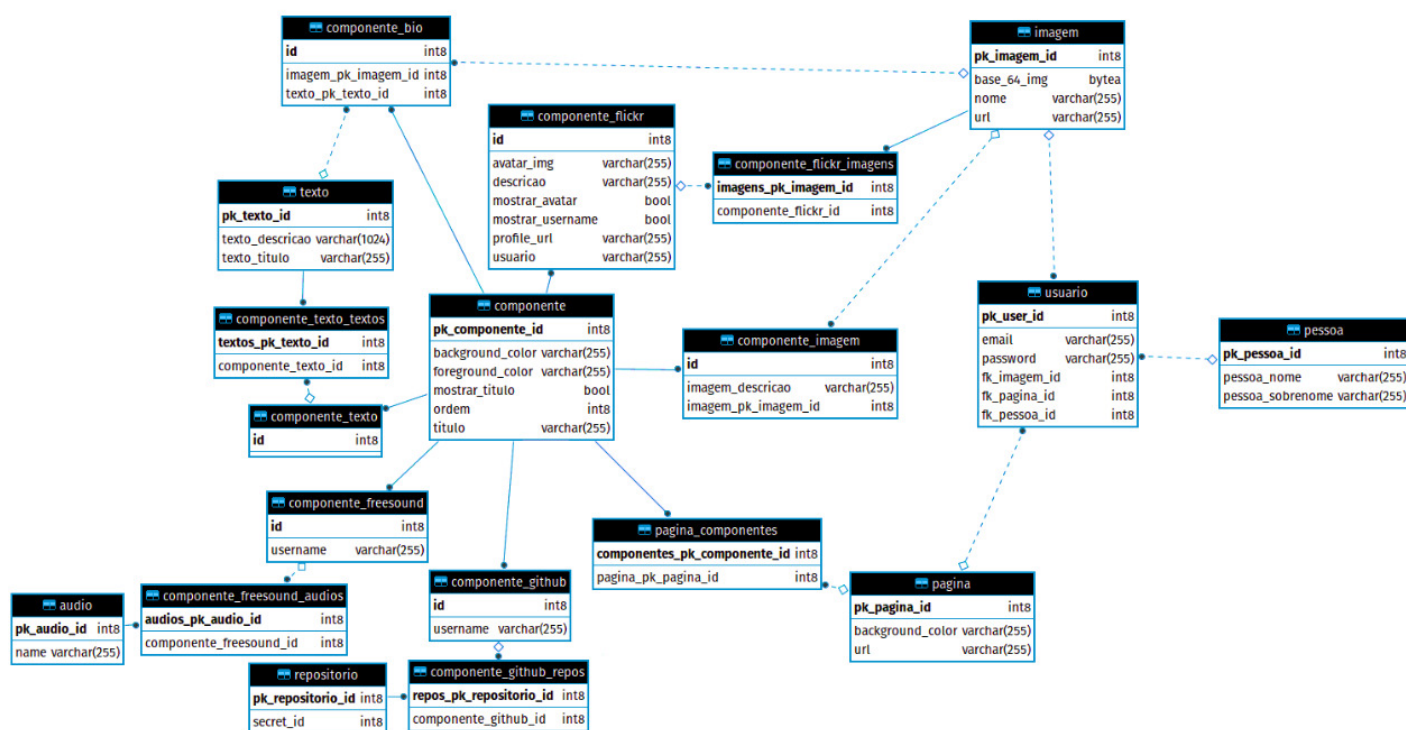
FIGURA 4 - DIAGRAMA DE CASO DE USO



FONTE: OS AUTORES (2021).

Relacional, que tem configuração em acordo com o Modelo de Entidade Relacional, constante no Apêndice A. No Sistema de Banco de Dados as informações serão armazenadas em diferentes tabelas, como exposto no Diagrama Físico do Banco de Dados Relacional apresentado na Figura 6.

FIGURA 6 - DIAGRAMA FÍSICO DO BANCO DE DADOS RELACIONAL

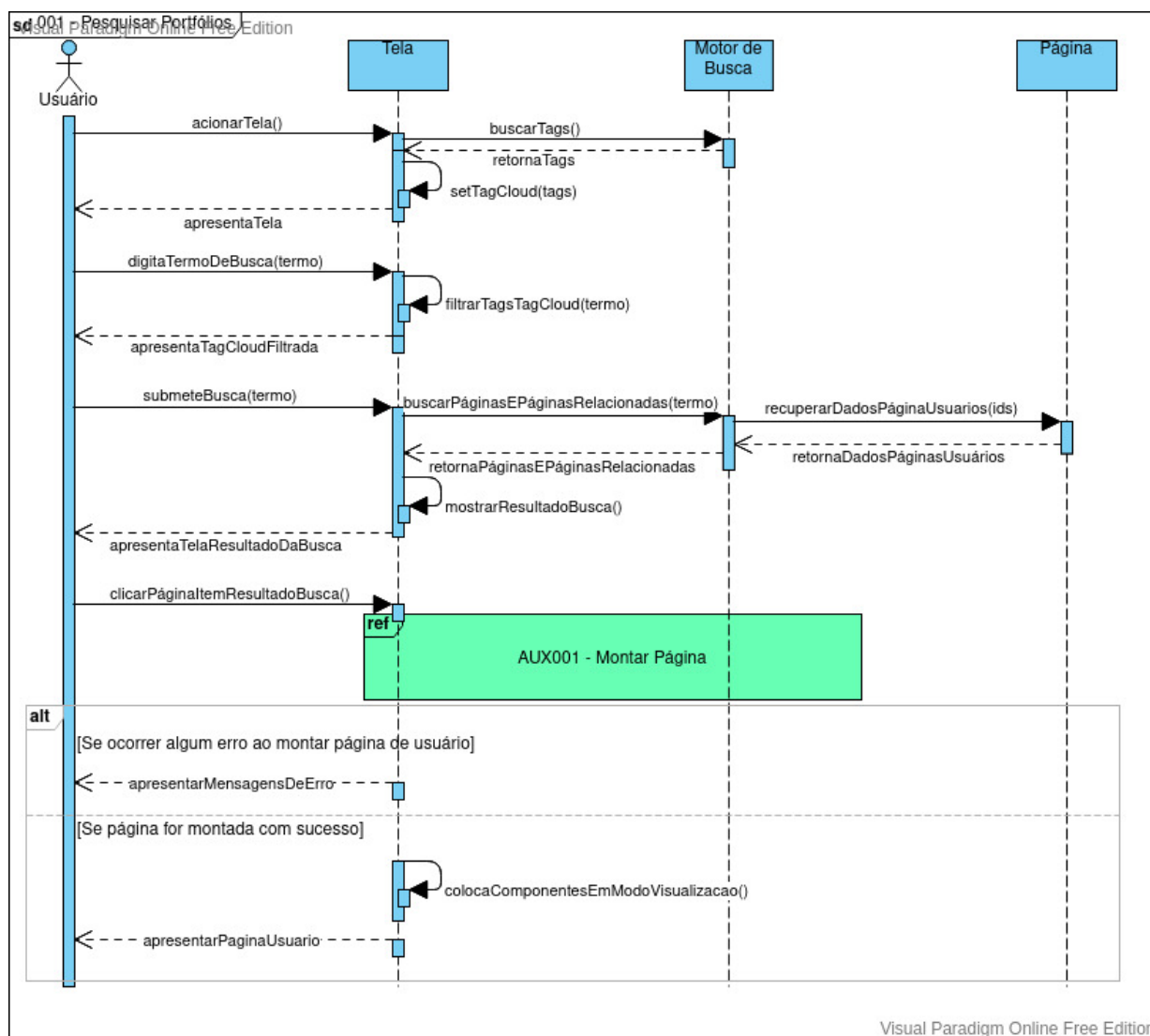


FONTE: OS AUTORES (2021).

Após a inserção das informações no Sistema de Banco de Dados, as páginas ficam disponíveis para pesquisa através do caso de uso Pesquisar Portfólios.

O funcionamento do motor de busca por trás da pesquisa fica a cargo da integração do sistema com o Banco de Dados em Grafos, que apresenta quais resultados mais se aproximam da necessidade do cliente. A comunicação entre esse banco de dados e o sistema pode ser percebido no Diagrama de Sequência 001 apresentado na Figura 7.

FIGURA 7 - DIAGRAMA SEQUÊNCIA 001 – PESQUISAR PORTFÓLIOS



FONTE: OS AUTORES (2021).

Os demais Diagramas de Sequência, que demonstram a forma com que as funções da aplicação são desempenhadas, especificando as interações entre os objetos do sistema para realizar uma função, se encontram no Apêndice B.

Por fim, o sistema apresenta como resultado da busca uma lista de profissionais contendo os requisitos solicitados pelo cliente, em conjunto com os profissionais que tem requisitos que mais se aproximam a demanda. Assim, o cliente tem a sua disposição, além dos profissionais buscados, outros profissionais potenciais que são relacionados a sua demanda.

3.3 LEVANTAMENTO DOS REQUISITOS FUNCIONAIS

Os requisitos funcionais especificam as funções que o software ou que os componentes do software serão capazes de executar durante operação de um usuário. Estes requisitos são apresentados no Quadro 5.

QUADRO 5 - REQUISITOS FUNCIONAIS

Requisito	Nome	Descrição
REQF001	Cadastrar-se como usuário	A pessoa irá cadastrar-se como um usuário que irá colocar seu portfolio na plataforma
REQF002	Cadastrar-se como recrutador	A pessoa irá cadastrar-se como um usuário que irá pesquisar por perfis específicos
REQF003	Renderizar site pessoal	O usuário gerará o site de acordo com o estado de edição na plataforma
REQF004	Adicionar componente	O usuário irá adicionar componentes no site de acordo com o que desejar
REQF005	Ordenar componentes	O usuário irá ordenar componentes no site de acordo com o que desejar
REQF006	Remover componentes	O usuário irá remover componentes no site de acordo com o que desejar
REQF007	Editar componente	O usuário irá remover componentes no site de acordo com o que desejar
REQF008	Mudar cor do background do site	O usuário pode mudar o tema de background do site
REQF009	Fazer upload de imagem	O usuário pode inserir imagens para utilizar em seus componentes
REQF010	Procurar candidato	O usuário recrutador poderá pesquisar por perfis de acordo com <i>tags</i> específicas do sistema
REQF011	Ver perfil do usuário	Ambos os usuários poderão navegar entre perfis existentes no sistem
REQF012	Filtrar perfil por tag	A pesquisa poderá ser filtrada por <i>tags</i> específicas
REQF013	Integrar com APIs externas	Na construção da página pessoal o sistema deve possibilitar a busca por dados em APIs externas (Github, Freeound e Flickr)

FONTE: OS AUTORES (2021).

3.4 LEVANTAMENTO DOS REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais especificam de que forma o software irá realizar suas funcionalidades, como em questões de desempenho, interface e qualidade. No Quadro 6 são apresentados os requisitos não funcionais do projeto:

QUADRO 6 - REQUISITOS NÃO FUNCIONAIS

Requisito	Nome	Descrição
REQNF001	Salvar <i>tags</i>	O sistema irá salvar <i>tags</i> de usuários e componentes
REQNF002	Busca por grafo	O sistema irá implementar uma busca por menor caminho de <i>tags</i> utilizando grafos
REQNF003	Upload de imagem	O sistema irá comportar o upload de imagens para utilização em componentes
REQNF004	Java 8	O sistema será desenvolvido utilizando java 8 no back
REQNF005	Angular 8	O sistema será desenvolvido utilizando Angular 8 no front
REQNF006	Spring boot	O backend terá o framework spring boot para desenvolvimnto das APIS de backend
REQNF007	Neo4j	O sistema utilizará neo4j para armazenamento de <i>tags</i>
REQNF008	Postgresql	O sistema utilizará postgresql para persistencia de dados básicos
REQNF009	Typescript	o sistema utilizará typescript no front end
REQNF010	JWT	Será utilizado JWT para controle de autorização e autenticação no front-end e no back-end

FONTE: OS AUTORES (2021).

3.5 PROCESSO DE TRABALHO

O processo de trabalho foi dividido em quatro grupos: planejamento do projeto, desenvolvimento do Backend, desenvolvimento do Frontend e conclusão, cada um com um conjunto de atividades a ser desenvolvido em um prazo determinado, como apresentado no Quadro 7.

Na fase de planejamento, foi elaborada a concepção inicial da aplicação identificando contexto, problema, justificativa e objetivos e também foram analisadas outras plataformas que atuam nesse contexto, com a finalidade de comparar e avaliar serviços semelhantes. O desenvolvimento da aplicação ficou dividido em Backend, que definiu a estrutura de servidor que suporta a aplicação e suas conexões com os banco de dados e Frontend, na qual foram desenvolvidas as interfaces da aplicação apresentadas ao usuário.

Por fim, na fase de integração e conclusão houve a integração do Backend com o Frontend, acompanhada por testes, correções e ajustes e a revisão final da documentação.

QUADRO 7 - CRONOGRAMA DE ATIVIDADES

	Atividades	Início	Duração	Término
Planejamento do Projeto	Identificação do Contexto da Aplicação (Problema, Justificativa e Objetivos)	4/1/2020	30	5/1/2020
	Revisão Literária	5/1/2020	15	5/16/2020
	Identificação, Comparação e Avaliação de Serviços Semelhantes	5/15/2020	15	5/30/2020
	Levantamento dos Requisitos da Aplicação	6/1/2020	30	7/1/2020
	Documentação inicial - Elaboração dos Diagramas da Aplicação	7/1/2020	30	7/31/2020
	Prototipação das Telas da Aplicação	7/1/2020	30	7/31/2020
Back end	Desenvolvimento da estrutura do Banco de Dados Relacional	8/1/2020	15	8/16/2020
	Desenvolvimento do Backend (Autenticação, Modelos, Services e Resources)	8/16/2020	15	8/31/2020
	Desenvolvimento da estrutura do Banco de Dados em Grafos e motor de busca	9/1/2020	30	10/1/2020
	Integração do Backend com os Banco de Dados	10/1/2020	30	10/31/2020
	Testes, correções e ajustes do Backend	11/1/2020	30	12/1/2020

FONTE: OS AUTORES (2021).

QUADRO 7 – CRONOGRAMA DE ATIVIDADES

(Conclusão)

	Atividades	Início	Duração	Término
Front end	Desenvolvimento da tela de Portfólio e componentes	8/1/2020	30	8/31/2020
	Desenvolvimento da tela de Preview do Portfólio	9/1/2020	15	9/16/2020
	Desenvolvimento da Landing Page e telas de login, cadastro e configurações	9/15/2020	15	9/30/2020
	Desenvolvimento das telas de busca e apresentação de resultado	10/1/2020	15	10/16/2020
	Testes, correções e ajustes do Frontend	10/16/2020	30	11/15/2020
Integração e Conclusão	Integração dos Sistemas	12/1/2020	60	1/30/2021
	Testes, correções e ajustes	2/1/2021	30	3/3/2021
	Revisão Final da Documentação	3/1/2021	22	3/23/2021

FONTE: OS AUTORES (2021).

O diagrama de Gantt, ilustrando o cronograma do projeto, sua linha do tempo ao longo do período de desenvolvimento das atividades, é apresentado no Apêndice F.

3.6 MODELO DE NEGÓCIO

Embora a aplicação descrita neste relatório seja apenas um objeto de Trabalho de Conclusão de Curso, o seu desenvolvimento é feito considerando a possibilidade de que a mesma seja disponibilizada ao mercado. Para este cenário, foi pensado em um modelo de negócio da seguinte forma: Inicialmente a aplicação será ofertada gratuitamente, estando em uma categoria beta e contando com o apoio de usuários através de feedback e manutenção financeira por meio de doações e anúncios. Para versões futuras, estuda-se a arrecadação financeira através de oferecimento de cursos de especialização para os usuários por meio de parcerias com instituições (como Udemy e Alura), além de uma modalidade Freemium, na qual determinados recursos são disponibilizados para

usuários pagantes (como por exemplo levantamento de dados detalhados acerca de tendências de busca). Vale ressaltar que a aplicação será disponibilizada no formato *Software as a Service* – SaaS – Software como serviço – portanto não será necessário que os usuários instalem qualquer software adicional, apenas um navegador moderno.

3.7 TECNOLOGIAS UTILIZADAS

Quanto às tecnologias utilizadas para o desenvolvimento do projeto destacam-se linguagens de programação para o back-end e para o front-end da aplicação, frameworks, bibliotecas e sistemas de gerenciamento de banco de dados.

3.7.1 VSCode (*Visual Studio Code*)

O Visual Studio Code é uma ferramenta de edição de texto que está disponível de forma gratuita para os sistemas Windows, macOS e Linux. Ela apresenta suporte nativo a linguagens como JavaScript, TypeScript e Node.js, e, além disso, conta com diversas extensões que possibilitam a customização do ambiente para melhor produtividade e integração com outras tecnologias, como: Java, HTML, CSS, JavaScript e TypeScript.

A ferramenta está disponível no Website do Visual Studio, que pode ser acessado pelo seguinte endereço: <https://code.visualstudio.com/docs>, que também traz vídeos introdutórios, guias de instalação, configuração e interface do usuário.

3.7.2 *Angular*

O Angular é um framework de design e desenvolvimento que traz como objetivo a criação eficiente e sofisticada de aplicações de uma página (*single-page apps*). É uma ferramenta gratuita que requer do desenvolvedor conhecimento prévio em HTML, CSS, JavaScript e TypeScript que é a linguagem nativa do framework.

A tecnologia está disponível no Website do Angular, que pode ser acessado pelo seguinte endereço: <https://angular.io/guide/setup-local>, com guias de instalação, configuração e um tutorial para criação de um aplicativo simples.

3.7.3 *Java*

O Java é uma linguagem de programação orientada a objetos utilizada para o desenvolvimento de aplicações. A plataforma de desenvolvimento Java é organizada para atender diversas necessidades, como: JavaSE para desenvolvimento desktop, Java ME para dispositivos móveis e embarcados e Java EE para soluções web e corporativas.

Para mais informações, pode-se consultar o Website da linguagem, no seguinte endereço: https://www.java.com/pt_BR/.

3.7.4 *HTML (HyperText Markup Language)*

HTML, ou Linguagem de Marcação de Hipertexto, é uma linguagem de marcação voltada para a construção da estrutura de páginas Web. Ela consiste em uma série de elementos que delimitam ou agrupam diferentes partes do conteúdo de uma página para estruturar uma página em um navegador.

Um Website que fornece diversos tutoriais sobre essa linguagem e seus elementos é o W3Schools, que pode ser acesso pelo seguinte endereço: <https://www.w3schools.com/html/>.

3.7.5 *CSS (Cascading StyleSheets)*

CSS, ou Folha de Estilo em Cascatas, é uma linguagem utilizada para definir estilos a elementos escritos em linguagem HTML. Por meio dela, é possível adicionar cor aos textos, aos planos de fundos, definir as fontes de texto, além de ajustar imagens e elementos aos tamanhos da tela.

Mais informações sobre essa ferramenta e como utilizá-la podem ser encontrados no Website da W3Schools, acessado pelo seguinte endereço: <https://www.w3schools.com/css/>.

3.7.6 *JavaScript*

O JavaScript é uma linguagem de script do lado do cliente que permite a inclusão de elementos dinâmicos em um Website. Utilizado para dinamização das páginas, possibilita a interação com o usuário por meio de caixas de diálogo e validação de dados.

Informações sobre a linguagem, como: sintaxe, elementos, operadores e demais funções podem ser encontrados no Website da W3Schools, no seguinte endereço: <https://www.w3schools.com/js/>.

3.7.7 *Bootstrap*

O Bootstrap é um framework para construção de Websites responsivos que permite a construção rápida de protótipos e aplicações completas. Ele traz um sistema de grid responsivo que permite organização de elementos da página e componentes pré-construídos que agilizam o desenvolvimento.

Mais informações sobre este framework podem ser obtidas no Website do Bootstrap, disponível no endereço: <https://getbootstrap.com.br/>.

3.7.8 *Spring Boot*

O Spring Boot é uma ferramenta que permite agilizar o processo de desenvolvimento de software facilitando configurações e publicações de aplicações. Por meio dela, o desenvolvedor pode indicar quais módulos deseja utilizar (Web, Template, Persistência e outros) para que o Spring Boot configure a aplicação.

Informações sobre essa ferramenta podem ser encontradas no Website do Spring, acesso pelo seguinte endereço: <https://spring.io/projects/spring-boot>.

3.7.9 *Hibernate*

O Hibernate é um framework Java para mapeamento de um modelo de objetos para um modelo de dados relacional. Além disso, por meio dessa ferramenta são facilitadas operações que envolvem o acesso a banco de dados. Informações complementares sobre a ferramenta podem ser obtidas no Website do Hibernate, disponível no seguinte endereço: <https://hibernate.org/>.

3.7.10 *PostgreSQL*

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional de código aberto. O sistema trabalha com a linguagem SQL, ou Structured Query Language, e permite que sejam realizadas consultas, inserções, remoções e diversas outras operações em um banco de dados. Mais informações sobre esse sistema podem ser encontradas no endereço: <https://www.postgresql.org/>.

3.7.11 *Neo4j*

A ferramenta Neo4j é um banco de dados de grafos nativo, que apresenta uma estrutura flexível definida pela relação entre dois ou mais registros de dados, ao contrário dos bancos de dados tradicionais com sua estrutura de linhas, colunas e tabelas. Mais informações podem ser obtidas no site do Neo4j, no endereço: <https://neo4j.com/>.

3.7.12 *GitHub*

O GitHub é um sistema de controle de versionamento de projetos, permitindo que uma equipe trabalhe em conjunto criando elementos de uma aplicação em um ambiente colaborativo. Dessa forma, as mudanças no código podem ser controladas e, caso necessário, pode-se realizar a restauração do código modificado. Informações sobre a

ferramenta podem ser encontradas no site do GitHub no seguinte endereço: <https://github.com/>.

3.7.13 *TypeScript*

É uma linguagem de programação que pode ser utilizada em front-end e back-end. Ela estende o Javascript adicionando tipos às variáveis e retornos de funções, facilitando o desenvolvimento ao se utilizar uma IDE compatível, pois alguns erros podem ser verificados durante o desenvolvimento evitando problemas em tempo de execução.

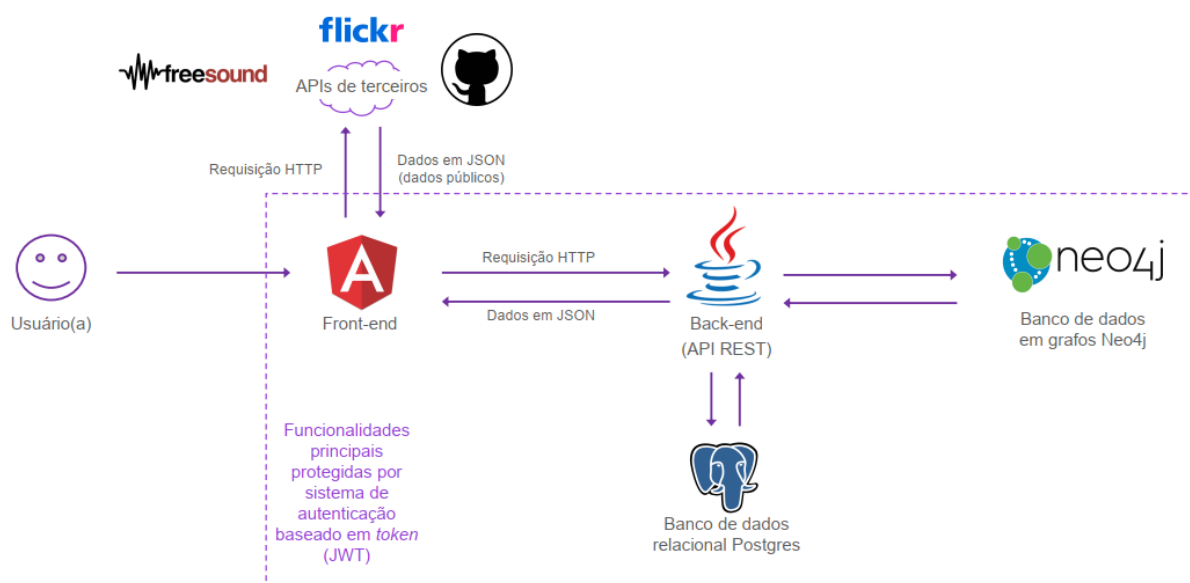
A aplicação em TypeScript após concluída é compilada gerando um código Javascript que, então é executado.

Mais informações sobre esta linguagem podem ser encontradas no seguinte endereço: <https://www.typescriptlang.org/>

4 IMPLEMENTAÇÃO

A implementação do sistema é dividida em duas partes, a primeira trata do back-end da aplicação, envolvendo: sistemas de login, banco de dados relacional para armazenar informações dos usuários, banco de dados em grafos para armazenar informações das *tags* dos perfis e serviços do lado do servidor. A segunda parte trata do front-end da aplicação, referente às páginas web e a interface apresentada ao usuário, e da integração com o back-end. A arquitetura da comunicação entre as partes da aplicação pode ser observada na Figura 8.

FIGURA 8 - ARQUITETURA DA APLICAÇÃO



FONTE: OS AUTORES (2021).

4.1 BACK-END

O back-end é a parte do projeto que funciona como estrutura da aplicação, sendo executado no lado do servidor, é responsável por intermediar funcionalidades entre o front-end e os sistemas de banco de dados, além de tratar questões de segurança, validação, persistência de dados e regras de negócio.

4.1.1 Sistema de Usuários

O sistema de usuários é responsável pela manutenção das contas usuários, permitindo que novos usuários sejam criados e que usuários existentes sejam atualizados, alterados ou excluídos.

Este sistema faz ligação entre as requisições do usuário e o banco de dados relacional que armazena as informações do usuário, e, com intuito de persistir e dar consistências aos dados, utiliza o mapeamento de modelos através de anotações do Hibernate.

O usuário é um objeto contendo endereço de e-mail, uma senha, um objeto “Pessoa” que guarda informações do cliente (nome e sobrenome), um objeto “Página” que guarda as páginas criadas pelo cliente e um objeto “Imagem” que guarda uma imagem do cliente.

4.1.2 Sistema de Login

O sistema de login é responsável pela validação dos dados enviados pelos usuários que acessam a aplicação, verificando existência do cadastro e autenticando a senha digitada.

O sistema utiliza criptografia para armazenar as senhas dos usuários através da funcionalidade *PasswordEncoder* da ferramenta Spring Boot, que encapsula as especificações da codificação e garante a segurança das credenciais dos usuários, prevenindo acessos não autorizados a aplicação.

Além disso, ao realizar o login no sistema, um Token é atribuído a sessão do usuário com intuito de autenticar a sessão e garantir maior segurança no acesso. O Token tem prazo de validade, e após expirada a validade o Token invalida a sessão do usuário.

4.1.3 Sistema de Gerenciamento de Componentes

O sistema de gerenciamento de componentes é utilizado para manter os componentes da página do usuário, os quais podem ser componentes de texto, imagem, biografia ou componentes GitHub, Flickr ou Freesound.

Os componentes são objetos do sistema e armazenam as características escolhidas pelo usuário ao personalizar as páginas, além de textos e imagens, o usuário poderá selecionar uma cor de fundo e cor de destaque para seus componentes, informação que também é guardada em uma instância desse objeto.

Os elementos que referenciam as redes GitHub, Flickr e SoundCloud apresentam as informações e trabalhos expostos nessas redes. Para isso o cliente fornece o nome da conta de acesso a esses sites e, por meio das interfaces de programação de aplicação (APIs), o sistema se comunica com essas redes e apresenta seu conteúdo dentro do componente.

4.1.4 Banco de Dados Relacional

O banco de dados relacional PostgreSQL é utilizado para armazenar informações dos usuários cadastrados no site, guardando e-mail cadastrado e senha criptografada e para armazenar as informações necessárias para construção de componentes e as páginas.

O sistema de login se comunica com o banco para autenticar as informações nos momentos de login, além das funções de criação, alteração ou remoção de contas de usuários.

4.1.5 Banco de Dados em Grafos

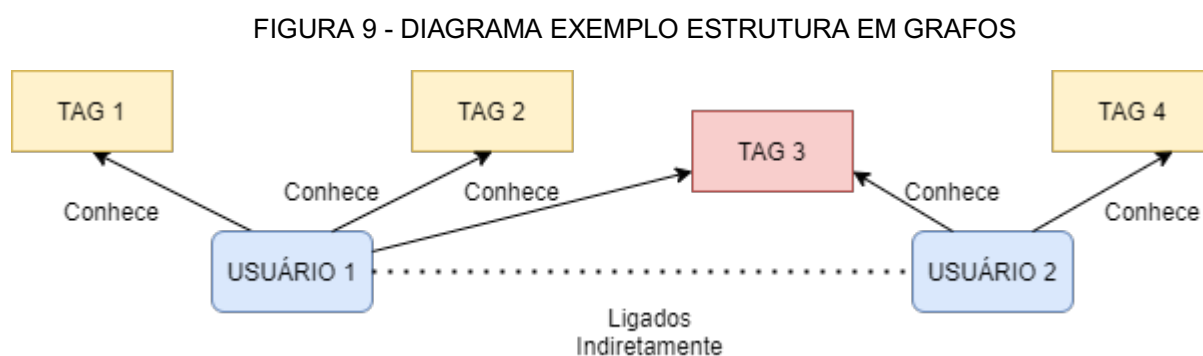
O banco de dados em grafos Neo4j é responsável por guardar as relações que um profissional tem com uma área de atuação ou tipo de serviço, representado por uma *Tag*. Partindo dessa função, o banco gera um grafo contendo as relações entre profissionais que compartilham *Tags* semelhantes.

O grafo gerado é utilizado para encontrar os profissionais que tem serviços ou áreas de atuação que mais se aproximam a descrição buscada por um cliente. Assim, o sistema apresenta mais resultados a quem busca por um serviço, mostrando serviços que são de alguma forma relacionados aos buscados e podem ser de interesse do cliente.

4.1.6 Geração do grafo de *Tags*

Para a criação do grafo das *Tags*, o front-end enviará uma requisição para o resource no back-end, que inclui essas *Tags*, ainda não relacionadas, no banco de dados em grafos. No momento em que as *Tags* são ligadas aos usuários, pode-se construir uma rede de usuários interligados por diferentes *Tags*, de forma, então, com que o grafo seja gerado. O banco de dados em grafo permite que diversas operações de buscas sejam realizadas, por meio do percurso estabelecido entre usuários e *Tags* interligados, de maneiras mais profundas do que as operações de um banco de dados relacional.

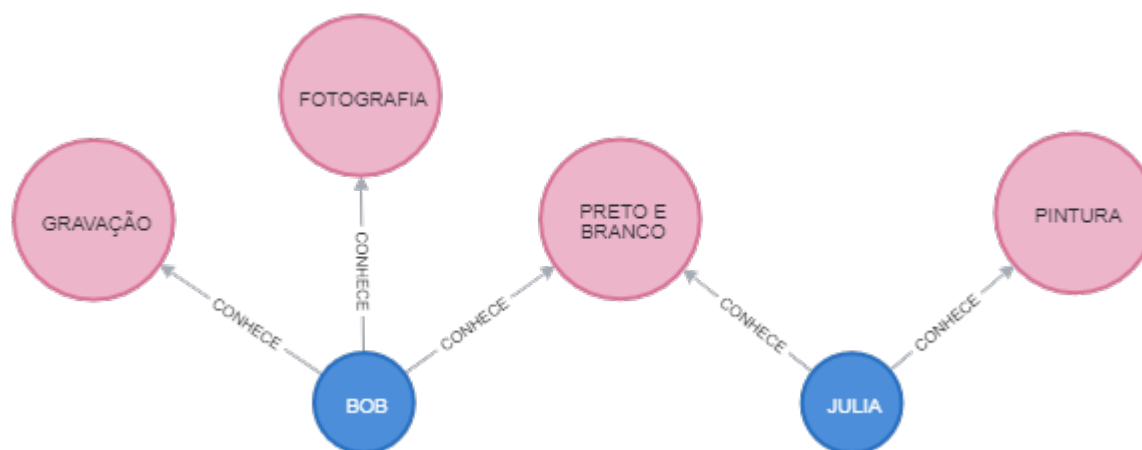
Como exemplo desse funcionamento, na Figura 9 apresenta-se uma situação em que um usuário realiza uma busca pela *Tag 2*, em resposta a essa requisição o sistema recomenda primeiramente o Usuário 1, por estar diretamente ligado a *Tag 2*. Em complemento, mostra o Usuário 2, que mesmo sem estar ligado a *Tag 2*, está relacionado com o Usuário 1 por meio da *Tag 3*.



FONTE: OS AUTORES (2021).

Com a implementação da base de dados no banco em grafos Neo4j, é gerada uma estrutura análoga a apresentada na Figura 10. Como exemplo inserido no contexto da aplicação, quando um usuário busca pelo termo “fotografia” ou “gravação”, pode receber uma indicação secundária do portfólio da usuária Julia, já que está ligado ao usuário “Bob” pelo nó “Preto e branco”, isto apenas após receber as indicações de portfólios ligadas diretamente ao termo da busca. Para sugerir as recomendações dessa maneira, o sistema faz uso do algoritmo de menor caminho, que é nativo do banco de dados em grafos Neo4J e de difícil implementação em bancos relacionais.

FIGURA 10 - GRAFOS GERADOS PELO NEO4J



FONTE: OS AUTORES (2021).

4.1.7 Motor de Busca por Tags

Para que sejam apresentados os perfis de profissionais que mais se aproximam ao perfil buscado pelo usuário, é necessário que o sistema tenha um motor de busca capaz de encontrar as palavras-chave melhor se encaixam na descrição procurada.

Por esse motivo, o sistema é integrado a um banco de dados em grafos, responsável por armazenar as *Tags* que melhor definem um profissional, sua área de atuação ou os tipos de serviços prestados.

Pela integração com esse banco de dados, o motor busca pelo perfil que mais se encaixa na descrição procurada e, pelos dados estarem armazenados na forma de

grafos, traz outros resultados relacionados com o principal encontrado em uma distância de até 3 nós.

Por fim, o motor de busca retorna o resultado principal e os que mais se assemelham a ele para o front-end e, assim, o usuário visualiza outros profissionais que potencialmente podem atender seus requisitos.

É importante ressaltar aqui que, como o motor de busca se trata de um sistema para recuperação de dados, há uma lógica na ordenação dos dados quando estes são recuperados. O primeiro critério de ordenação é se as páginas retornadas possuem em sua lista de *tags* aquela que foi buscada, separando assim as páginas em dois conjuntos: páginas que possuem a *tag* em sua lista e páginas que foram recuperadas por possuírem uma ligação indireta com páginas do primeiro conjunto. Após essa ordenação é realizada a ordenação por ordem alfabética em cada conjunto de páginas.

4.2 FRONT-END

O front-end é a parte do projeto responsável pela apresentação da interface ao usuário final, envolvendo a estrutura das páginas e funcionalidades apresentadas ao cliente. Ressalta-se que, para a implementação do front-end, optou-se por um modelo do tipo Single Page Application – SPA (Aplicação de página única) utilizando o framework Angular 2+.

4.2.1 Tela de apresentação

A tela inicial apresentada ao usuário contém em seu centro um campo para buscar por palavras acompanhado de uma nuvem de palavras (comumente conhecida em aplicações web como *Tag Cloud* ou *Word Cloud*), que trará as *Tags* atualmente registradas no banco de dados de grafo organizadas conforme o número de ocorrências.

A tela também apresenta as opções de login e de cadastrar um novo usuário. Dessa maneira, ela permite com que um usuário qualquer realize buscas por portfólios profissionais sem ter necessidade de estar cadastrado, e possibilita a entrada ou

cadastro de usuário para aquele que deseja criar uma página para exposição de seu portfólio.

4.2.2 Tela de personalização de portfólios

Ao realizar o login no sistema o usuário pode acessar a tela de personalização de sua página através do menu flutuante localizando no canto superior direito da tela. Ao selecionar a opção correspondente à edição de sua página, é apresentada ao usuário a tela de seu portfólio inicialmente apenas com um componente Bio, com um texto simples, gerado automaticamente no momento da criação do seu cadastro. Logo, neste momento o usuário pode adicionar e personalizar componentes, por meio da edição de conteúdo textual, cor de fundo e de destaque do componente (a partir do seletor de cores nativo do navegador ou sistema operacional) e carregamento de imagens. Os tipos de componentes existentes são descritos a seguir, demonstrando suas ações até então:

- a) Texto: componente no qual o usuário pode descrever brevemente suas características, competências, algo relacionado ao seu perfil ou ainda um texto a livre escolha permitindo, inclusive, o formato *Markdown*. nele é permitida a inserção de até 2 textos, cada um contendo um título.
- b) Flickr: componente no qual o usuário pode apresentar seus trabalhos que estão disponíveis no site Flickr, especializado na hospedagem e compartilhamento de imagens como fotografias e ilustrações. O componente utilizará a API disponível pelo Flickr e, a partir da informação do nome de usuário do Flickr, apresentará em formato de galeria os trabalhos públicos do usuário registrados no Flickr, podendo o usuário escolher quais trabalhos, daqueles que estão publicamente disponíveis, deseja apresentar na aplicação.
- c) Github: componente no qual o usuário pode apresentar seus trabalhos que estão disponíveis no site Github, que é uma aplicação voltada para programadores e especializado na hospedagem, compartilhamento e controle de versionamento de códigos e arquivos. O componente utiliza a API do Github e, a partir da informação do nome de usuário do Github, apresenta em forma

- de lista de cartões os repositórios públicos do usuário dentro da plataforma, dando ênfase ao título, descrição e data de criação, sendo que o usuário poderá escolher quais repositórios, entre os que estão disponíveis de forma pública, deseja exibir em sua página.
- d) Freesound: componente no qual o usuário pode apresentar seus trabalhos que estão disponíveis no site Freesound, voltado para produtores musicais e especializado na hospedagem e compartilhamento de amostras de áudios. O componente utiliza a API do Freesound e, a partir da informação do nome de usuário do Freesound, apresenta em forma de *playlist* (lista de músicas), juntamente com uma interface para execução de áudio, os trabalhos públicos do usuário dentro da plataforma, sendo possível ao usuário escolher quais, entre os áudios publicamente disponíveis, são apresentados em sua página.
 - e) Foto: componente em que o usuário pode utilizar para fazer upload de uma imagem diretamente do seu dispositivo. Esta imagem será apresentada ampliada no próprio componente.
 - f) Bio: componente voltado para que o usuário possa escrever um texto, utilizando inclusive linguagem *Markdown*, voltado para ser um tipo de apresentação pessoal, ou ainda colocar dados de contato. Porém, vale ressaltar que o conteúdo deste texto é de escolha do usuário. Além do texto, este componente possibilita a inserção de uma imagem através de upload do arquivo a partir do dispositivo do usuário. Esta imagem será posicionada à esquerda do texto,

Todos os componentes apresentam propriedades que podem ser personalizadas, como o texto do título e se o mesmo deve ser exibido, cor de fundo, cor de destaque e, após sua inserção em tela, a ordem que ficaram dispostos no portfólio. Os componentes podem ser adicionados, removidos e alterados pelos usuários, e, com exceção dos componentes de Texto, Bio e Foto, necessitam a informação do nome de usuário utilizados nos sites, para que seu conteúdo seja carregado via API.

Ao finalizar a personalização da página, o usuário pode visualizar uma prévia de como sua página ficará, através de um botão *preview*, e, quando satisfeito com as alterações, pode finalizar a edição e salvar a página criada.

4.2.3 Tela de *preview* de portfólio

Após terminada a personalização desejada pelo o usuário, por meio do botão *preview*, é apresentada uma tela mostrando uma versão prévia da página publicada. O conteúdo apresentado é uma pré-visualização do resultado do portfólio do usuário com base nos componentes adicionados e editados, caso o usuário esteja de acordo com a configuração da página ele poderá voltar a tela de edição da página para salvar e publicar a página para que outros possam visualizar.

5 APRESENTAÇÃO DO SISTEMA

Ao acessar o sistema, é apresentada a tela de Pesquisar Portfólios, trazendo ao centro um campo que permite com que um usuário realize uma busca por texto. Por meio do campo de busca, o sistema permite ao usuário realizar pesquisas por profissionais de áreas específicas, definidas por *Tags* ou diretamente pelo nome do usuário.

Abaixo do campo de busca, a aplicação apresenta uma nuvem de palavras contendo as especialidades, representadas na forma de *Tags*, mais recorrentes armazenadas no Sistema de Banco de Dados conforme apresentado na Figura 1.

FIGURA 11 - DATA VIEW 01: TELA DE PESQUISAR PORTFÓLIOS



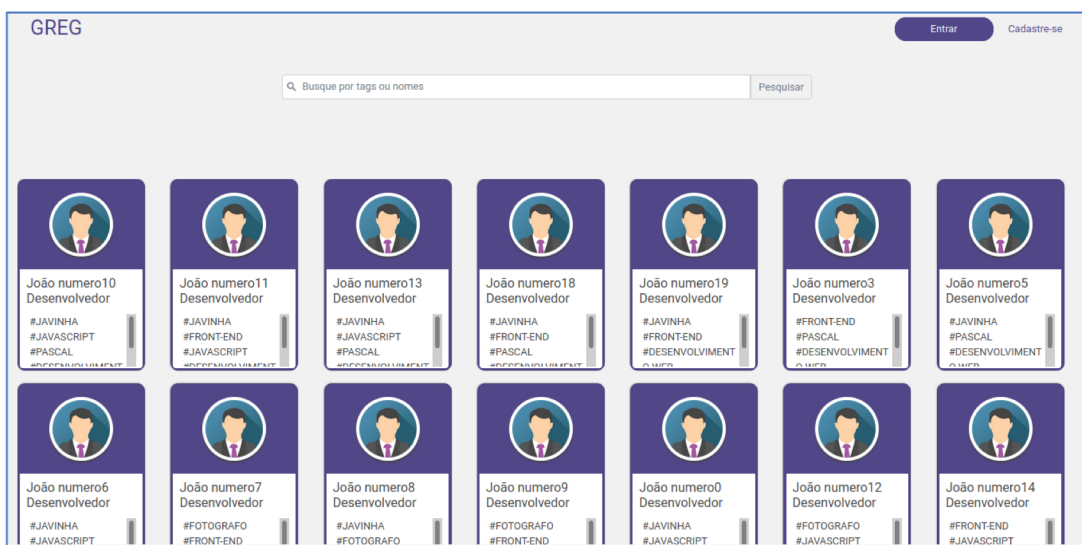
FONTE: OS AUTORES (2021).

Ao pesquisar por uma especialidade, o sistema se comunica com o Banco de Dados em Grafos, que é capaz de encontrar a especialidade que mais se adequa ao termo da pesquisa, e que, em especial, consegue encontrar especialidades com características semelhantes a buscada. Dessa forma, além do resultado buscado pelo usuário, o sistema também apresenta resultados que podem interessar a quem pesquisa.

Exemplificando os resultados semelhantes, pode-se mencionar a situação hipotética em que um usuário busca por um termo como: “Fotografia Preto e Branco”, e,

ao verificar que não consta nenhum resultado exato para a busca, o sistema recebe sugestões do Banco de Dados em Grafos como: “Fotografia Colorida”, “Filmagem Preto e Branco” ou “Pintura Preto e Branco”. Após concluída a pesquisa, o usuário pode acessar o perfil dos profissionais compatíveis, que estarão dispostos conforme apresentado na Figura 12.

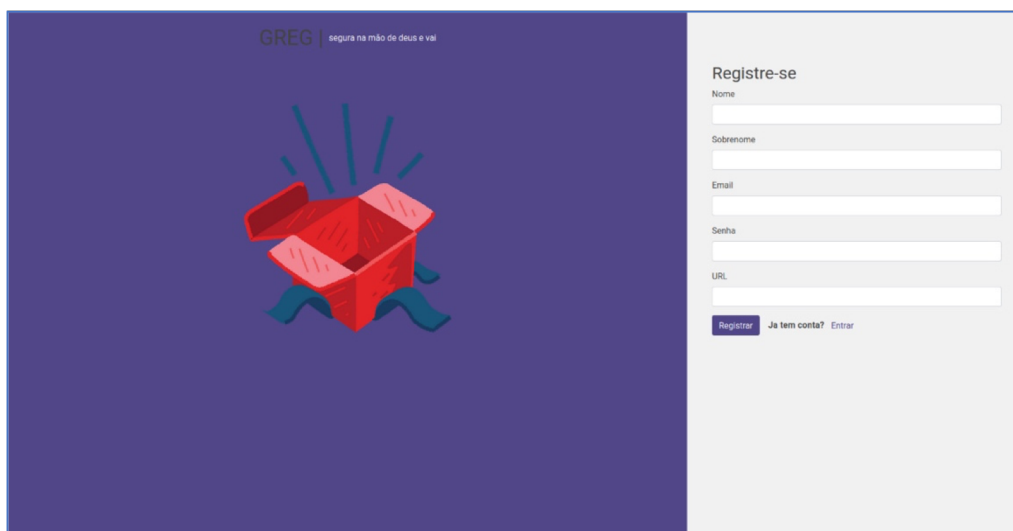
FIGURA 12 - DATA VIEW 02: TELA DE APRESENTAR RESULTADOS



FONTE: OS AUTORES (2021).

Para criar e editar uma página, o usuário deve cadastrar uma conta e realizar login, por meio dos botões no canto direito superior da página. Para o cadastro, o sistema requisita algumas informações do usuário conforme demonstrado nas Figura 13 e Figura 14.

FIGURA 13 - DATA VIEW 03: TELA DE REALIZAR CADASTRO



GREG | segura na mão de deus e vai

Registre-se

Nome

Sobrenome

Email

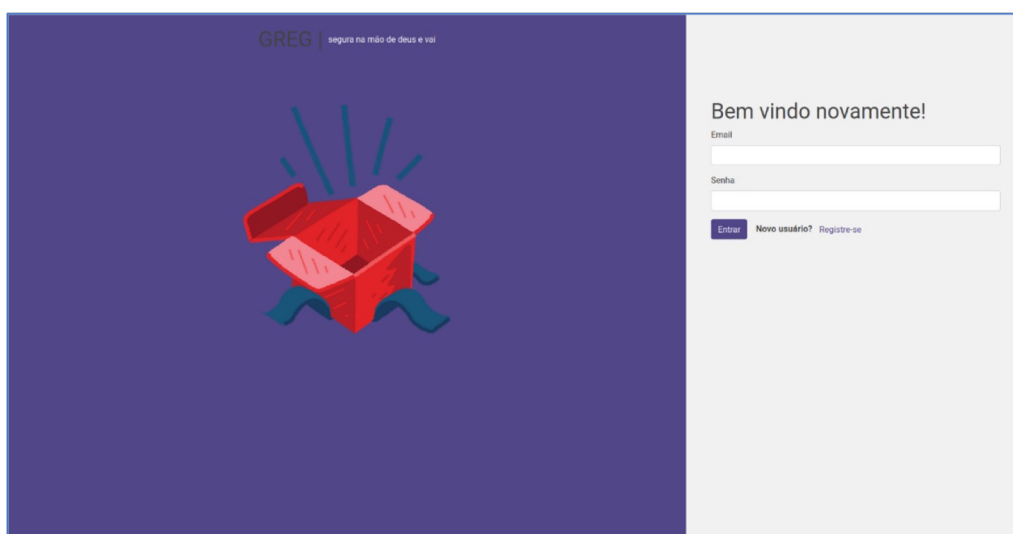
Senha

URL

[Ja tem conta?](#) [Entrar](#)

FONTE: OS AUTORES (2021).

FIGURA 14 - DATA VIEW 04: TELA DE REALIZAR LOGIN



GREG | segura na mão de deus e vai

Bem vindo novamente!

Email

Senha

[Novo usuário?](#) [Registre-se](#)

FONTE: OS AUTORES (2021).

Após o login, o sistema volta a tela Pesquisar Portfólios, que apresenta em seu canto superior direito uma imagem do usuário, contendo um menu para ações de editar os dados, editar a página do portfólio e encerrar a sessão do usuário (logout). Este menu pode ser observado na Figura 15.

FIGURA 15 - DATA VIEW 01: TELA DE PESQUISAR PORTFÓLIOS (USUÁRIO LOGADO)



FONTE: OS AUTORES (2021).

No menu do usuário, por meio da opção “Configurações” podem ser acessadas as informações do usuário, permitindo alterações como mudança de url da página ou inserção de uma imagem de usuário, conforme apresentado na Figura 16.

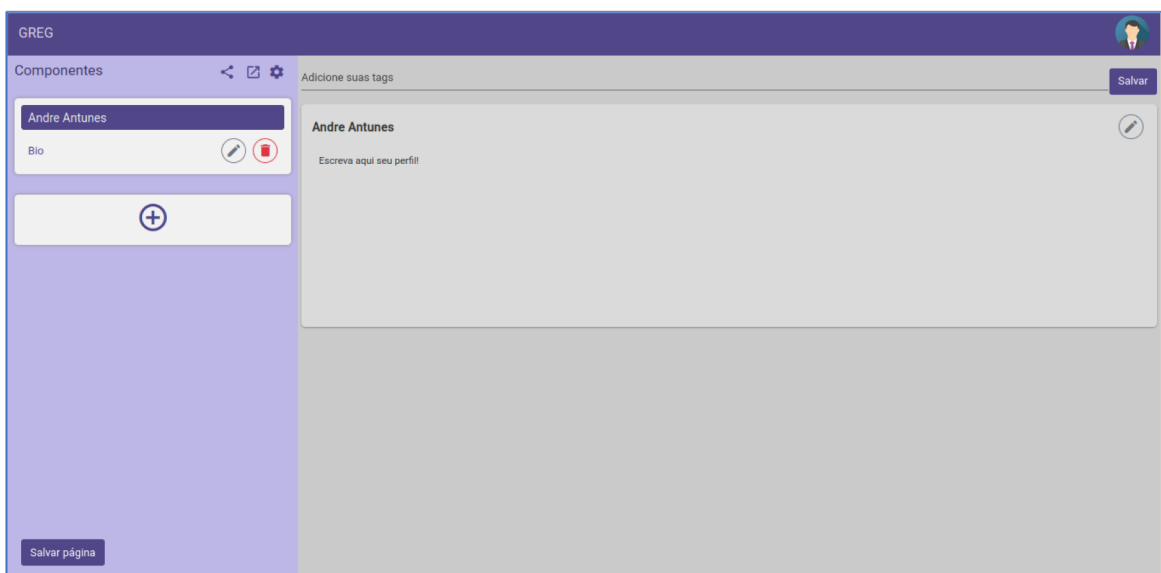
FIGURA 16 - DATA VIEW 06: TELA DE MANTER USUÁRIO

The screenshot shows the "Configurações" (Settings) page. On the left, there is a section for "Imagem principal" (Main image) with a "Browse..." button and the text "No file selected.". The main content area is divided into two sections: "Informações de conta" (Account information) and "Informações da página" (Page information). The "Informações de conta" section includes fields for "Nome" (Name) with the value "Andre", "Sobrenome" (Surname) with the value "Antunes", and "Email" with the value "andre@ufpr.br". The "Informações da página" section includes a field for "URL da página" (Page URL) with the value "andre.antunes". A "Submit" button is located at the bottom left of the form.

FONTE: OS AUTORES (2021).

Por meio da opção “Editar Página”, o sistema apresenta a Tela de Manter Página, que permitirá ao usuário montar sua página com os componentes desejados. Esta tela é apresentada na Figura 17.

FIGURA 17 - DATA VIEW 08: TELA DE MANTER PÁGINA

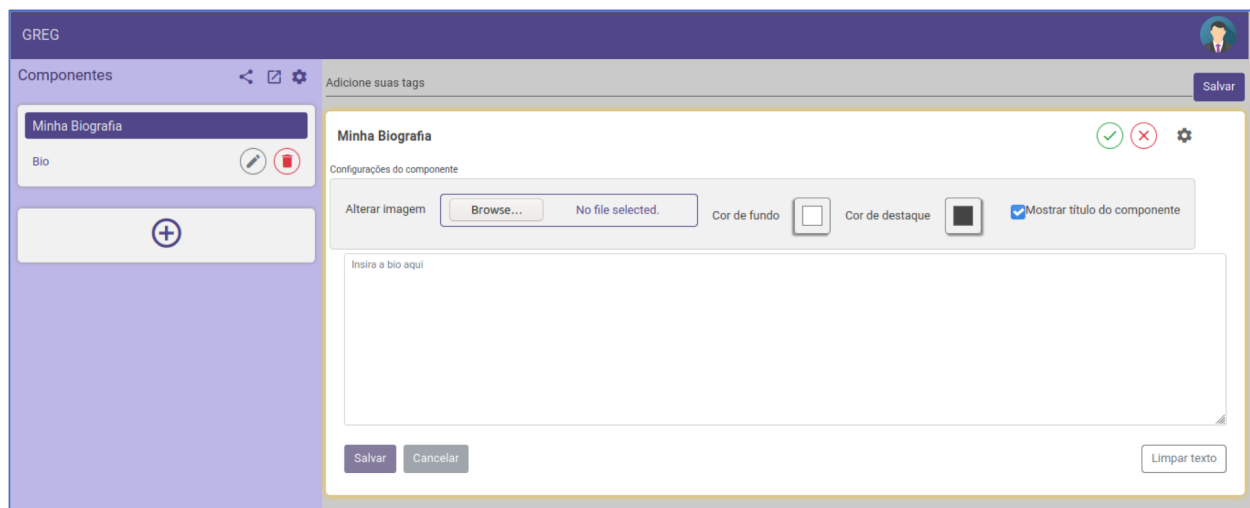


FONTE: OS AUTORES (2021).

Na Tela de Manter Página, o usuário pode adicionar *tags*, que descrevem suas especialidades, e personalizar a página adicionando componentes pelo menu lateral na esquerda da página. Além da inclusão, alteração e exclusão de componentes, o menu permite visualizar uma prévia da página e gera um link para compartilhamento da página.

Ao adicionar um componente, o usuário deve selecionar o tipo de componente desejado, entre: Componentes de Texto, Imagem, Biografia, GitHub, Flickr e Freesound. Todos os componentes podem ser personalizados, incluindo mudança na cor de fundo, cor de destaque e opção por mostrar ou não o título do componente. Além disso, a ordem dos componentes na página pode ser alterada através da reorganização dos elementos no menu da página. Na Figura 18 é mostrado, como exemplo, a edição do componente Bio com as suas opções de personalização expostas.

FIGURA 18 - DATA VIEW 12: TELA DE MANTER BIOGRAFIA DO USUÁRIO



FONTE: OS AUTORES (2021).

Ao final da edição, o usuário pode salvar as alterações, que podem ser revistas a qualquer momento. Dessa forma, a página de portfólio criada já está disponível para pesquisa e visualização dos demais usuários. As demais telas que ilustram as funções restantes do sistema são apresentadas no Apêndice D – Protótipo das Telas.

6 CONSIDERAÇÕES FINAIS

O objetivo desse trabalho foi desenvolver uma plataforma que permita ao usuário a criação de uma carta de apresentação profissional focada na divulgação de seus trabalhos de áreas da tecnologia da informação, artes visuais e produção musical, contendo alto nível de personalização e que possibilite a realização de buscas por profissionais e serviços que mais se adequem às necessidades do interessado.

A aplicação desenvolvida permite que profissionais dessas áreas compartilhem seus trabalhos em um ambiente único por meio da construção de páginas personalizadas que possibilitam a exposição de seus trabalhos. Além disso, através do mecanismo de busca, o portfólio de um profissional fica disponível para visualização de qualquer usuário que realiza uma pesquisa no sistema, fornecendo visibilidade de suas produções aos interessados.

Durante o desenvolvimento da aplicação, foram utilizadas ferramentas essenciais para o desenvolvimento de software, fundamentando a estrutura e funcionalidades do sistema, como o levantamento de requisitos, a identificação de casos de uso essenciais, a análise comparativa de opções disponíveis no mercado.

Por fim, pode ser levada em consideração para o desenvolvimento de futuros trabalhos a criação de novos componentes que realizem a integração com mais plataformas de exposição de trabalhos, como o Gitlab, DevianArt ou Soundcloud, disponibilização de novas ferramentas para o usuário visando levantamento de dados baseados nas buscas realizadas, criação de um perfil do tipo “Recrutador” que poderá favoritar e avaliar páginas de usuários e prover um sistema para possibilitar o contato entre usuários.

REFERÊNCIAS

- ALVARENGA, G. M.; ARAUJO, Z. R. **Portfólio: conceitos básicos e indicações para utilização**; Estudos em Avaliação Educacional, v. 17, n. 33. Londrina, 2006.
- BISCHOFF K., FIRAN C. S., NEJDL W., e PAIU R.; **Can all tags be used for search?**; CIKM 2008: Proceeding of the 17th ACM conference on Information and knowledge management, páginas 193–202, Nova York, NY, USA, 2008. ACM.
- BRANSKI, R. M.; **Localização de informações na internet: características e formas de funcionamento dos mecanismos de busca**; Transinformação, v. 12, nº 1, p.11-19. Campinas, 2000.
- CIRIBELI, J. P., PAIVA, V. H. P.; **Redes e mídias sociais na internet: realidades e perspectivas de um mundo conectado**. Revista Mediação, v. 13. Belo Horizonte, 2011.
- CHANG, S. E., LIU A. Y., SHEN, W. C.; **User trust in social networking services: a comparison of Facebook and LinkedIn**. Institute of Technology Management, National Chung Hsing University. Taiwan, 2016.
- CONTO, J. M. de; **O sistema de gêneros da seleção de candidatos a emprego no contexto empresarial**; V Simpósio Internacional de Estudos de Gêneros Textuais; Caxias do Sul, RS. 2009.
- COSTA, E.; **O papo aqui é profissional**. Info, São Paulo, n. 268, p. 48-49, jun. 2008.
- DAVIS JR., C.A. (1997) **Aumentando a eficiência da solução de problemas de caminho mínimo em SIG**. In: GIS BRASIL, 3. Editora Sagres, Curitiba, 1997.
- DAYLEY, B., DAYLEY, B., DAYLEY, C.; **Node.js, MongoDB and Angular web development: the definitive guide to using the mean stack to build web applications**. 2 Edição. Pearson Education, 2018.
- DEVIANART. Disponível em: <https://about.deviantart.com/>, acesso em 17/08/2020.
- ELMASRI, R.E.; NAVATHE, S.B. **Sistemas de banco de dados**. 4.ed. São Paulo: Pearson Education do Brasil Ltda, 2005.
- FERNÁNDEZ-PÉREZ, V., ALONSO-GALICIA, P. E., RODRÍQUEZ-ARIZA, L., & del Mar FUENTES-FUENTES, M.; **Professional and personal social networks: A bridge to entrepreneurship for academics?**; European Management Journal, 2015.
- FLANAGAN, David. **Javascript: o guia definitivo**. 6. ed. Porto Alegre: Bookman, 2013.

Fivver. Fivver Announces Fourth Quarter and Full Year 2019 Results, fevereiro 2020. Disponível em: <https://investors.fiverr.com/press-releases/press-releases-details/2020/Fiverr-Announces-Fourth-Quarter-and-Full-Year-2019-Results/>, acesso em 17/08/2020.

GITHUB. Disponível em: <https://github.com/about>, acesso em 17/08/2020.

GITLAB. Disponível em: <https://about.gitlab.com/company/>, acesso em 17/08/2020.

LINKEDIN. Disponível em: <https://www.linkedin.com/company/linkedin>, acesso em 17/08/2020.

LUCCHESI, C.L. **Introdução à teoria dos grafos.** Rio de Janeiro: Instituto de Matemática Pura e Aplicada, 1979.

MILETTO, E. M., BERTAGNOLLI, S. C.; **desenvolvimento de software ii: introdução ao desenvolvimento web com Html, Css, Javascript e Php.** Eixo: Informação e Comunicação. Porto Alegre: Bookman, 2014.

PENTEADO, R. M. R.; SCHROEDER, R.; HOSS, D.; NANDE, J.; MAEDA, R. M.; COUTO, W. O. e HARA, C. S.; **Um estudo sobre bancos de dados em grafos nativos.** X ERBD-Escola Regional de Banco de Dados, 2014.

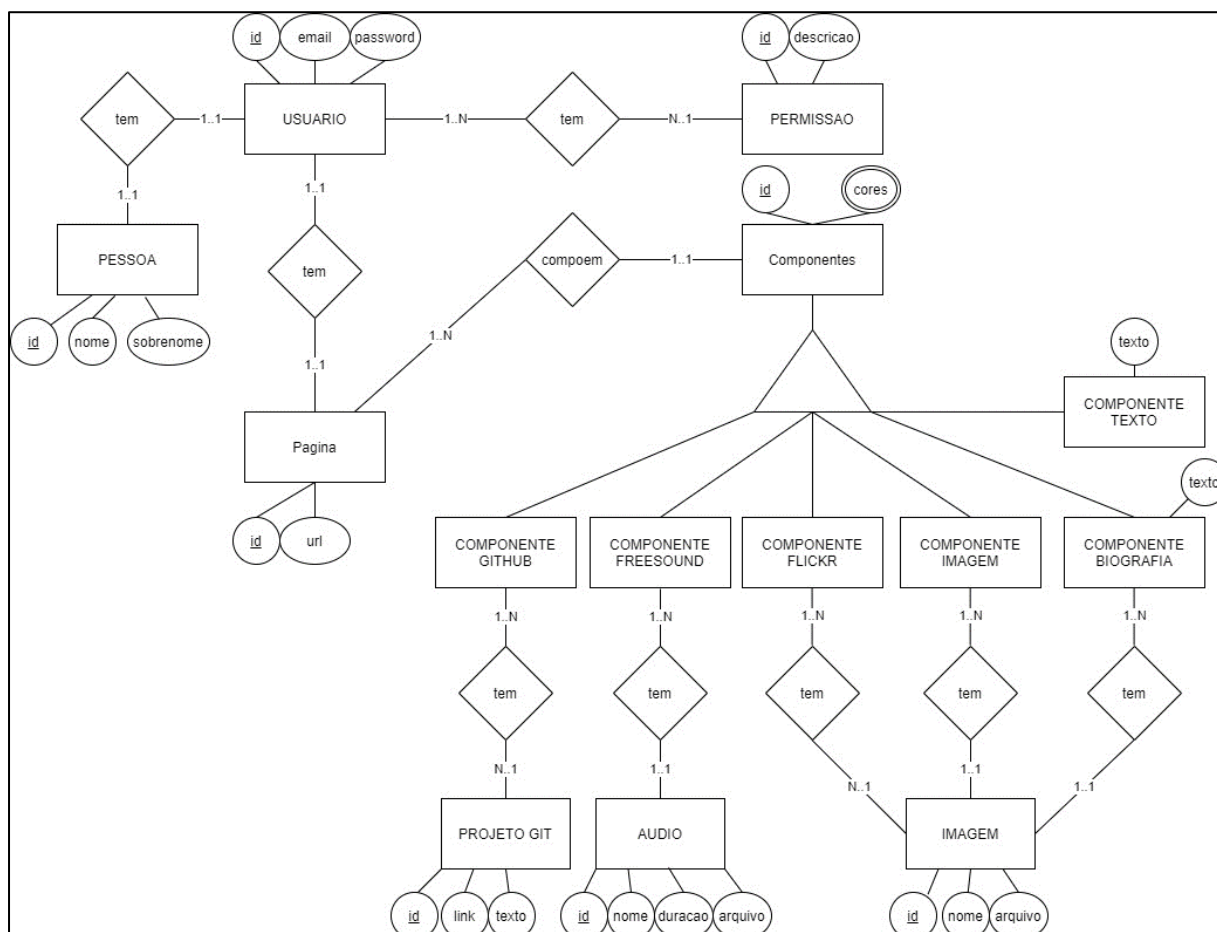
SOUNDCLOUD. Disponível em: <https://soundcloud.com/pages/contact>, acesso em 17/08/2020.

WATTHIER, L.; **Manifestações da linguagem oral na produção escrita do gênero carta de apresentação: aspectos discursivos/textuais e didáticos;** Tese (Doutorado em Letras); Cascavel, PR. UNIOESTE, 2016.

WATERMAN, M. A. **Teaching portfólios for summative and peer evaluation.** In: American association for higher education conference on assessment for higher education, 6. São Francisco, 1991.

APÊNDICE A – DIAGRAMA DE ENTIDADE RELACIONAMENTO

O diagrama de entidade relacionamento é utilizado para modelagem do banco de dados, e pode auxiliar o entendimento sobre como o software vai operar. No caso, cada usuário será uma pessoa com determinadas permissões sobre as páginas, e cada página de um usuário poderá conter componentes como os de texto, música e imagem.



FONTE: OS AUTORES (2021).

APÊNDICE B – DIAGRAMA DE SEQUÊNCIA

Diagrama de Sequência: 002 – Acessar Conta (login)

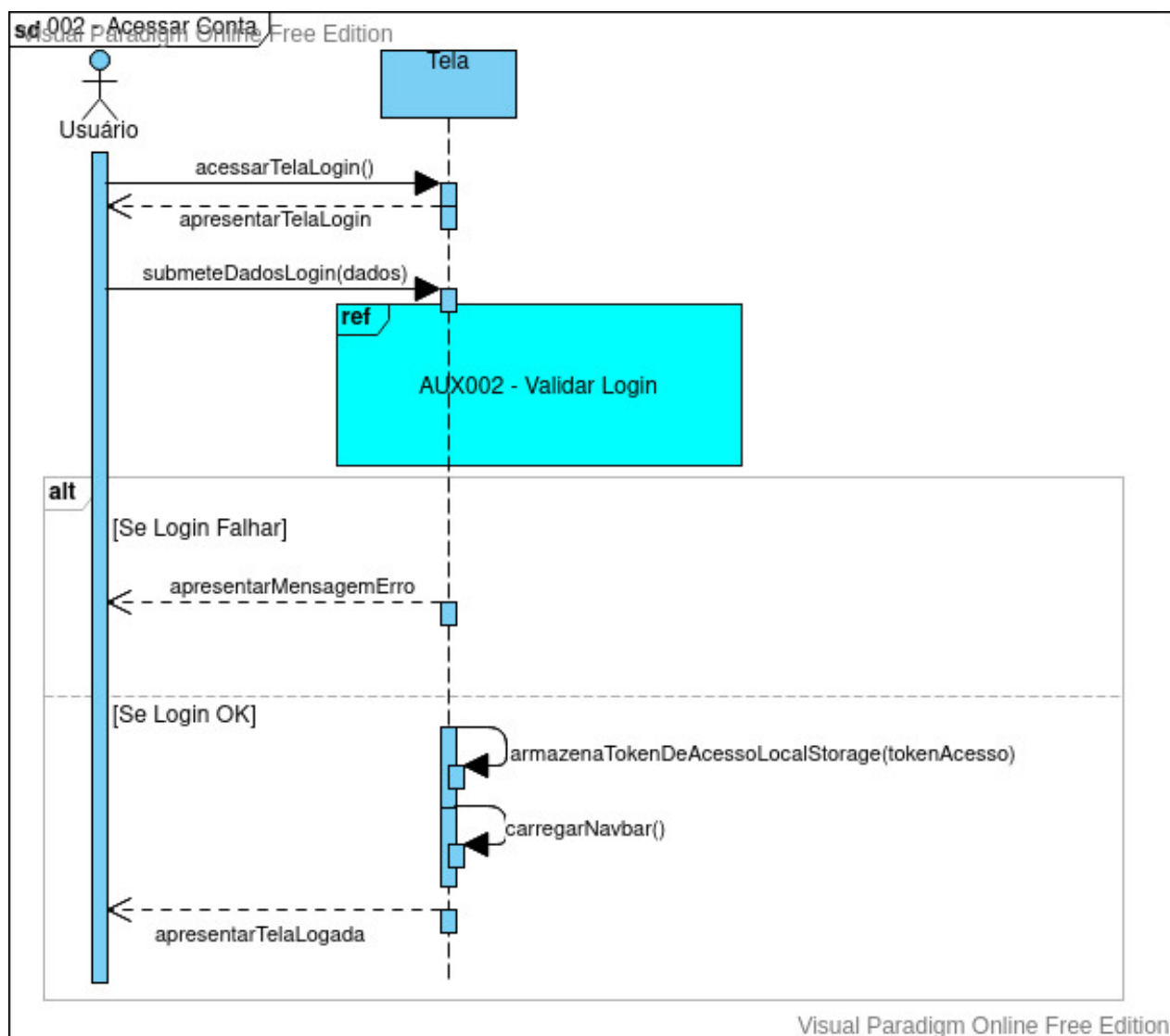


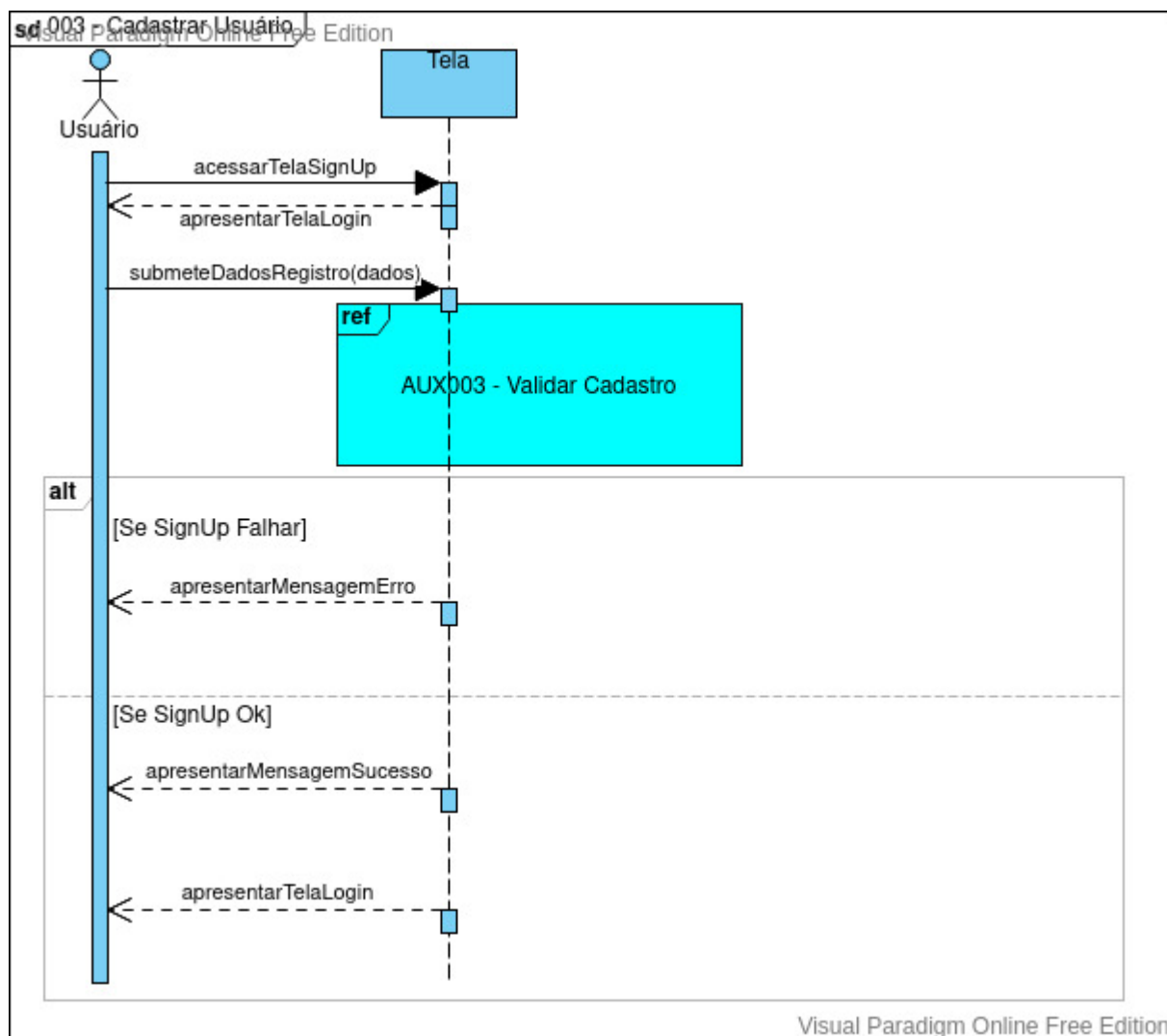
Diagrama de Sequência: 003 – Cadastrar Usuário

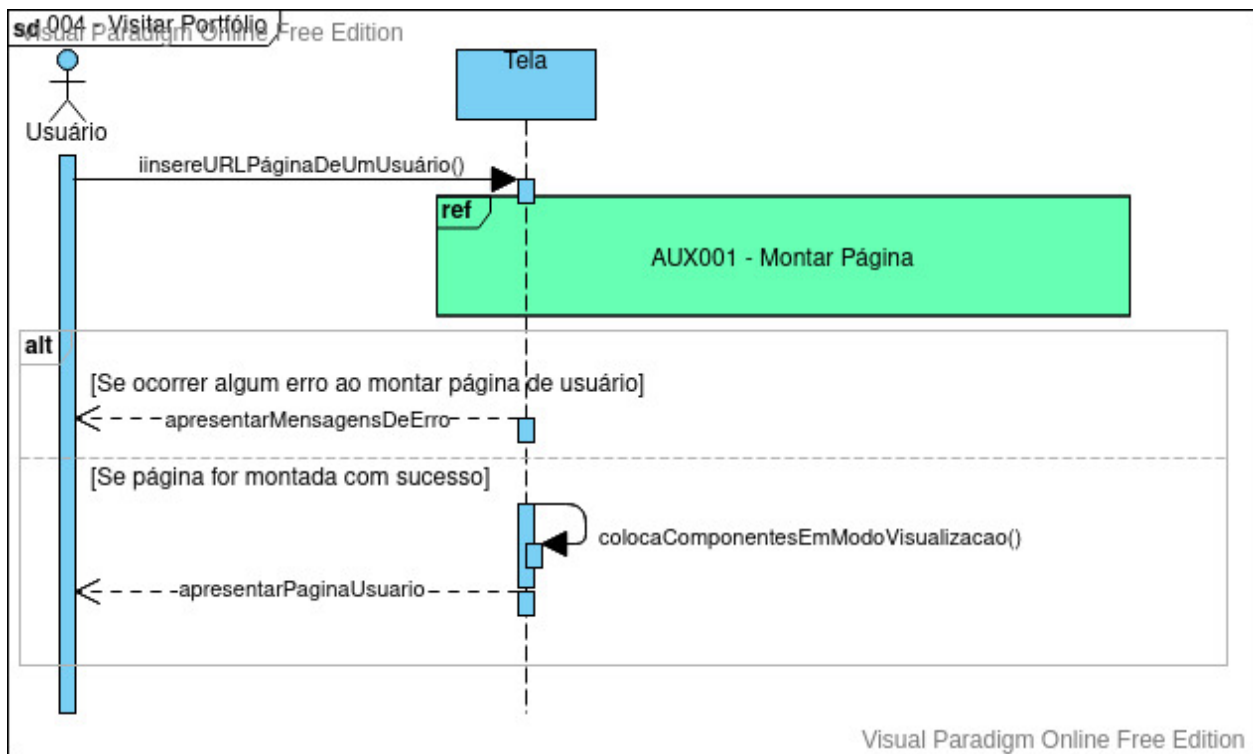
Diagrama de Sequência: 004 – Visitar Portfólio

Diagrama de Sequência: 005 – Manter Configuração de Usuário

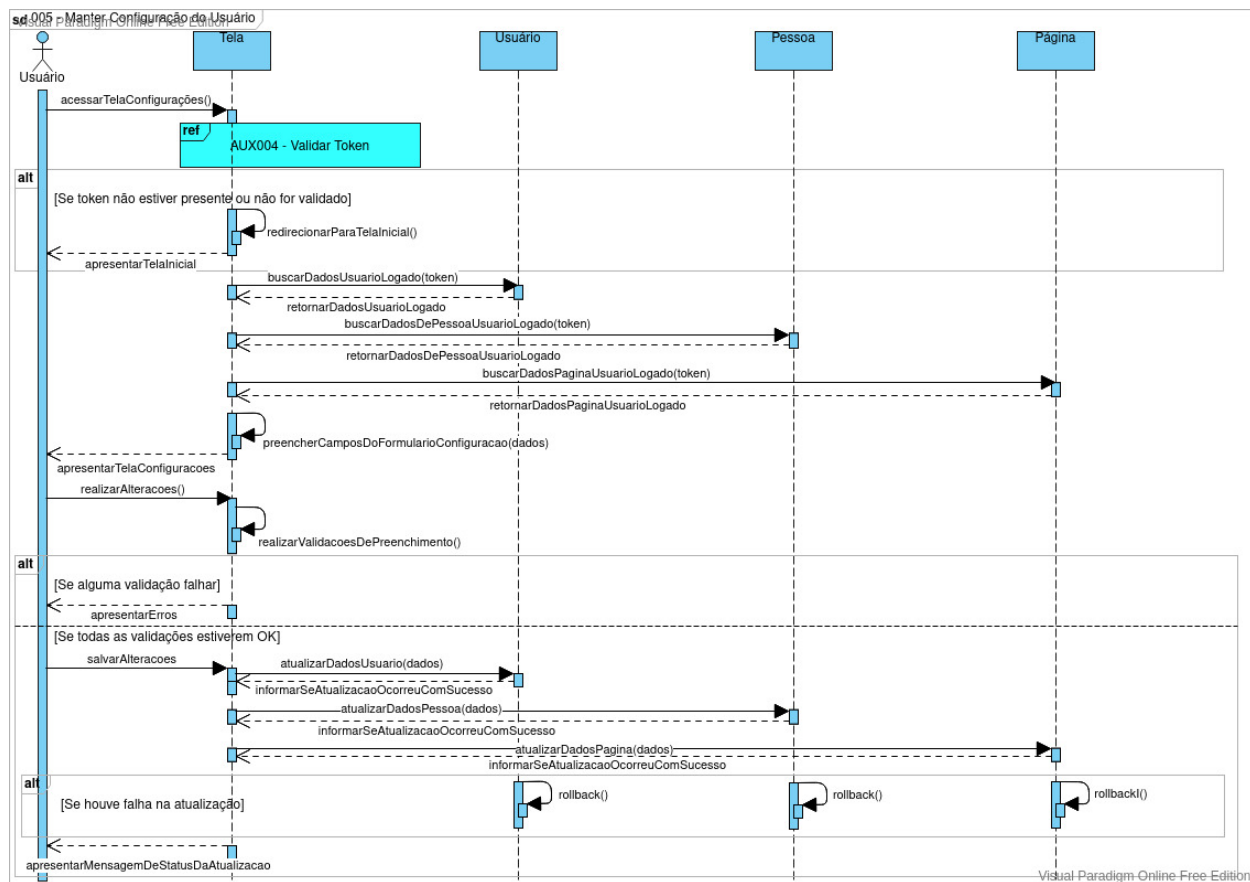


Diagrama de Sequência: 006 – Manter Componentes da Página

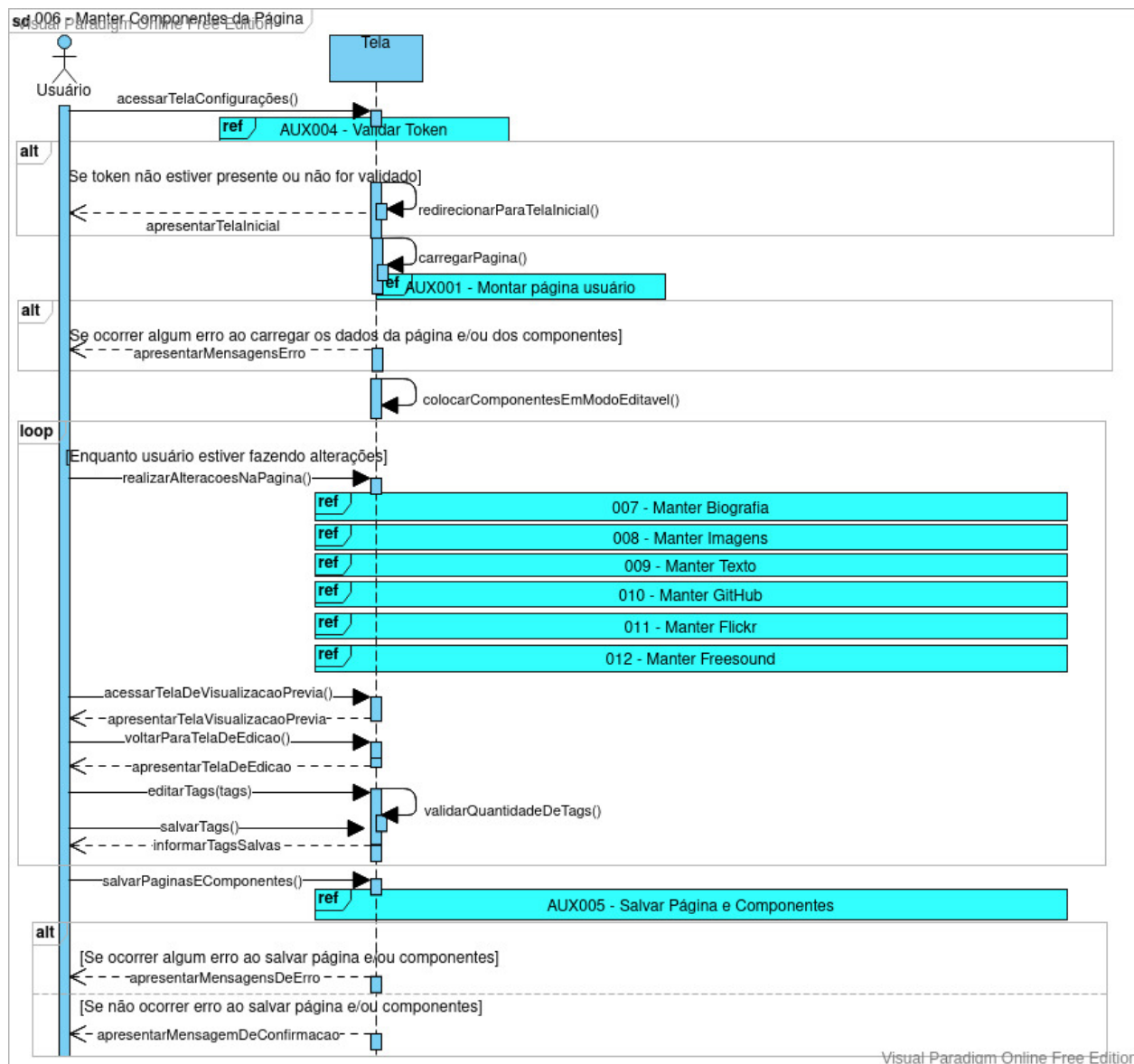


Diagrama de Sequência: 007 – Manter Biografia

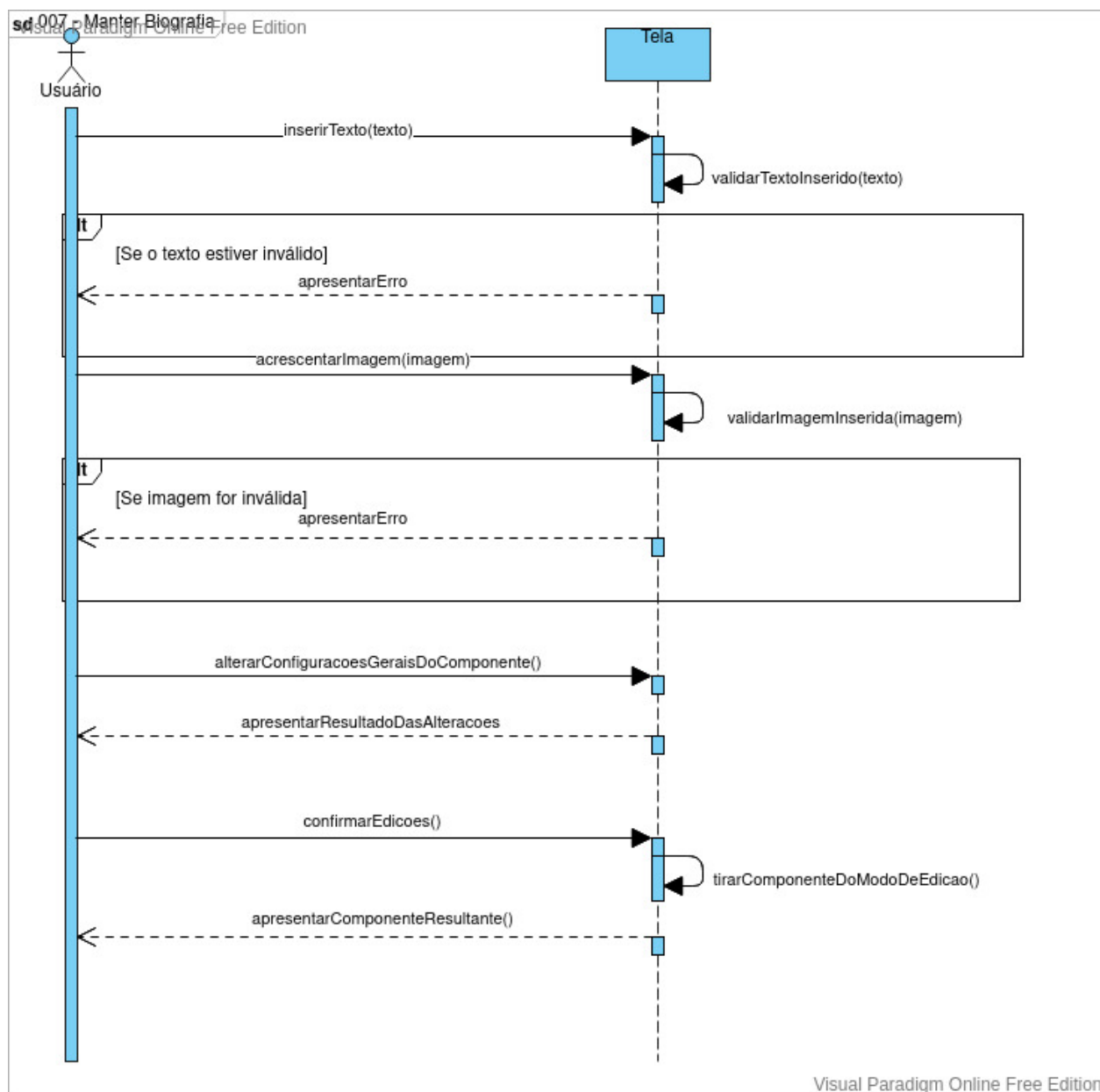


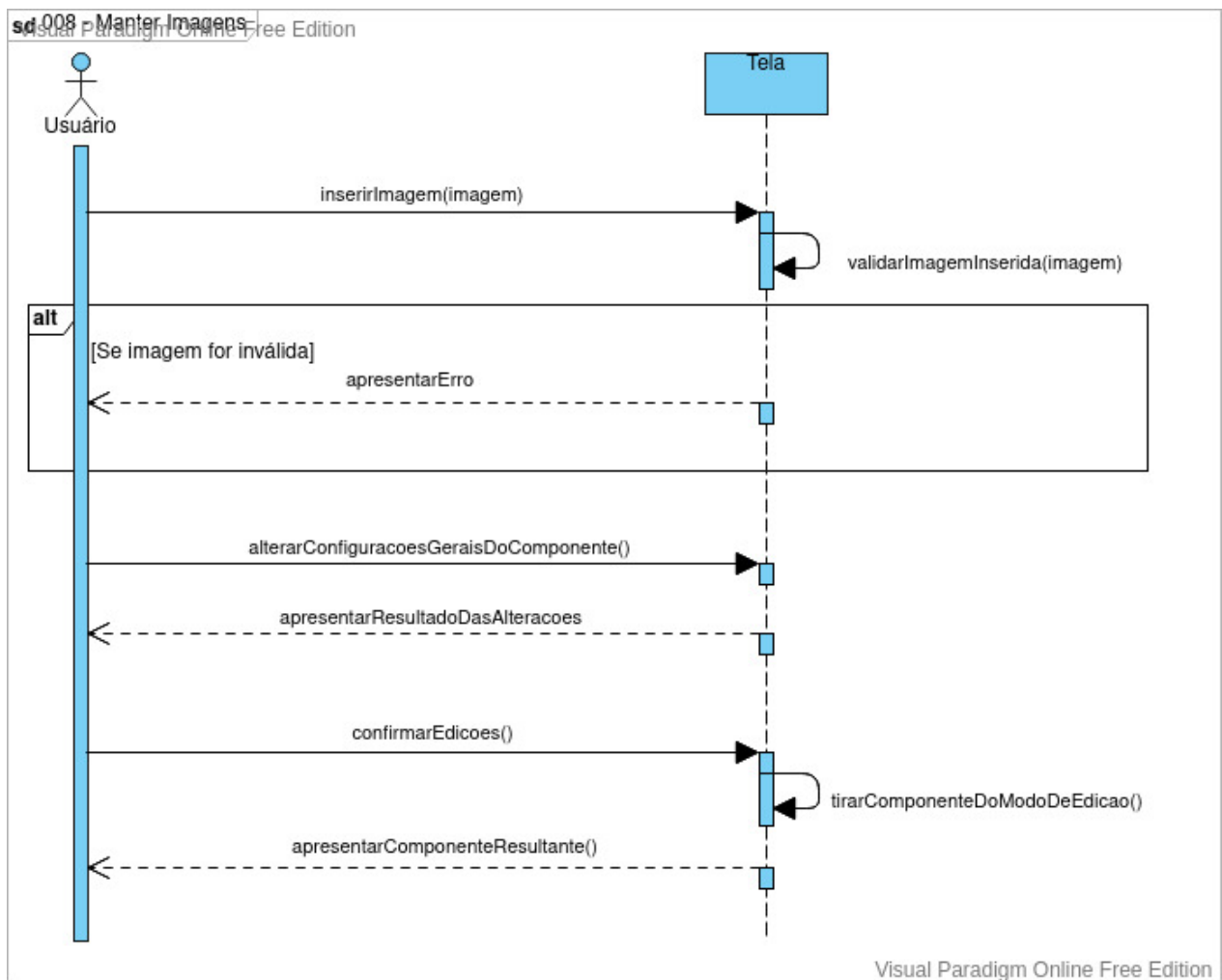
Diagrama de Sequência: 008 – Manter Imagens

Diagrama de Sequência: 009 – Manter Texto

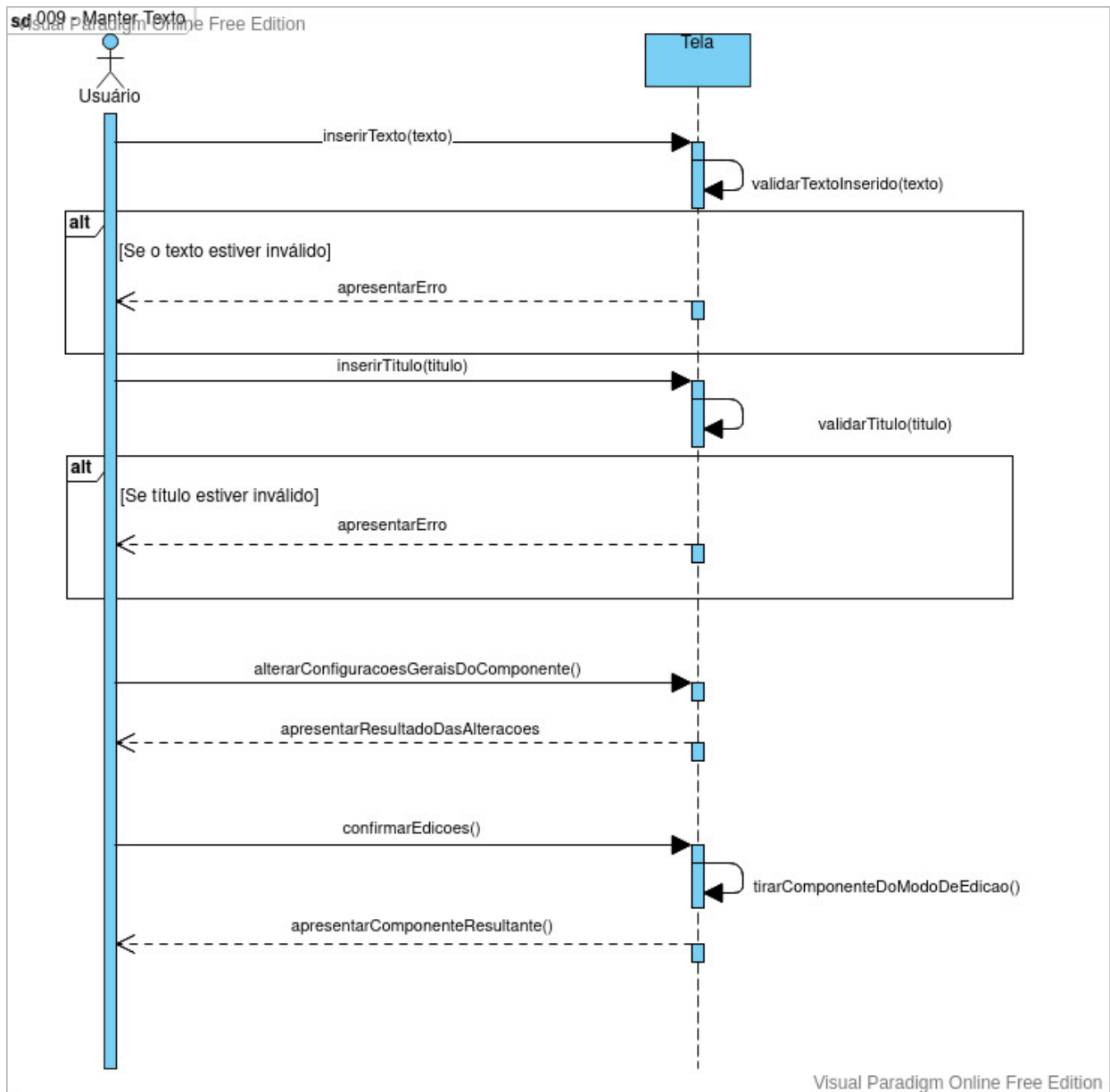


Diagrama de Sequência: 010 – Manter GitHub

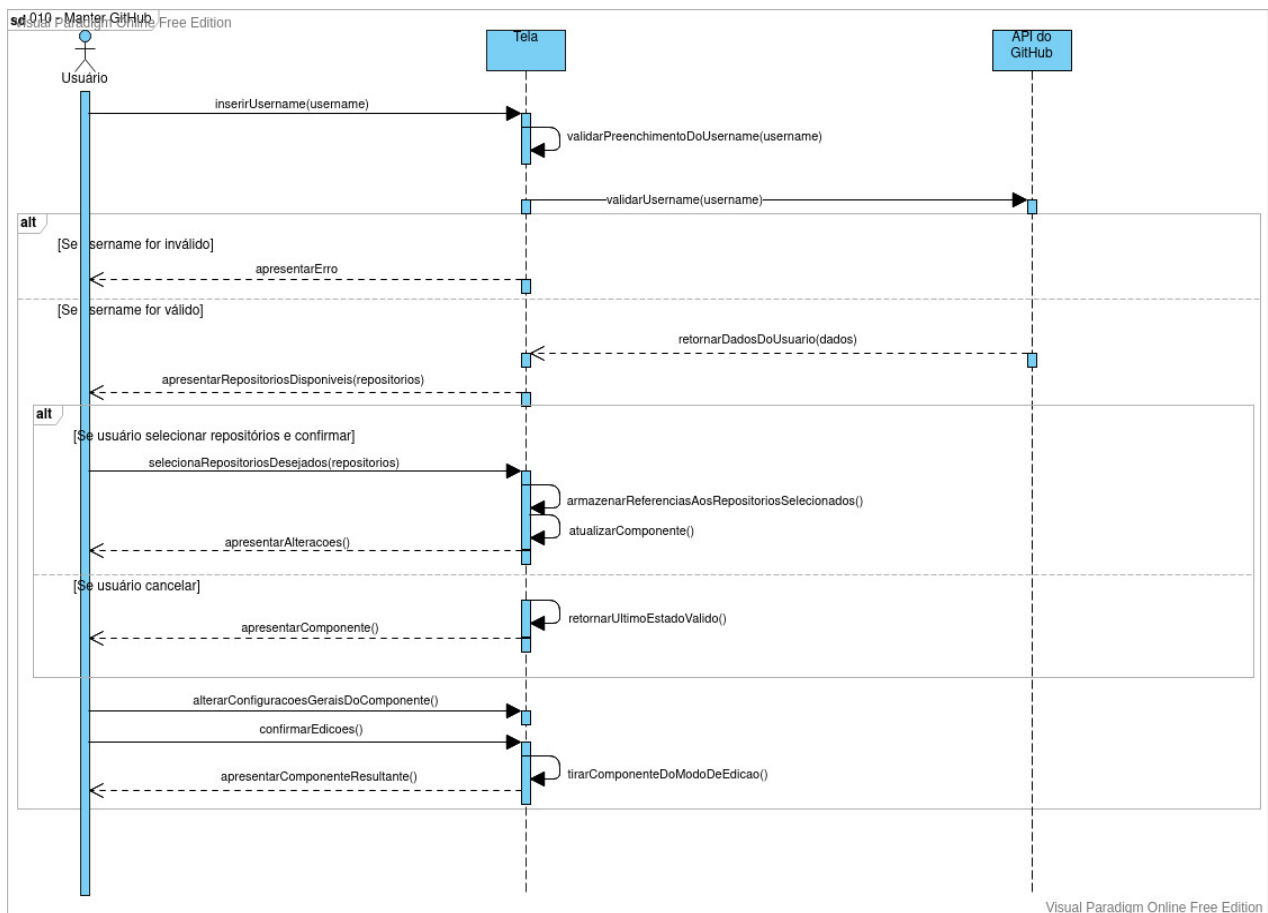


Diagrama de Sequência: 011 – Manter Flickr

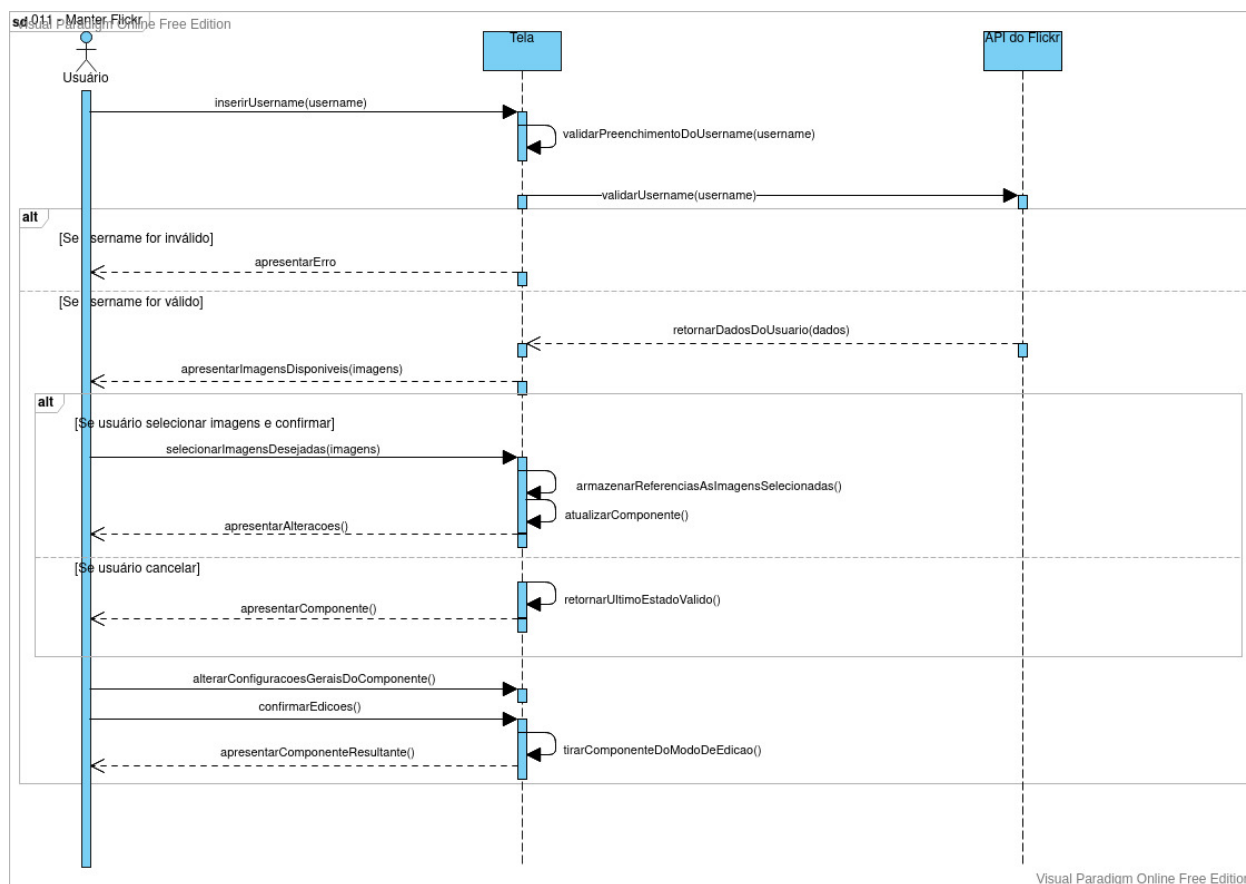


Diagrama de Sequência: 012 – Manter Freesound

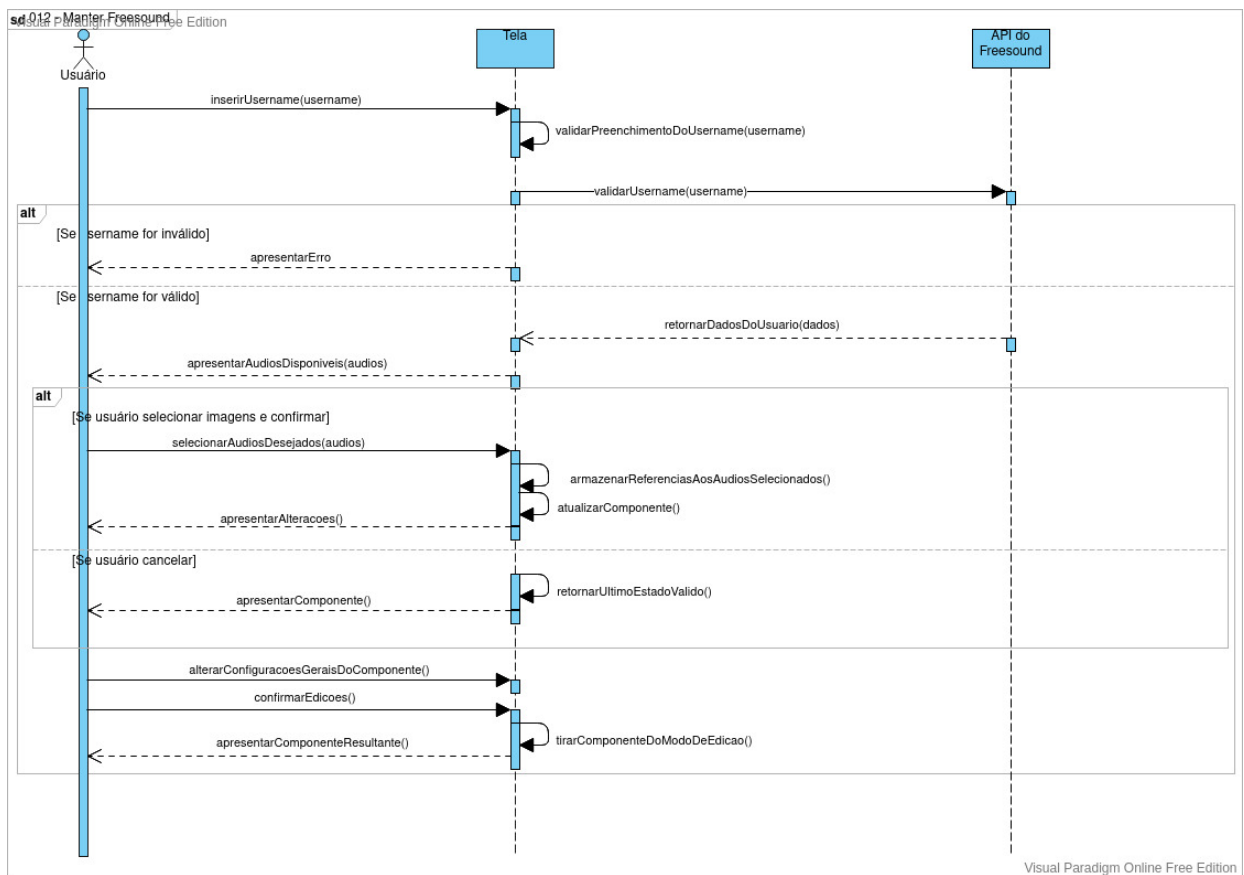


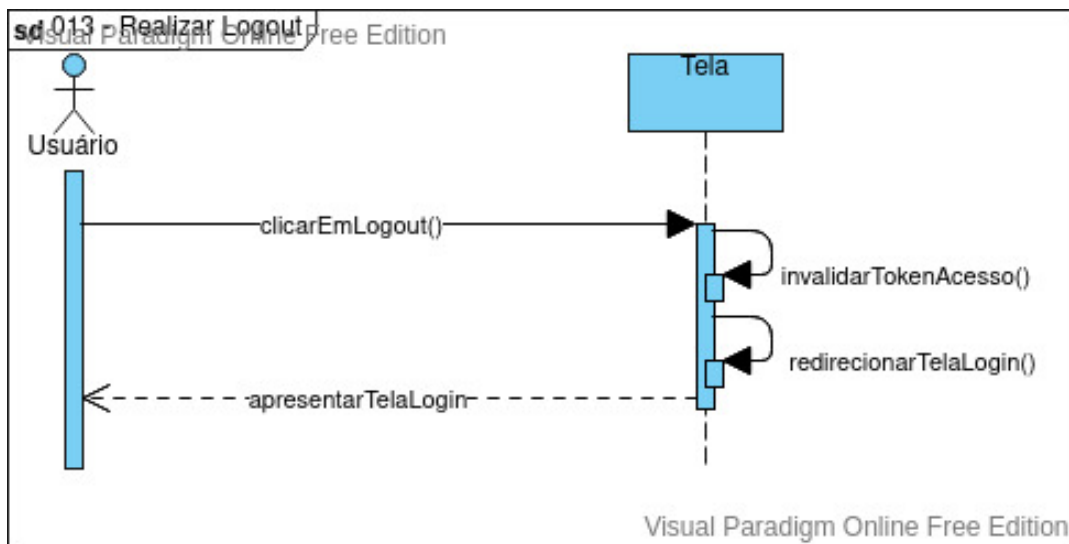
Diagrama de Sequência: 013 – Realizar Logout

Diagrama de Sequência: AUX001 – Montar Página

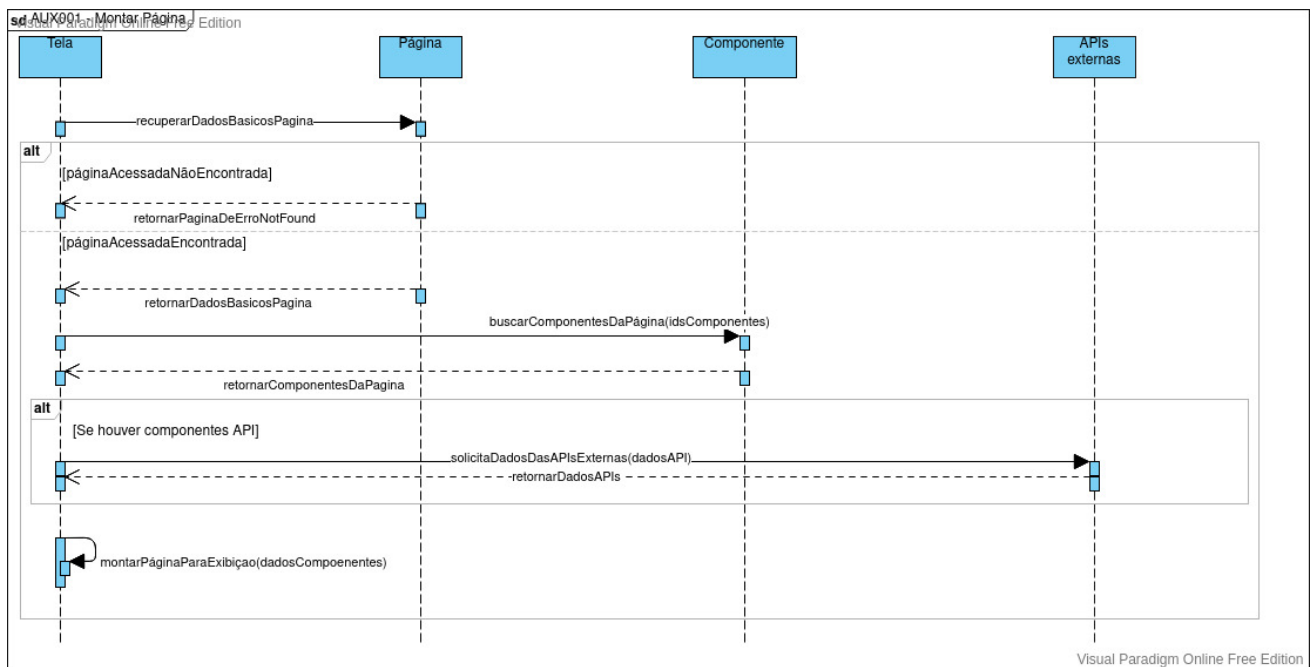


Diagrama de Sequência: AUX002 – Validar Login

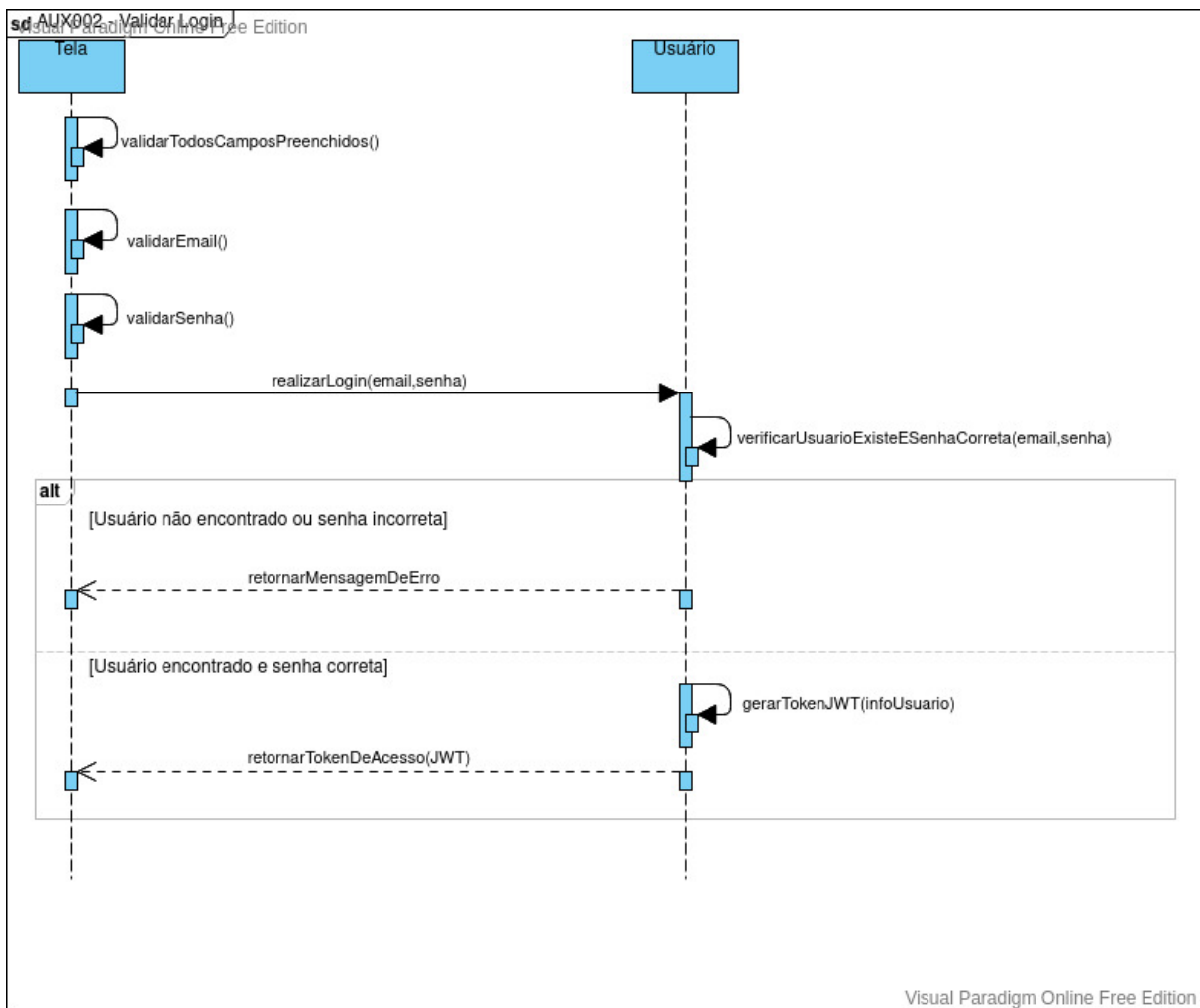


Diagrama de Sequência: AUX003 – Validar Cadastro

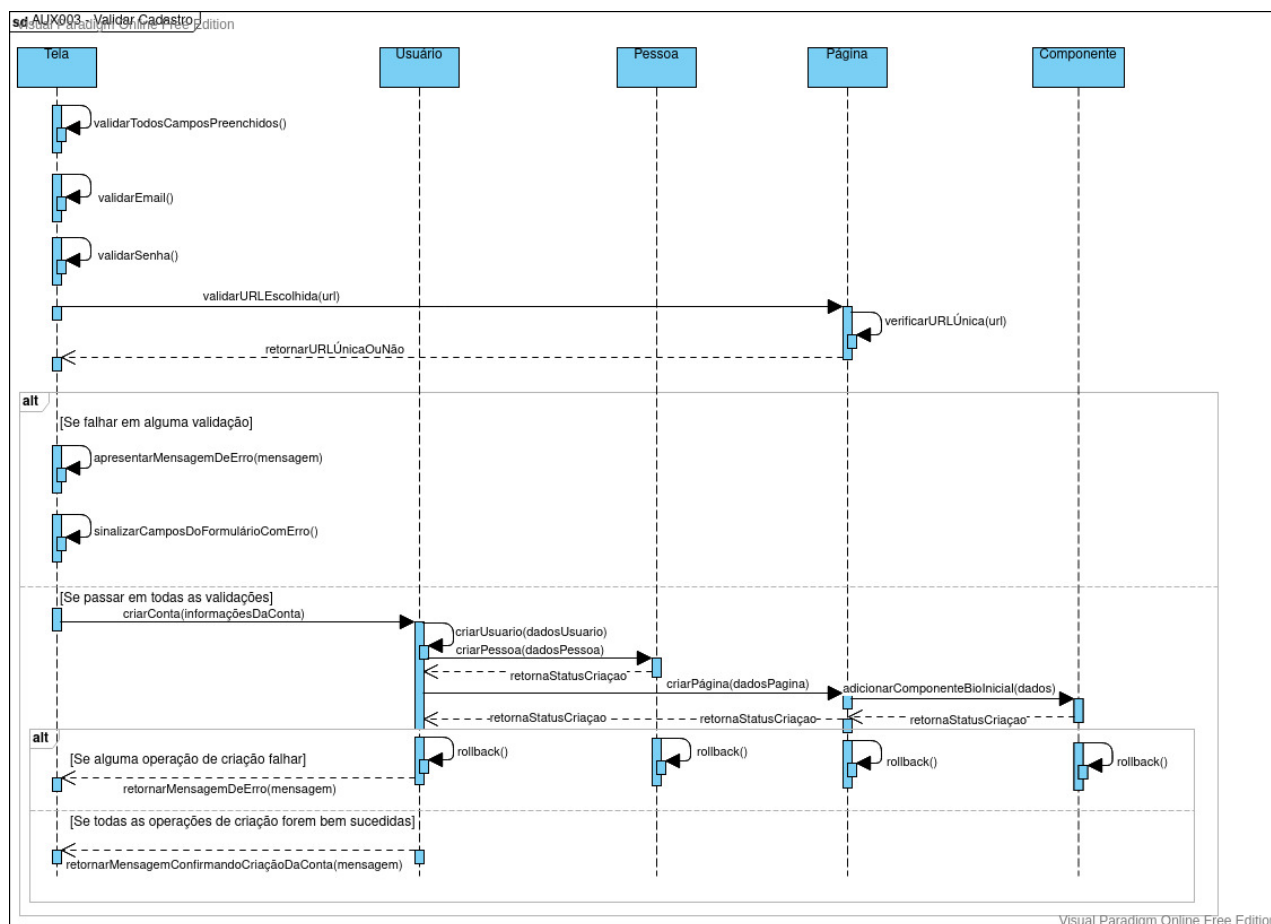


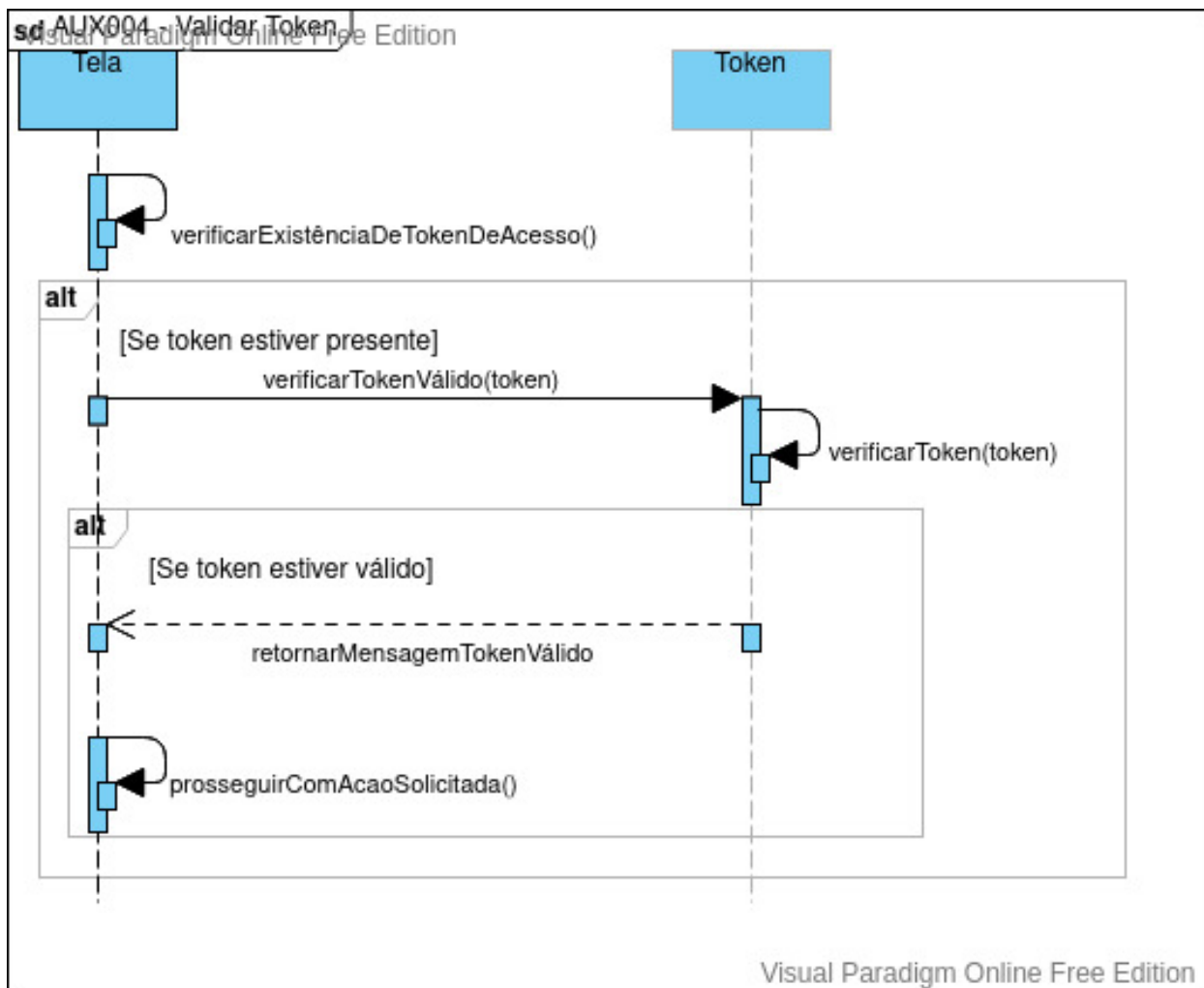
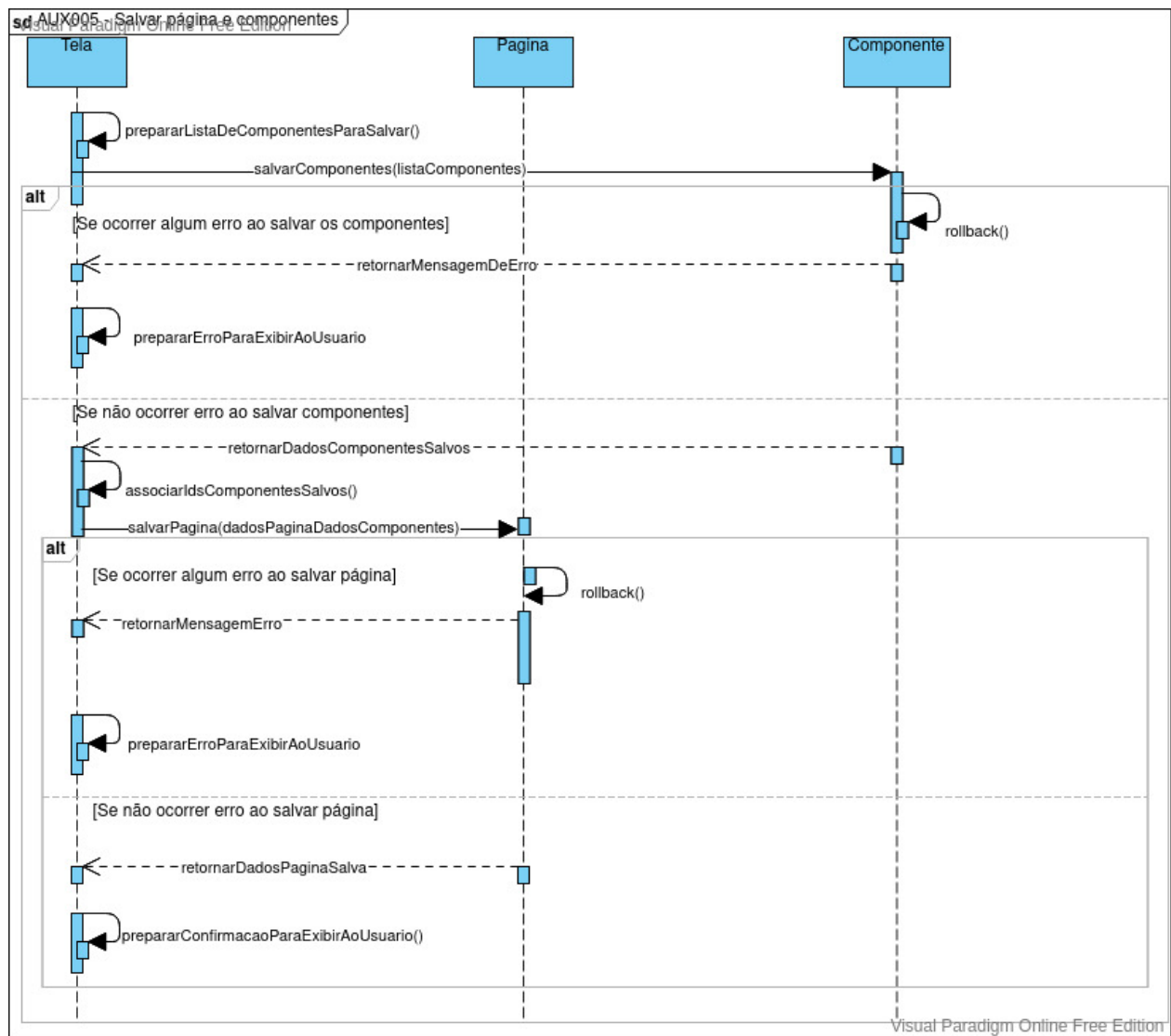
Diagrama de Sequência: AUX004 – Validar Token

Diagrama de Sequência: AUX005 – Salvar página e componentes

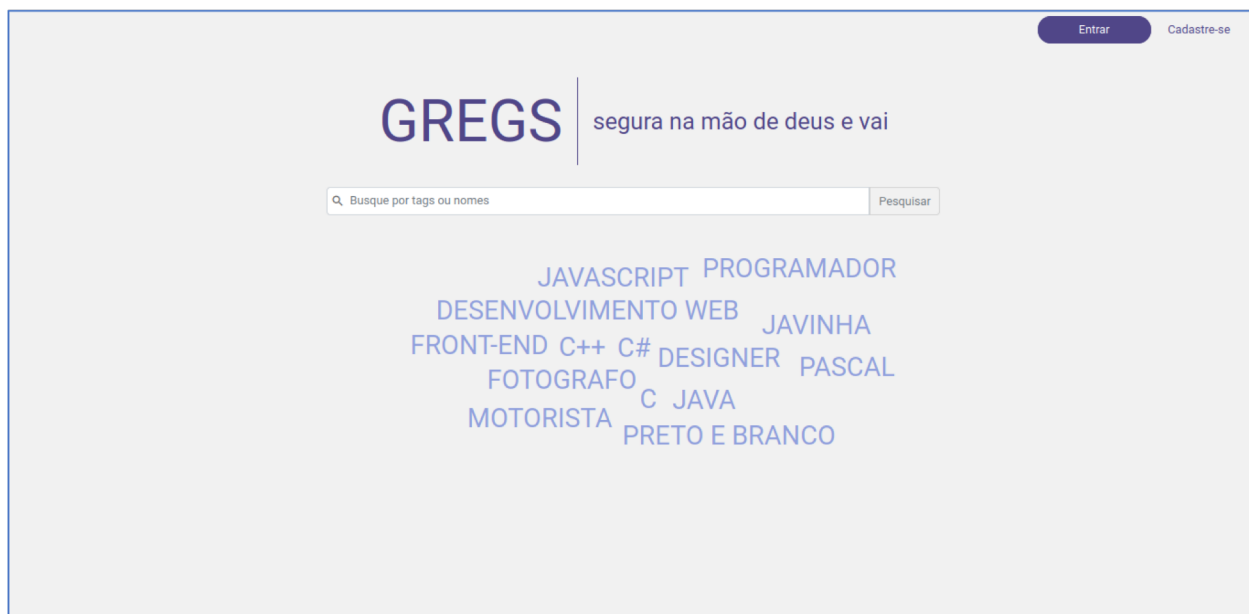


APÊNDICE C – ESPECIFICAÇÃO DOS CASOS DE USO

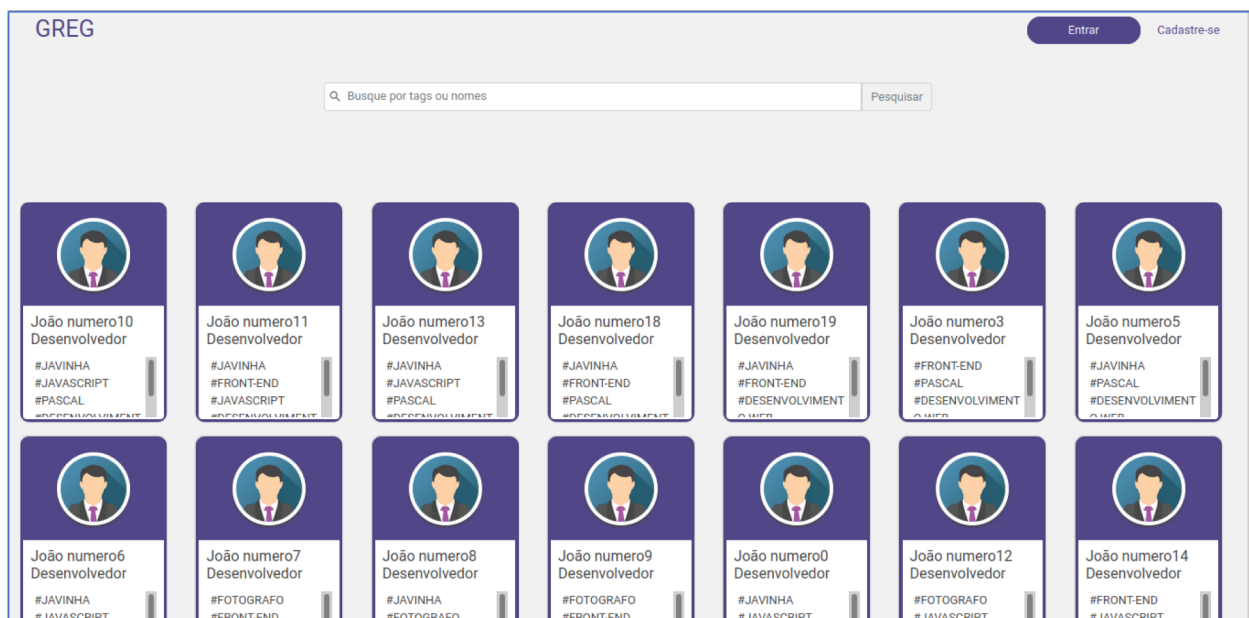
Especificação do Caso de Uso: UC 001 – Pesquisar Portfólios

Descrição: Este caso de uso serve para efetuar a pesquisa de Portfólios de Usuários na base de dados.

Data View: DV1 - Tela de Pesquisar Portfólios.



Data View: DV2 - Tela de Apresentar Resultados Pesquisados.



Ator Primário: Cliente

Fluxo de Eventos Principal:

1. O sistema busca as *Tags* mais recorrentes e preenche a Word Cloud.
2. O sistema apresenta a tela DV1.
3. O usuário preenche o campo de busca com o texto desejado.
4. O usuário pressiona o botão “Pesquisar”.
5. O sistema busca os resultados semelhantes ao texto inserido no campo de busca.
6. O sistema apresenta a tela DV2.
7. O caso de uso é encerrado.

Fluxos Alternativos:

A1: Botão “Entrar” pressionado:

1. O sistema chama o caso de uso UC 002 – Realizar Login.
2. O caso de uso é encerrado.

A2: Botão “Cadastra-se” pressionado:

1. O sistema chama o caso de uso UC 003 – Cadastrar Usuário.
2. O caso de uso é encerrado.

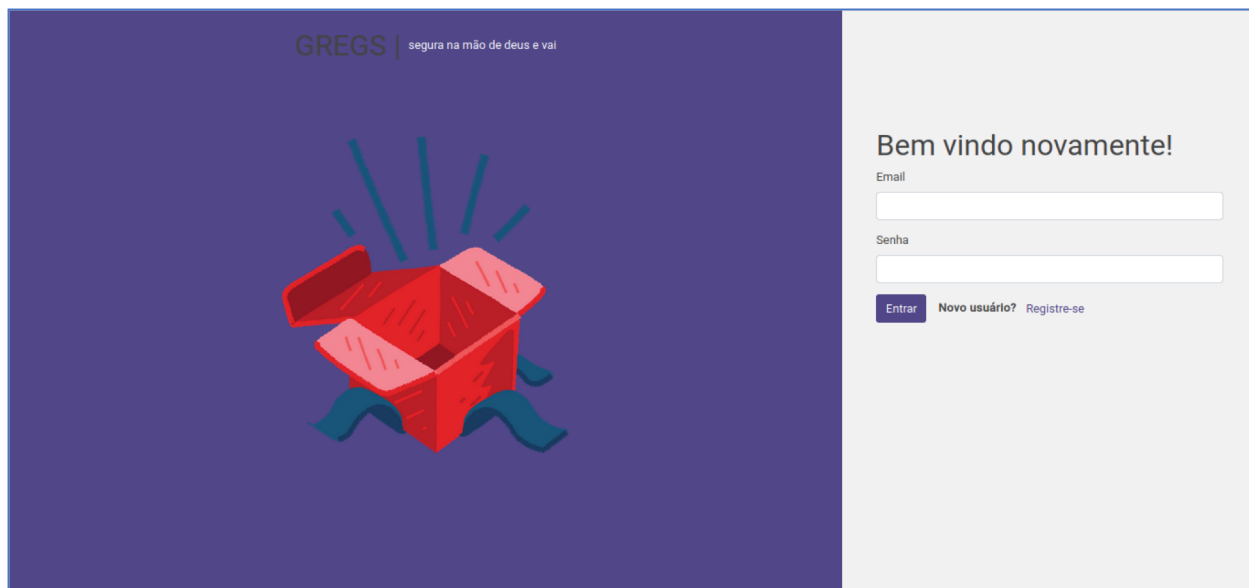
A3: Navegar diretamente a um portfólio:

1. O usuário insere no navegador a URL de um Portfólio.
2. O sistema chama o caso de uso UC 004 – Visitar Portfólio.
3. O caso de uso é encerrado.

Fluxos de Exceção:

E1: Texto não preenchido e botão “Pesquisar” pressionado

1. O sistema emite a mensagem “Preencha ao menos um caractere”

Especificação do Caso de Uso: UC 002 – Acessar conta (login)**Descrição:** Este caso de uso serve para que o Usuário realize o Login no Sistema.**Data View:** DV3 - Tela de Realizar Login.**Ator Primário:** Usuário**Fluxo de Eventos Principal:**

1. O sistema apresenta a tela DV3.
2. O usuário preenche os campos Email e Senha com suas informações.
3. O usuário pressiona o botão “Entrar”.
4. O sistema chama o caso de uso UC 001 – Pesquisar Portfólios.
5. O caso de uso é encerrado.

Fluxos Alternativos:**A1:** Botão “Registre-se” pressionado:

1. O sistema chama o caso de uso UC 003 – Cadastrar Usuário.
2. O caso de uso é encerrado.

Fluxos de Exceção:**E1:** Usuário informa email ou senha não cadastrados:

1. O sistema emite a mensagem “Email ou senha incorretos”
2. O uso de caso é reiniciado.

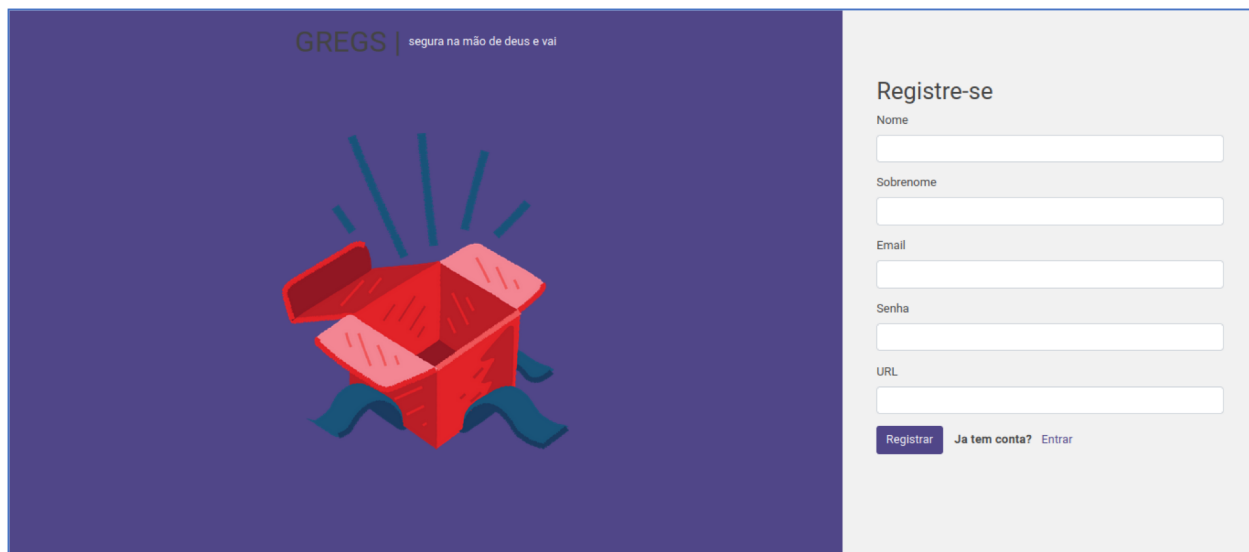
E2: Usuário não preenche email ou senha e pressiona “Entrar”:

1. O sistema emite a mensagem “Email e senha tem que ser preenchidos”

Especificação do Caso de Uso: UC 003 – Cadastrar Usuário

Descrição: Este caso de uso serve para que o Usuário realize o Cadastro no Sistema.

Data View: DV4 - Tela de Realizar Cadastro.



The image shows a web interface for user registration. On the left, there is a dark purple banner with the logo 'GREGS | segura na mão de deus e vai' and an illustration of a red gift box with blue ribbons and rays. On the right, there is a light gray panel titled 'Registre-se' containing five input fields: 'Nome', 'Sobrenome', 'Email', 'Senha', and 'URL'. Below these fields are three buttons: 'Registrar' (dark purple), 'Ja tem conta?' (light gray), and 'Entrar' (light gray).

Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema apresenta a tela DV4.
2. O usuário preenche o campo Nome.
3. O usuário preenche o campo Sobrenome.
4. O usuário preenche o campo Email.
5. O usuário preenche o campo Senha.
6. O usuário preenche o campo Url com o nome desejado para seu portfólio.
7. O usuário pressiona o botão “Registrar”.
8. O sistema chama o caso de uso UC 002 – Realizar Login.
9. O caso de uso é encerrado.

Fluxos Alternativos:

A1: Botão “Entrar” pressionado:

1. O sistema chama o caso de uso UC 002 – Realizar Login.
2. O caso de uso é encerrado.

Fluxos de Exceção:

E1: Usuário não preenche algum campo solicitado:

1. O sistema emite a mensagem “Todos os campos devem ser preenchidos”

E2: Usuário não preenche o campo email conforme a regra de negócio:

1. O sistema emite a mensagem “O campo email deve ser preenchido no formato nome@domínio”

E3: Usuário não preenche o campo senha conforme a regra de negócio:

1. O sistema emite a mensagem “A senha deverá conter no mínimo 6 dígitos”

Regras de Negócio:

R1: Formato do Email:

1. O email deve estar no padrão “nome@dominio”.

R2: Formato da Senha:

1. A senha deverá conter no mínimo 6 caracteres.

Especificação do Caso de Uso: UC 004 – Visitar Portfólios

Descrição: Este caso de uso serve para apresentar o Portfólio de um usuário.

Data View: DV5 - Tela do Portfólio do Usuário.



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema busca as Informações do Portfólio e preenche a página com os componentes
2. O sistema apresenta a tela DV5.
3. O caso de uso é encerrado.

Fluxos Alternativos:

A1: Botão “Voltar” pressionado:

3. O sistema chama o caso de uso UC 001 – Pesquisar Portfólios.
4. O caso de uso é encerrado.

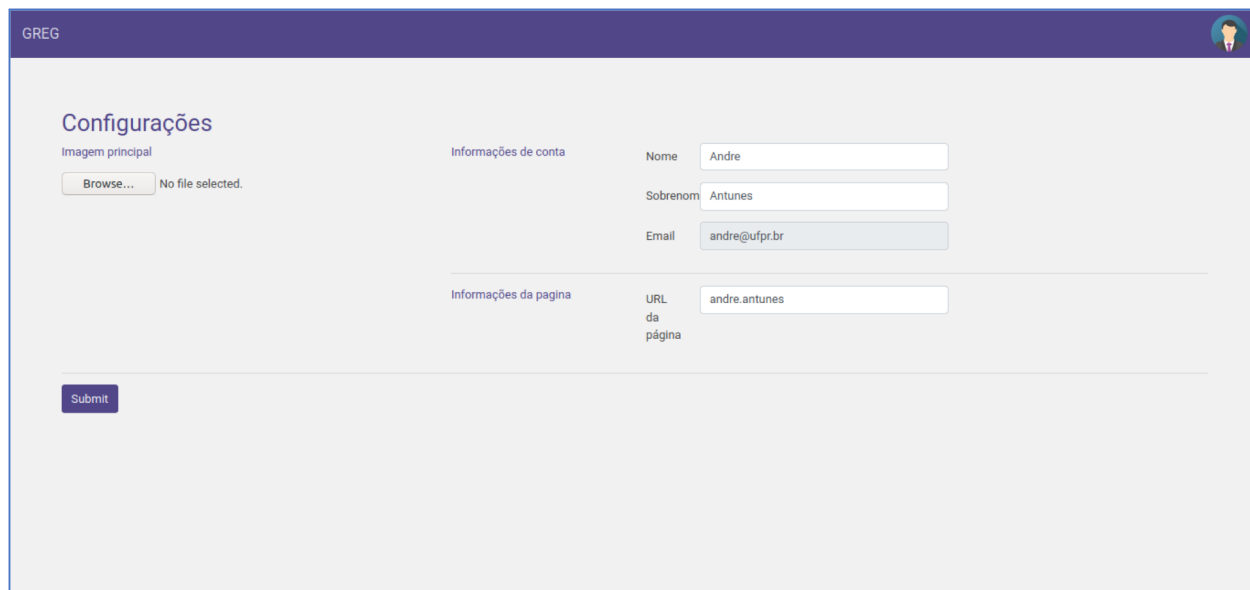
A2: A página possui componentes que buscam dados em APIs externas:

1. O sistema faz a requisição HTTP para a API desejada.
2. O sistema filtra os dados conforme os dados sobre o componente salvos no banco de dados
3. O sistema apresenta a tela DV5
4. O caso de uso é encerrado

Especificação do Caso de Uso: UC 005 – Manter Configuração de Usuário

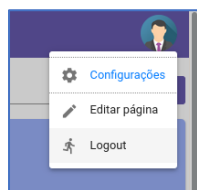
Descrição: Este caso de uso serve para manter as informações do usuário.

Data View: DV6 - Tela de Manter Usuário.



The screenshot shows a web interface titled 'Configurações' (Settings) under the 'GREG' header. On the left, there is a section for 'Imagem principal' (Main image) with a 'Browse...' button and the text 'No file selected.'. To the right, under 'Informações de conta' (Account information), there are three input fields: 'Nome' (Name) with 'Andre', 'Sobrenom' (Surname) with 'Antunes', and 'Email' with 'andre@ufpr.br'. Below this, under 'Informações da página' (Page information), there is an input field for 'URL da página' (Page URL) with 'andre.antunes'. A 'Submit' button is located at the bottom left of the form area.

Data View: DV7 – Menu do Usuário



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema busca as Informações do Usuário e preenche a página com o nome, sobrenome, email e url cadastradas pelo Usuário.
2. O sistema apresenta a tela DV6.
3. O usuário pressiona o botão “Browse” sob o título “Imagem Pessoal”.
4. O usuário seleciona um arquivo de imagem de seu sistema.
5. O usuário seleciona o campo “Nome” e, se desejar, altera o texto;
6. O usuário seleciona o campo “Sobrenome” e, se desejar, altera o texto;
7. O usuário seleciona o campo “URL da página” e, se desejar, altera o texto;
8. O usuário pressiona o botão “Submit”
9. O sistema inclui a alteração dos dados.
10. O sistema emite a mensagem “Dados atualizados com sucesso”

11. O caso de uso é encerrado.

Fluxos Alternativos:

A1: No Menu do Usuário – Canto Superior Direito, sob a imagem do Usuário, Botão “Configurações” pressionado:

1. O caso de uso é reiniciado.

A2: No Menu do Usuário – Canto Superior Direito, sob a imagem do Usuário, Botão “Editar Página” pressionado:

1. O sistema chama o caso de uso UC 006 – Manter Página.
2. O caso de uso é encerrado.

A3: No Menu do Usuário – Canto Superior Direito, sob a imagem do Usuário, Botão “Logout” pressionado:

1. O sistema chama o caso de uso UC 001 – Pesquisar Portfólios.
2. O caso de uso é encerrado.

Fluxos de Exceção:

E1: Usuário não preenche algum campo solicitado:

2. O sistema emite a mensagem “Todos os campos devem ser preenchidos”

E2: Usuário não preenche o campo email conforme a regra de negócio:

2. O sistema emite a mensagem “O campo email deve ser preenchido no formato nome@domínio”

E3: Usuário não preenche o campo senha conforme a regra de negócio:

2. O sistema emite a mensagem “A senha deverá conter no mínimo 6 dígitos”

Regras de Negócio:

R1: Formato do Email:

2. O email deve estar no padrão “nome@dominio”.

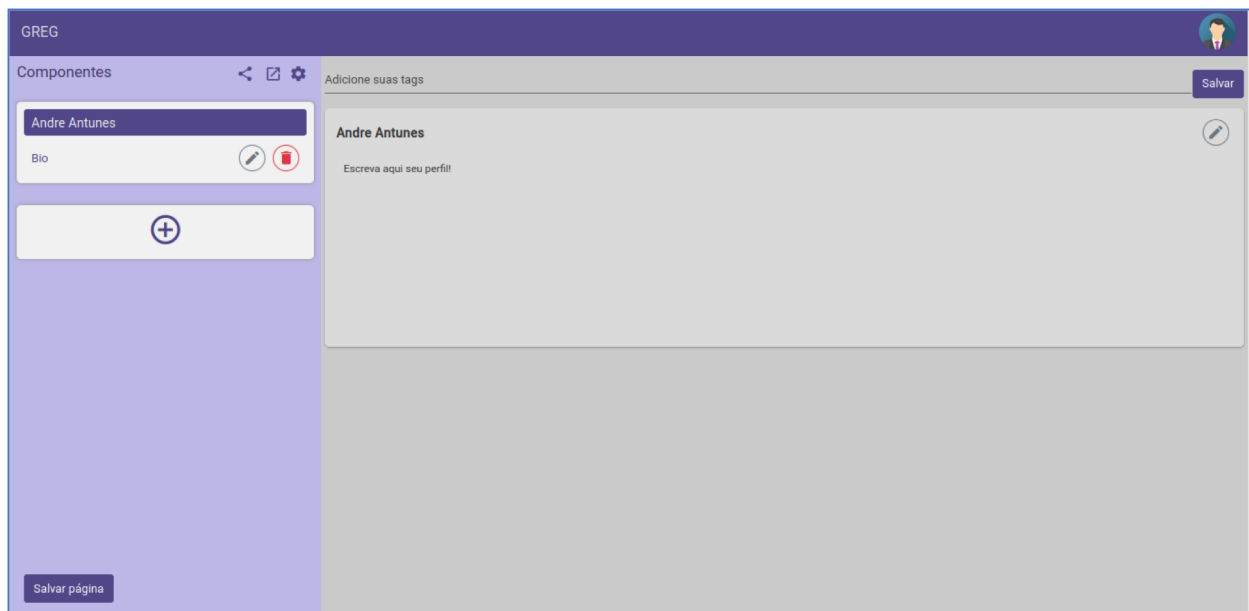
R2: Formato da Senha:

2. A senha deverá conter no mínimo 6 caracteres.

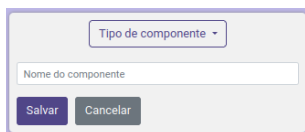
Especificação do Caso de Uso: UC 006 – Manter Componentes da Página

Descrição: Este caso de uso serve para manter as informações dos componentes da página de Portfólio do usuário.

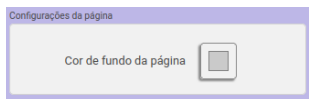
Data View: DV8 - Tela de Manter Página.



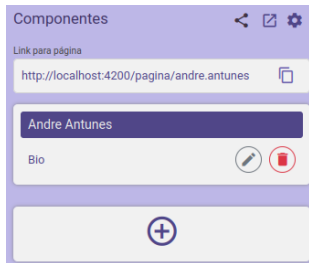
Data View: DV9 – Adicionar Componentes.



Data View: DV10 – Configurações da Página



Data View: DV11 – Menu da Página



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema cria um componente vazio Biografia (BIO)
2. O sistema busca as informações do Usuário e preenche o título do Componente Biografia com o nome do Usuário.
3. O sistema apresenta a tela DV8.
4. O usuário seleciona o campo *Tags*, sob o placeholder “Adicione suas *tags*”.
5. O usuário insere no campo *Tags* o texto correspondente a *Tag* que deseja incluir.
6. O usuário pressiona o botão “Enter” para confirmar a inclusão da *Tag*.
7. O usuário pressiona “Salvar”.
8. O sistema inclui a alteração no Componente de *Tags* da página.
9. O usuário pressiona “Salvar Página”.
10. O sistema emite a mensagem “Página atualizada com sucesso”
11. O caso de uso é encerrado.

Fluxos Alternativos:

A1: No Menu do Usuário – Canto Superior Direito, sob a imagem do Usuário, Botão “Configurações” pressionado:

1. O sistema chama o caso de uso UC 005 – Manter Usuário.
2. O caso de uso é encerrado.

A2: No Menu do Usuário – Canto Superior Direito, sob a imagem do Usuário, Botão “Editar Página” pressionado:

1. O caso de uso é reiniciado.

A3: No Menu do Usuário – Canto Superior Direito, sob a imagem do Usuário, Botão “Logout” pressionado:

1. O sistema chama o caso de uso UC 001 – Pesquisar Portfólios.
2. O caso de uso é encerrado.

A4: Botão “Compartilhar” pressionado:

1. O sistema busca informações sobre a URL do usuário.
2. O sistema apresenta a url para da página na tela DV11 – Compartilhar Página

A5: Botão “Configurar” pressionado:

1. O sistema busca informações sobre a cor de fundo da página do usuário.
2. O sistema apresenta a tela DV10 – Conifgurações de Página
3. O usuário seleciona a cor de fundo (Color Picker)
4. O sistema inclui a alteração no Componente Página

A6: Botão “Prévia” pressionado:

1. O sistema busca informações sobre a URL do usuário logado.
2. O sistema chama o caso de uso UC 004 – Visitar Portfólios.
3. O caso de uso é encerrado.

A7: Adicionar Componente

1. O usuário pressiona o Botão “+”.
2. O sistema busca informações sobre os tipos de Componentes disponíveis: “Texto”, “Biografia”, “Imagem”, “Flickr”, “Freesound” e “GitHub”.
3. O sistema apresenta a tela DV9 – Adicionar Componentes.
4. O usuário seleciona o tipo de Documento desejado.
5. O usuário preenche o nome do componente.
6. O usuário pressiona salvar.
7. O sistema inclui a adição do componente na página.
8. O sistema chama o caso de uso Manter Componente específico para o tipo de Componente adicionado.

A8: Remover Componente

1. O usuário pressiona o Botão de Deletar (Lixeira) do Componente desejado;
2. O sistema exclui o componente da página

A9: Editar Componente

1. O usuário pressiona o Botão de Editar (Lápis) do Componente desejado
2. O sistema busca as informações sobre o componente desejado
3. O sistema chama o caso de uso Manter Componente específico para o Componente selecionado.

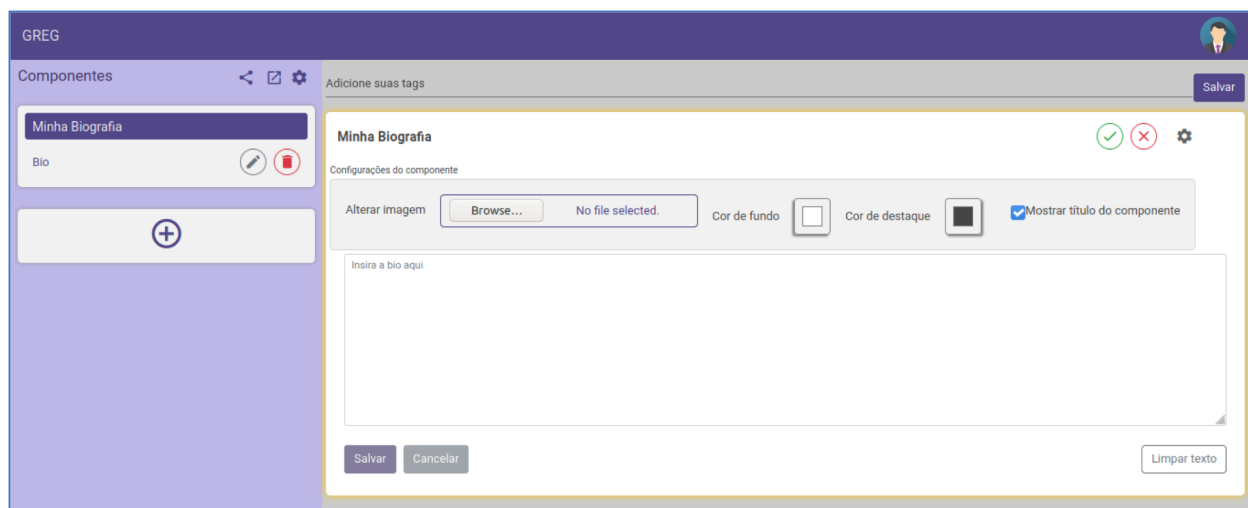
Regras de Negócio:**R1:** Título:

1. O título do componente conter até 255 caracteres.

Especificação do Caso de Uso: UC 007 – Manter Biografia

Descrição: Este caso de uso serve para manter as informações do Componente de Biografia do usuário.

Data View: DV12 - Tela de Manter Biografia do Usuário.



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema apresenta a tela DV12.
2. O usuário insere o texto desejado no campo de texto sob o placeholder “Insira a biografia aqui”
3. O usuário pressiona o botão “Configurações” (Engrenagem).
4. O usuário seleciona um arquivo de imagem de seu sistema.
5. O usuário seleciona a cor de fundo.
6. O usuário seleciona a cor de destaque.
7. O usuário seleciona mostrar ou não o título do componente na Checkbox.
8. O usuário pressiona o botão “Salvar” ou “Visto” (Verde).
9. O sistema inclui a alteração dos dados.
10. O sistema emite a mensagem “Componente atualizado com sucesso”
11. O caso de uso é encerrado.

Fluxos Alternativos:

A1: O usuário pressiona “Cancelar” ou “X” (Vermelho):

1. O caso de uso é encerrado.

A2: O usuário pressiona “Limpar Texto”:

1. O sistema exclui qualquer texto do campo de texto.

Fluxos de Exceção:**E1:** Usuário não preenche o campo de texto:

1. O sistema emite a mensagem “O campo de texto Biografia deve ser preenchido”

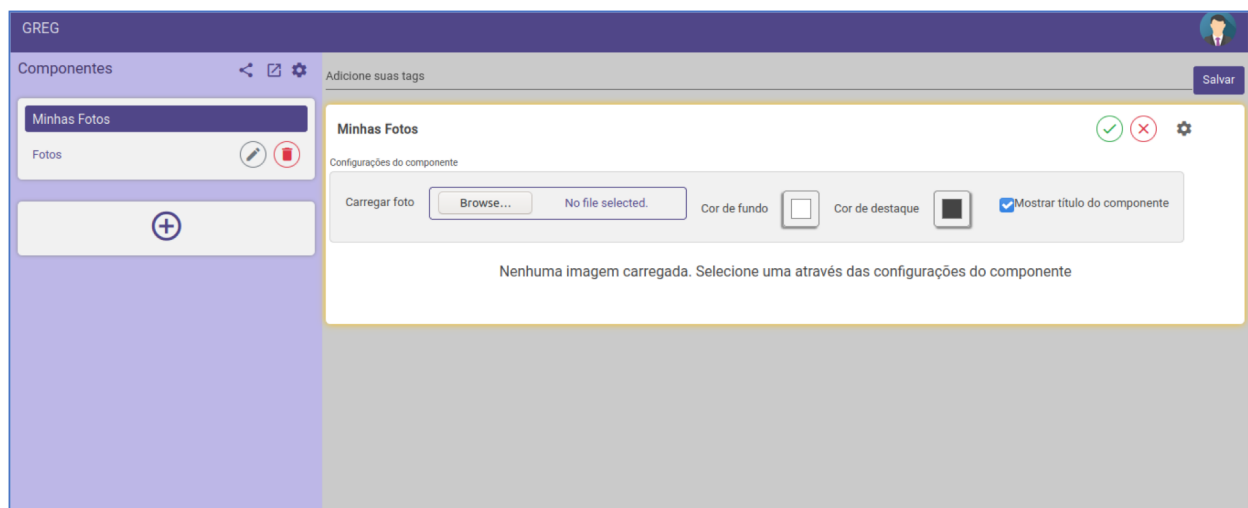
Regras de Negócio:**R1:** Biografia:

2. O texto da biografia pode conter até 1024 caracteres.

Especificação do Caso de Uso: UC 008 – Manter Imagens

Descrição: Este caso de uso serve para manter as informações do Componente de Fotos do usuário.

Data View: DV13 - Tela de Manter Fotos do Usuário.



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema busca as Informações do Usuário e preenche o título do componente cadastrado.
2. O sistema apresenta a tela DV13.
3. O usuário pressiona o botão “Browse” sob o título “Carregar Foto”.
4. O usuário seleciona um arquivo de imagem de seu sistema.
5. O usuário seleciona a cor de fundo.
6. O usuário seleciona a cor de destaque.
7. O usuário seleciona mostrar ou não o título do componente na Checkbox.
8. O usuário pressiona o botão “Visto” (Verde).
9. O sistema inclui a alteração dos dados.
10. O sistema emite a mensagem “Dados atualizados com sucesso”
11. O caso de uso é encerrado.

Fluxos Alternativos:

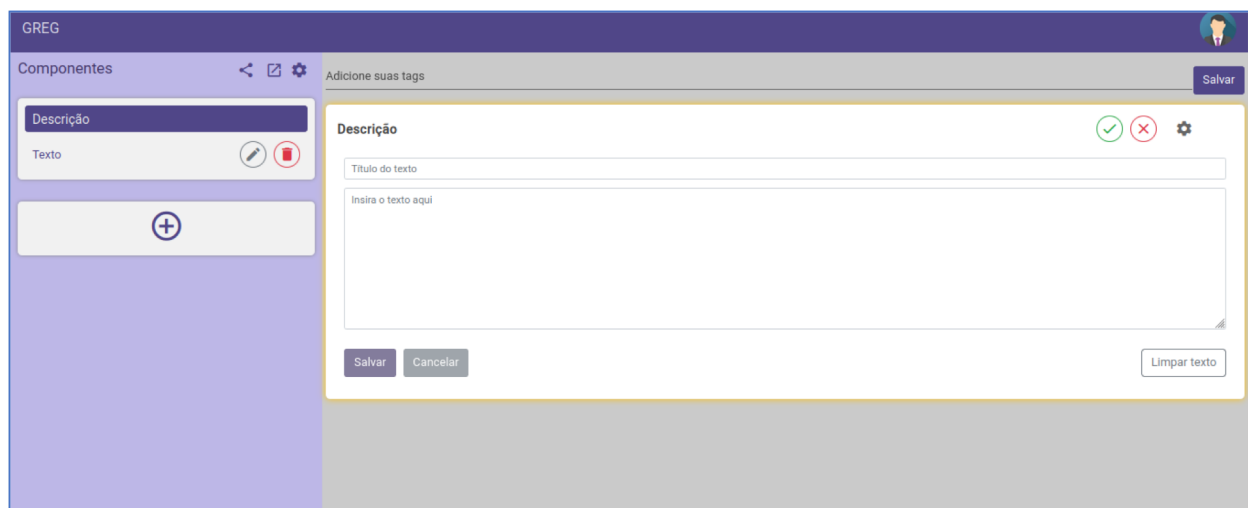
A1: “Cancelar” ou “X” (Vermelho):

1. O usuário pressiona “Cancelar” ou “X” (Vermelho):
2. O caso de uso é encerrado.

Especificação do Caso de Uso: UC 009 – Manter Texto

Descrição: Este caso de uso serve para manter as informações do Componente de Texto do usuário.

Data View: DV14 - Tela de Manter Texto do Usuário.



Ator Primário: Usuário

Fluxo de Eventos Principal:

3. O sistema busca as Informações do Usuário e preenche o título do componente cadastrado.
4. O sistema apresenta a tela DV14.
5. O usuário preenche o Título do texto sob o placeholder “Título do Texto”.
6. O usuário preenche o texto sob o placeholder “Insira o texto aqui”.
7. O usuário pressiona o botão “Configurações” (Engrenagem).
8. O usuário seleciona um arquivo de imagem de seu sistema.
9. O usuário seleciona a cor de fundo.
10. O usuário seleciona a cor de destaque.
11. O usuário seleciona mostrar ou não o título do componente na Checkbox.
12. O usuário pressiona o botão “Salvar” ou “Visto” (Verde).
13. O sistema inclui a alteração dos dados.
14. O sistema emite a mensagem “Dados atualizados com sucesso”
15. O caso de uso é encerrado.

Fluxos Alternativos:

A1: “Cancelar” ou “X” (Vermelho):

1. O usuário pressiona “Cancelar” ou “X” (Vermelho):
2. O caso de uso é encerrado.

A2: “Limpar Texto”:

1. O usuário pressiona “Limpar Texto”
2. O sistema exclui qualquer texto do campo de texto.

Fluxos de Exceção:

E1: Usuário não preenche o campo de título de texto:

1. O sistema emite a mensagem “O campo de Título de Texto deve ser preenchido”

E1: Usuário não preenche o campo de texto:

1. O sistema emite a mensagem “O campo de texto deve ser preenchido”

Regras de Negócio:

R1: Título do Texto:

1. O texto da biografia pode conter até 255 caracteres.

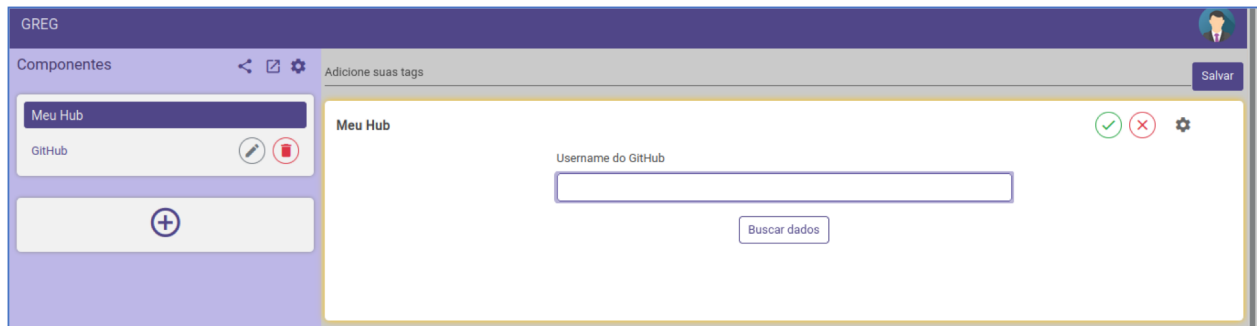
R1: Texto:

1. O texto do componente texto pode conter até 1024 caracteres.

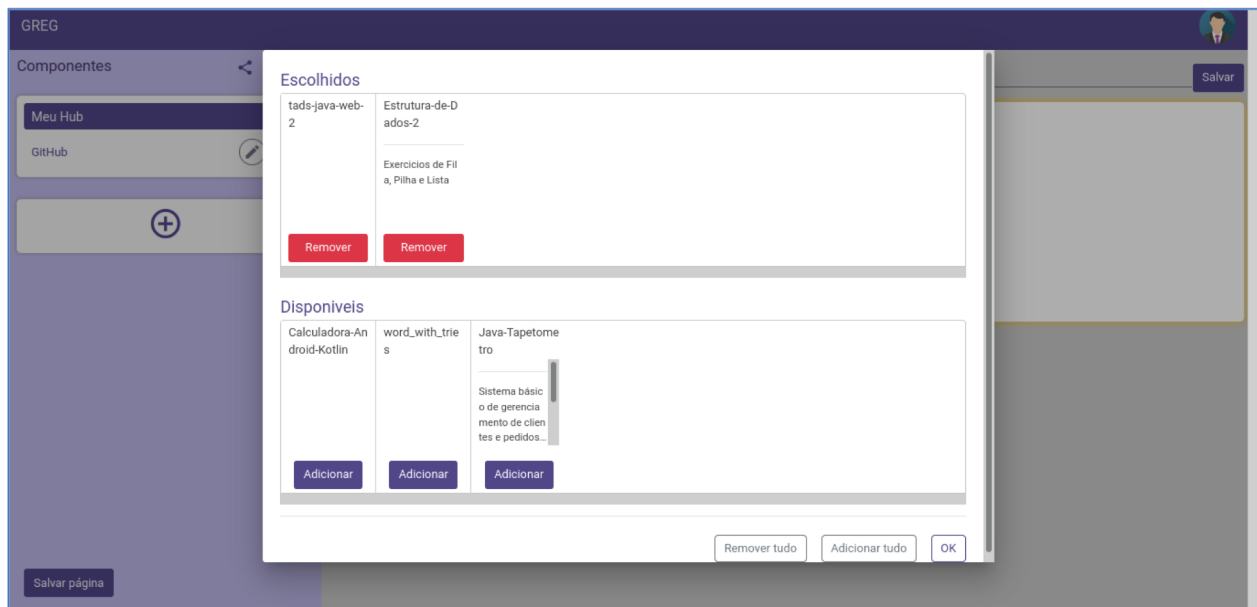
Especificação do Caso de Uso: UC 010 – Manter GitHub

Descrição: Este caso de uso serve para manter as informações do Componente de GitHub do usuário.

Data View: DV15 - Tela de Manter GitHub do Usuário.



Data View: DV16 – Selecionar repositórios GitHub.



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema busca as Informações do Usuário e preenche o título do componente cadastrado.
2. O sistema apresenta a tela DV15.
3. O usuário preenche o nome do usuário do GitHub.
4. O usuário pressiona o botão “Buscar dados”.
5. O sistema busca os repositórios do Usuário no GitHub, através de uma API.
6. O sistema preenche os dados dos repositórios disponíveis.
7. O sistema apresenta a tela DV16

8. O usuário seleciona os repositórios desejados pressionando o Botão “Adicionar”.
9. O usuário pressiona o Botão “Ok”.
10. O sistema inclui os repositórios no Componente GitHub.
11. O usuário pressiona o botão “Configurações” (Engrenagem).
12. O usuário seleciona a cor de fundo.
13. O usuário seleciona a cor de destaque.
14. O usuário seleciona mostrar ou não o título do componente na Checkbox.
15. O usuário pressiona o botão “Visto” (Verde).
16. O sistema inclui a alteração dos dados do componente.
17. O sistema emite a mensagem “Componente atualizado com sucesso”
18. O caso de uso é encerrado.

Fluxos Alternativos:

A1: “Cancelar” ou “X” (Vermelho):

1. O usuário pressiona “Cancelar” ou “X” (Vermelho):
2. O caso de uso é encerrado.

A2: Na tela DV16 – Selecionar Repositórios – Adicionar tudo:

1. O usuário pressiona o Botão “Adicionar tudo”
2. O sistema inclui todos os repositórios disponíveis no Componente GitHub.

A3: Na tela DV16 – Selecionar Repositórios – Remover tudo:

1. O usuário pressiona o Botão “Remover tudo”
2. O sistema remove todos os repositórios selecionados do Componente GitHub.

A4: Na tela DV16 – Selecionar Repositórios – Remover:

1. O usuário pressiona o Botão “Remover” em um repositório selecionado.
2. O sistema remove o repositório selecionado do Componente GitHub.

Regras de Negócio:

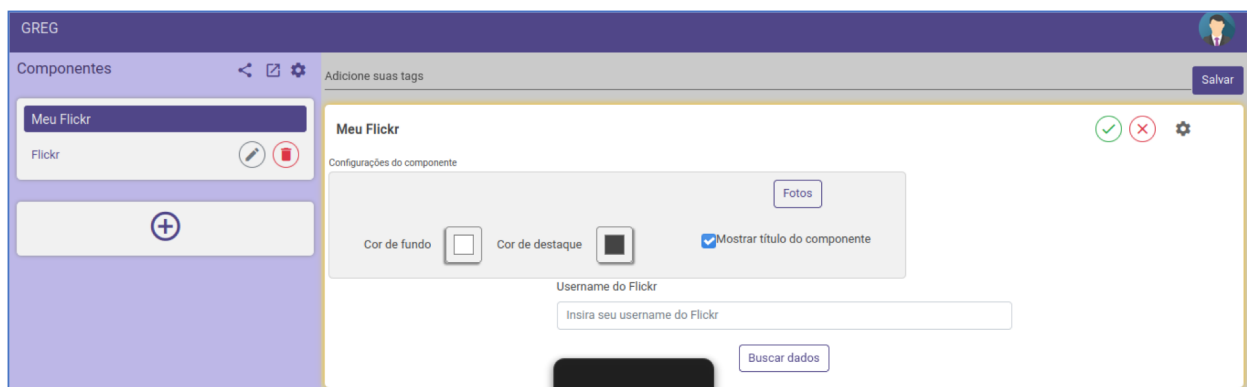
R1: Nome de usuário GitHub:

1. O nome de usuário pode conter até 255 caracteres.

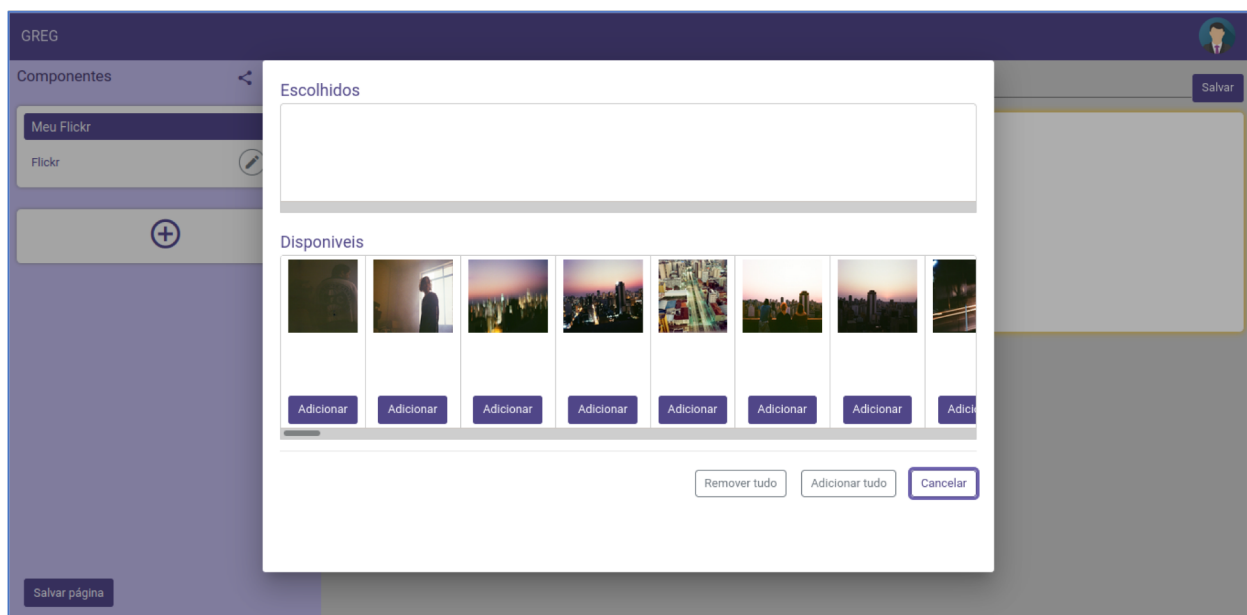
Especificação do Caso de Uso: UC 011 – Manter Flickr

Descrição: Este caso de uso serve para manter as informações do Componente de Flickr do usuário.

Data View: DV17 - Tela de Manter Flickr do Usuário.



Data View: DV18 – Selecionar fotos Flickr.



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema busca as Informações do Usuário e preenche o título do componente cadastrado.
2. O sistema apresenta a tela DV17.
3. O usuário preenche o nome do usuário do Flickr.
4. O usuário pressiona o botão “Buscar dados”.
5. O sistema busca os repositórios do Usuário no Flickr, através de uma API.

6. O sistema preenche os dados das fotos disponíveis.
7. O sistema apresenta a tela DV18
8. O usuário seleciona as fotos desejadas pressionando o Botão “Adicionar”.
9. O usuário pressiona o Botão “Ok”.
10. O sistema inclui as fotos no Componente Flickr.
11. O usuário pressiona o botão “Configurações” (Engrenagem).
12. O usuário seleciona a cor de fundo.
13. O usuário seleciona a cor de destaque.
14. O usuário seleciona mostrar ou não o título do componente na Checkbox.
15. O usuário pressiona o botão “Visto” (Verde).
16. O sistema inclui a alteração dos dados do componente.
17. O sistema emite a mensagem “Componente atualizado com sucesso”
18. O caso de uso é encerrado.

Fluxos Alternativos:

A1: “Cancelar” ou “X” (Vermelho):

1. O usuário pressiona “Cancelar” ou “X” (Vermelho):
2. O caso de uso é encerrado.

A2: Na tela DV18 – Selecionar Fotos – Adicionar tudo:

1. O usuário pressiona o Botão “Adicionar tudo”
2. O sistema inclui todas as fotos disponíveis no Componente Flickr.

A3: Na tela DV18 – Selecionar Fotos – Remover tudo:

1. O usuário pressiona o Botão “Remover tudo”
2. O sistema remove todas as fotos selecionadas do Componente Flickr.

A4: Na tela DV16 – Selecionar Fotos – Remover:

1. O usuário pressiona o Botão “Remover” em uma foto selecionada.
2. O sistema remove a foto selecionada do Componente Flickr.

Regras de Negócio:

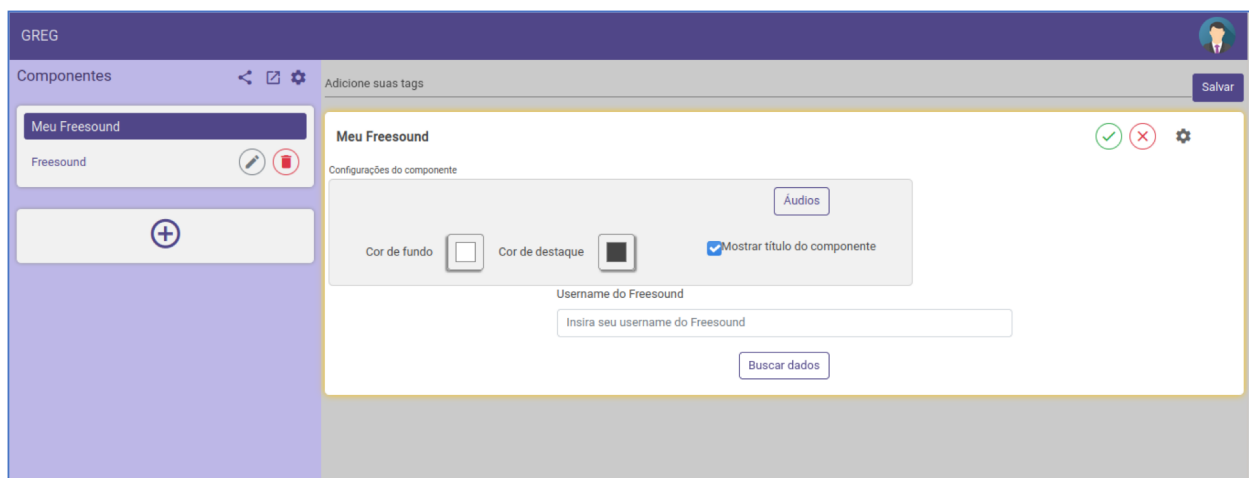
R1: Nome de usuário Flickr:

1. O nome de usuário pode conter até 255 caracteres.

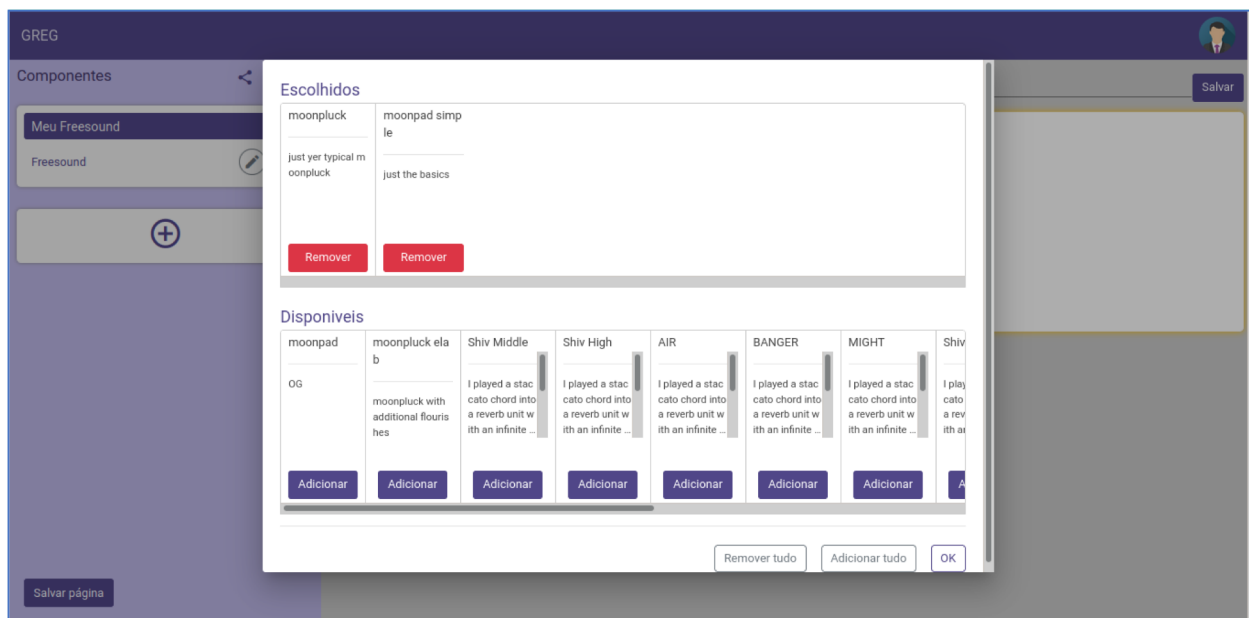
Especificação do Caso de Uso: UC 012 – Manter Freesound

Descrição: Este caso de uso serve para manter as informações do Componente de Freesound do usuário.

Data View: DV19 - Tela de Manter Freesound do Usuário.



Data View: DV20 – Selecionar Audio Freesound.



Ator Primário: Usuário

Fluxo de Eventos Principal:

1. O sistema busca as Informações do Usuário e preenche o título do componente cadastrado.
2. O sistema apresenta a tela DV19.
3. O usuário preenche o nome do usuário do Freesound.

4. O usuário pressiona o botão “Buscar dados”.
5. O sistema busca os áudios do Usuário no Freesound, através de uma API.
6. O sistema preenche os dados dos áudios disponíveis.
7. O sistema apresenta a tela DV20.
8. O usuário seleciona os áudios desejados pressionando o Botão “Adicionar”.
9. O usuário pressiona o Botão “Ok”.
10. O sistema inclui os áudios no Componente Flickr.
11. O usuário pressiona o botão “Configurações” (Engrenagem).
12. O usuário seleciona a cor de fundo.
13. O usuário seleciona a cor de destaque.
14. O usuário seleciona mostrar ou não o título do componente na Checkbox.
15. O usuário pressiona o botão “Visto” (Verde).
16. O sistema inclui a alteração dos dados do componente.
17. O sistema emite a mensagem “Componente atualizado com sucesso”
18. O caso de uso é encerrado.

Fluxos Alternativos:

A1: “Cancelar” ou “X” (Vermelho):

1. O usuário pressiona “Cancelar” ou “X” (Vermelho):
2. O caso de uso é encerrado.

A2: Na tela DV18 – Selecionar Áudio – Adicionar tudo:

1. O usuário pressiona o Botão “Adicionar tudo”
2. O sistema inclui todos os áudios disponíveis no Componente Freesound.

A3: Na tela DV18 – Selecionar Áudio – Remover tudo:

1. O usuário pressiona o Botão “Remover tudo”
2. O sistema remove todos os áudios selecionados do Componente Freesound.

A4: Na tela DV16 – Selecionar Áudio – Remover:

1. O usuário pressiona o Botão “Remover” em uma foto selecionada.
2. O sistema remove o áudio selecionado do Componente Freesound.

Regras de Negócio:

R1: Nome de usuário Freesound:

1. O nome de usuário pode conter até 255 caracteres.

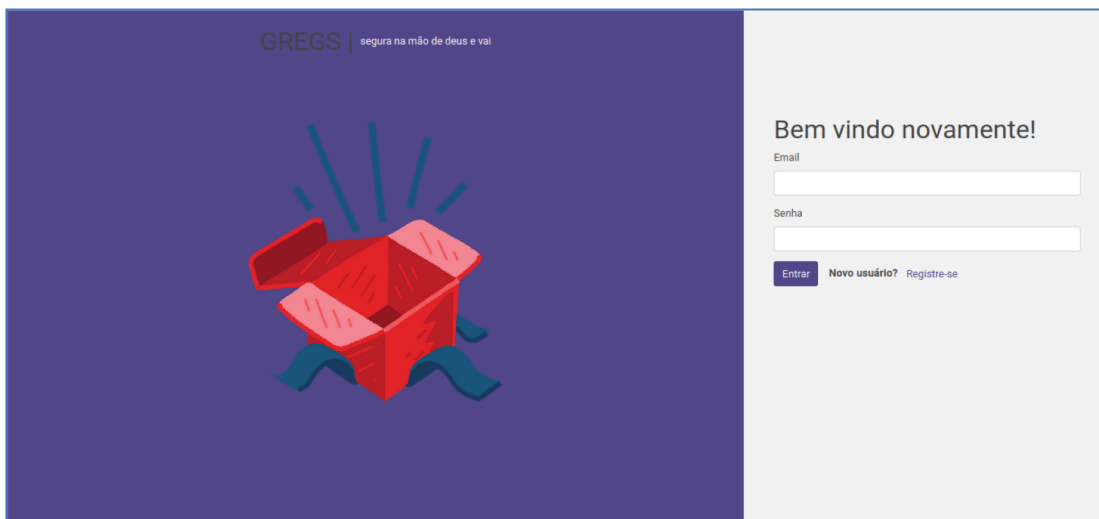
Especificação do Caso de Uso: UC 013 – Realizar *Logout*

Descrição: Este caso de uso serve para que o usuário possa realizar o *logout* da aplicação

Data View: DV21 – Tela de menu do usuário



Data View: DV22 – Tela de Login



Ator Primário: Usuário

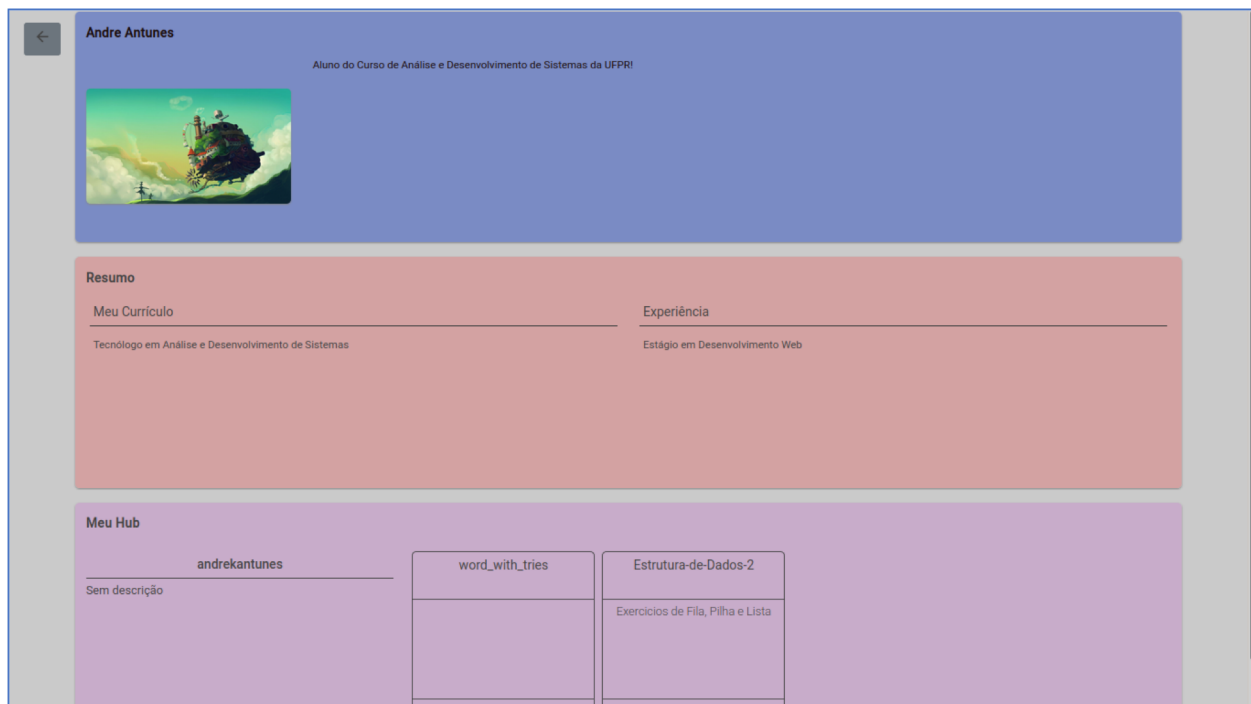
Fluxo de Eventos Principal:

1. O usuário clica no botão com a imagem de perfil para abrir o menu de usuário
2. O sistema apresenta a tela DV21
3. O usuário clica em Logout

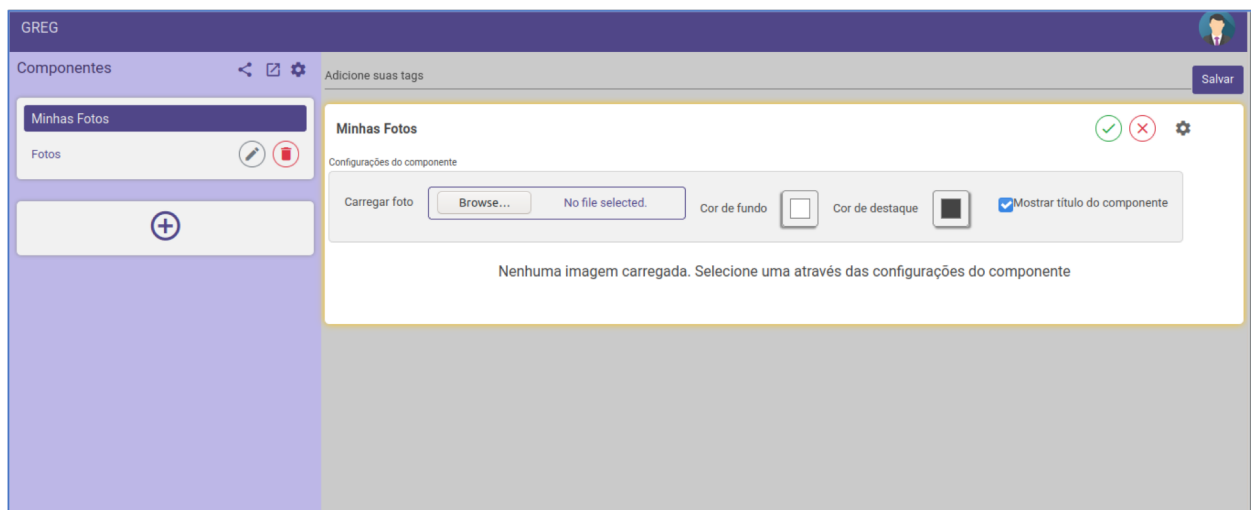
4. O sistema invalida o token de acesso do usuário
5. O sistema redireciona o usuário para o login
6. O sistema apresenta a tela de login DV22
7. O caso de uso é encerrado

APÊNDICE D – PROTÓTIPOS DAS TELAS

Data View: DV5 - Tela do Portfólio do Usuário.



Data View: DV13 - Tela de Manter Fotos do Usuário.



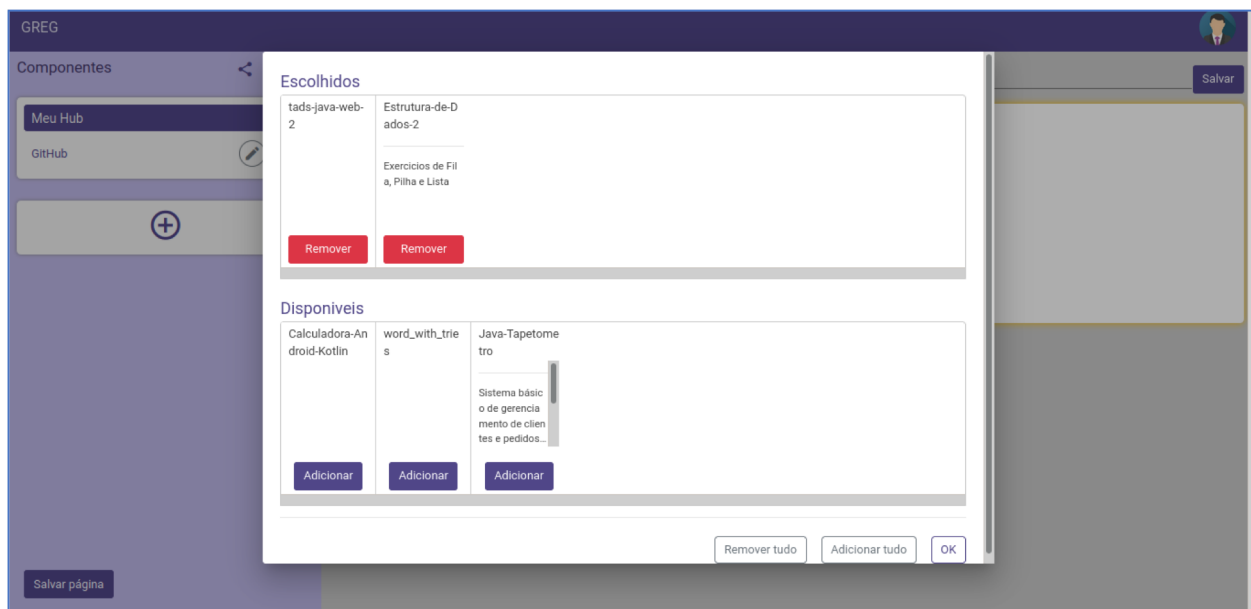
Data View: DV14 - Tela de Manter Texto do Usuário.

The screenshot shows a web application interface for maintaining user text. The top navigation bar is dark purple with the name 'GREG' on the left and a user profile icon on the right. Below the navigation bar is a light purple sidebar labeled 'Componentes' (Components) containing a 'Descrição' (Description) component with a 'Texto' (Text) field and a plus sign button. The main content area is white and titled 'Adicione suas tags' (Add your tags) with a 'Salvar' (Save) button in the top right. The main content area is divided into two sections: a top section titled 'Descrição' with a green checkmark, a red X, and a gear icon, and a bottom section with a text input field labeled 'Título do texto' (Text title) and a larger text area labeled 'Insira o texto aqui' (Enter text here). At the bottom of the main content area are buttons for 'Salvar' (Save), 'Cancelar' (Cancel), and 'Limpar texto' (Clear text).

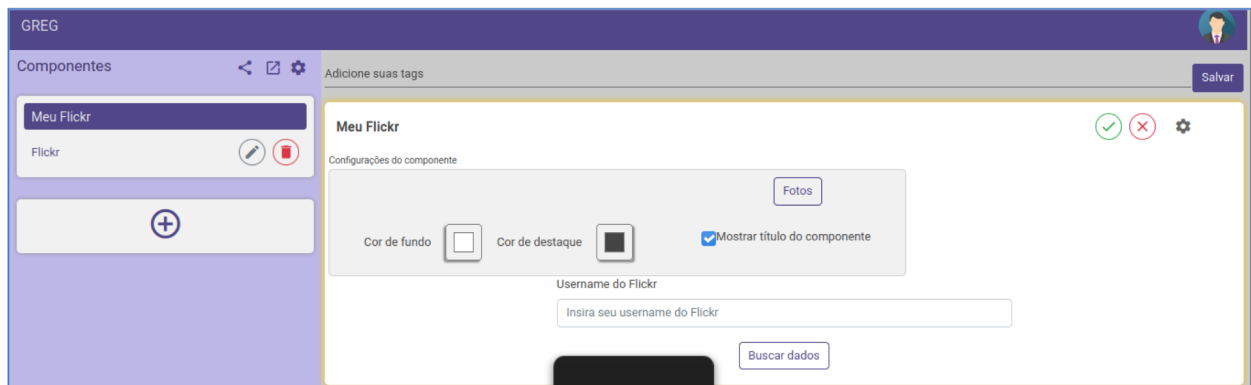
Data View: DV15 - Tela de Manter GitHub do Usuário.

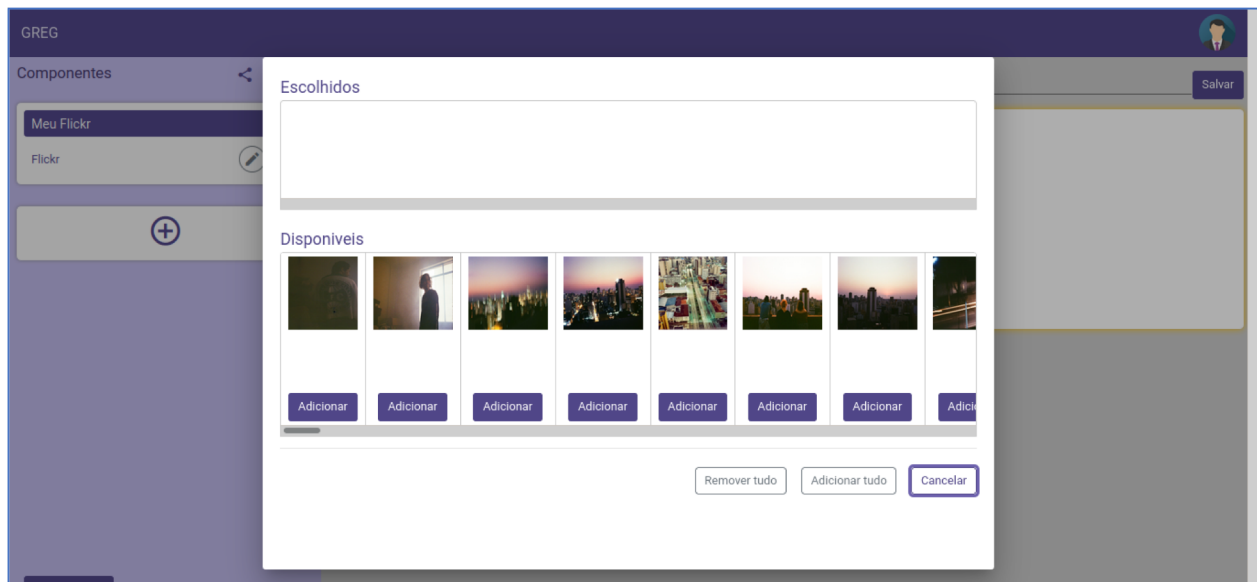
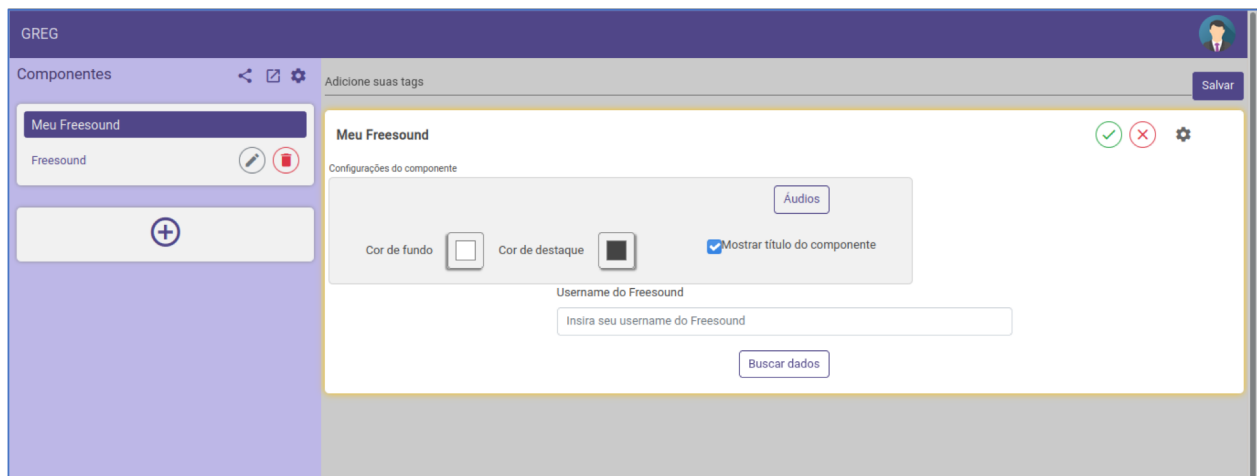
The screenshot shows a web application interface for maintaining user GitHub information. The top navigation bar is dark purple with the name 'GREG' on the left and a user profile icon on the right. Below the navigation bar is a light purple sidebar labeled 'Componentes' (Components) containing a 'Meu Hub' (My Hub) component with a 'GitHub' field and a plus sign button. The main content area is white and titled 'Adicione suas tags' (Add your tags) with a 'Salvar' (Save) button in the top right. The main content area is divided into two sections: a top section titled 'Meu Hub' with a green checkmark, a red X, and a gear icon, and a bottom section with a text input field labeled 'Username do GitHub' (GitHub Username) and a 'Buscar dados' (Search data) button.

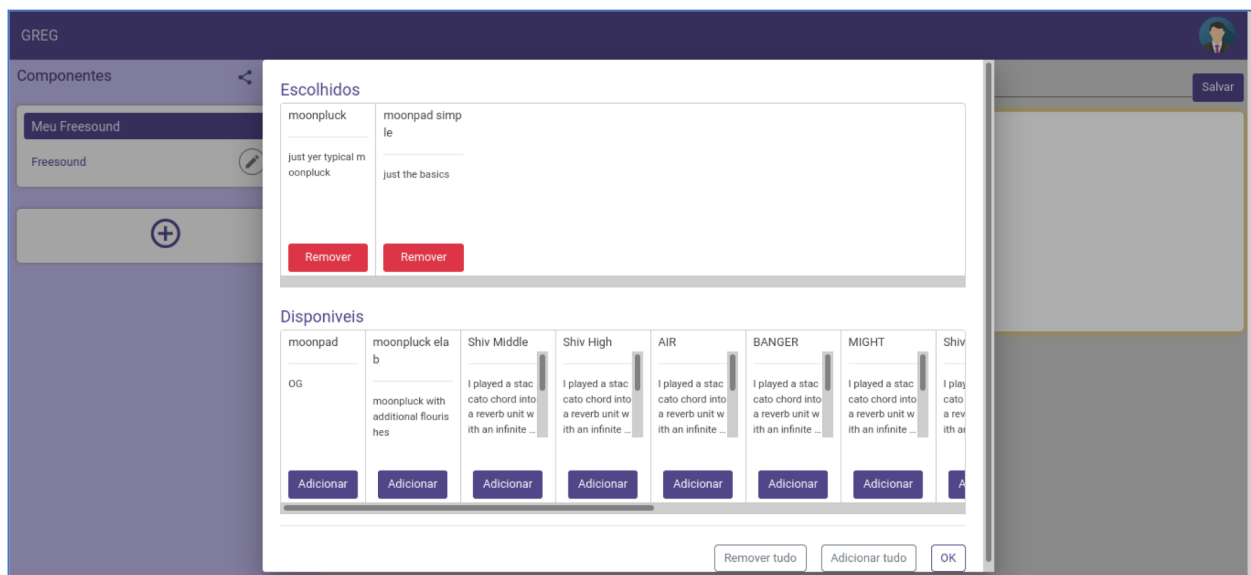
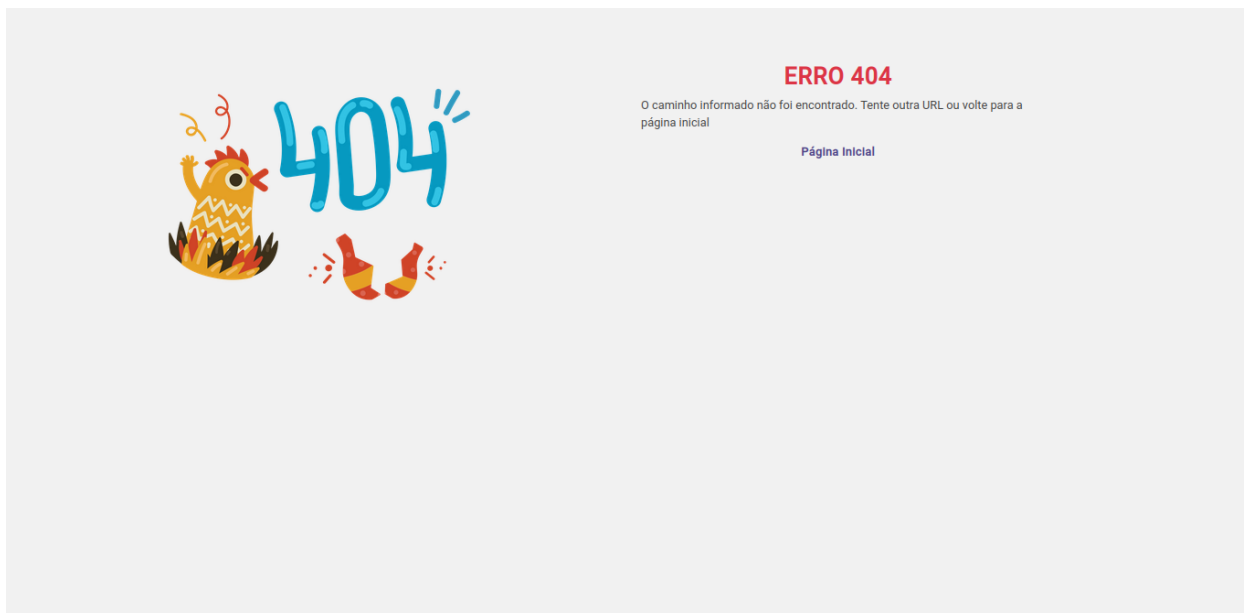
Data View: DV16 – Selecionar repositórios GitHub.



Data View: DV17 - Tela de Manter Flickr do Usuário.



Data View: DV18 – Selecionar fotos Flickr.**Data View: DV19 - Tela de Manter Freesound do Usuário.**

Data View: DV20 – Selecionar Audio Freesound.**Data View: DV21 – Tela de Erro.**

APÊNDICE E – ENTRY POINTS DA API

Entry Point 01: GET /login

Obtém usuário logado no Sistema

'Content-Type:'application/json'

'Authorization':

TokenNumber

Response:

```
{
  "msg": "Login Efetuado com Sucesso",
  "data": {
    "id": 01,
    "email": "email",
    "pessoa": {
      "id": 22,
      "nome": " nome ",
      "sobrenome": " sobrenome ",
      "nomeCompleto": " nome completo "
    },
    "pagina": {
      "id": 01,
      "url": "url",
      "backgroundColor": null,
      "componentes": [
        {
          "id": 01,
          "tipo": "TipoComponente"
        }
      ]
    },
    "imagemUsuario": {
      "id": 01,
      "nome": "nome",
      "base64Img": null,
      "url": "url"
    }
  },
  "tags": []
}
```

Entry Point 02: POST /login

Obtém token válido se o usuário e senha estiverem corretos.

Headers:

```
Content-Type: application/json
```

Body:

```
{
  "email": "email",
  "password": "senha"
}
```

Response:

```
{
  "msg": "Login Efetuado com Sucesso",
  "data": {
    "authorization":
    "TokenNumber"
  }
}
```

Entry Point 03: GET /usuarios/{busca}

Retorna usuários pelo motor de busca do sistema, que pesquisa por *Tags* e nome, se a busca resultar em vazio, retorna todos os usuários cadastrados no sistema.

Response:

```
{
  "msg": "Resultados Encontrados",
  "data": [
    {
      "id": 01,
      "email": "email",
      "pessoa": {
        "id": 01,
        "nome": "nome",
        "sobrenome": " sobrenome ",
        "nomeCompleto": " nome completo "
      },
      "pagina": {
        "id": 01,
        "url": " url ",
        "backgroundColor": null,

```

```

        "componentes": [
            {
                "id": 01,
                "tipo": "TipoComponente"
            }
        ]
    },
    "imagemUsuario": {
        "id": 0,
        "nome": "nome",
        "base64Img": null,
        "url": "url"
    },
    "tags": []
},
{
    "id": 01,
    "email": "nome@dominio.com",
    "pessoa": {
        "id": 01,
        "nome": "Nome de Usuário",
        "sobrenome": "Sobrenome do Usuário",
        "nomeCompleto": " Nome Completo de Usuário "
    },
    "pagina": {
        "id": 01,
        "url": "url",
        "backgroundColor": null,
        "componentes": [
            {
                "id": 01,
                "tipo": " TipoComponente "
            },
            {
                "id": 02,
                "tipo": " TipoComponente "
            }
        ]
    },
    "imagemUsuario": {
        "id": 01,
        "nome": "imagem do usuário",
        "base64Img": null,
        "url": "url"
    },
    "tags": [

```

```

        "TAG 01",
        "TAG 02",
        "TAG 03",
    ]
}
}
}

```

Entry Point 04: POST /usuarios

Inserir usuário no sistema

Body:

```

{
  "email": "email",
  "nome": "nome",
  "password": "senha",
  "sobrenome": "sobrenome",
  "url": "url"
}

```

Response:

```

{
  "msg": "Usuario Inserido com Sucesso",
  "data": {
    "id": 01,
    "email": "email",
    "pessoa": {
      "id": 01,
      "nome": "nome",
      "sobrenome": " sobrenome ",
      "nomeCompleto": " nome completo "
    },
    "pagina": {
      "id": 01,
      "url": "url",
      "backgroundColor": null,
      "componentes": [
        {
          "id": 01,
          "tipo": "TipoComponente"
        }
      ]
    }
  },
}

```



```

        "imagemUsuario": {
            "id": 01,
            "nome": "nome",
            "base64Img": null,
            "url": "url"
        },
        "tags": []
    }
}

```

Entry Point 05: PUT /usuarios

Atualiza os dados do usuário no sistema

Headers:

Content-Type: application/json

Authorization:

TokenNumber

Body:

```

{
    "nome": "nome",
    "sobrenome": "sobrenome",
    "email": "email",
    "url": "url"
}

```

Response:

```

{
    "msg": "Usuario Atualizado com Sucesso",
    "data": {
        "id": 01,
        "email": " email ",
        "pessoa": {
            "id": 01,
            "nome": "nome",
            "sobrenome": " sobrenome ",
            "nomeCompleto": " nome completo "
        },
        "pagina": {
            "id": 01,
            "url": "url",
            "backgroundColor": null,

```

```

        "componentes": [
            {
                "id": 01,
                "tipo": "TipoComponente"
            }
        ]
    },
    "imagemUsuario": {
        "id": 01,
        "nome": "nome",
        "base64Img": null,
        "url": "url"
    },
    "tags": []
}

```

Entry Point 06: GET /usuarios/config

Retorna as configurações do usuário

Headers:

Content-Type: application/json

Authorization:

TokenNumber

Response

```

{
    "msg": "Configurações Obtidas com Sucesso",
    "data": {
        "nome": "nome",
        "sobrenome": " sobrenome ",
        "email": " email ",
        "imagemUsuario": null,
        "urlPagina": "url",
        "senhaAntiga": null,
        "senhaNova": null
    }
}

```

Entry Point 07: GET /pagina/{url}

Retorna a página da url correspondente

Response:

```
{
  "msg": "Pagina Encontrada!",
  "data": {
    "pagina": {
      "id": 01,
      "url": "url",
      "backgroundColor": null,
      "componentes": [
        {
          "id": 01,
          "tipo": "TipoComponente"
        }
      ]
    }
  }
}
```

Entry Point 08: PUT /pagina

Atualiza a página ligada ao usuário com os componentes passados e demais dados passados no body.

Headers:

Content-Type: application/json

Authorization:

TokenNumber

Response:

```
{
  "msg": "Página Atualizada com Sucesso",
  "data": {
    "pagina": {
      "id": 01,
      "url": "url",
      "backgroundColor": "color",
      "componentes": [
        {

```

```

        "id": 01,
        "tipo": "TipoComponente"
    }
  ]
}

```

Entry Point 09: GET /tags

Retorna todas as *tags* do Sistema.

Response:

```

{
  "msg": "Resultados Encontrados",
  "data": [
    "TAG 01",
    "TAG 02",
    "TAG 03",
  ]
}

```

Entry Point 010: PUT /tags

Adiciona a lista de *Tags* passadas no body e as liga ao usuário obtido a partir do Authorization.

Headers:

Content-Type: application/json

Authorization:

TokenNumber

Response:

```

{
  "msg": "Tags Inseridas com sucesso",
  "data": null
}

```

Entry Point 011: GET /componente/{id}

Retorna o componente com id correspondente.

Authorization:

TokenNumber

Response:

```
{
  "msg": "Sucesso",
  "data": {
    "id": 1,
    "ordem": 0,
    "titulo": "Titulo do Projeto",
    "mostrarTitulo": true,
    "backgroundColor": "color",
    "foregroundColor": "color",
    "imagem": {
      "id": 1,
      "nome": "nome ",
      "base64Img": null,
      "url": "url"
    },
    "descricao": " Descrição",
    "tipo": "TipoComponente"
  }
}
```

Entry Point 012: POST /componente/

Insere componente de acordo com passado no corpo.

Headers:

Content-Type: application/json

Authorization:

TokenNumber

Response:

```
{
  "msg": "Sucesso",
  "data": {
    "id": 1,
    "ordem": 0,
```

```

        "titulo": "Titulo do Projeto",
        "mostrarTitulo": true,
        "backgroundColor": "color",
        "foregroundColor": "color",
        "imagem": {
            "id": 2,
            "nome": "nome ",
            "base64Img": null,
            "url": "url"
        },
        "descricao": "descrição",
        "tipo": "TipoComponente"
    }
}

```

Entry Point 013: PUT /componente/{id}

Atualiza o componente de id com os valores passado no body.

Headers:

Content-Type: application/json

Response:

```

{
    "msg": "Componente Inserido com Sucesso",
    "data": {
        "id": 01,
        "ordem": 0,
        "titulo": "titulo",
        "mostrarTitulo": true,
        "backgroundColor": "color",
        "foregroundColor": "color",
        "imagem": null,
        "texto": {
            "id": 01,
            "titulo": "titulo",
            "descricao": "descrição"
        },
        "tipo": "TipoComponente"
    }
}

```

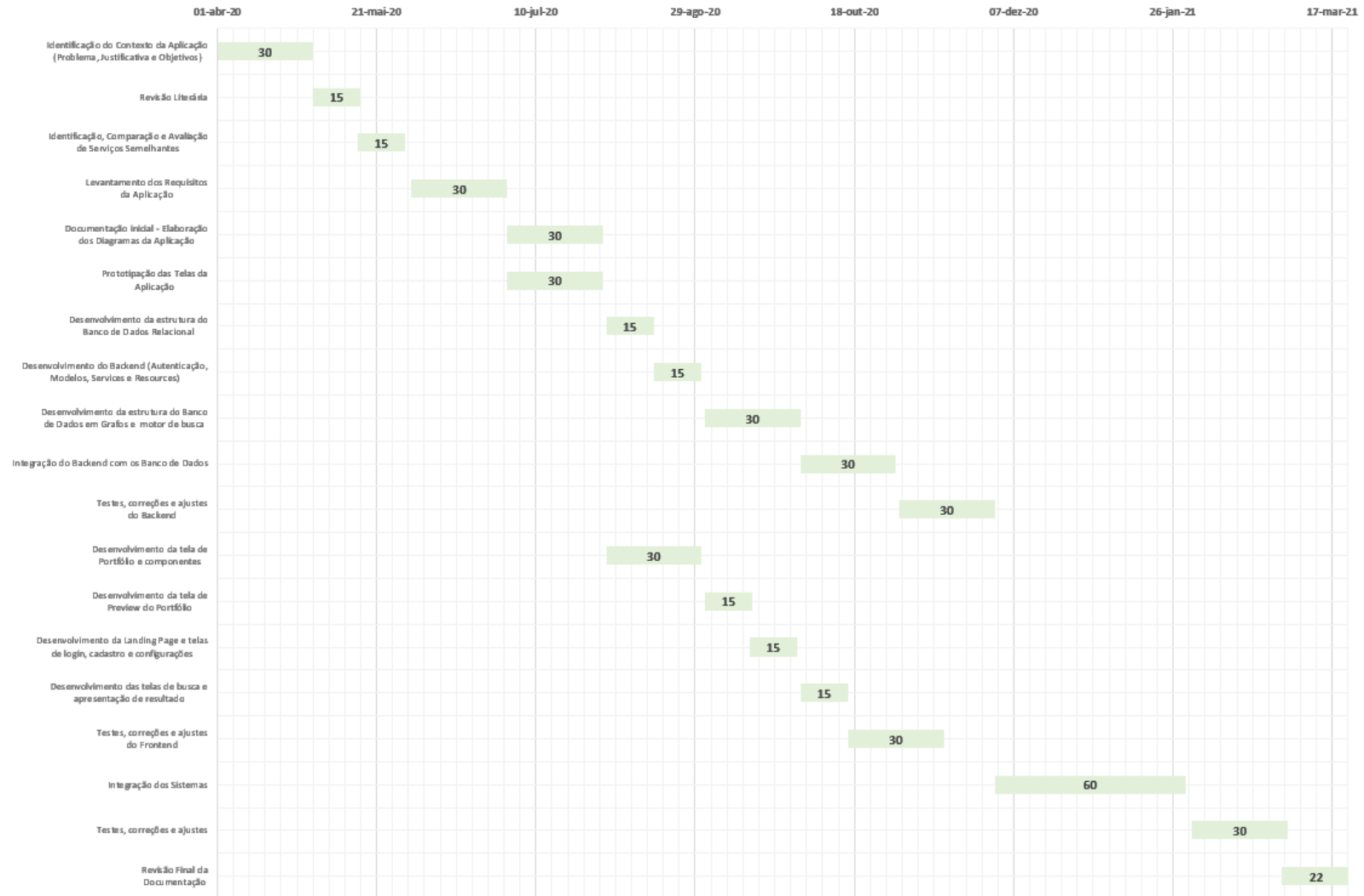
Entry Point 014: GET /health

Testa se o servidor está online.

Response:

```
{  
  "msg": "Hello, World!",  
  "data": null  
}
```

APÊNDICE F – DIAGRAMA DE GANTT



FONTE: OS AUTORES (2021)