

Banco de Imagens Online: Impulsionando a Presença Digital de Empreendedores com Designs Atraentes

Luiz Eduardo Rodrigues Correia



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

João Pessoa, PB

2024

Luiz Eduardo Rodrigues Correia

Banco de Imagens Online: Impulsionando a Presença Digital de Empreendedores com Designs Atraentes

Relatório Técnico apresentado ao curso Engenharia da Computação do Centro de Informática, da Universidade Federal da Paraíba, como requisito para a obtenção do grau de Bacharel em Engenharia da Computação

Orientadora: Thaís Gaudencio do Rêgo

Novembro de 2024

Catálogo na publicação
Seção de Catalogação e Classificação

C824b Correia, Luiz Eduardo Rodrigues.

Banco de imagens online: impulsionando a presença digital de empreendedores com designs atraentes / Luiz Eduardo Rodrigues Correia. - João Pessoa, 2024.

44 f. : il.

Orientação: Thaís Gaudencio do Rêgo.

Coorientação: Yuri de Almeida Malheiros Barbosa.
TCC (Graduação) - UFPB/CI.

1. Plataforma Web. 2. Empreendedores. 3. Design. 4. Personalização. 5. Google Cloud Functions. I. Rêgo, Thaís Gaudencio do. II. Barbosa, Yuri de Almeida Malheiros. III. Título.

UFPB/CI

CDU 004.92



CENTRO DE INFORMÁTICA
UNIVERSIDADE FEDERAL DA PARAÍBA

Trabalho de Conclusão de Curso de Engenharia da Computação intitulado ***Banco de Imagens Online: Impulsionando a Presença Digital de Empreendedores com Designs Atraentes*** de autoria de Luiz Eduardo Rodrigues Correia, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dra. Thaís Gaudencio do Rêgo
Universidade Federal da Paraíba

Prof. Dr. Yuri de Almeida Malheiros Barbosa
Universidade Federal da Paraíba

Prof. Dr. Lincoln David Nery e Silva
Universidade Federal da Paraíba

João Pessoa, 16 de novembro de 2024

A todos que participaram desta trajetória.

RESUMO

A crescente influência das redes sociais no processo de vendas, torna indispensável, a presença de pequenos e médios empreendedores nesse meio. Diante disso, o design e a presença online assumem papéis essenciais, para que empreendedores consigam atrair mais pessoas em seus negócios. O presente relatório técnico propõe o desenvolvimento de uma plataforma *Web* destinada a apoiar esses empreendedores fornecendo um banco de imagens, permitindo a personalização das mesmas e oferecendo ferramentas educacionais sobre design estratégico. A aplicação foi desenvolvida utilizando tecnologias como ReactJS no *front-end*, *Google Cloud Functions* no *back-end* e *Firestore* para armazenamento, garantindo escalabilidade e performance.

Palavras-chave: plataforma *Web*, *Google Cloud Functions*, empreendedores, design, personalização.

ABSTRACT

The growing influence of social media in the sales process makes the presence of small and medium-sized entrepreneurs essential in this space. Consequently, design and online presence become crucial to standing out from the competition and attracting more people to their businesses. This technical report proposes the development of a web platform aimed at supporting these entrepreneurs by providing an image bank, enabling their customization, and offering educational tools on strategic design. The application was built using technologies such as ReactJS for the front-end, Google Cloud Functions for the back-end, and Firestore for storage, ensuring scalability and performance.

Key-words: web platform, Google Cloud Functions, entrepreneurs, design, customization.

LISTA DE FIGURAS

1	Arquitetura do sistema.	30
2	À esquerda: Área para <i>login</i> . À direita: Área para solicitar recuperação de senha.	31
3	Tela inicial (<i>Home</i>).	32
4	Tela inicial com menu suspenso.	33
5	Tela do curso.	34
6	Tela de exibição de aula.	35
7	À esquerda: Tela de seleção de nichos e pesquisa das imagens. À direita: Tela de seleção de imagens.	36
8	À esquerda: Menu para alteração de opacidade. À direita: Menu para alteração de cor.	37
9	À esquerda: Imagens com cores alteradas. À direita: Tela de imagens favoritas	38

LISTA DE TABELAS

1	Caso de teste [CT01].	40
2	Caso de teste [CT02].	40
3	Caso de teste [CT03].	41
4	Caso de teste [CT04].	41
5	Caso de teste [CT05].	41
6	Caso de teste [CT06].	42
7	Caso de teste [CT07].	42
8	Caso de teste [CT08].	42
9	Caso de teste [CT09].	43
10	Caso de teste [CT10].	43
11	Caso de teste [CT11].	43

LISTA DE ABREVIATURAS

API — *Application Programming Interface*

Blob — *Binary Large Object*

CSS — *Cascading Style Sheets*

FGV — Fundação Getúlio Vargas

GCP — *Google Cloud Platform*

HTML — *Hypertext Markup Language*

JS — *JavaScript*

JSON — *JavaScript Object Notation*

LGPD — Lei Geral de Proteção de Dados Pessoais

NoSQL — *No Structured Query Language*

SGBD — Sistema Gerenciador de Banco de Dados

SQL — *Structured Query Language*

UI — *User Interface*

Sumário

1	INTRODUÇÃO	17
1.1	Tema	18
1.2	Problema	18
1.3	Objetivo geral	18
1.4	Objetivos específicos	19
1.5	Estrutura do relatório técnico	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	Conceitos Gerais	20
2.1.1	<i>Front-end e back-end</i>	20
2.1.2	<i>Serverless Computing</i>	20
2.1.3	Banco de dados	21
2.1.4	Blob	21
2.2	Ferramentas utilizadas	21
2.2.1	JavaScript	22
2.2.2	ReactJS	22
2.2.3	<i>Tailwind CSS</i>	22
2.2.4	Vite	22
2.2.5	<i>Firestore Database</i>	23
2.2.6	Hospedagem e <i>back-end</i> da aplicação	23
3	METODOLOGIA	24
3.1	Visão Geral	24
3.2	Requisitos Funcionais	24
3.2.1	[RF01] Cadastro de usuário	24
3.2.2	[RF02] <i>Login</i> de usuário	25
3.2.3	[RF03] Recuperação de senha	25
3.2.4	[RF04] Visualização de imagens	25
3.2.5	[RF05] Busca de imagens por texto	25

3.2.6	[RF06] Favoritar uma imagem	25
3.2.7	[RF07] Visualizar imagens favoritadas	25
3.2.8	[RF08] Alterar a cor de uma imagem	25
3.2.9	[RF9] Alterar a opacidade de uma imagem	26
3.2.10	[RF10] Copiar uma imagem	26
3.2.11	[RF11] Acesso aos cursos	26
3.2.12	[RF12] Acesso aos módulos	26
3.2.13	[RF13] Acesso às aulas	26
3.2.14	[RF14] Editar informações pessoais	26
3.3	Requisitos Não Funcionais	26
3.3.1	Usabilidade	27
3.3.1.1	[RNF01] Facilidade de Navegação	27
3.3.1.2	[RNF02] <i>Feedback</i> do Sistema	27
3.3.1.3	[RNF03] Responsividade	27
3.3.2	Confiabilidade	27
3.3.2.1	[RNF04] <i>Backup</i> de Dados	27
3.3.2.2	[RNF05] Alta Disponibilidade	27
3.3.3	Segurança	28
3.3.3.1	[RNF06] Criptografia de Dados	28
3.3.3.2	[RNF07] Adequação à Lei Geral de Proteção de dados Pes- soais	28
3.3.3.3	[RNF08] Controle de Acesso	28
3.3.4	Disponibilidade	28
3.3.4.1	[RNF09] Escalabilidade Automática	28
3.3.4.2	[RNF10] Conexão ao <i>Firestore</i>	28
3.3.5	Manutenibilidade	29
3.3.5.1	[RNF11] Modularidade do Código	29
3.3.6	Compatibilidade	29
3.3.6.1	[RNF12] Suporte Multiplataforma	29
3.3.7	Desempenho	29

3.3.7.1	[RNF13] Tempo de carregamento de páginas	29
3.3.7.2	[RNF14] Otimização de Consultas	30
3.4	Arquitetura	30
3.5	Interface do sistema	30
3.5.1	Telas da aplicação	31
3.5.1.1	Tela de acesso ao sistema	31
3.5.1.2	Tela inicial	32
3.5.1.3	Tela de curso	33
3.5.1.4	Tela de aulas	34
3.5.1.5	Tela de seleção de nicho e pesquisa das imagens	35
3.5.1.6	Tela de seleção de imagens	35
3.5.1.7	Tela de favoritos	38
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	39
4.1	Plano de testes	39
4.2	Casos de teste	39
4.3	Análise dos resultados	44
5	CONCLUSÕES E TRABALHOS FUTUROS	45
	REFERÊNCIAS	46

1 INTRODUÇÃO

Nos últimos anos, o mundo empresarial, sejam grandes ou pequenos empreendimentos, passou por uma transformação radical, impulsionada por uma revolução digital crescente. Junto à pandemia do Coronavírus e as medidas restritivas que foram necessárias, entre elas a necessidade de distanciamento social, o mundo digital foi uma ferramenta primordial para a continuidade da prestação de serviços, com o menor contato possível, acelerando assim, o uso do meio digital para as mais diversas necessidades [1].

A demanda de uma presença digital pelos empreendedores tornou-se essencial para o sucesso de qualquer empreendimento. Uma pesquisa realizada pelo Sebrae e da Fundação Getúlio Vargas (FGV) estima que, no primeiro semestre de 2022, 71% das micro e pequenas empresas brasileiras, incluindo microempreendedores individuais, faziam a utilização de meios digitais como canais de vendas [1].

Além disso, de acordo com o levantamento feito pela Comscore, no ano de 2022, o Brasil era o 3º país que mais consome redes sociais no mundo [2]. Ainda no mesmo ano, foi estimado que o Brasil possuía 150,4 milhões de usuários em redes sociais, ou seja, 70,3% da população do país [2]. Deste modo, vê-se a importância do uso de redes sociais, principalmente para microempreendedores, fornecendo uma maior facilidade na divulgação e vendas de produtos e serviços, bem como na maximização de um canal de vendas, com potencial para atingir grandes quantidades de usuário, por um baixo custo [3].

Com milhões de pessoas e empreendimentos já conectados em redes sociais, surge o desafio de chamar atenção e reter novos usuários para a sua marca. Visto que o processamento visual é uma das funções mais importantes do cérebro [4], o design e a forma ao qual as empresas se comportam online têm uma importância significativa na atração e retenção de novos usuários para os negócios [5].

Diante desse cenário, plataformas como a *Envato*, através do seu serviço *Envato Elements*, oferecem aos empreendedores uma vasta gama de recursos visuais para suprir as necessidades de *branding* e *marketing* [26]. O *Envato Elements* atua como um banco de imagens, vídeos, modelos de design e outros conteúdos criativos, facilitando o acesso a materiais visuais. A proposta deste trabalho assemelha-se a essa ferramenta, porém, com foco exclusivo em redes sociais e no apoio a pequenos e médios empreendedores.

No contexto apresentado, vê-se que o papel do design e da presença online das empresas torna-se crucial para se destacar em meio à concorrência e capturar a atenção do público-alvo. Um design visualmente atrativo e uma identidade de marca consistente podem criar uma primeira impressão positiva e aumentar a probabilidade de engajamento por parte dos usuários. Além disso, a forma como uma empresa se comporta online,

desde a qualidade do conteúdo publicado até a interação com os seguidores, influencia diretamente na percepção da marca e na construção de relacionamentos duradouros com os clientes. Portanto, investir em uma estratégia de design eficaz e uma presença online bem elaborada não é apenas uma vantagem competitiva, mas uma necessidade para os empreendedores que desejam crescer e prosperar em um mundo cada vez mais digital e visual.

1.1 Tema

Diante da crescente necessidade de micro e pequenas empresas se inserirem no ambiente digital, impulsionada pela popularização de ferramentas que permitem a realização de diversas atividades de maneira remota, este trabalho tem como objeto de estudo o desenvolvimento de uma plataforma *Web* destinada a apoiar esses empreendedores na construção de uma presença digital mais sólida e eficaz. A ferramenta proposta tem como objetivo oferecer suporte educacional, capacitando os usuários a navegar com competência no cenário digital e fornecer um banco de imagens de alta qualidade, que poderá ser utilizado de forma estratégica, para aprimorar o design visual de suas ofertas e otimizar a exposição de seus negócios no ambiente online. Essa combinação visa não apenas facilitar a adaptação dessas empresas ao mundo digital, mas também elevar o padrão estético e comunicacional de suas marcas, contribuindo para o fortalecimento de sua competitividade no mercado.

1.2 Problema

A digitalização de pequenos e microempreendedores no Brasil é uma tendência crescente, mas, ao mesmo tempo, muitas dessas empresas enfrentam dificuldades para se destacar no ambiente digital. Embora a maioria dos empreendedores reconheça a importância das redes sociais como um canal de vendas e divulgação, há uma clara lacuna na capacidade de utilizar estratégias visuais de maneira eficaz. O desafio se dá pela falta de acesso a recursos de design de alta qualidade e pela carência de conhecimento em como utilizá-los para criar uma presença digital forte e coesa. Isso resulta em um baixo nível de competitividade, dificultando a atração e retenção de clientes em um mercado saturado [6].

1.3 Objetivo geral

Como consequência dessa realidade, o trabalho propõe uma solução que ofereça a esses empreendedores, não apenas um banco de imagens atrativas, mas também uma plataforma que os capacite a utilizar esses recursos, de forma estratégica para potencializar sua presença digital. A dificuldade em criar uma identidade visual consistente e

profissional, sem um orçamento elevado, é um problema recorrente para esse público, o que justifica a proposta de uma ferramenta financeiramente acessível, intuitiva e voltada para a educação e o empoderamento digital.

1.4 Objetivos específicos

- Desenvolver uma aplicação *Web*;
- Implementar ferramentas para personalização das imagens da plataforma;
- Levantamento de requisitos funcionais e não funcionais da aplicação;
- Apresentar as soluções e tecnologias utilizadas para o desenvolvimento do projeto;
- Elaborar casos de testes, a fim de garantir o bom funcionamento do sistema.

1.5 Estrutura do relatório técnico

A estrutura de divisão adotada ao relatório, foi de cinco capítulos. Em seu primeiro capítulo, o trabalho introduz a sua temática, expõe o problema a ser resolvido e lista seus objetivos. Para o segundo, são apresentados conceitos gerais acerca do projeto a ser desenvolvido. No terceiro capítulo, a modelagem dos requisitos da aplicação e sua estrutura do banco de dados são os itens a serem expostos. No quarto capítulo, é realizada a apresentação dos planos e casos de testes, bem como uma análise do projeto desenvolvido. E por fim, em seu quinto capítulo é expressa a conclusão, bem como, propostas para melhorias futuras na aplicação.

2 FUNDAMENTAÇÃO TEÓRICA

No presente capítulo, serão apresentados conceitos e definições das ferramentas utilizadas para o desenvolvimento da aplicação.

2.1 Conceitos Gerais

Neste tópico serão descritos conceitos que envolvem a construção da estrutura do aplicativo.

2.1.1 *Front-end e back-end*

O *front-end* e o *back-end* são termos que representam diferentes vertentes na construção de uma aplicação *Web*. *Front-end* refere-se à parte visual da aplicação, ou seja, é a camada responsável por apresentar visualmente, ao usuário, os elementos que fazem parte da aplicação [7]. É por meio desta camada que usuários podem interagir e se comunicar com o servidor, de forma visual. As tecnologias que são comumente utilizadas para a construção do *front-end* são o HTML (*Hypertext Markup Language*), que é uma linguagem de marcação, o CSS (*Cascading Style Sheets*), que é utilizado para adição de estilos ao HTML, e por fim, o JS (JavaScript), que possibilita a interação do usuário com as estruturas visuais presentes na página [7].

Já o *back-end*, por sua vez, está relacionado à capacidade de processamento e armazenamento de informações em uma aplicação, bem como possibilitar a comunicação da ferramenta com serviços externos. É atribuída ao *back-end* a função de controlar o fluxo de dados provenientes do *front-end*, garantindo que as informações sejam manipuladas e armazenadas com boa performance e garantindo a segurança [8].

2.1.2 *Serverless Computing*

Serverless Computing é um paradigma de computação em nuvem, que permite a criação de aplicações, sem a necessidade de gerenciamento explícito do *back-end*, ou da infraestrutura de servidores. Nessa arquitetura, a alocação de recursos e a execução de tarefas são realizadas de forma dinâmica por provedores de serviços em nuvem, como por exemplo o *Google Cloud Functions* [9].

A alta escalabilidade e flexibilidade podem ser citados entre as principais vantagens do uso do *serverless computing*. Sua escalabilidade automática permite que a aplicação ajuste o uso de recursos com base em sua demanda [10]. Além disso, esse paradigma auxilia na produtividade do desenvolvimento, permitindo que os desenvolvedores se concentrem

no desenvolvimento da aplicação, deixando a cargo dos provedores a alocação de recursos [9].

2.1.3 Banco de dados

Um banco de dados é um sistema estruturado que permite o armazenamento, gerenciamento e recuperação de grandes volumes de informações de maneira eficiente, através de um SGBD (sistema gerenciador de banco de dados). Ele organiza os dados em formatos específicos, como tabelas, documentos, grafos ou outros modelos, facilitando a manipulação e consulta, por meio de linguagens de consulta estruturadas, como SQL (*Structured Query Language*) para bancos de dados relacionais. O principal objetivo de um banco de dados é garantir a integridade e consistência dos dados, permitindo que as informações possam ser acessadas e manipuladas de maneira segura e eficiente, por múltiplos usuários e aplicações, simultaneamente [11].

Os bancos de dados são classificados em várias categorias, como bancos de dados relacionais (SQL) e não-relacionais (NoSQL), dependendo de como as informações são armazenadas e organizadas. Os bancos de dados relacionais armazenam os dados em tabelas com esquemas predefinidos, o que facilita a integridade e a realização de operações complexas. Já os bancos de dados NoSQL, como por exemplo, MongoDB e o *Firestore Database*, oferecem maior flexibilidade no armazenamento de dados semi-estruturados ou não estruturados, sendo amplamente utilizados em cenários que exigem alta escalabilidade e adaptação a mudanças rápidas de requisitos [12].

2.1.4 Blob

O formato blob (*Binary Large Object*) refere-se a uma coleção de dados binários armazenados como uma única entidade em um sistema. Blobs são amplamente utilizados para armazenar grandes volumes de dados, como imagens, vídeos e outros tipos de mídia, que não podem ser representados como texto simples [17]. Essa abordagem é especialmente útil em sistemas que exigem o armazenamento e a manipulação de dados não estruturados de forma eficiente, como no caso de imagens em plataformas *Web*, permitindo sua manipulação, transferência e cópia para dispositivos do usuário.

2.2 Ferramentas utilizadas

As linguagens, *frameworks* e serviços utilizados para o desenvolvimento da aplicação serão explicitados neste tópico.

2.2.1 JavaScript

A principal linguagem utilizada para o desenvolvimento do *front-end* do projeto, foi o JavaScript, que é uma linguagem de programação de alto nível amplamente utilizada no desenvolvimento de aplicações *Web*, tanto no *front-end*, quanto no *back-end*. Criada originalmente em 1995 como uma linguagem de script para o navegador Netscape, JavaScript evoluiu ao longo dos anos e tornou-se uma das linguagens mais populares e versáteis para a criação de aplicações *Web* e *mobile*. Sua funcionalidade vai muito além da simples manipulação de elementos de página, permitindo o desenvolvimento de sistemas completos, incluindo servidores e APIs (*Application Programming Interface*) [18].

2.2.2 ReactJS

O ReactJS é uma biblioteca JavaScript de código aberto projetada para construir interfaces de usuário (UI) reativas e interativas, desenvolvida originalmente pelo *Facebook*. Diferentemente de *frameworks* tradicionais, o React adota uma abordagem baseada em componentes, o que permite aos desenvolvedores dividir a interface em blocos reutilizáveis e independentes. Essa estrutura modular facilita a manutenção e a escalabilidade de aplicações. Por meio deste *framework* JavaScript, a estrutura das interfaces da aplicação foi desenvolvida de maneira organizada e eficiente [19].

2.2.3 Tailwind CSS

O *Tailwind* CSS é um *framework* de CSS de código aberto, desenvolvido com a abordagem “*utility-first*”, o que significa que se baseia em classes utilitárias para estilizar elementos de interface. Diferentemente de outros *frameworks* tradicionais, como o Bootstrap, que fornecem componentes pré-construídos, o *Tailwind* permite que os desenvolvedores construam estilos aplicando pequenas classes de utilidade diretamente no HTML. Essa metodologia oferece flexibilidade e rapidez no desenvolvimento de interfaces, uma vez que os desenvolvedores podem compor *layouts* complexos, utilizando um conjunto consistente de classes de utilidade. A estilização dos componentes utilizados foi realizada por meio do *Tailwind CSS* [20].

2.2.4 Vite

O *Vite* é uma ferramenta de inicialização de *frameworks* moderna, bem como um ambiente de desenvolvimento projetado para otimizar o fluxo de trabalho de aplicações *Web front-end*. A ferramenta tem como principal objetivo fornecer um servidor de desenvolvimento mais rápido e eficiente para aplicações modernas, utilizando *frameworks* como React [21].

2.2.5 *Firestore Database*

O *Firestore* é um banco de dados NoSQL e *serverless* fornecido pelo *Google*, projetado para armazenar e gerenciar dados de maneira flexível e escalável. Ele utiliza um modelo de documentos e coleções, permitindo que desenvolvedores criem e sincronizem informações em tempo real, facilitando a integração com aplicações *Web* e móveis. Sua arquitetura *serverless* simplifica o desenvolvimento ao gerenciar, automaticamente, a escalabilidade e o provisionamento de recursos.

2.2.6 Hospedagem e *back-end* da aplicação

Durante o desenvolvimento desta aplicação, foi escolhida a arquitetura *serverless* para o *back-end*, utilizando o *Google Cloud Functions*, que é um serviço que permite a construção de funções no lado do servidor, para a execução de tarefas, como por exemplo, alterações no banco de dados, conexões com serviços externos, entre outros [23]. O serviço é útil para construção de APIs, automações e integrações, sem a necessidade de gerenciar servidores. Cada função é executada de maneira isolada e escalada automaticamente, conforme o volume de solicitações, oferecendo alta disponibilidade e eficiência de custos.

Essa abordagem *serverless* elimina a necessidade de gerenciar servidores dedicados, uma vez que o provedor de nuvem cuida da escalabilidade e do provisionamento de recursos, conforme a demanda. Essa escolha reduz a complexidade de infraestrutura, ao passo que garante uma integração fluida com outros serviços do *Google Cloud*, como o *Firestore*, para armazenamento e o *Firebase*, para autenticação.

O *Google Cloud Hosting* foi a ferramenta utilizada para a hospedagem da aplicação, sendo um serviço oferecido pelo *Google Cloud Platform* (GCP), projetado para fornecer uma infraestrutura flexível, escalável e confiável para aplicações *Web* e móveis. Utilizando a mesma infraestrutura que alimenta produtos como *Google* e *YouTube*, o *Google Cloud Hosting* suporta uma ampla gama de serviços, desde hospedagem de sites estáticos, até aplicações complexas de microsserviços. Essa plataforma permite a distribuição global de conteúdo, gerenciando automaticamente a escalabilidade e alta disponibilidade das aplicações [23].

Essa infraestrutura completa fornece o suporte necessário para garantir a estabilidade, a alta disponibilidade e o desempenho eficiente da nossa aplicação, permitindo que ela seja acessada globalmente, de maneira segura e contínua, atendendo às demandas de crescimento e garantindo uma experiência de usuário satisfatória.

3 METODOLOGIA

3.1 Visão Geral

O aplicativo *Web* desenvolvido tem como objetivo apoiar pequenos empreendedores na criação de conteúdos visuais de qualidade para suas redes sociais, facilitando o acesso a designs, que podem ser personalizados, refletindo a identidade visual de suas marcas. A plataforma funciona como um banco de imagens que permite a personalização rápida e prática, adaptando-se ao estilo de cada usuário. No sistema, as imagens são chamadas de “figurinhas”, que são divididas em diferentes categorias, que por sua vez, são chamadas de “nichos” dentro da aplicação. As modificações destes termos foram realizadas devido a uma comunicação já existente em outras aplicações semelhantes.

Ademais, a aplicação permite que o usuário favorite suas imagens, permitindo um acesso mais rápido, através da criação de uma pasta personalizada com suas imagens favoritas. Além disso, a aplicação possui um componente educativo, orientando os usuários na construção de designs eficientes e na utilização estratégica de imagens, conforme as demandas de seu nicho de mercado. Este componente educativo é formado por uma estrutura de cursos, onde cada curso poderá ter um ou mais módulos e, cada módulo, uma ou mais aulas.

O desenvolvimento da plataforma foi realizado com a utilização de tecnologias modernas, garantindo uma experiência de usuário fluida e eficiente. A interface foi construída com ReactJS, uma biblioteca JavaScript voltada para a criação de interfaces de usuário dinâmicas e responsivas. No *back-end*, utilizou-se serviços do *Google Cloud*, para a implementação de uma arquitetura *serverless*, facilitando a escalabilidade e o gerenciamento das funcionalidades. Além disso, o armazenamento de dados é feito por meio do *Firestore*.

3.2 Requisitos Funcionais

Os requisitos funcionais de um sistema representam as especificações das funcionalidades que a aplicação deve oferecer aos usuários. Eles descrevem os serviços, tarefas e operações que o sistema deve executar, para atender às necessidades e objetivos propostos. Dessa forma, tais requisitos indicam de maneira clara como o usuário poderá interagir com a plataforma e quais ações serão permitidas [24]. Neste tópico serão apresentados os requisitos funcionais da aplicação.

3.2.1 [RF01] Cadastro de usuário

O sistema deve permitir que novos usuários se cadastrem na plataforma, após o pagamento ser aprovado por um serviço externo. Cada usuário deve ter um nome, e-mail e senha. Estes dados serão repassados para o *back-end* da plataforma criar o usuário.

3.2.2 [RF02] *Login* de usuário

O sistema deve permitir que o usuário faça *login* utilizando e-mail e senha cadastrados.

3.2.3 [RF03] Recuperação de senha

O sistema deve permitir que o usuário realize a alteração de sua senha, tanto após estar autenticado, quanto na tela de *login*. Para o último caso, basta o fornecimento do e-mail.

3.2.4 [RF04] Visualização de imagens

O usuário deve conseguir ver as imagens do banco, para posterior uso. As imagens podem ser pesquisadas através de filtros dentro do aplicativo, ou através de busca textual.

3.2.5 [RF05] Busca de imagens por texto

O usuário poderá pesquisar por imagens, em uma caixa de pesquisa por texto, de acordo com descrições previamente indexadas às imagens, que por sua vez, estarão previamente armazenadas no banco de dados.

3.2.6 [RF06] Favoritar uma imagem

O sistema deve permitir que o usuário consiga adicionar imagens à uma aba de favoritos, para facilitar o posterior uso da mesma.

3.2.7 [RF07] Visualizar imagens favoritadas

O sistema deve ter uma aba específica, que contenha todas as imagens adicionadas como “favoritas” pelo usuário.

3.2.8 [RF08] Alterar a cor de uma imagem

O usuário deve conseguir alterar a cor de qualquer imagem, para que se adeque à sua paleta de cores, ou, qualquer cor desejada pelo usuário. Isto também deve ser aplicado às imagens marcadas como “favoritas”.

3.2.9 [RF9] Alterar a opacidade de uma imagem

O usuário deve conseguir alterar a opacidade de qualquer imagem. Isto também deve ser aplicado às imagens marcadas como “favoritas”.

3.2.10 [RF10] Copiar uma imagem

O usuário deve conseguir copiar qualquer imagem em um clique, mesmo que ela esteja com alteração de cor, opacidade, ou ainda, que esteja entre as imagens favoritas.

3.2.11 [RF11] Acesso aos cursos

O usuário deve conseguir acessar qualquer curso que esteja na plataforma.

3.2.12 [RF12] Acesso aos módulos

O usuário deve conseguir acessar qualquer módulo que esteja na plataforma, para o curso selecionado.

3.2.13 [RF13] Acesso às aulas

O usuário deve conseguir acessar qualquer aula que esteja na plataforma, para o módulo selecionado.

3.2.14 [RF14] Editar informações pessoais

O sistema deve permitir que o usuário altere suas informações pessoais, tais como nome, e-mail e celular.

3.3 Requisitos Não Funcionais

Os requisitos não funcionais são elementos cruciais no desenvolvimento de sistemas de *software*, pois definem as qualidades e restrições que um sistema deve atender. Eles abrangem aspectos como usabilidade, desempenho, segurança, confiabilidade, manutenibilidade e portabilidade, entre outros, sendo considerados subjetivos [13] [14]. Nesta seção, serão descritos os requisitos não funcionais elaborados para a aplicação deste trabalho, onde serão divididos como requisitos de: usabilidade, confiabilidade, segurança, disponibilidade, manutenibilidade, compatibilidade e desempenho.

3.3.1 Usabilidade

A usabilidade refere-se à facilidade com que os usuários interagem com um sistema ou aplicação. Um sistema com alta usabilidade é intuitivo, simples de navegar e requer um esforço mínimo para ser utilizado, mesmo por pessoas sem experiência técnica [27].

3.3.1.1 [RNF01] Facilidade de Navegação

A interface deve ser intuitiva, permitindo que usuários sem experiência técnica utilizem todas as funcionalidades, de forma simples e rápida.

3.3.1.2 [RNF02] *Feedback* do Sistema

O sistema deve fornecer *feedback* claro e imediato para todas as ações do usuário, como confirmações de operações ou erros.

3.3.1.3 [RNF03] Responsividade

A plataforma deve se adaptar automaticamente a diferentes tamanhos de tela, garantindo uma experiência fluida e eficiente, em dispositivos móveis e *desktop*.

3.3.2 Confiabilidade

A confiabilidade de um sistema refere-se à capacidade de operar consistentemente sem falhas. Ela abrange a integridade dos dados, a aptidão de recuperar informações após falhas e a continuidade do serviço. Sistemas confiáveis minimizam o tempo de inatividade e garantem que os dados do usuário sejam mantidos seguros e intactos, mesmo em casos de falhas [27].

3.3.2.1 [RNF04] *Backup* de Dados

Os dados do usuário devem ser salvos regularmente e armazenados em *backups* automáticos, para prevenir perdas em caso de falhas.

3.3.2.2 [RNF05] Alta Disponibilidade

O sistema deve estar disponível para os usuários o máximo de tempo possível, garantindo acesso contínuo e minimizando períodos de inatividade.

3.3.3 Segurança

Segurança no desenvolvimento de sistemas diz respeito às medidas tomadas para proteger os dados e as informações de usuários, contra acessos não autorizados e vulnerabilidades. Isso inclui a implementação de criptografia, controle de acesso e conformidade com leis [27].

3.3.3.1 [RNF06] Criptografia de Dados

Todos os dados sensíveis, como senhas, tanto em trânsito quanto em repouso, devem ser criptografados utilizando algoritmos de criptografia robustos, como AES-256.

3.3.3.2 [RNF07] Adequação à Lei Geral de Proteção de dados Pessoais

O sistema deve garantir o cumprimento da Lei Geral de Proteção de Dados (LGPD), implementando medidas de segurança para proteger os dados pessoais dos usuários. Isso inclui a coleta e o tratamento de dados com o consentimento do usuário, a possibilidade de exclusão ou correção de informações, e a implementação de criptografia para dados sensíveis.

3.3.3.3 [RNF08] Controle de Acesso

O sistema deve implementar diferentes níveis de acesso para usuários, garantindo que informações pessoais sejam acessadas apenas pelos usuários que possuem controle sobre as informações.

3.3.4 Disponibilidade

A disponibilidade de uma aplicação refere-se à sua capacidade de estar acessível e funcional para os usuários a maior parte do tempo, minimizando períodos de inatividade, em condições normais ou adversas [28].

3.3.4.1 [RNF09] Escalabilidade Automática

A infraestrutura deve ser capaz de escalar automaticamente conforme a demanda, assegurando o funcionamento eficiente, mesmo durante picos de uso.

3.3.4.2 [RNF10] Conexão ao *Firestore*

A conexão com o *Firestore* deve ser mantida de forma contínua e estável, garantindo a sincronização em tempo real dos dados. O sistema deve assegurar baixa latência

nas operações de leitura e escrita, permitindo uma experiência fluida para os usuários.

3.3.5 Manutenibilidade

A manutenibilidade refere-se à facilidade com que uma aplicação pode ser atualizada, corrigida ou aprimorada ao longo do tempo, sem comprometer seu funcionamento [27].

3.3.5.1 [RNF11] Modularidade do Código

O sistema deve ser desenvolvido com uma arquitetura modular, permitindo que componentes possam ser alterados ou substituídos, sem afetar outras partes da aplicação.

3.3.6 Compatibilidade

Sistemas que possuem compatibilidade asseguram que a experiência do usuário seja consistente, independentemente do ambiente ou do dispositivo usado para acessá-los [29].

3.3.6.1 [RNF12] Suporte Multiplataforma

O sistema deve ser compatível com os principais navegadores (*Chrome*, *Firefox*, *Safari* e *Edge*), a partir de versões que suportam o uso de JavaScript, garantindo uma experiência fluida em todos os principais navegadores.

3.3.7 Desempenho

O desempenho de uma plataforma *Web* refere-se à sua capacidade de responder rapidamente às solicitações dos usuários e realizar operações de maneira eficiente, mesmo sob alta demanda. Isso inclui o tempo de carregamento das páginas e a otimização de consultas ao banco de dados para minimizar o uso de recursos [27].

3.3.7.1 [RNF13] Tempo de carregamento de páginas

As páginas da aplicação devem carregar completamente em menos de 3 segundos. Este tempo foi definido com base em dados divulgados pelo Google, que indicam que 53% das visitas a sites móveis são abandonadas quando o tempo de carregamento ultrapassa esse limite [25].

3.3.7.2 [RNF14] Otimização de Consultas

As consultas ao banco de dados devem ser otimizadas para minimizar o tempo de execução e reduzir o consumo de recursos.

3.4 Arquitetura

A Figura 1, apresenta a arquitetura do sistema. O cliente, que seria a interface construída com React, interage inicialmente com o *Google Cloud Hosting*, que por sua vez, hospeda a interface de usuário e serve como a camada de apresentação da aplicação. As solicitações enviadas são direcionadas para o *Cloud Functions*, responsável pela execução da lógica de negócio e pela integração com APIs externas, quando necessário. Para o armazenamento e gerenciamento de dados, a comunicação é feita com o *Firestore*, que armazena informações dos usuários e as imagens manipuladas na aplicação, e quando requisitado, recupera os dados armazenados e os envia para o servidor, repassando por sua vez ao usuário.

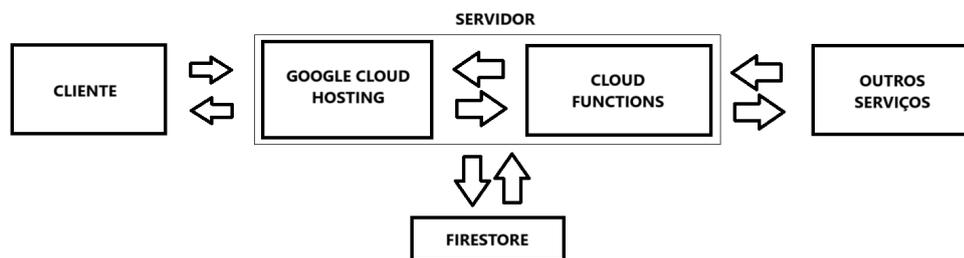


Figura 1: Arquitetura do sistema.

Fonte: De autoria própria.

3.5 Interface do sistema

Na sequência, será apresentada a interface do sistema, que foi projetada com foco na usabilidade e no minimalismo, proporcionando uma experiência fluida e intuitiva para o usuário. O design é responsivo, adaptando-se a diversos tamanhos de tela, porém, com ênfase em dispositivos móveis, telas em que focaremos neste trabalho, por ser o principal meio de acesso às redes sociais [16]. A navegação é clara, facilitando o acesso aos cursos e recursos. O sistema também adota um *layout* escuro, que visa aumentar o conforto visual, especialmente em ambientes de baixa luminosidade. Estudos indicam que *layouts* escuros reduzem a fadiga ocular e minimizam o ofuscamento, proporcionando uma experiência de uso mais agradável, especialmente em interações prolongadas [15].

3.5.1 Telas da aplicação

Nesta seção, serão detalhadas as telas que compõem o sistema, oferecendo uma visão completa de sua estrutura e funcionalidade.

3.5.1.1 Tela de acesso ao sistema

A página de acesso ao sistema foi projetada para ser simples e objetiva, como apresentado no lado direito da Figura 2, garantindo que o usuário tenha uma experiência rápida e eficiente ao acessar o sistema. A interface conta com os campos de e-mail e senha, onde o usuário deve inserir suas credenciais para acessar o sistema. Abaixo dos campos, há links de suporte, como “Problemas para acessar?” e “Falar com suporte”, fornecendo alternativas caso o usuário enfrente dificuldades no *login*. Ao selecionar “Problemas para acessar” o usuário é redirecionado à tela onde pode solicitar um link de recuperação de senha, conforme mostrado no lado direito da Figura 2. O design segue o padrão minimalista e utiliza o *layout* escuro, alinhado com o restante do sistema, promovendo conforto visual.

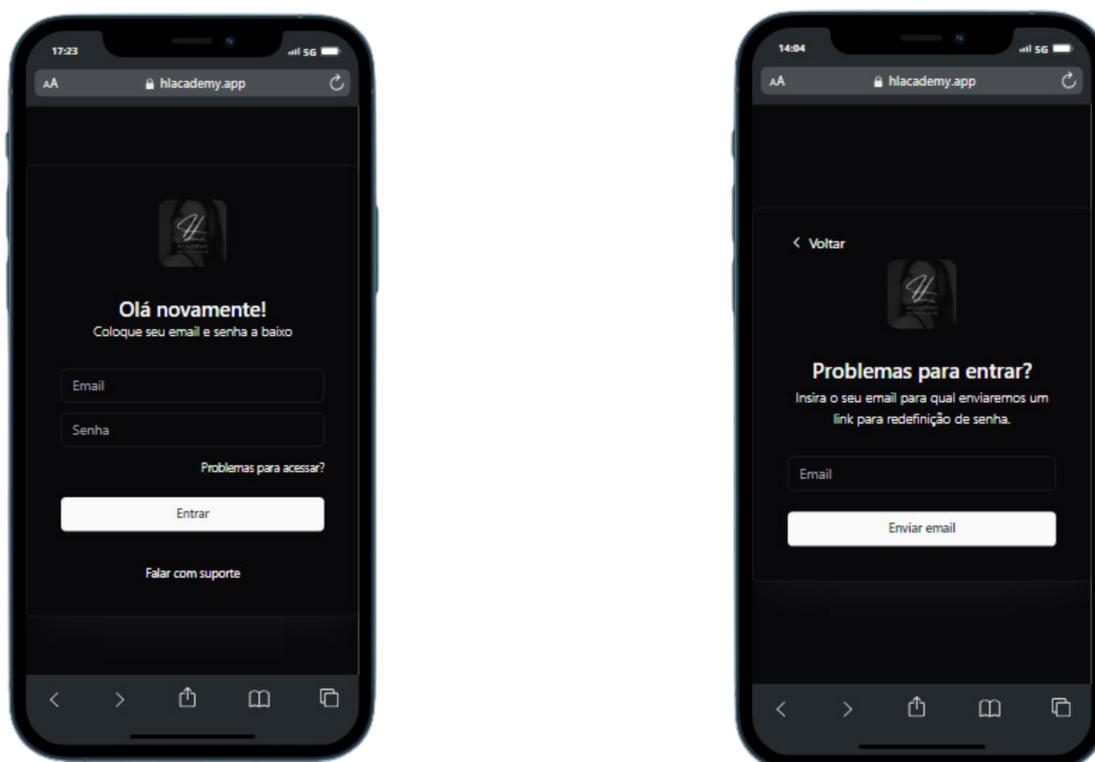


Figura 2: À esquerda: Área para *login*. À direita: Área para solicitar recuperação de senha.

Fonte: De autoria própria.

3.5.1.2 Tela inicial

A tela inicial (*home page*), conforme ilustrado na Figura 3, é a primeira interface visual apresentada ao usuário, após a autenticação bem-sucedida. No topo da tela, exibe-se uma saudação personalizada com o nome do usuário, proporcionando uma interação mais amigável. Logo abaixo, é possível visualizar os cursos disponíveis e que o usuário possui acesso, apresentados em uma seção dedicada, permitindo fácil acesso ao conteúdo. A navegação é facilitada através de uma barra inferior com ícones que direcionam o usuário para as principais seções da plataforma, que são as seções de: favoritos, *home* e figurinhas.

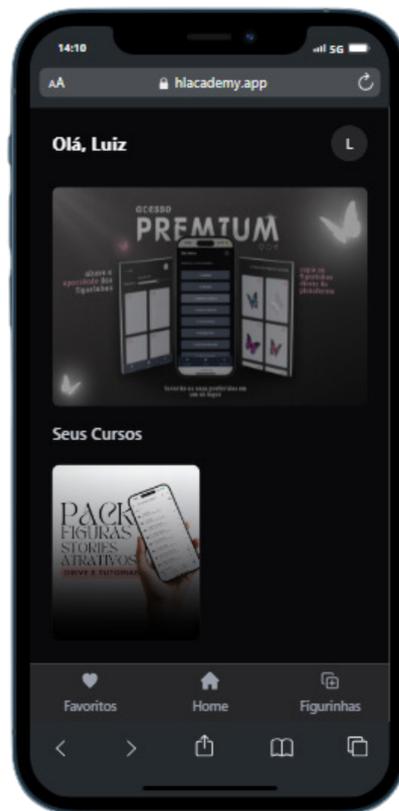


Figura 3: Tela inicial (*Home*).

Fonte: De autoria própria.

No canto superior direito da figura anterior temos um ícone circular contendo a inicial do nome do usuário, que ao ser clicado, exibe um menu suspenso com as opções de início, configurações, suporte e também *logout*, que pode ser feito através do botão “sair”, conforme mostra a Figura 4. Esse menu proporciona ao usuário um acesso rápido a funcionalidades importantes, como ajustes de configurações da conta e suporte técnico, além de permitir o *logout* de forma prática.



Figura 4: Tela inicial com menu suspenso.
Fonte: De autoria própria.

3.5.1.3 Tela de curso

Na Figura 5, é ilustrada a tela de informações de um curso, que é exibido após o usuário selecionar algum dos conteúdos disponíveis. Nesta tela, são apresentadas informações detalhadas sobre o conteúdo oferecido ao usuário. Na parte superior, está o título do curso, acompanhado de uma breve descrição que informa o objetivo, bem como, a quantidade de módulos disponíveis. Imediatamente abaixo, há uma lista de módulos organizada em formato de acordeão, permitindo que o usuário expanda cada um para visualizar os tópicos específicos. As informações contidas na figura foram borradas, para preservar políticas de privacidade da aplicação.

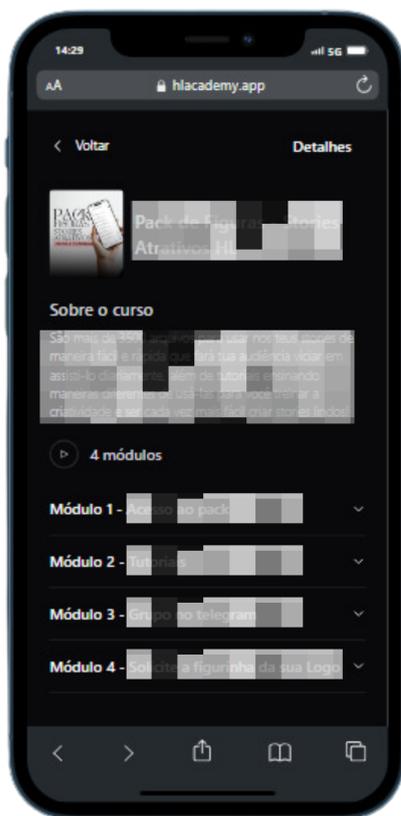


Figura 5: Tela do curso.
Fonte: De autoria própria.

3.5.1.4 Tela de aulas

A tela que permite uma visão das aulas, conforme ilustrado na Figura 6, apresenta o conteúdo em vídeo, sendo o principal componente um *player* de vídeo, que permite assistir às aulas diretamente pela plataforma. Abaixo do *player* de vídeo, o usuário encontra uma descrição detalhada do conteúdo da aula. Assim como na seção anterior, as informações contidas na figura foram borradas, para preservar as políticas de privacidade do sistema.

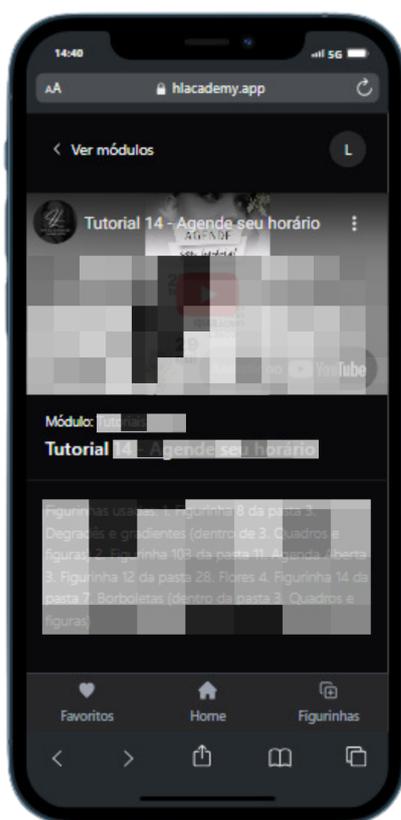


Figura 6: Tela de exibição de aula.
Fonte: De autoria própria.

3.5.1.5 Tela de seleção de nicho e pesquisa das imagens

Para a próxima tela são listados diferentes nichos das imagens possuídas pelo sistema, como exibido ao lado esquerdo da Figura 7. Estas imagens são chamadas de “figurinhas” e o “nicho” representa a categoria a qual a imagem pertence. A aplicação organiza as imagens em categorias específicas para facilitar a navegação e seleção pelo usuário. Cada seção corresponde a um nicho temático, como sombras, quadros e figuras, frases e palavras, entre outros. Os botões que dão acesso às imagens são dispostos em formato de lista, proporcionando uma navegação intuitiva. Além disso, há um campo de pesquisa na parte superior, permitindo que o usuário encontre figurinhas específicas com base no texto digitado.

3.5.1.6 Tela de seleção de imagens

A tela de seleção de imagens, como representada ao lado direito da Figura 7, permitem a exibição em 2 colunas, onde o usuário poderá escolher a arte a ser copiada. Cada imagem apresenta um ícone de coração, que permite a marcação e inclusão da imagem na aba de “favoritos”, da mesma maneira que possui um ícone de engrenagem

para acessar a funcionalidade que configura a opacidade da imagem, proporcionando personalização ao usuário. A interação que copia a imagem para o dispositivo do usuário ocorre ao tocar na imagem desejada.

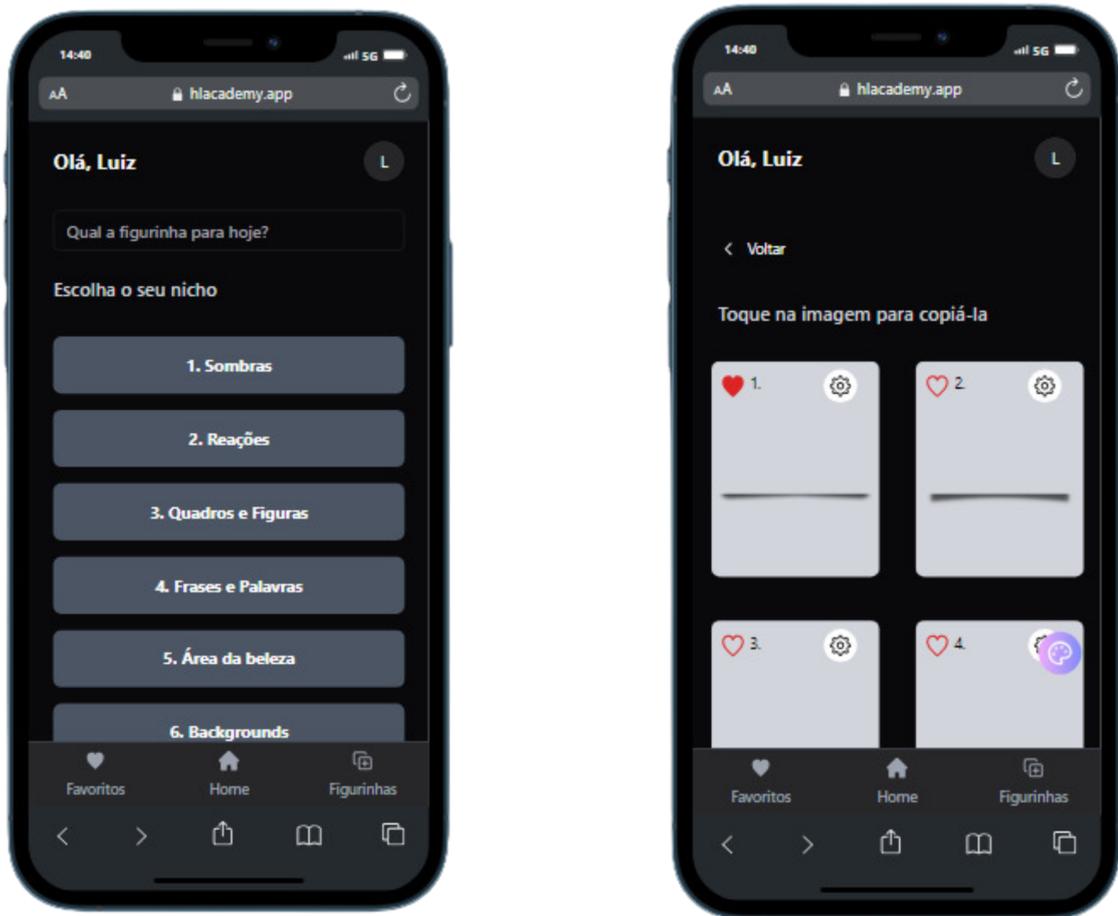


Figura 7: À esquerda: Tela de seleção de nichos e pesquisa das imagens. À direita: Tela de seleção de imagens.

Fonte: De autoria própria.

Ao selecionar uma imagem, ela é transformada em blob, um formato de dados que facilita o armazenamento temporário no navegador, permitindo que o usuário copie a imagem diretamente para o dispositivo. A transformação acontece por meio da execução de uma função feita em JavaScript, que é invocada após ocorrer um evento de clique na imagem selecionada. Essa abordagem garante que as figurinhas possam ser manipuladas de maneira eficiente e rápida, oferecendo uma experiência de usuário mais prática.

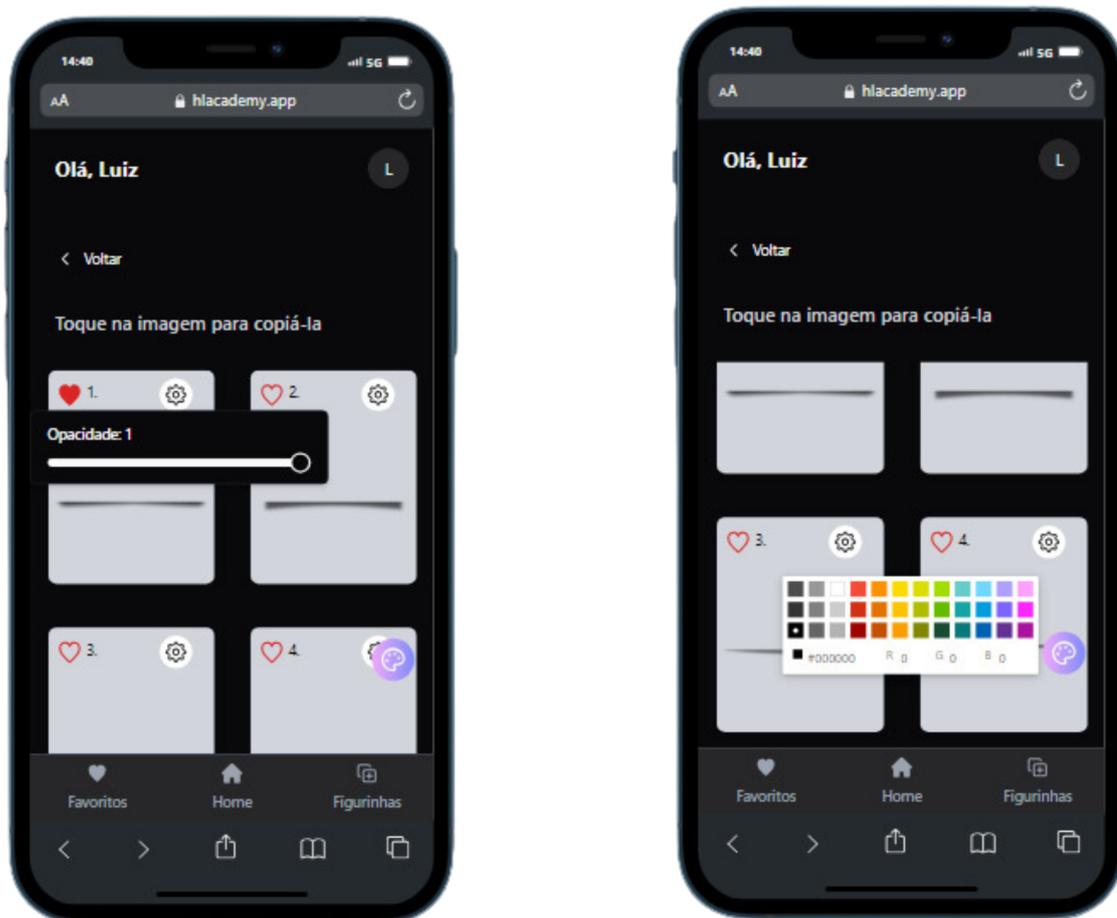


Figura 8: À esquerda: Menu para alteração de opacidade. À direita: Menu para alteração de cor.

Fonte: De autoria própria.

Na Figura 8, é possível observar os menus para alteração de opacidade e cor, respectivamente, da esquerda para a direita. O menu de opacidade é ativo ao clicar no ícone de engrenagem, que fica disponível na parte superior de cada imagem. Já a opção que permite a alteração de cor, é exibida após a interação do usuário com o botão de cor degradê, que possui um ícone de paleta de cores localizado no interior. O lado esquerdo da Figura 9 mostra uma tonalidade de vermelho selecionada, onde é possível perceber que todas as imagens têm sua cor alterada, com base na escolha do usuário.

Ademais, é importante notar que o coração preenchido com a cor vermelha representa que a imagem foi adicionada à aba de favoritos, já o coração sem preenchimento interno, representa a não adição, isto é, a configuração padrão de cada imagem. Essa condição pode ser observada no lado esquerdo da Figura 8.

3.5.1.7 Tela de favoritos

A tela de imagens adicionadas aos favoritos, como ilustrado ao lado direito da Figura 9, exibe todas as quais foram previamente marcadas com o ícone de coração. Essas imagens são exibidas em formato de grade, tal qual a tela de seleção de figurinhas, facilitando o acesso rápido aos conteúdos favoritos do usuário. Assim como na tela de seleção de imagem, neste painel o usuário consegue personalizar e selecionar qualquer imagem.

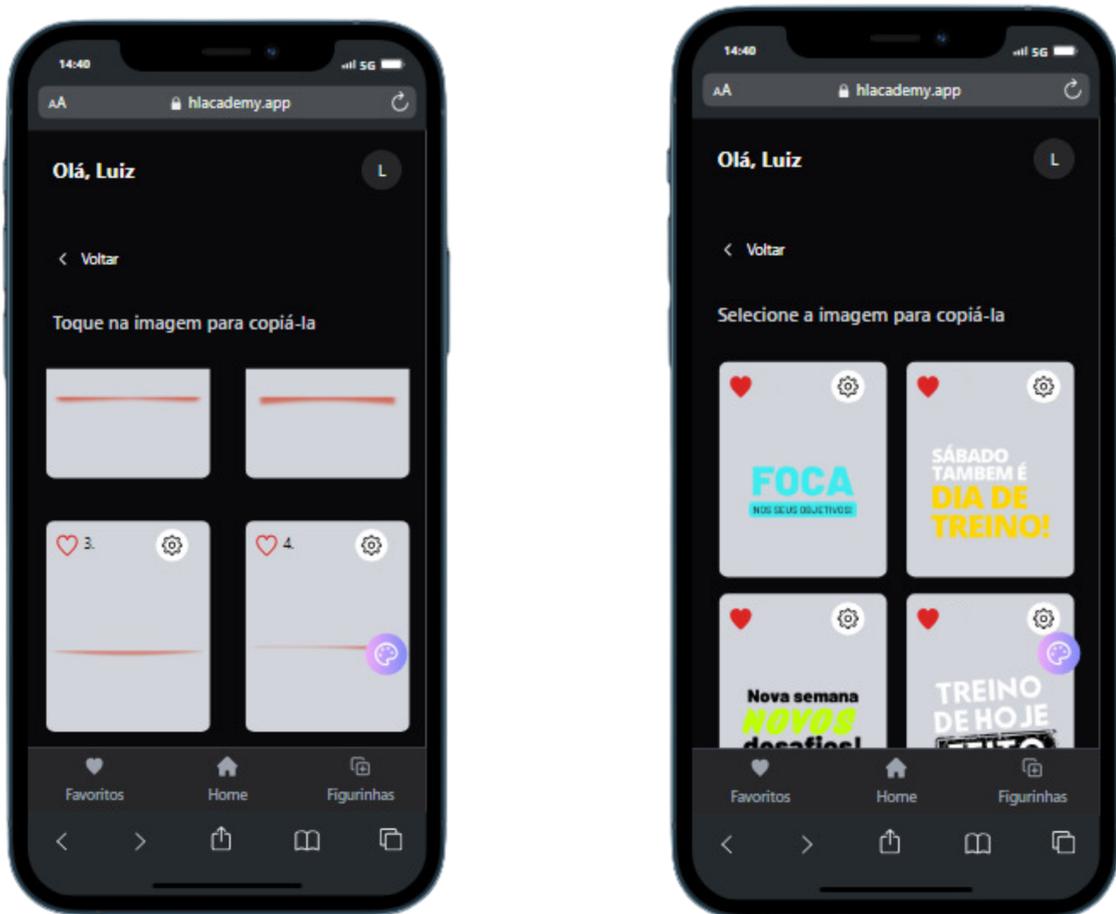


Figura 9: À esquerda: Imagens com cores alteradas. À direita: Tela de imagens favoritas

Fonte: De autoria própria.

4 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Para análise e exibição dos resultados, foi realizada a elaboração de um plano de testes, para assegurar o bom funcionamento da plataforma. O plano elaborado será exposto neste capítulo.

4.1 Plano de testes

A elaboração do plano de testes visa identificar, de maneira antecipada, eventuais falhas e defeitos na aplicação desenvolvida, contribuindo para a qualidade do produto final, visando minimizar o número de falhas. Para isso, foram realizados testes em funcionalidades consideradas fundamentais para o bom funcionamento da aplicação.

Para execução do plano, o *front-end* foi hospedado no *Google Cloud Hosting*, já para as funcionalidades e processos que devem ser executadas no *back-end*, utilizou-se o *Google Cloud Functions*. A partir dessa infraestrutura, os casos de testes foram efetuados majoritariamente em *smartphones*, com o acesso sendo realizado nos principais navegadores, incluindo *Google Chrome*, *Firefox*, *Edge* e *Safari*. O objetivo foi garantir que a aplicação mantivesse seu comportamento e bom funcionamento em diferentes navegadores e dispositivos móveis com diferentes sistemas operacionais. Outrossim, os testes foram conduzidos pelo autor deste trabalho.

4.2 Casos de teste

Os casos de testes listados nas Tabelas 1-11 foram selecionados para abranger funcionalidades cruciais para o correto funcionamento do sistema, bem como para atender às necessidades que o sistema se propõe a solucionar.

CT01	Logar usuário.
Descrição	Avaliar o processo de <i>login</i> do usuário, verificando tanto o sucesso quanto a falha na autenticação.
Pré-condições	Conta de usuário válida já cadastrada.
Passos do teste	<ul style="list-style-type: none"> • Inserir credenciais válidas (e-mail e senha) e clicar em “Entrar”. • Inserir credenciais inválidas (e-mail ou senha incorretos) e tentar o <i>login</i>.
Resultado esperado	<ul style="list-style-type: none"> • Com credenciais corretas: O usuário deve ser autenticado e redirecionado para a tela inicial; • Com credenciais incorretas: Uma mensagem de erro “Credenciais inválidas” deve ser exibida.
Resultado obtido	Êxito.

Tabela 1: Caso de teste [CT01].

CT02	Navegar na tela inicial.
Descrição	Verificar a navegação na tela inicial.
Pré-condições	Usuário autenticado.
Passos do teste	<ul style="list-style-type: none"> • Clicar nos ícones de “Favoritos”, “Home” e “Figurinhas”.
Resultado esperado	<ul style="list-style-type: none"> • Usuário navega corretamente entre as seções.
Resultado obtido	Êxito.

Tabela 2: Caso de teste [CT02].

CT03	Acessar curso.
Descrição	Testar a exibição correta dos cursos cadastrados.
Pré-condições	Usuário autenticado e curso cadastrado no sistema..
Passos do teste	<ul style="list-style-type: none"> • Acessar a tela inicial, clicar em um curso e verificar se o curso é exibido corretamente.
Resultado esperado	<ul style="list-style-type: none"> • O curso é exibido corretamente.
Resultado obtido	Êxito.

Tabela 3: Caso de teste [CT03].

CT04	Marcação de uma imagem como favorita.
Descrição	Testar a funcionalidade de marcar uma figurinha como favorita.
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • Clicar no ícone de coração sem preenchimento interno, de alguma imagem.
Resultado esperado	<ul style="list-style-type: none"> • A imagem é marcada como favorita, alterando o ícone para um coração preenchido de vermelho e é salva na seção “Favoritos”.
Resultado obtido	Êxito.

Tabela 4: Caso de teste [CT04].

CT05	Buscar imagem por texto.
Descrição	Verificar o funcionamento do campo de busca textual de imagens.
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • Inserir um termo de busca e verificar os resultados retornados.
Resultado esperado	<ul style="list-style-type: none"> • Imagens relacionadas ao termo inserido são exibidas corretamente.
Resultado obtido	Êxito.

Tabela 5: Caso de teste [CT05].

CT06	Copiar uma imagem
Descrição	Testar a funcionalidade de cópia de imagens.
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • Selecionar uma imagem e clicar para copiá-la.
Resultado esperado	<ul style="list-style-type: none"> • A imagem é transformada em blob e copiada para o dispositivo do usuário.
Resultado obtido	Êxito.

Tabela 6: Caso de teste [CT06].

CT07	Alterar a cor de uma imagem
Descrição	Verificar a personalização por cor das imagens
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • Selecionar o ícone de paleta de cores e selecionar alguma cor.
Resultado esperado	<ul style="list-style-type: none"> • A imagem tem sua cor alterada, conforme escolha do usuário.
Resultado obtido	Êxito.

Tabela 7: Caso de teste [CT07].

CT08	Alterar a opacidade de uma imagem
Descrição	Verificar a personalização da opacidade das imagens
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • Selecionar o ícone de engrenagem de uma imagem e altera o valor da opacidade.
Resultado esperado	<ul style="list-style-type: none"> • A imagem tem sua opacidade alterada, conforme escolha do usuário.
Resultado obtido	Êxito.

Tabela 8: Caso de teste [CT08].

CT09	Visualizar categorias das imagens
Descrição	Verificar a visualização correta das categorias (nichos) de imagens.
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • Acessar a seção de “Figurinhas” e verificar se todas as categorias estão sendo exibidas.
Resultado esperado	<ul style="list-style-type: none"> • Todas as categorias são listadas corretamente.
Resultado obtido	Êxito.

Tabela 9: Caso de teste [CT09].

CT10	Seleção de uma categoria
Descrição	Testar a abertura das imagens, caso uma categoria seja selecionada
Pré-condições	Usuário autenticado e imagens cadastradas.
Passos do teste	<ul style="list-style-type: none"> • O usuário seleciona uma categoria para visualizar as imagens.
Resultado esperado	<ul style="list-style-type: none"> • Usuário consegue visualizar as imagens da categoria selecionada.
Resultado obtido	Êxito.

Tabela 10: Caso de teste [CT10].

CT11	Acesso às aulas de um curso
Descrição	Verificar se as informações das aulas dos cursos são exibidas corretamente
Pré-condições	Usuário autenticado.
Passos do teste	<ul style="list-style-type: none"> • Selecionar um curso, acessar um módulo e selecionar um aula.
Resultado esperado	<ul style="list-style-type: none"> • Usuário consegue acessar as informações da aula.
Resultado obtido	Êxito.

Tabela 11: Caso de teste [CT11].

4.3 Análise dos resultados

Os testes realizados na aplicação indicaram que as principais funcionalidades operaram conforme o esperado, garantindo uma experiência estável e eficiente para os usuários. O sistema apresentou uma boa adaptação a diferentes tamanhos de tela, sem comprometer a usabilidade, porém, é importante salientar que o foco da aplicação são dispositivos móveis, conforme apontado no capítulo anterior.

No entanto, foi identificado um comportamento específico relacionado à cópia de imagens em dispositivos móveis, em um cenário onde existiu uma baixa velocidade de rede. Neste caso específico, a operação apresentou ligeira latência, mas sem comprometer o resultado final. O processo de manipulação das imagens, convertendo-as em formato blob para facilitar a cópia para o dispositivo do usuário, foi realizado com sucesso.

A aplicação também foi testada em diferentes navegadores e dispositivos, mostrando uma compatibilidade satisfatória em todas as combinações de tecnologia. Em suma, os testes foram concluídos com êxito e a aplicação apresentou-se apta a oferecer um banco de imagens personalizáveis, para empreendedores. Além disso, é importante salientar que, embora os testes tenham sido bem sucedidos, apenas foram realizados casos de testes em funções cruciais para o atingir o objetivo do sistema.

5 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresentou o desenvolvimento de uma aplicação *Web* voltada para pequenos empreendedores, com foco em proporcionar ferramentas de design visual para redes sociais. A aplicação foi projetada para facilitar a personalização de imagens, além de oferecer funcionalidades educacionais sobre a criação de conteúdo visual. A análise do tema revelou a importância de soluções simples e acessíveis para esse público, visando a ampliação da presença digital de pequenos negócios.

Apesar dos resultados positivos, foram identificados pequenos desafios, como uma leve latência na cópia de figurinhas em conexões de internet lenta e também a necessidade de ampliação dos casos de testes para o sistema. No entanto, essas questões não comprometeram o funcionamento geral da aplicação, que se mostrou eficiente e funcional para o que se propõe.

A aplicação oferece espaço para melhorias e evolução, com potencial para incluir novas funcionalidades e otimizações, conforme o *feedback* dos usuários e as demandas de mercado. Para isso, algumas melhorias que podem ser acrescentadas em trabalhos futuros são:

- Introduzir um algoritmo para recomendação de imagens, com base nas imagens utilizadas e perfil do usuário.
- Auxílio na confecção de paletas de cores personalizadas, com base nas cores preferidas do usuário.
- Buscador baseado em imagem, para que o usuário consiga encontrar imagens semelhantes, com base em uma imagem previamente carregada.
- Criação de um agente inteligente, para auxiliar os usuários em outros processos de design, como criação de conteúdos, para suas redes sociais.

REFERÊNCIAS

- [1] Qual a importância de fortalecer sua presença digital - Sebrae. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/qual-a-importancia-de-fortalecer-sua-presenca-digital,d4bbf6fe59164810VgnVCM100000d701210aRCRD>. Acesso em: 2 de out. de 2024.
- [2] Tendências Digitais 2023. Disponível em: <https://www.comscore.com/por/Insights/Apresentacao-e-documentos/2023/Tendencias-Digitais-2023>. Acesso em: 2 de out. de 2024.
- [3] CERQUEIRA, M. S. N. **Mídias sociais e microempreendedores individuais: uma interação cada vez mais globalizada para o empreendedor**. Trabalho de Conclusão de Curso (Bacharel em Administração), 2020.
- [4] HOFFMANN, M. **Vision: Elementary and complex visual processing**. Em: **Cognitive, Conative and Behavioral Neurology**. Springer International Publishing, 2016. p. 51–82. Disponível em: https://doi.org/10.1007/978-3-319-33181-2_4.
- [5] Entenda o poder das imagens no marketing. Disponível em: <https://www.alura.com.br/artigos/entenda-o-poder-das-imagens-no-marketing>. Acesso em: 29 de ago. de 2024.
- [6] OPOKU, E. et al. **Enhancing small and medium-sized businesses through digitalization**. World Journal of Advanced Research and Reviews, v. 23, n. 2, p. 222–239, 2024.
- [7] Brandon, Atkinson. 4. **Custom SharePoint Solutions with HTML and JavaScript: For SharePoint 2013 and SharePoint Online**. (2015).
- [8] Ari, Shapiro., Miriam, Efrach. **System and method for accessing functionality of a backend system from an application server**. (2002).
- [9] EISMANN, S. et al. **Serverless applications: Why, when, and how?** **IEEE software**, v. 38, n. 1, p. 32–39, 2021. Disponível em: <https://doi.org/10.1109/ms.2020.3023302>.
- [10] SCHLEIER-SMITH, J. et al. **What serverless computing is and should become: The next phase of cloud computing**. Communications of the ACM, v. 64, n. 5, p. 76–84, 2021. Disponível em: <https://doi.org/10.1145/3406011>.
- [11] PANDA, R. M.; DAYA SAGAR, B. S. **Database management system**. Em: **Encyclopedia of Earth Sciences Series**. Cham: Springer International Publishing, 2023. p. 247–250. Disponível em: https://doi.org/10.1007/978-3-030-85040-1_80.

- [12] BINANI, S.; GUTTI, A.; UPADHYAY, S. **SQL vs. NoSQL vs. NewSQL- A Comparative Study**. Communications on applied electronics, v. 6, n. 1, p. 43–46, 2016. Disponível em: <https://doi.org/10.5120/cae2016652418>.
- [13] GILNEK, V.; SANTANDER, V. **Elaboração e avaliação de um Catálogo de Requisitos Não-Funcionais Críticos em um Contexto de Domínio de Aplicação de Diplomas Digitais**. Anais do Workshop em Engenharia de Requisitos, 2023.
- [14] G. Costa and D. Nejeliski. **Definição de requisitos não funcionais em projetos da economia criativa**. Diálogo Com a Economia Criativa, vol. 3, no. 9, p. 10-21, 2018. Disponível em: <https://doi.org/10.22398/2525-2828.3910-21>
- [15] B. Cidral and M. Cinelli. **(des)conforto visual: uma análise do ofuscamento**. Blucher Design Proceedings, p. 5346-5357, 2019. Disponível em: https://doi.org/10.5151/ped2018-7.2_aco_10
- [16] B. Pêgo, J. Rocha, & G. Santos. **Prevalência de fear of missing out e fatores associados em estudantes universitários: sociabilidade, bemestar e cultura digital**. Interfaces Científicas - Humanas E Sociais, vol. 9, no. 3, p. 410-425, 2022.
- [17] P. Matri, A. Costan, G. Antoniu, J. Montes, & M. Pérez. **Blob storage meets built-in transactions**. SC16: International Conference for High Performance Computing, Networking, Storage and Analysis, 2016.
- [18] JavaScript. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 2 de out. de 2024.
- [19] React. Disponível em: <https://react.dev/>. Acesso em: 2 de out. de 2024.
- [20] Tailwind UI - official Tailwind CSS components & templates. Disponível em: <https://tailwindui.com/>. Acesso em: 2 de out. de 2024.
- [21] Vite. Disponível em: <https://vite.dev/>. Acesso em: 2 de out. de 2024.
- [22] Firestore. Disponível em: <https://firebase.google.com/docs/firestore?hl=pt-br>. Acesso em: 2 de out. de 2024.
- [23] Compute Engine. Disponível em: <https://cloud.google.com/products/compute?&hl=pt.br>. Acesso em: 2 de out. de 2024.
- [24] RIOS, F. L. DE C.; JANISSEK-MUNIZ, R. **UMA PROPOSTA DE RELAÇÃO DE REQUISITOS FUNCIONAIS PARA UM SOFTWARE DE APOIO AO PROCESSO DE INTELIGÊNCIA**. v. 20, n. 2, 2014.

- [25] Mobile site abandonment after delayed load time. Disponível em: <https://www.thinkwithgoogle.com/consumer-insights/consumer-trends/mobile-site-load-time-statistics/>. Acesso em: 15 de out. de 2024.
- [26] Envato. Disponível em: <https://elements.envato.com/pt-br/>. Acesso em: 22 de out. de 2024.
- [27] SOMMERVILLE, I. **Software Engineering, Global Edition**. 10. ed. Pearson Education, 2015.
- [28] IBM. Disponível em: <https://www.ibm.com/docs/pt/i/7.4?topic=roadmap-availability-concepts>. Acesso em: 22 de out. de 2024.
- [29] BENSLIMANE, D.; DUSTDAR, S.; SHETH, A. **Services mashups: The new generation of web applications**. [s.l.] IEEE Internet Computing, 2008.