



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Algoritmia: Aprendizagem Gamificada para Estudantes de Computação

Kesley Kenny Vasques Guimarães
Pedro Henrique de Brito Agnes

Monografia apresentada como requisito parcial
para conclusão do Curso de Computação — Licenciatura

Orientador
Prof. Dr. Jan Mendonça Correa

Brasília
2023

Dedicatória

Dedicamos este trabalho primeiramente a Deus, por concessão de vida, saúde e fé e às nossas famílias por sempre estarem presentes em nossas vidas e por todo o apoio que nos deram ao longo de toda esta caminhada.

Agradecimentos

A Deus pela saúde e força para superar as dificuldades.

Ao nosso orientador Jan Mendonça Correa por tornar este trabalho possível e por toda a ajuda oferecida.

A todos os professores que nos lecionaram durante todos esses anos na UnB, que nos permitiram um acesso facilitado ao conhecimento.

Aos nossos amigos e colegas que participaram desta jornada acadêmica, e juntos, superamos as diversas dificuldades encontradas.

Às nossas famílias, por todo o apoio incondicional.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Este trabalho descreve a criação e funcionamento da plataforma intitulada **Algoritmia**. Trata-se de um aplicativo móvel para dispositivos com o sistema Android para a aprendizagem de computação de forma gamificada para estudantes e uma aplicação *Web* para a criação de cursos de computação com ferramentas que facilitam a implementação da gamificação para professores. O objetivo principal do projeto é que seja uma ferramenta gratuita que facilite o aprendizado da computação de forma a tentar aumentar a motivação e o engajamento dos alunos por meio da gamificação.

Palavras-chave: software educacional, gamificação, ensino de computação

Abstract

This work describes the creation and functionalities of the platform named **Algoritmia**. It is a mobile application developed for Android devices to help students learn computer related topics using gamification and a Web application to help teachers implement gamification on the creation of computer related courses. The main goal of the project is to be a free tool that makes the learning of computer science easier by trying to increase student's motivation and engagement through gamification.

Keywords: educational software, gamification, teaching computer science

Sumário

1	Introdução	1
1.1	Problema	1
1.2	Justificativa	1
1.3	Objetivo Geral	2
1.4	Objetivos Específicos	2
1.5	Metodologia	2
1.6	Trabalhos Relacionados	3
2	Gamificação	5
2.1	Jogos	5
2.2	Jogos eletrônicos	6
2.3	A Gamificação no processo de Ensino e Aprendizagem	6
2.3.1	Elementos de <i>Design</i> de Jogos que podem ser aplicados na Gamifi- cação	7
2.4	O Ensino de Computação	8
2.5	Gamificação no Algoritmia	9
3	Processo de Desenvolvimento do Aplicativo	11
3.1	Processo	11
3.1.1	Figma	13
3.1.2	Markdown	17
3.1.3	<i>Lucid Chart</i>	18
3.1.4	PlantUML	21
3.1.5	Notion	23
3.1.6	Git e Git Flow	23
3.2	Tecnologias Utilizadas	25
3.2.1	Android	25
3.2.2	Typescript	26
3.2.3	React e React Native	27

3.2.4	<i>Material UI</i>	27
3.2.5	Expo	28
3.2.6	Jest	28
3.2.7	Visual Studio Code	29
3.2.8	Firebase	29
4	O Aplicativo Algoritmia	30
4.1	Front Office	31
4.1.1	Login e Registro	31
4.1.2	<i>Home</i> e Navegação	33
4.1.3	<i>Ranking</i> e Pontuação	34
4.1.4	Dados de Usuário	35
4.1.5	Conversa	36
4.1.6	Tópicos e Desafios	36
4.2	Back Office	39
4.2.1	Login e Registro	39
4.2.2	<i>Home</i> e Navegação	40
4.2.3	Perfil	42
4.2.4	Times	42
4.2.5	Participantes e Tópicos	44
4.2.6	Subtópicos	45
4.2.7	Desafios	46
4.3	Testes Preliminares	48
4.3.1	Nível de ensino em computação	49
4.3.2	Nota para a experiência de uso do aplicativo	49
4.3.3	O aplicativo possui uma interface intuitiva?	49
4.3.4	Você acredita que a plataforma é capaz de auxiliar os estudantes e professores no ensino e aprendizagem de computação?	50
4.3.5	Você acha que o formato dos exercícios e do ranking serviram como fatores motivadores durante a resolução dos desafios?	50
4.3.6	Pontos positivos e negativos e sugestões	50
5	Considerações Finais	52
5.1	Trabalhos Futuros	53
	Referências	54

Lista de Figuras

3.1	Protótipos de telas pertencentes às rotas de autenticação.	14
3.2	Protótipos de telas contendo a <i>Home</i> do aplicativo e seleção de Tópicos. . .	15
3.3	Protótipos de telas contendo a página de tópico com opções para navegação para os desafios cadastrados e para o <i>ranking</i>	15
3.4	Protótipos de telas contendo as funções de envio e recebimento de mensagens no <i>Algoritmia</i>	16
3.5	Protótipos de telas pertencentes às rotas de gerenciamento de conta.	17
3.6	Fluxo de utilização do aplicativo móvel <i>Algoritmia</i> na visão do usuário. .	19
3.7	Fluxo de utilização da aplicação <i>Web Algoritmia</i> na visão do usuário. . .	20
3.8	Diagrama de classes do <i>Algoritmia</i>	22
3.9	Exemplo visual do Git Flow.	24
4.1	Telas de <i>Login</i> , Registro e Pré- <i>Home</i> - <i>Front Office</i>	32
4.2	Telas acessíveis pela barra de navegação: <i>Home</i> , Conta e Mensagens - <i>Front Office</i>	33
4.3	Telas da funcionalidade de <i>Ranking</i> : <i>Home</i> e <i>Ranking</i> - <i>Front Office</i>	34
4.4	Telas para gerenciamento de conta - <i>Front Office</i>	35
4.5	Telas para a funcionalidade de <i>Chat</i> - <i>Front Office</i>	36
4.6	Telas relacionadas à funcionalidade de Tópicos - <i>Front Office</i>	37
4.7	Telas de Desafio e <i>Feedback</i> - <i>Front Office</i>	38
4.8	Tela de Login - <i>Back Office</i>	39
4.9	Tela de Cadastro - <i>Back Office</i>	40
4.10	Tela <i>Home</i> - <i>Back Office</i>	41
4.11	Tela <i>Home</i> - <i>Back Office</i>	41
4.12	Tela de gerenciamento de Perfil - <i>Back Office</i>	42
4.13	Tela de Times - <i>Back Office</i>	43
4.14	Tela de Times - Novo time cadastrado - <i>Back Office</i>	43
4.15	Tela do Time selecionado - Participantes - <i>Back Office</i>	44
4.16	Tela do Time selecionado - Tópicos - <i>Back Office</i>	45
4.17	Tela do Tópico selecionado - Subtópicos - <i>Back Office</i>	46

4.18 Tela do Subtópico selecionado - Desafios - <i>Back Office</i>	47
4.19 Tela do Desafio selecionado - <i>Back Office</i>	48

Capítulo 1

Introdução

Com os avanços tecnológicos observados no decorrer das últimas décadas, surgem, cada vez mais, oportunidades de melhorar os modelos existentes de ensino e aprendizagem. Nos últimos anos, a tecnologia vem se tornando uma das principais aliadas dos estudantes dentro e fora das salas de aula. Existem diversas ferramentas disponíveis para diferentes nichos educacionais, mas nem todas são gratuitas, acessíveis e, tampouco, despertam o interesse dos alunos.

Este projeto tem como objetivo, criar uma plataforma digital, intitulada **Algoritmia** para auxiliar estudantes e professores no ensino e aprendizagem de tópicos do ramo computacional. **Algoritmia** utiliza uma abordagem gamificada [1] para tornar o aprendizado mais interessante e, conseqüentemente, mais fácil de alcançar o aluno.

1.1 Problema

Existe uma carência de ferramentas capazes de superar as barreiras para o aprendizado de temáticas computacionais. A aprendizagem de computação está comumente relacionada com longas horas de estudos em um ambiente silencioso e com bons equipamentos. Embora estes fatores possam facilitar o processo de aprendizagem, eles não devem ser responsáveis por limitar a busca por conhecimento. Entretanto, a falta de tempo, e de um ambiente adequado, tornam-se grandes impeditivos, capazes de gerar nos estudantes, frustrações ao iniciarem seus estudos sobre determinados assuntos.

1.2 Justificativa

Devido a grande necessidade da construção de recursos didáticos capazes de auxiliar estudantes na busca por conhecimento de elementos da computação, este presente trabalho tem como objetivo, atender, de forma acessível, os estudantes que desejam aperfeiçoar

ou adquirir conhecimentos acerca de diversas temáticas computacionais. Para isso, foi desenvolvido uma plataforma digital, que engloba um aplicativo para dispositivos móveis gamificado, onde os estudantes poderão, gratuitamente, aprender de forma prática, diversos tópicos relacionados às áreas de seu interesse. A plataforma conta também, com uma aplicação *web* para auxiliar professores em práticas pedagógicas.

1.3 Objetivo Geral

Construir uma plataforma digital para solucionar problemas relacionados ao processo de aprendizagem que os estudantes enfrentam no âmbito computacional. Além disso, a plataforma serve de auxílio para professores que desejam utilizar recursos digitais em suas práticas docentes.

1.4 Objetivos Específicos

A fim de atender os objetivos finais do projeto, foram estabelecidos os seguintes objetivos específicos:

- Desenvolver um aplicativo que integre elementos de gamificação em um ambiente de ensino de computação para aumentar a motivação e o envolvimento dos alunos;
- Fornecer uma ferramenta para que os professores usem a tecnologia de maneira efetiva e integrem-na em sua prática docente para melhorar a aprendizagem dos alunos e tornar o ensino mais eficiente e interessante;
- Oferecer um mecanismo de *feedback* imediato e eficaz para os alunos;
- Acompanhar o progresso dos alunos e fornecer relatórios detalhados aos professores.

1.5 Metodologia

Este trabalho foi desenvolvido segundo a linha de pesquisa exploratória. Foram visitadas fontes diversas a respeito das falhas no processo de ensino e aprendizagem de diversas áreas da computação, em específico, relacionadas ao desenvolvimento do pensamento computacional. As etapas da construção do projeto estão relacionadas em:

- Pesquisa sobre os principais pontos de defasagem no ensino de computação;
- Estudo sobre a abordagem da gamificação no processo de aprendizagem;
- *Benchmark* com as principais ferramentas de *e-learning*;

- Estabelecer critérios de aceite;
- Construção de protótipos, fluxogramas e documentação;
- Desenvolvimento das aplicações móvel e web;
- Validação da plataforma por meio de testes unitários, funcionais e exploratórios;
- Avaliar os impactos da plataforma na aprendizagem dos estudantes, e no auxílio aos professores de computação.

1.6 Trabalhos Relacionados

Para que o escopo das aplicações fosse definido, e para fins de referência, foram levantados aplicativos educacionais de diversos segmentos, utilizando-se como base as fontes de pesquisa do Play Store¹, que contém os aplicativos em questão para baixar e o Google Scholar², que contém artigos acadêmicos e livros com estudos de caso sobre eles. As principais palavras-chave utilizadas como parâmetros de busca foram: *Coding Apps*, *E-learning*, *Game-based Learning* e Jogos Educacionais. Com base nas pesquisas, os aplicativos que mais aproximaram-se do contexto do projeto são:

- **Duolingo**³ [2]: aplicativo para ensino de línguas estrangeiras; Possui foco em gamificação; Alto engajamento popular; O diferencial do aplicativo é sua trilha de aprendizado, possibilitando o usuário monitorar o seu progresso por meio de uma interface amigável; Disponível gratuitamente em diversos idiomas, incluindo Português;
- **Khan Academy**⁴ [3]: aplicativo para ensino de conteúdos diversos, em especial, conteúdos acadêmicos e de desenvolvimento pessoal; Possui elementos de gamificação; Alto engajamento popular; O diferencial do aplicativo é sua seção de ensino, com professores especializados em diversas áreas do conhecimento; Disponível gratuitamente em diversos idiomas, incluindo Português;
- **Beecrowd**⁵: aplicativo web para ensino de algoritmos e programação, focado em competição; Possui elementos de gamificação; Alto engajamento popular; O diferencial do aplicativo é seu intenso foco em programação competitiva, fazendo com que os estudantes se mantenham engajados na busca por um bom posicionamento no *ranking*; Disponível gratuitamente em diversos idiomas, incluindo Português.

¹<https://play.google.com>

²<https://scholar.google.com/>

³<https://pt.duolingo.com/>

⁴<https://pt.khanacademy.org>

⁵<https://www.beecrowd.com.br>

O Algoritmia se diferencia de outras plataformas educacionais por sua abordagem gamificada, que atende tanto estudantes quanto docentes. Além disso, é uma plataforma totalmente gratuita e de código aberto com a licença MIT, permitindo maior liberdade àqueles que desejam usá-la.

Capítulo 2

Gamificação

A gamificação [1] é o processo de implementação de elementos de *design* de jogos em contextos do mundo real para propósitos diversos, fora do contexto dos jogos em si. O principal propósito dela é para gerar uma maior motivação nas pessoas ao executarem determinada atividade. Neste capítulo, serão definidos conceitos necessários para a gamificação e serão exploradas as aplicações da mesma no processo de ensino e aprendizagem e como ela foi aplicada no projeto **Algoritmia**.

2.1 Jogos

Jogo é uma expressão que define diversos conceitos diferentes, onde o significado depende de elementos como faixa etária dos participantes, local, contexto, entre muitos outros fatores [4]. Pesquisadores do *Laboratoire de Recherche sur le Jeu et le Jouet*, da *Université Paris-Nord* como Gilles Brougère [5] e Jacques Henriot [6] endereçaram a definição do termo jogo por meio de três níveis de diferenciações, sendo eles:

1. O resultado de um sistema linguístico que funciona dentro de um contexto social;
2. Um sistema de regras;
3. Um objeto.

A definição acima se dá devido à mistura que é feita entre jogos, brincadeiras e brinquedos, que são tratados como um só na maioria dos contextos. Com os níveis definidos, é possível classificar os jogos, assim como diferenciá-los, pois mesmo se dois jogos diferentes utilizam o mesmo objeto para jogar como cartas de baralho, ainda é possível distinguí-los pelas regras, por exemplo.

2.2 Jogos eletrônicos

Jogos eletrônicos, popularmente conhecidos como videogames, são formas de entretenimento que se expressam no meio virtual. São jogos que permitem a interação por meio de um dispositivo eletrônico, como computadores, consoles, *smartphones* e *arcades*.

Segundo dados transmitidos em 2022 da Associação de *softwares* de entretenimento [7], bilhões de pessoas utilizam os jogos eletrônicos como forma de entretenimento diariamente na atualidade, transcendendo barreiras de idade, gênero, cultura e até distância no caso de jogos de múltiplos jogadores *online*. Devido à grande presença dos videogames na atualidade, este se tornou um dos mercados mais rentáveis dos últimos tempos, e com um potencial ainda maior de crescimento [8].

Na atualidade, o mercado de jogos tem sido dominado, principalmente pelos jogos para *smartphones* [9], possuindo 52% da rentabilidade total dos jogos no ano de 2021. Isso se dá pois os celulares são os dispositivos tecnológicos que têm estado cada vez mais presente no cotidiano das pessoas devido à sua portabilidade e popularidade. Segundo dados do IBGE de 2019 [10], 97,4% dos estudantes utilizam um aparelho celular para estudar, incluindo a rede pública e privada. O *Algoritmia* tenta ser uma plataforma que facilita o uso de um *smartphone* como instrumento de aprendizagem.

2.3 A Gamificação no processo de Ensino e Aprendizagem

Com os avanços tecnológicos dos últimos tempos, diversos setores da sociedade se viram forçados a realizarem mudanças para a inclusão das tecnologias mais atuais para possibilitar maior acessibilidade ao público e realizar inovações. Atualmente, muitos jovens do século XXI tiveram acesso a diversas tecnologias no cotidiano durante todo ou grande parte do seu desenvolvimento cognitivo. Visto isso, as escolas e os professores são motivados a se adaptar à nova realidade dos alunos para ensinar de forma mais inclusiva, possibilitando ao outro, o acesso à informação e ao conhecimento [11].

A tecnologia da comunicação pode ser aplicada nas salas de aula de forma a dinamizar o processo de ensino. Os tempos atuais mudaram os padrões educacionais, transformando a educação em um sistema voltado para o desenvolvimento de competências e habilidades. Isso pode ajudar os alunos a compreenderem melhor a realidade, conseqüentemente, tendo a capacidade de causar um impacto positivo no presente e até no futuro dos estudantes [11].

2.3.1 Elementos de *Design* de Jogos que podem ser aplicados na Gamificação

Diversos autores apresentam vários aspectos diferentes para o *design* completo de um jogo, porém uma seleção dos principais itens apresentados recorrentemente pelos autores são: pontuação, distintivos, *ranking*, gráficos de performance, histórias significativas, avatares e colegas [1]. A seguir, cada um dos itens serão descritos detalhadamente.

- **Pontuação** é um elemento básico de uma grande quantidade de jogos e aplicações gamificadas, que representa o sucesso em cumprir determinadas atividade dentro do ambiente do jogo, representando o progresso do jogador. Dentre os principais propósitos dos pontos, estão o de permitir um *feedback* para o jogador e o de servir como uma recompensa pelo trabalho feito no ambiente do jogo;
- **Distintivos** são representações visuais de uma ou mais conquistas e podem ser obtidos ou coletados dentro do ambiente do jogo. Têm como função, confirmar a conquista do jogador, simbolizando o mérito;
- ***Ranking*** é uma classificação de jogadores que é feita com base no seu sucesso relativo, sendo medido de acordo com algum critério determinado pelo jogo, como por exemplo, a própria pontuação. Os efeitos motivadores de um sistema de classificação podem variar bastante, podendo causar uma grande motivação em pessoas que estão a poucos passos de passar de posição, mas também servindo como um desmotivador para os jogadores que se encontram no fundo da classificação [12]. Apesar disso, a competição causada por um ranqueamento, pode criar uma pressão social para aumentar o nível de engajamento do jogador e, conseqüentemente, ter um efeito construtivo na participação e resultado [13];
- **Gráficos de Performance** são mais usados em jogos de simulação ou estratégia e permitem aos jogadores, o acesso à informações sobre a sua evolução no jogo, comparando a performance da partida atual com as anteriores. Ao contrário de um *ranking*, o gráfico de performance não compara a performance do jogador com outros jogadores, mas sim avalia a evolução de sua performance ao longo do tempo;
- **Histórias Significativas** não se relacionam com a performance do jogador. O contexto narrativo em que uma aplicação gamificada pode ser incorporada, contextualiza as atividades e características do jogo, aplicando um significado que vai além da tarefa de obter pontos e conquistas, o que pode gerar resultado positivos para a inspiração e engajamento dos jogadores, principalmente se a história estiver de acordo com seus interesses pessoais [14];

- **Avatares** são representações visuais do jogador dentro do ambiente do jogo, que podem ser escolhidos ou criados pelo jogador. Avatares permitem que os jogadores adotem ou criem uma nova identidade dentro do jogo;
- **Colegas** podem introduzir no jogo, o conflito, a competição e a cooperação [15]. A cooperação pode ser obtida por meio da introdução de times, ao definir grupos de jogadores que trabalham em conjunto para atingir um determinado objetivo mútuo [12].

2.4 O Ensino de Computação

A área de Tecnologia da Informação e Comunicação (TIC) é um dos setores que mais tem crescido nos últimos anos em relação à demanda, oportunidades e, conseqüentemente, procura por qualificação profissional [16]. Com isso, cursos da área de computação, em especial têm mostrado uma alta procura, porém, ao mesmo tempo, mostram uma das maiores taxas de evasão dentre as formações superiores e técnicas [17]. Por exemplo, o curso de sistemas para internet analisado do Instituto Federal do Tocantins apresentou uma taxa de evasão média anual de 51,46% entre 2015 e 2019 e o curso de ciência da computação ofertado pela Universidade Federal do Tocantins uma taxa de 27,08% neste mesmo período.

Segundo um estudo conduzido por Raphael Magalhães Hoed com a orientação do Prof. Dr. Marcelo Ladeira [18], a evasão na UnB nos cursos de computação também se mostrou bem elevada ao analisar uma amostra entre os anos de 2005 a 2015. O estudo mostra uma evasão de 40,7% no curso de Ciência da Computação, 55,6% no curso de Licenciatura da Computação, 34,5% no curso de Engenharia da Computação e 21,3% no curso de Engenharia de Software.

Os motivos para a evasão escolar podem ser diversos, sendo classificados como internos ou externos, mas um dos grandes fatores que têm levado os alunos à evasão [18] são relacionadas às dificuldades enfrentadas pelos alunos em disciplinas de matemática de algoritmos, também influenciando na probabilidade do aluno não evadir mediante a aprovação nessas disciplinas. Cabe à instituição de ensino tomar as medidas necessárias para diminuir as altas taxas de evasão ao analisar os motivos para chegar em uma solução. Todavia, um método de ensino mais inclusivo poderia ter a capacidade de manter os alunos engajados no curso e promover um melhor aprendizado.

Com isso em mente, o **Algoritmia** foi desenvolvido, utilizando dos conceitos de gamificação para auxiliar no processo de aprendizagem. As ferramentas para o uso da gamificação são disponibilizadas para que o professor que decide usar a plataforma seja

capaz de montar o seu curso e matricular os alunos, assim como acompanhar os seus desempenhos. O detalhamento destas funcionalidades será descrito a seguir.

2.5 Gamificação no Algoritmia

A maioria dos elementos de gamificação presentes na compilação apresentada na Seção 2.3.1 foram aplicados de alguma forma no aplicativo **Algoritmia**. Porém, por se tratar de uma plataforma onde um professor pode criar um curso, alguns dos fatores para uma aplicação mais efetiva da gamificação dependem do conhecimento do professor sobre o tema e como ele irá aplicar o uso de conceitos de jogos nos exercícios propostos. A seguir, está descrito como foi aplicado cada um dos aspectos existentes no aplicativo sobre o que se refere à gamificação.

- A **Pontuação** é individual para cada aluno e é diferente para cada time ou curso cadastrado na plataforma. Deste modo, o sistema de ranqueamento de cada time não sofrerá interferência dos outros, garantindo uma plataforma individual para cada um. O usuário ganha pontos ao resolver corretamente os desafios propostos, mas não obtém ponto algum ao responder incorretamente o desafio.
- O **Ranking** foi implementado individualmente para cada curso ou time cadastrado na plataforma conforme mencionado anteriormente. A classificação é gerada unicamente com base na quantidade de pontos obtidos por cada aluno. A quantidade de pontos obtidos em cada desafio é definida pelo professor, porém é limitada a um *range* para não gerar diferenças muito grandes de pontuação em cada desafio.
- A **Performance** dos alunos pode ser visualizada pelos professores para localizar possíveis dificuldades generalizadas ou individuais. O aluno também pode visualizar a sua performance ao finalizar os desafios de um tópico, podendo acessar um *feedback* individual para cada questão ou as estatísticas gerais dos desafios realizados como a porcentagem de acerto e erro e a pontuação total obtida. Ao visualizar os seus erros e acertos, o estudante pode identificar as suas próprias dificuldades, o que permite o auto-aperfeiçoamento ao estudar mais o que lhe falta, assim como pode também recorrer à ajuda dos outros para um crescimento mútuo.
- **Histórias Significativas** não foram implementadas diretamente no aplicativo pois se trata de algo individual para cada contexto e, portanto, cabe ao professor a decisão de incluir a história ou não. O professor poderá seguir a sua própria metodologia de ensino para despertar o interesse e engajamento dos alunos.

- Os **Colegas** de um determinado curso aparecem no *ranking* da plataforma, permitindo que os alunos se comuniquem entre si por meio da ferramenta de conversa por mensagens implementada para solicitar ajuda ou interagir da forma que bem entenderem.

O capítulo seguinte discute aspectos mais técnicos sobre o desenvolvimento e o funcionamento do aplicativo **Algoritmia**.

Capítulo 3

Processo de Desenvolvimento do Aplicativo

O desenvolvimento do aplicativo móvel e da plataforma Web *Algoritmia* foi feito utilizando diversas tecnologias e ferramentas, assim como metodologias. A seguir neste capítulo, serão descritas cada uma das etapas do processo de criação do *app*.

3.1 Processo

O processo de *software* [19] é visto por uma sequência de atividades que produzem uma variedade de documentos, resultando em um programa satisfatório e executável. Cada vez mais, os processos se mostram tão importantes quanto o *software* em si, sendo indispensáveis para a produção de um produto de maior qualidade. O desenvolvimento da aplicação foi feito utilizando a metodologia ágil Scrum [20].

O movimento ágil da indústria de *software* [21] se iniciou com a publicação do manifesto ágil¹ em 2001 por um grupo de praticantes e consultores de *software*. O manifesto agrega os seguintes valores:

- **Indivíduos e Interações** acima de processos e ferramentas;
- **Software em funcionamento** acima de uma documentação compreensiva;
- **Colaboração com o cliente** acima de negociações de contrato;
- **Responder a mudanças** mais que seguir um plano;

Scrum [20] é um *framework* que ajuda as pessoas, times e organizações a gerar valor através de soluções adaptáveis para problemas complexos. De forma resumida, o Scrum

¹<https://agilemanifesto.org/>

requer uma pessoa denominada *Scrum Master* para controlar um ambiente caracterizado por:

1. O dono do produto (*Product Owner*): solicita o trabalho para um item complexo que é enviado para o *Product Backlog*;
2. O time: transforma os itens selecionados em subitens de valor incremental durante uma fase denominada *Sprint*;
3. O time, juntamente com as partes interessadas: avaliam os resultados e fazem os planejamentos e ajustes necessários para a próxima *Sprint*;
4. Processo se repete com a nova *Sprint*.

O Scrum aplicado no **Algoritmia** foi adaptado às necessidades do projeto. Foram usadas *Sprints* para as diversas fases do projeto, incluindo a revisão das tarefas concluídas e o planejamento das próximas tarefas, onde as tarefas do *backlog* são selecionadas para entrar no escopo da *Sprint* conforme necessidade. É comum no Scrum, a ocorrência de reuniões diárias para que a situação das tarefas sejam compartilhadas entre a equipe e os impedimentos sejam expostos para que seja buscada uma solução, porém, por se tratar de uma equipe de apenas duas pessoas e sem a presença de um *Scrum Master*, a frequência das reuniões foi diminuída, ocorrendo, no mínimo uma vez por semana.

O *framework* Scrum não tem uma lista de artefatos de documentação obrigatórios a se produzir, pois se trata de uma metodologia ágil, caracterizada pelo contato pessoal e o funcionamento do produto acima de uma documentação expressiva. Porém, as documentações não são proibidas, e devem ser produzidas conforme a necessidade da equipe e do cliente, servindo como critérios de aceite ou como referência interna para auxiliar no desenvolvimento ou manutenções futuras. Para o projeto, foram desenvolvidos artefatos conforme a necessidade surgia.

Diversos recursos e programas foram utilizados para auxiliar nos processos de desenvolvimento do **Algoritmia** sendo todos de licença livre ou de uso gratuito. Antes de iniciar o desenvolvimento do *software*, foram criados protótipos de telas do projeto, a fim de facilitar a criação das telas funcionais posteriormente. Durante esta etapa, o estilo do aplicativo foi definido, incluindo as cores predominantes, diferentes telas pertencentes à interface de usuário, rotas de navegação e as funções acessíveis por cada uma das telas projetadas. Tudo isso conferiu uma identidade ao projeto de *software*. A seguir, estão descritos os demais artefatos produzidos durante o desenvolvimento do **Algoritmia**:

- Protótipos de telas (3.1.1);
- Fluxo de utilização da aplicação (3.1.3);

- Histórias de Usuário (3.1.2);
- Diagrama de classes (3.1.4).

Com o protótipo inicial do aplicativo móvel criado e aprovado, foi utilizado um *software* de organização de tarefas chamado *Notion* (3.1.5) para realizar o cadastro dos itens a serem implementados no aplicativo e foram criadas histórias de usuário para documentar os critérios de aceite e as funcionalidades a serem desenvolvidas. Com o início da primeira *Sprint*, os primeiros *cards* pendentes de resolução foram divididos entre a dupla e foi iniciado o desenvolvimento da plataforma em questão.

A seguir, algumas das ferramentas utilizadas no processo de planejamento e desenvolvimento do *Algoritmia* são descritas, assim como os artefatos produzidos em cada uma delas.

3.1.1 Figma

Segundo Connel e Shaffer [22], um protótipo de *software* é um modelo dinâmico visual que oferece uma ferramenta de comunicação para o cliente e para o desenvolvedor que é bem mais efetiva do que qualquer prosa narrativa ou modelos estáticos visuais para representar funcionalidade. O *software* Figma² é uma plataforma de design hospedada em formato de aplicação *Web* que facilita a criação de protótipos para aplicativos *mobile*, *web*, entre outros.

A criação dos protótipos do aplicativo móvel foi feita utilizando o Figma de forma gratuita. A plataforma não é de licença livre e permite a criação de conta para o uso gratuito, porém também possui planos pagos de assinatura. Com as ferramentas disponibilizadas pelo Figma, foram prototipadas todas as telas do aplicativo de acordo com a ideia inicial do projeto, assim como o fluxo de navegação por parte do usuário final. Não foram prototipadas as telas da aplicação *Web* (*Back Office*) no Figma devido à utilização dos elementos de *UI* da biblioteca *Material UI* [23]. O protótipo do aplicativo *mobile* pode ser dividido em 4 partes:

Autenticação

Composta por 3 telas: Login, Registro e *Reset* de Senha. As telas são navegáveis entre si por meio dos *links* clicáveis de cor azul ou cinza.

²<https://www.figma.com/>

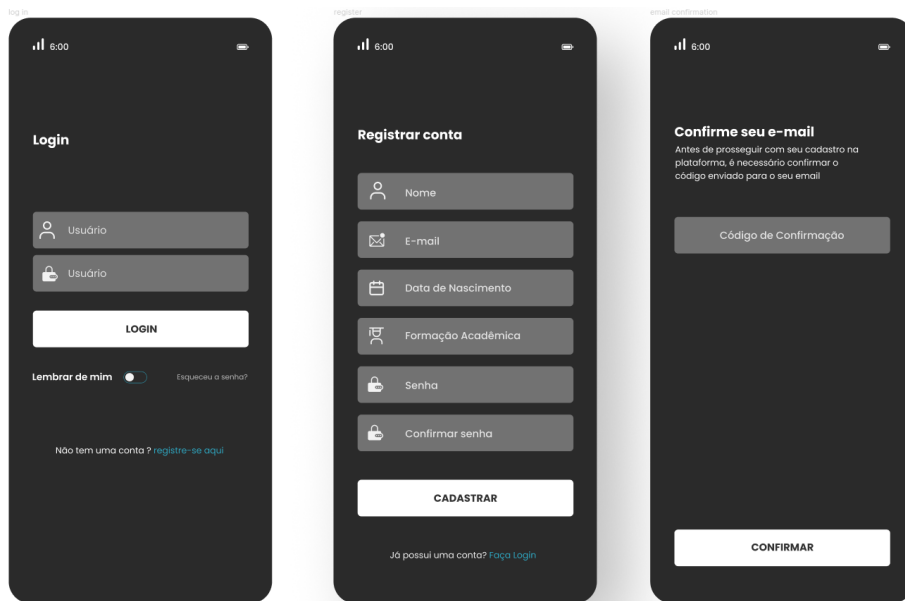


Figura 3.1: Protótipos de telas pertencentes às rotas de autenticação.

Cursos

Definitivamente a parte mais importante do aplicativo, sendo composta por Figura 3.2 e Figura 3.3. Foram prototipadas 6 telas para este item, incluindo a tela inicial chamada de *Pré-Home*, onde o usuário é redirecionado após efetuar o *login*. A tela *Home* é carregada por meio do botão para acessar a área de cada curso, onde se encontram os principais dados do curso, como os tópicos e o *ranking*. Ao acessar um tópico a partir da tela *Home*, o usuário é redirecionado para a tela deste tópico, onde pode selecionar um subtópico para resolver desafios ou visualizar o *ranking* local.

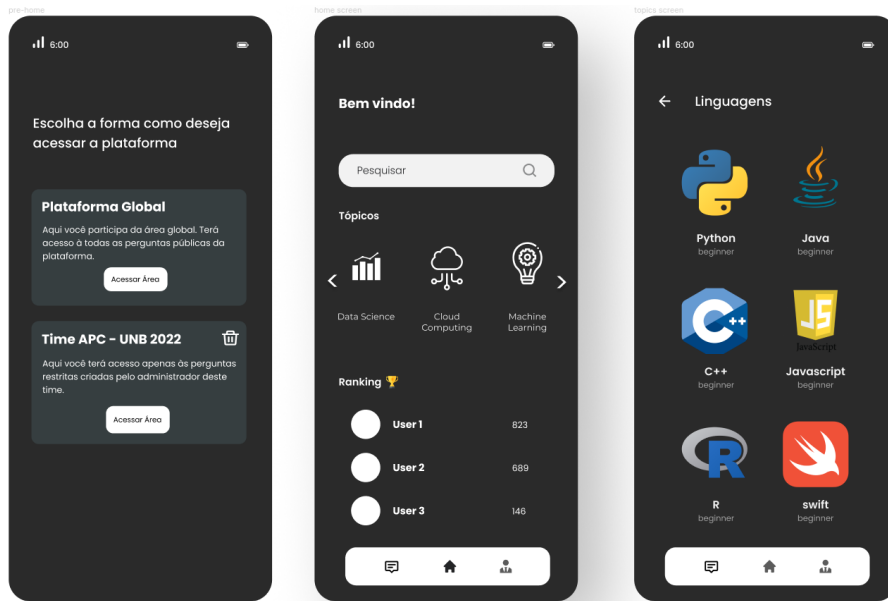


Figura 3.2: Protótipos de telas contendo a *Home* do aplicativo e seleção de Tópicos.

Pela Figura 3.3, é possível visualizar a tela do subtópico com as opções de acessar os desafios disponíveis e o *ranking*. A tela de desafio contém um enunciado com uma imagem opcional e campos para resposta. Por fim, a tela de classificação vai conter os usuários ordenados pelo melhor aproveitamento do curso.

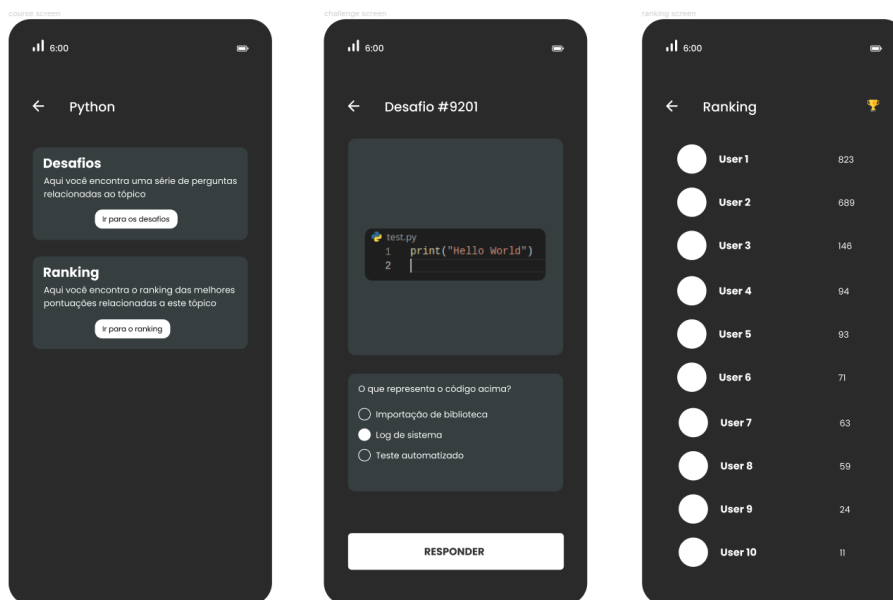


Figura 3.3: Protótipos de telas contendo a página de tópico com opções para navegação para os desafios cadastrados e para o *ranking*.

Chat

Acessível pela barra de navegação, a tela de *Chats* contém todas as pessoas com as quais o usuário logado se comunicou pelo aplicativo. Conforme mostrado na Figura 3.4, a tela de conversa é carregada ao selecionar um dos usuários listados.

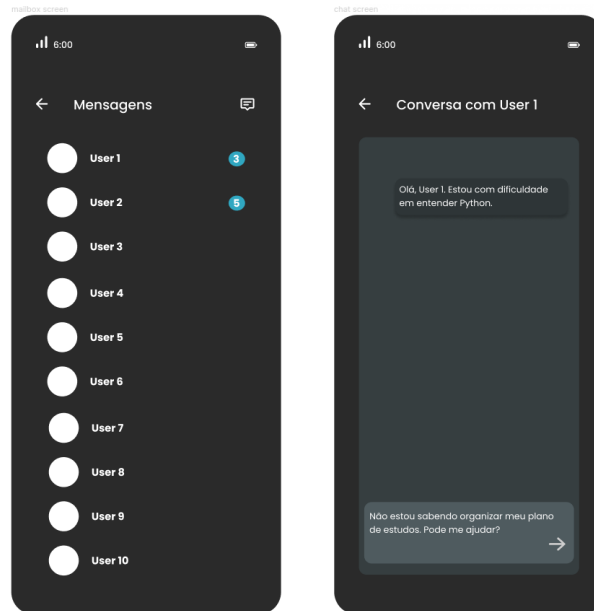


Figura 3.4: Protótipos de telas contendo as funções de envio e recebimento de mensagens no Algoritmia.

Gerenciamento de Conta

Rota composta por 3 telas mostradas na Figura 3.5, acessíveis pela barra de navegação, sendo uma para visualização dos dados de usuário, com opções para alteração dos dados e para fazer o *Logout*, enquanto as outras são para realizar a alteração efetiva dos dados pessoais e senha.

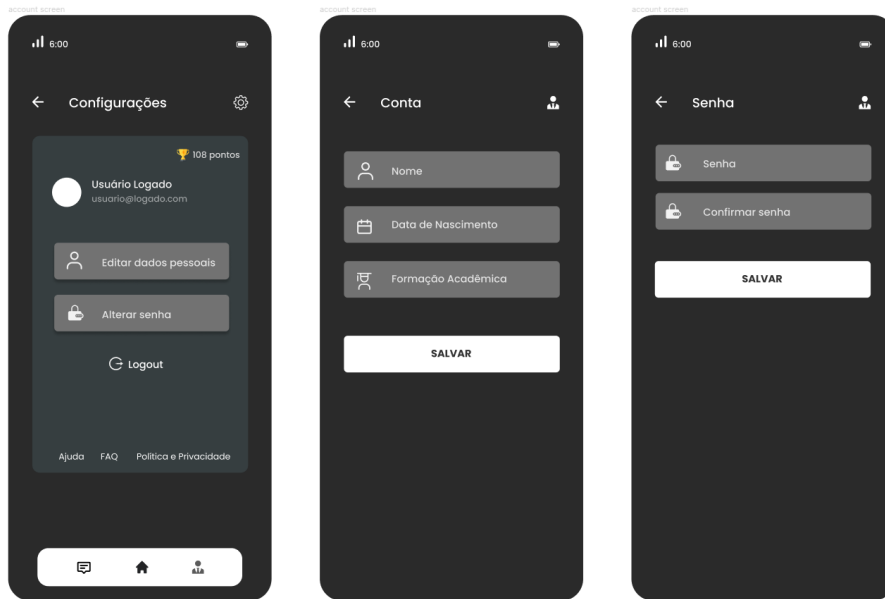


Figura 3.5: Protótipos de telas pertencentes às rotas de gerenciamento de conta.

3.1.2 Markdown

Markdown [24] é uma linguagem de marcação de texto leve para a criação de textos formatados utilizando um simples editor de texto. O formato tem a licença *open-source* BSD, que permite a redistribuição do código-fonte desde que mantidos os direitos autorais. As histórias de usuário da aplicação foram criadas utilizando a linguagem Markdown conforme o exemplo a seguir, que descreve a funcionalidade de realizar desafios no *Front Office* pela visão do usuário:

Como estudante, **eu quero** ser capaz de realizar os desafios existentes, **para** consolidar os meus conhecimentos e aumentar a minha colocação no *ranking*.

Critério de Aceitação 01:

1. Usuário logado acessa o time que contém o curso que deseja fazer;
2. Usuário acessa os tópicos do curso pela tela *Home*;
3. Usuário visualiza os subtópicos pertencentes ao tópico pai com seu progresso de conclusão dos desafios presentes a cada um;
4. Usuário acessa os desafios de um subtópico;
5. Usuário é levado para o primeiro desafio do tópico, onde terá acesso às funções de avançar para o próximo desafio ou voltar para o anterior;

6. Se o desafio ainda não estiver respondido, o usuário será capaz de responder, senão, o botão de responder será desabilitado e o usuário apenas poderá visualizar a sua resposta e qual é a resposta correta;
7. Ao pular ou responder o último desafio, o usuário será redirecionado para a tela de *feedback*;
8. Será exibida a quantidade de questões acertadas, com erro e não respondidas, assim como a quantidade de pontos obtida em cada desafio respondido;
9. Usuário será capaz de revisar as respostas ou retornar aos subtópicos por meio de botões presentes na extremidade inferior da tela.
10. Ao visualizar o *ranking* novamente, a sua pontuação terá sido alterada com base nos pontos ganhos mostrados no *feedback*, tendo a possibilidade de subir de posição.

As demais histórias de usuário podem ser acessadas por meio do repositório auxiliar³ criado para o projeto no GitHub (3.1.6).

Markdown também foi usado no aplicativo e aplicação *Web* para a criação e exibição dos desafios, de forma a permitir ao professor uma maior versatilidade. Para deixar a aplicação mais acessível a professores que não têm conhecimentos sobre a linguagem, foi usado um editor mais visual no *Back Office*, com as opções de formatação em botões e uma visualização do texto formatado ao lado.

3.1.3 *Lucid Chart*

O *Lucid Chart*⁴ é uma ferramenta *online* com plano gratuito para realizar o desenho de fluxos, podendo ser feito de forma colaborativa. Com as ferramentas disponibilizadas pela plataforma, foi criado o fluxograma da Figura 3.6 para o aplicativo móvel, que mostra as funcionalidades acessíveis pelo usuário final por cada tela. Também é mostrado o funcionamento básico dos desafios incluídos na plataforma, que fornecem uma pontuação ao usuário somente quando a pergunta é respondida corretamente, não incrementando os pontos quando o desafio é respondido incorretamente.

³https://pedenite.github.io/monografia-pages/user_stories.pdf

⁴<https://www.lucidchart.com/>

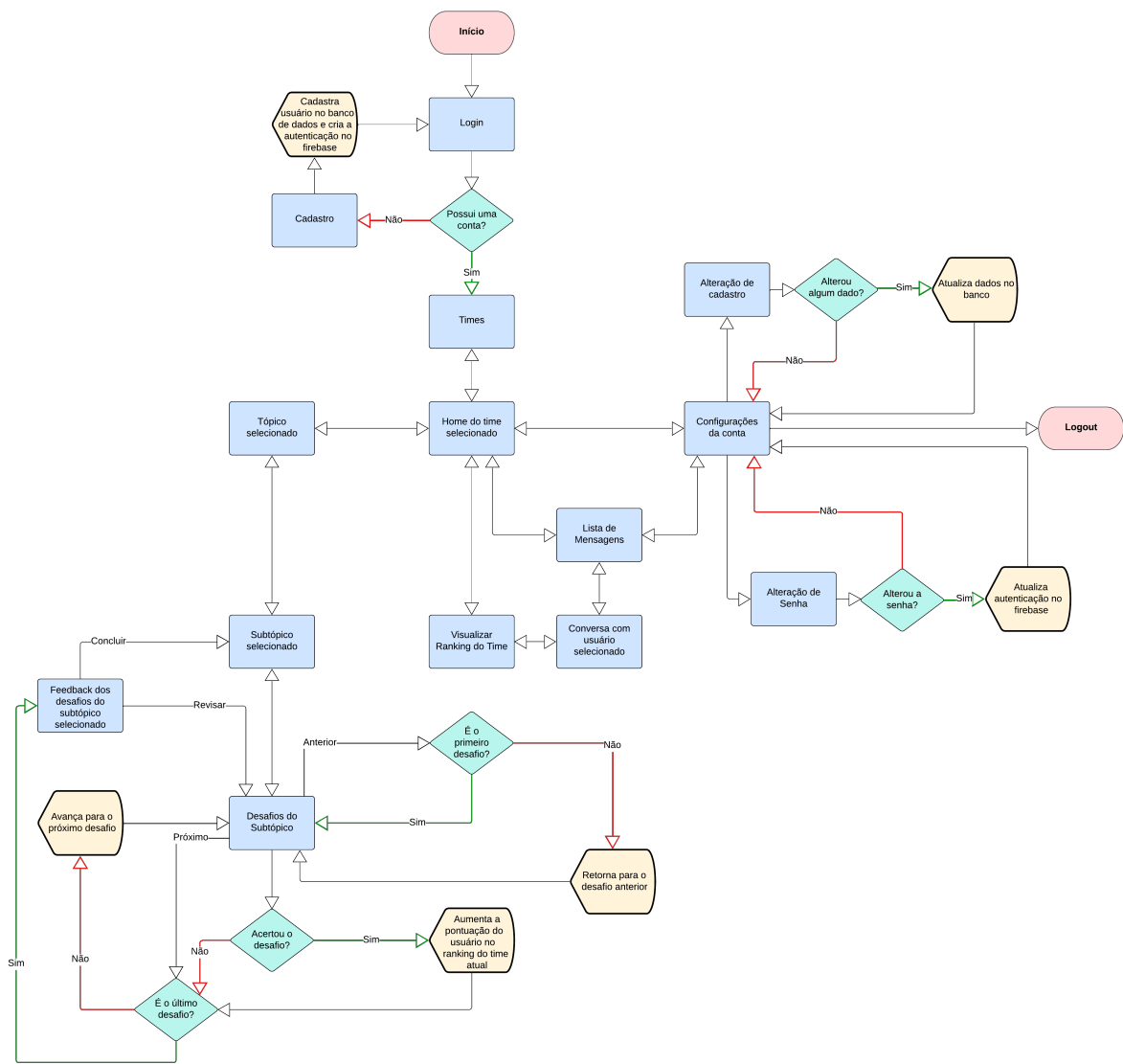


Figura 3.6: Fluxo de utilização do aplicativo móvel *Algoritmia* na visão do usuário.

Também foi desenvolvido um fluxograma para a aplicação *Web* seguindo o mesmo padrão do aplicativo. O fluxo está desenhado na Figura 3.7.

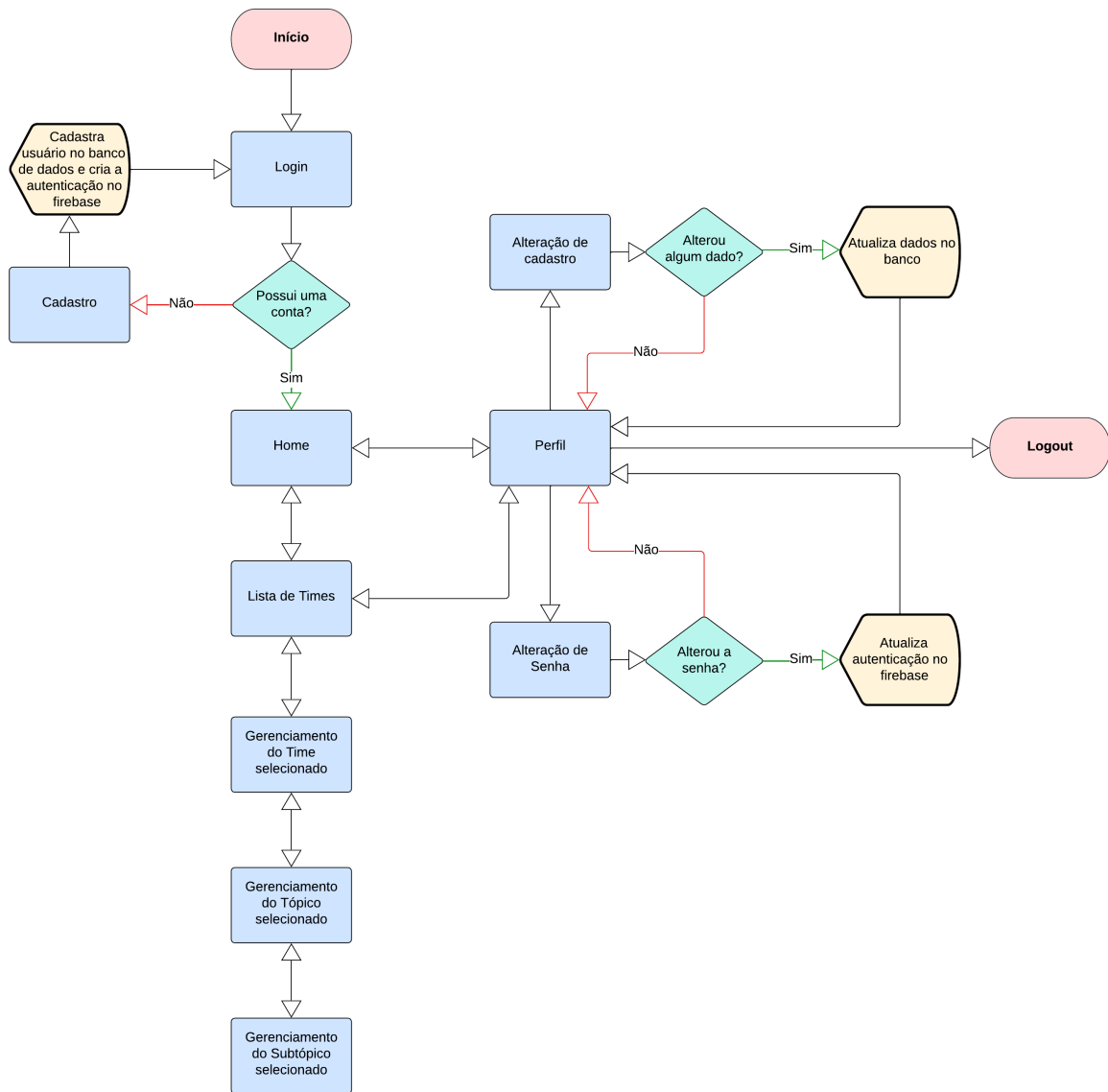


Figura 3.7: Fluxo de utilização da aplicação *Web Algoritmia* na visão do usuário.

Pela Figura 3.6 e Figura 3.7, é possível visualizar o início e o fim do fluxo de usabilidade do *Front Office* e *Back Office* em cor vermelha, sendo o início, o *login* e o fim, o *logout*. As telas foram representadas por retângulos de cor azul e as condições por losangos de cor azul claro. Por fim, funcionalidades específicas são representadas pelo item de cor amarelada.

3.1.4 PlantUML

Unified Modeling Language (UML) [25] é uma linguagem de modelagem usada no campo de engenharia de *software* que visa criar uma maneira padrão de visualizar o *design* de um sistema. É uma ferramenta *open-source* sob a licença *GNU General Public License*. Dentre os tipos de diagramas que fazem parte do UML, estão o diagrama de classe, de objeto, de sequência, de caso de uso, de atividade, entre muitos outros. Para o contexto do projeto, foi gerado o diagrama de classes seguindo o UML.

PlantUML ⁵ é uma ferramenta que permite escrever diagramas UML por meio de uma sintaxe textual. Ele também suporta outros diversos diagramas que não fazem parte do UML. A ferramenta foi usada para gerar o diagrama de classes da aplicação conforme Figura 3.8.

⁵<https://plantuml.com/>

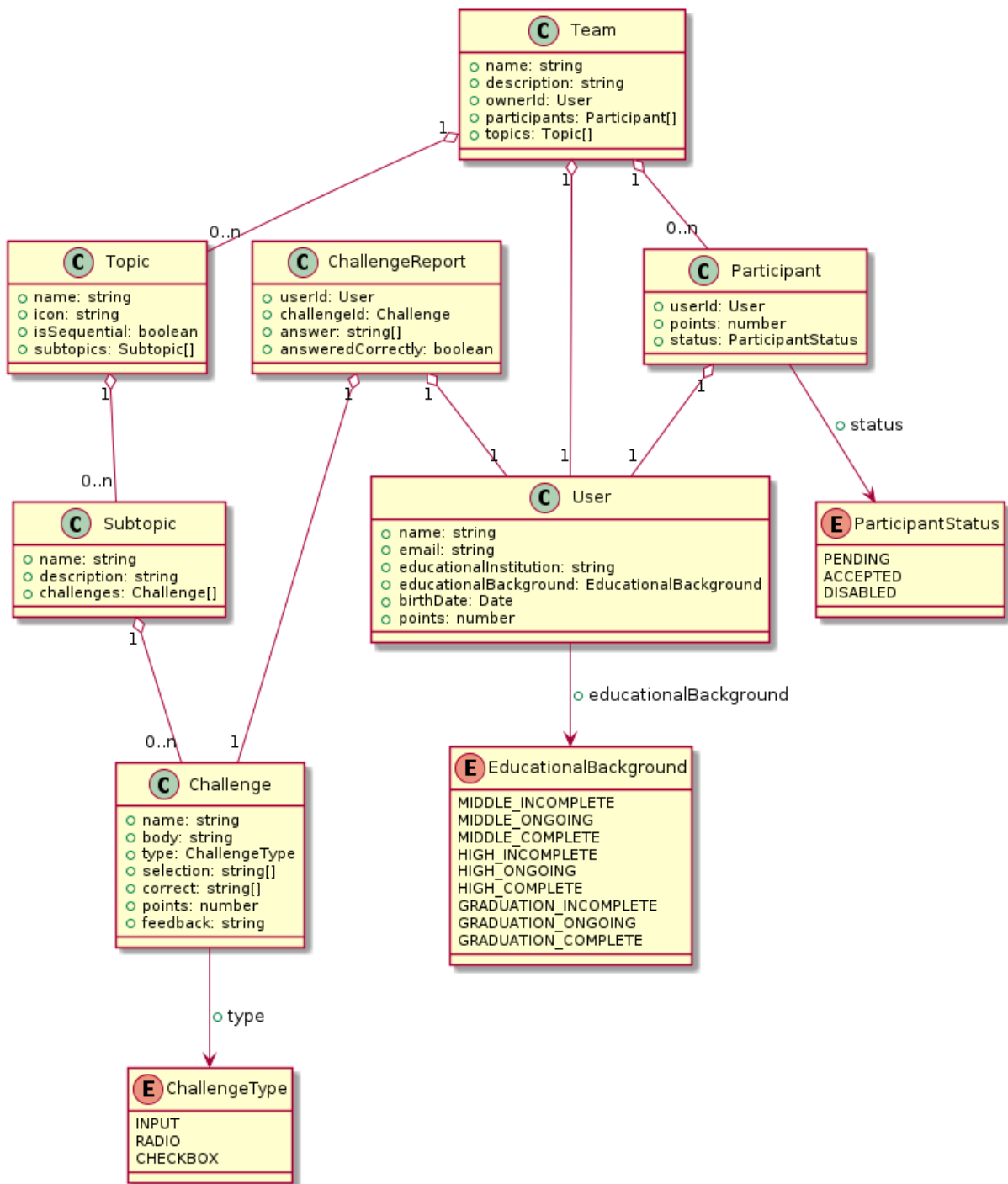


Figura 3.8: Diagrama de classes do Algoritmia.

Pelo diagrama de classes, é possível visualizar as classes utilizadas no projeto, tanto na versão *Front Office* quanto *Back Office*, que refletem diretamente na modelagem do banco de dados não-relacional utilizado. Devido à categoria de banco de dados utilizada (3.2.8), não foram gerados documentos adicionais para uma descrição mais visual da base.

3.1.5 Notion

O software *Notion*⁶ é uma plataforma de produtividade utilizada principalmente para organizar trabalhos em equipe. Apenas os recursos gratuitos da plataforma foram usados, atendendo todas as expectativas para o projeto. A ferramenta dispõe de diversos recursos relacionados a metodologias ágeis, documentação, métricas, entre outros, porém para o *Algoritmia*, foi utilizado o quadro kanban para a divisão e organização das tarefas do projeto, assim como para acompanhar as etapas do processo do Scrum.

3.1.6 Git e Git Flow

Git⁷ [26] é uma ferramenta de controle de versão particularmente poderosa e flexível que torna o desenvolvimento colaborativo mais eficiente. A ferramenta foi inventada por Linus Torvalds para suportar o desenvolvimento do *kernel* do Linux, mas se mostrou de grande valor para uma vasta gama de projetos. O Git é distribuído sob a licença *GNU General Public License version 2.0*, que garante a liberdade de compartilhamento e modificação para assegurar a gratuidade de uso por todos os usuários.

Durante todo o desenvolvimento do projeto, foi utilizada a ferramenta Git para o controle de versão do código fonte. Para o compartilhamento do código e diferentes versões entre a dupla, foi usado o *GitHub*⁸ como repositório remoto. A plataforma é de uso livre e funciona no formato *Web*, facilitando o uso do Git, assim como acrescentando funcionalidades de organização de projetos que ajudam muito nos processos de desenvolvimento. Foram criados quatro repositórios no total para o projeto no GitHub de forma a promover uma melhor separação de cada plataforma e permitir até permitir o reuso de recursos com o repositório auxiliar.

1. Front Office⁹;
2. Back Office¹⁰;
3. Monografia¹¹;
4. Auxiliar¹².

O primeiro repositório listado é o que armazena todo o código-fonte do aplicativo móvel, assim como o segundo, que armazena o código da aplicação *Web*. O terceiro item,

⁶<https://www.notion.so/>

⁷<https://git-scm.com/>

⁸<https://github.com/>

⁹<https://github.com/KesleyK/monografia-app>

¹⁰<https://github.com/KesleyK/monografia-backoffice/tree/master>

¹¹<https://github.com/Pedenite/Monografia>

¹²<https://github.com/Pedenite/monografia-pages>

intitulado Monografia contém todos os arquivos para gerar esta monografia e o último foi usado para incluir artefatos adicionais do projeto como páginas *Web* que incluem a política de privacidade, PDFs com as histórias de usuário, entre outros. O repositório auxiliar utiliza de uma funcionalidade do GitHub chamada *GitHub Pages* [27], que permite a publicação de páginas *Web* diretamente do repositório. Isso permitiu que as políticas de privacidade do aplicativo fossem publicadas como um *site*¹³, assim como os outros artefatos incluídos.

Em todos os repositórios criados, com exceção da Monografia, foi incluída a licença do MIT, garantindo a livre distribuição e modificação do código-fonte. O repositório da Monografia foi criado por meio de um *fork* do repositório de mesmo nome de autoria do UnB-CIC¹⁴, que contém o template para a criação da monografia para os cursos do departamento de computação da Universidade de Brasília.

Adicionalmente, para aprimorar o uso efetivo do Git, foi utilizado um fluxo de criação de *branches* intitulado *Git Flow*, que consiste de regras de nomenclatura e de hierarquia para as ramificações criadas em um projeto. Na Figura 3.9 é possível visualizar as nomenclaturas das *branches* utilizadas no projeto *Algoritmia*, assim como de onde são feitos os *checkouts* e *merges* para cada uma.

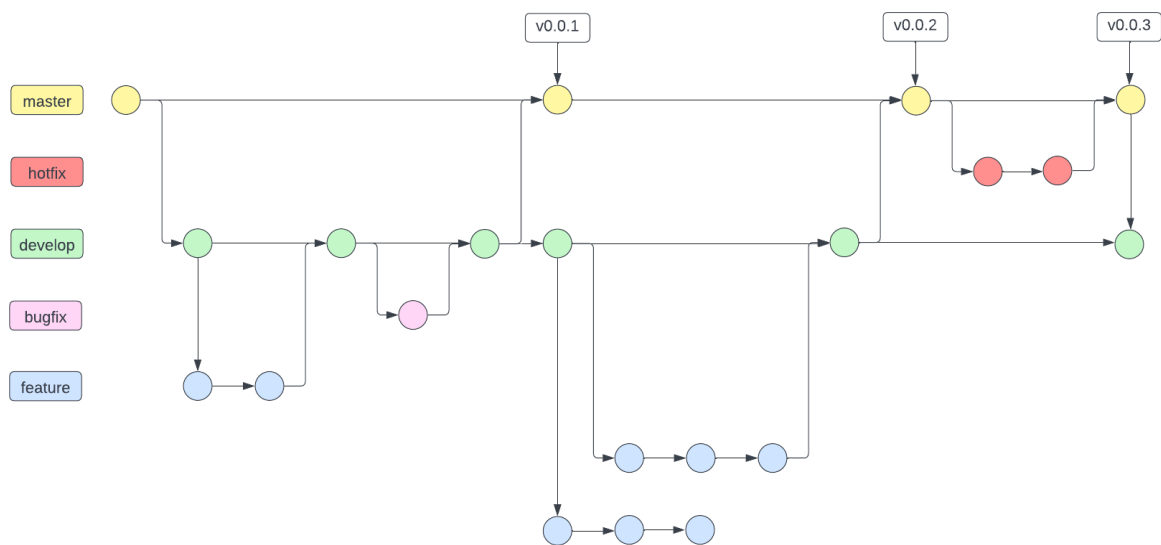


Figura 3.9: Exemplo visual do Git Flow.

- *Master*: é a principal e não deve ter *commits* feitos diretamente nela, sendo usada para as *releases* do projeto.

¹³<https://pedenite.github.io/monografia-pages/>

¹⁴<https://github.com/UnB-CIC/Monografia>

- *Hotfix*: é usada quando são descobertos erros na *Master*, para corrigir o mais rápido possível.
- *Develop*: vai acompanhar a *Master*, inclusive sendo gerada a partir dela no início do projeto, e a partir dela que são geradas as *branches* de *Feature* e *Bugfix*, caso necessário.
- *Bugfix*: usadas para corrigir problemas encontrados na *branch Develop* antes de realizar o *merge* com a *Master*, que pode representar um ambiente de produção, assim evitando problemas possivelmente catastróficos.
- *Feature*: representam os recursos incrementais do projeto como novas funcionalidades ou melhorias. Foi o tipo de *branch* mais usado no projeto, sendo criada uma *branch* para cada *card* que não represente um defeito no Notion.

O uso do *Git Flow* permitiu uma organização melhor do projeto para a realização do desenvolvimento de diversas funcionalidades simultaneamente sem interferência de uma nas outras. Com o uso dos *Pull requests* do Github, foi possível aprimorar ainda mais o processo de *merge* com a *branch Develop* após o desenvolvimento de funcionalidades incluindo o *Code Review* para a aprovação do item, permitindo uma melhor integração da dupla e conhecimento sobre o *status* do projeto.

3.2 Tecnologias Utilizadas

Para o desenvolvimento do projeto de *software*, diversas tecnologias foram utilizadas, incluindo linguagens de programação, *frameworks*, ferramentas e ambientes. Nas seções seguintes, estão descritas cada uma delas.

3.2.1 Android

Android¹⁵ é o sistema operacional para *smartphones* mais utilizado na atualidade, possuindo uma quota de mercado de 71,96% segundo dados de novembro de 2022 [28]. Trata-se de um sistema *open-source*, distribuído, em maior parte, sob a licença *Apache 2.0*, mas por ser baseado em Linux, tem o seu *kernel* distribuído sob a licença *GPLv2*. Android é composto por uma pilha criada para suportar uma vasta quantidade de dispositivos e aspectos dos mesmos [29]. A hierarquia do sistema pode ser vista da seguinte forma:

- **Aplicativos do sistema**: Todos os aplicativos que interagem diretamente com o usuário como calculadora, câmera, *e-mail*, entre outros;

¹⁵<https://www.android.com/>

- **Java API Framework:** Provedores de conteúdo, sistema de *views* e gerenciadores responsáveis por manter os aplicativos funcionando para o usuário final;
- **Bibliotecas C/C++ nativas e Runtime do Android:** Bibliotecas e funcionalidades requeridas pelos componentes e serviços do Android como o OpenGL e *Android Core Libraries*;
- **Hardware Abstract Layer (HAL):** Camada de abstração de *hardware*, é responsável por garantir o correto funcionamento das funcionalidades disponibilizadas pelo Android por meio de qualquer equipamento pertencente ao aparelho como câmera, microfone, saída de áudio, *bluetooth*, entre outros.
- **Kernel do Linux:** A fundação de todo o sistema Android, fornece funcionalidades de segurança, concorrência, gerenciamento de memória, entre muitos outros recursos de baixo nível para todas as camadas acima. Por ser um produto que já está no mercado a algumas décadas, também facilita para os fabricantes desenvolverem aparelhos para um *Kernel* mais conhecido.

Todo o código do aplicativo **Algoritmia** foi testado no sistema Android, porém, devido a utilização do React Native (3.2.3) no código-fonte, o aplicativo é capaz de executar em dispositivos iOS também. Porém, devido à falta de testes para a plataforma do iOS, não há garantias do correto funcionamento de todas as funcionalidades, ao contrário da versão para sistemas Android.

3.2.2 Typescript

Typescript¹⁶ é uma linguagem de programação *open-source* sob a licença *Apache 2.0*, fortemente tipada que foi desenvolvida e é mantida pela Microsoft, sendo construída em cima da linguagem Javascript, mas permitindo um melhor uso ferramental a qualquer escala [30]. Por sua vez, Javascript [31] é uma linguagem leve, interpretada ou compilada *just-in-time* com funções de primeira classe. É uma linguagem multi-paradigma que é usada vastamente em páginas da *Web*, mas também tem ganhado uma grande popularidade em sistemas *back-end* e *Mobile* com tecnologias como Node.js e React Native.

A escolha da linguagem se deu por razões como a facilidade do uso e devido às opções que o typescript proporciona em relação à tipagem, que têm a capacidade de aumentar a produtividade ao se utilizar editores de texto avançados, assim como diminuir a quantidade de erros em tempo de execução.

¹⁶<https://www.typescriptlang.org/>

3.2.3 React e React Native

React¹⁷ é uma biblioteca Javascript para realizar a criação de interfaces de usuário. Trata-se de um projeto *open-source*, sob a licença do MIT, criado e mantido pela empresa Meta. A ideia geral da biblioteca é ser declarativa, baseada em componentes e multi-plataforma, permitindo a criação de um código modularizado mais simples de debugar, e que pode ser convertido para outras plataformas de forma relativamente fácil [32].

A sintaxe do React utiliza a extensão de sintaxe do Javascript chamada JSX, que se assemelha muito com a sintaxe de linguagens de marcação de texto como HTML e XML, porém traduzindo a sintaxe para códigos Javascript. Por este motivo, a biblioteca pode ser aprendida com uma certa facilidade por desenvolvedores que já estejam familiarizados com o desenvolvimento *Web* nativo usando HTML.

React Native¹⁸ compartilha dos mesmos princípios, funcionalidades e sintaxe do React para o desenvolvimento de aplicativos nativos. Também é um *framework* distribuído sob a licença do MIT, garantindo que seu código seja considerado aberto. Os componentes presentes no código escrito em react native são transformados em elementos de interfaces de usuário utilizando a *API* da própria plataforma onde será executado. O *framework* facilita o desenvolvimento para plataformas móveis de forma a utilizar o mesmo código tanto no Android como no iOS, não exigindo muito retrabalho ao criar suporte a outras plataformas para o aplicativo. Assim como React, o React Native é uma biblioteca *open-source* mantida pela Meta [33].

A escolha do React e do React Native para o projeto **Algoritmia** foi devido à familiaridade com elas por parte da dupla, não exigindo um treinamento extensivo para seu uso no projeto. A aplicação móvel, chamada de *Front Office* foi desenvolvida usando React Native, enquanto a versão *Web*, *Back Office*, foi desenvolvida utilizando React.

3.2.4 Material UI

Material UI [23] é uma biblioteca que disponibiliza elementos gráficos para páginas *Web* em React. Os elementos de *UI* foram utilizados extensivamente por todas as telas da aplicação. Os principais motivos da escolha da biblioteca foi a facilidade de uso, o suporte para o Typescript e React e os *designs* agradáveis dos componentes.

¹⁷<https://reactjs.org/>

¹⁸<https://reactnative.dev>

3.2.5 Expo

Expo¹⁹ é uma plataforma *open-source*, distribuída sob a licença BSD para a criação de aplicativos nativos para Android, iOS e *Web* com Javascript e React [34]. A plataforma foi usada na versão *mobile* do *Algoritmia* para facilitar o uso de tecnologias nativas do Android, mas também facilitaria para o iOS, porém este não foi testado e, portanto não existem garantias de funcionamento.

Com o uso do Expo, foi também instalado um aplicativo chamado *Expo Go*²⁰ para facilitar o teste durante o desenvolvimento. O aplicativo em questão permite a execução de aplicativos que utilizam a plataforma expo sem a necessidade de instalá-los no dispositivo, bastando acessar um *link* disponibilizado pela ferramenta ou realizar a leitura de um *QR Code* com a câmera do dispositivo, permitindo também a distribuição do *app* por este meio. O aplicativo do *Algoritmia* foi disponibilizado para execução por meio do aplicativo *Expo Go* por meio de uma *URL* gerada²¹, permitindo a distribuição para testes.

3.2.6 Jest

Softwares devem ser testados para transmitir a confiabilidade, ou seja, garantir o correto funcionamento nos ambientes intencionados [35]. A automação dos testes pode reduzir significativamente o esforço necessário para a realização adequada dos testes, assim como aumentar a quantidade de testes realizados durante um espaço de tempo limitado.

Jest²² é um *framework* de testes em javascript que tem como foco, a simplicidade [36]. É uma ferramenta que tem uma boa documentação e apresenta suporte para projetos que utilizam Typescript e React, dentre vários outros. O Jest é mantido pela Meta sob a licença do MIT, fazendo-o *open-source*, logo, o seu uso é liberado gratuitamente, assim como o seu código fonte.

A escolha do Jest como ferramenta de testes se deu principalmente pelo suporte nativo do Typescript e do React, que são tecnologias que foram utilizadas no desenvolvimento do *Algoritmia*. O projeto apresenta testes unitários, de integração e features. O uso de testes automatizados facilitou as mudanças no código no decorrer do projeto, de forma a garantir o funcionamento dos códigos mais antigos após incrementos ou correções sem muito esforço adicional.

¹⁹<https://expo.dev/>

²⁰<https://play.google.com/store/apps/details?id=host.exp.exponent>

²¹<https://expo.dev/@pedenite/algoritmia?serviceType=classic&distribution=expo-go>

²²<https://jestjs.io/pt-BR/>

3.2.7 Visual Studio Code

Visual Studio Code²³ é um editor de código fonte leve, mas com recursos poderosos, disponível gratuitamente para os sistemas operacionais Windows, Linux e macOS. A ferramenta *open-source* com a licença do MIT tem suporte *built-in* para JavaScript, TypeScript e Node.js e possui um ecossistema rico de extensões para outras linguagens e ambientes de execução [37].

A escolha do editor foi feita por preferência pessoal dos autores do *Algoritmia*, sendo os principais fatores decisivos, a familiaridade com a ferramenta, o suporte nativo para Typescript e React e a gratuidade. O Visual Studio Code também inclui o *Intellisense*, que é uma ferramenta que provê *auto-complete* inteligente para o código, baseando-se em análises semânticas da linguagem usada e na análise do código-fonte escrito [38]. O Visual Studio Code juntamente com o *Intellisense* permitiu uma maior produtividade durante todo o processo de desenvolvimento do *Algoritmia*.

3.2.8 Firebase

Firebase²⁴ é uma plataforma de desenvolvimento de aplicações que auxilia na criação e no crescimento delas [39]. A plataforma pertence ao Google e é gratuita de forma limitada, sendo necessário aderir a um plano pago para continuar utilizando após atingir uma quota. Dentre os serviços oferecidos, podem ser encontrados serviços de autenticação e gerenciamento de usuários e banco de dados. A plataforma adere à categoria de serviço *low-code*, onde o programador tem acesso a um sistema de *back-end* utilizando poucas linhas de código, sendo necessário apenas criar a aplicação que irá consumir os serviços.

A plataforma *Algoritmia* utiliza os serviços de autenticação, banco de dados e *hosting* providos pelo Firebase. O banco de dados NoSQL, denominado Firestore[40], funciona em nuvem de forma flexível e escalável para armazenar e sincronizar dados para o desenvolvimento do lado do cliente e do servidor Firestore. Por se tratar de um banco de dados NoSQL, não foram produzidos artefatos adicionais de documentação para demonstrar a forma como os dados estão estruturados e como são relacionados, pois o diagrama de classes na Figura 3.8 já contém todas as informações necessárias para a modelagem do banco. O serviço de *hosting* foi utilizado para que a aplicação *Web*, o *Back Office*²⁵, fosse disponibilizada amplamente através da *internet*.

A escolha do Firebase se deu pela flexibilidade e as boas condições do plano gratuito, que atenderam todas as necessidades do projeto.

²³<https://code.visualstudio.com/>

²⁴<https://firebase.google.com/>

²⁵<https://monografia-app.firebaseio.com/>

Capítulo 4

O Aplicativo Algoritmia

Neste capítulo, serão descritos o aplicativo móvel e a aplicação *Web Algoritmia*, assim como todas as suas funcionalidades, telas e aplicabilidade.

O aplicativo móvel foi desenvolvido com o intuito de ensinar computação para qualquer pessoa interessada de forma mais descontraída e gamificada, enquanto a versão *Web* visa disponibilizar as ferramentas necessárias para o professor criar um curso com estas características. O projeto foi disponibilizado no GitHub com a licença do MIT [41], que é uma licença simples e curta com a única condição sendo de preservação do *copyright* e notas da licença. A licença em si é *open-source* e garante a qualquer pessoa, o uso comercial, distribuição, modificação e uso privado do projeto, porém, não provê garantia de forma alguma.

O *Algoritmia* foi dividido em dois módulos, denominados *Back Office* e *Front Office*. O *Front Office* foi o módulo desenvolvido para dispositivos móveis Android e é o elemento principal da aplicação, representado o local onde o aluno irá realizar os desafios, consultar sua colocação, interagir com outros alunos, entre outros. Já o *Back Office* é uma versão *Web* da aplicação cujo público alvo são professores que queiram utilizar da plataforma para ensinar os seus alunos. Os professores serão capazes de criar cursos, que são compostos por tópicos, que por si, podem possuir diversos desafios, cada um valendo uma quantidade determinável de pontos que os alunos poderão obter. Por se tratar de um aplicativo em que o conteúdo pode ser adicionado a qualquer momento por um professor que se cadastre na plataforma *Web*, os conhecimentos prévios para o uso não podem ser determinados como um todo, mas sim por curso, a ser determinado pelo professor em questão.

O aplicativo conta com um curso já pré-cadastrado que pode ser acessado pela tela inicial por meio da denominação "Plataforma global". Foram criados exercícios de Python para iniciantes. Para a criação dos exercícios, foi usado como base, o formato dos exercícios sobre a linguagem de programação Python, presentes na plataforma W3schools¹, porém

¹<https://www.w3schools.com/>

com diversas adaptações e modificações para adequar à proposta do aplicativo, incluindo uma história de autoria da dupla para tentar gerar mais engajamento por parte dos alunos [1].

As próximas seções mostram as telas e funcionalidades do **Algoritmia** módulo *Front Office* e *Back Office* em mais detalhes.

4.1 Front Office

Versão *mobile* do **Algoritmia**, possui as seguintes características:

- **Público-alvo:** Estudantes de computação que estejam interessados em aprender de forma gamificada ou alunos de professores que utilizam a plataforma como elemento de aprendizagem de seus respectivos cursos;
- **Plataforma:** Dispositivos móveis, principalmente *smartphones* Android;
- **Objetivo geral:** Aprendizagem gamificada de computação.

A seguir, estão listadas as diferentes telas e funcionalidades do *Front Office* do **Algoritmia**.

4.1.1 Login e Registro

Ao acessar o aplicativo pela primeira vez, será requisitado que o usuário realize o *Login* na plataforma, conforme mostrado na primeira tela da Figura 4.1.

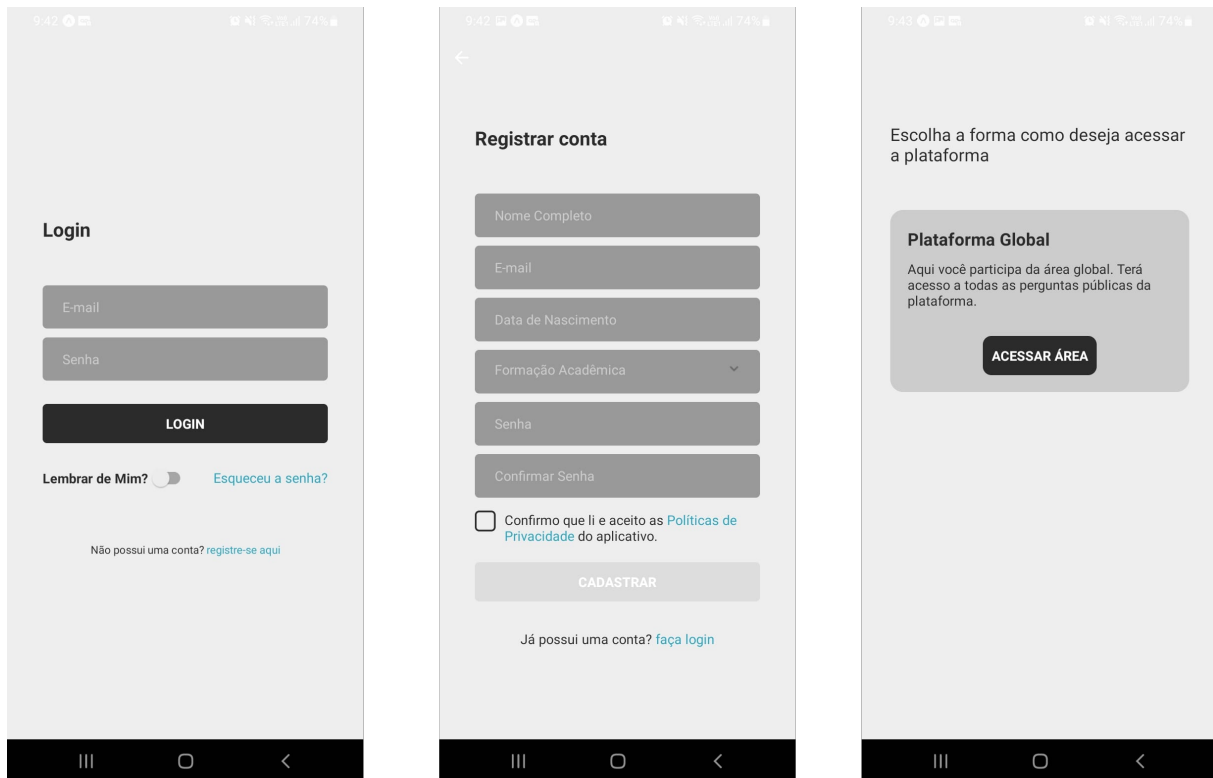


Figura 4.1: Telas de *Login*, Registro e Pré-Home - *Front Office*.

A tela possui os campos de *E-mail* e *Senha* para que seja possível a validação dos dados para realizar o *login* do usuário ou exibir uma mensagem de erro caso os dados estejam incorretos. Abaixo do botão de *login*, existe um seletor para lembrar do usuário que tem a função de diminuir a incidência de *logins* e também possui um elemento clicável para recuperar a senha de usuário pelo e-mail cadastrado. Por fim, no final da página, encontra-se o botão que leva à tela de registro de usuário. Ao clicar no botão "registre-se aqui", o usuário será levado à tela de registro.

Na tela de Registro, o usuário será capaz de se cadastrar, bastando preencher os dados de nome completo, *e-mail*, data de nascimento, formação acadêmica e senha, sendo necessária confirmação. O usuário somente poderá se cadastrar se aceitar as políticas de privacidade, que podem ser acessadas pelo *link* presente na *checkbox* no final do formulário. Esta medida serve para deixar o *software* em conformidade com a Lei Geral da Proteção de Dados (LGPD) [42]. O modelo utilizado para a criação da política de privacidade foi disponibilizado gratuitamente na plataforma Jusbrasil pelo usuário Pensador Jurídico [43].

Com a conta criada e validada corretamente, o aplicativo vai redirecionar para a tela de *login*, onde o usuário deverá digitar as informações de *login* e senha para finalmente acessar as funcionalidades do aplicativo, iniciando pela tela Pré-Home. Nesta tela, o

estudante poderá escolher a plataforma que irá acessar. Todos os usuários terão acesso à plataforma global, mas os professores podem cadastrar novos times no *Algoritmia* por meio do *Back Office* e matricular estudantes. À medida que o estudante é matriculado nos times, mais opções aparecem na *Pré-Home* abaixo da plataforma global para o acesso.

4.1.2 *Home* e Navegação

Ao acessar o time selecionado, o usuário é redirecionado para a tela *Home* do curso em específico. Nesta tela, existe uma barra de navegação que permite o acesso às telas de *Home*, *Conta* e *Mensagens*.

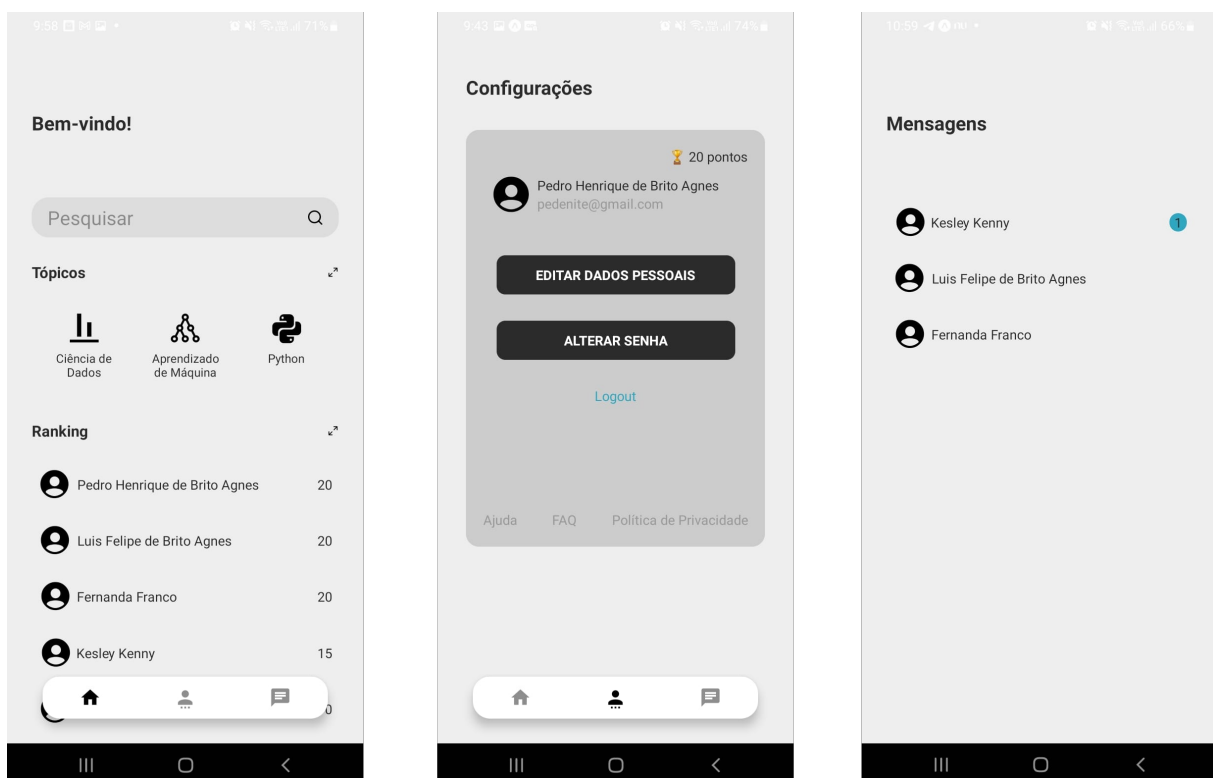


Figura 4.2: Telas acessíveis pela barra de navegação: *Home*, *Conta* e *Mensagens* - *Front Office*.

Pela tela *Home*, é possível visualizar uma barra de pesquisa para filtrar os tópicos exibidos do curso logo abaixo e o *ranking* de usuários, que é ligado aos pontos obtidos, em conformidade com os elementos de *design* de jogos. Pressionando o subtítulo "Tópicos", é possível acessar uma tela que exhibe todos os tópicos. O mesmo vale para a seção de *ranking*.

O segundo ícone da barra de navegação leva à tela de *Conta*, onde o usuário pode visualizar e editar as suas informações pessoais, assim como fazer o *logout*. Ao final da

seção em destaque da tela, também existem *links* que levam para *sites* que contêm a política de privacidade, perguntas frequentes e ajuda da aplicação.

Já o terceiro ícone, representa a tela de mensagens. Nela, é possível visualizar todas as conversas que foram iniciadas com os diferentes usuários da plataforma. Esta etapa está em conformidade com o elemento de *design* de jogos que se refere aos colegas, onde um aluno pode solicitar ajuda a outros, ou até oferecer essa ajuda ao ver posições no *ranking*. Se houverem mensagens não lidas, um ícone azul com a quantidade de mensagens não lidas irá aparecer ao lado direito do nome do usuário. Selecionando um usuário listado, é possível continuar a conversa com o mesmo por meio da ferramenta de *Chat*.

4.1.3 *Ranking* e Pontuação

Conforme mostrado na Seção 4.1.2, a tela *Home* exibe o *ranking* do time, porém limitado a dez registros que são ordenados pelos pontos obtidos. Ao pressionar o subtítulo *Ranking*, o usuário é levado à página dedicada ao *ranking* de usuário, onde é possível visualizar todos os participantes do time e suas respectivas posições.

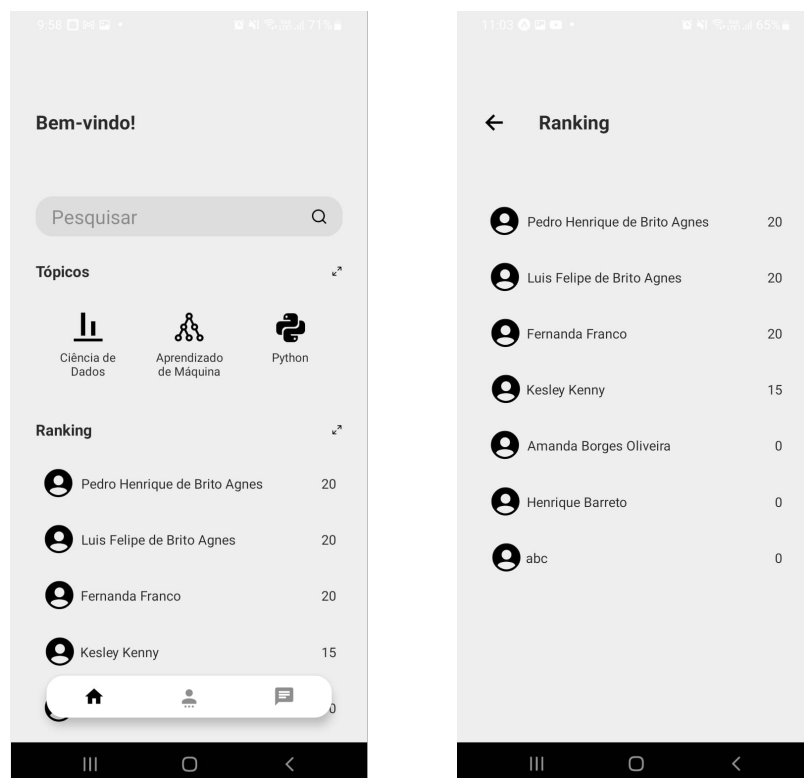


Figura 4.3: Telas da funcionalidade de *Ranking*: *Home* e *Ranking* - *Front Office*.

Os estudantes podem obter pontos por meio dos desafios do curso selecionado, que são acessíveis pelos tópicos (4.1.6). Ao acertar um desafio, o usuário ganha uma quantidade

determinada de pontos, que é definida pelo professor que cadastrou o desafio. A quantidade de pontos que podem ser configurados para um desafio é limitada no *Back Office* (4.2).

4.1.4 Dados de Usuário

Os usuários do aplicativo são capazes de alterar os seus dados de usuário por meio da tela de Conta, pelo botão "Editar dados pessoais". Para alterar a senha, existe o botão "Alterar Senha".

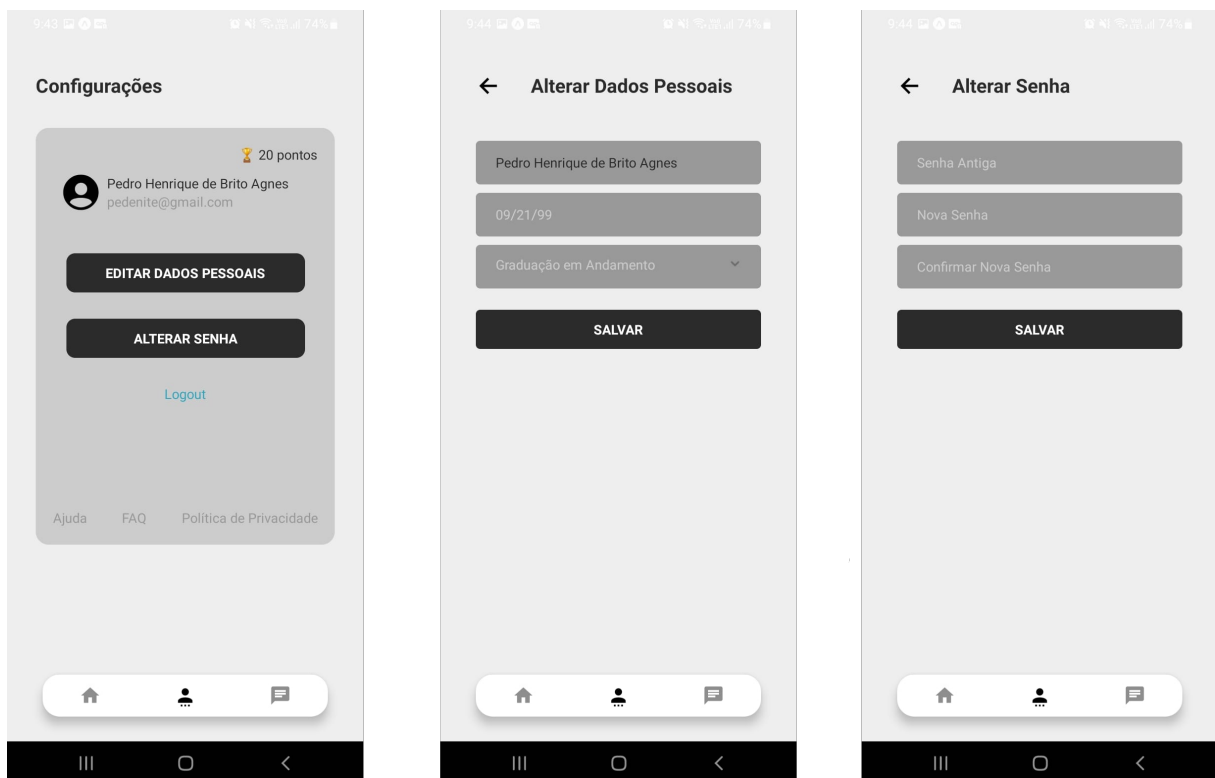


Figura 4.4: Telas para gerenciamento de conta - *Front Office*.

Nas telas de alterar dados e senha, o usuário tem a opção de efetivar a alteração realizada, assim como a opção de descartá-las por meio do botão de voltar no canto superior esquerdo. Todas as opções disponíveis e os dados coletados estão descritos nas políticas de privacidade do aplicativo, assim como é descrita a incapacidade do usuário de deletar a sua conta diretamente ou alterar o endereço de *e-mail*. Porém, nos *links* disponíveis ao final da página, está descrito como proceder para a exclusão da conta e alteração de *e-mail*, que seria por meio do contato por *e-mail* com os responsáveis pelo aplicativo.

4.1.5 Conversa

A funcionalidade de *Chat* da plataforma serve para intermediar a conversa por mensagens entre dois usuários. Para iniciar uma conversa com uma pessoa, basta encontrar e selecionar a pessoa na tela de *ranking* (4.3). Desta forma a conversa é criada e a tela de Conversa com esta pessoa é aberta. A partir do momento em que se iniciou uma conversa com a pessoa, seu nome irá aparecer na tela de conversas para mais fácil acesso ao histórico de mensagens.

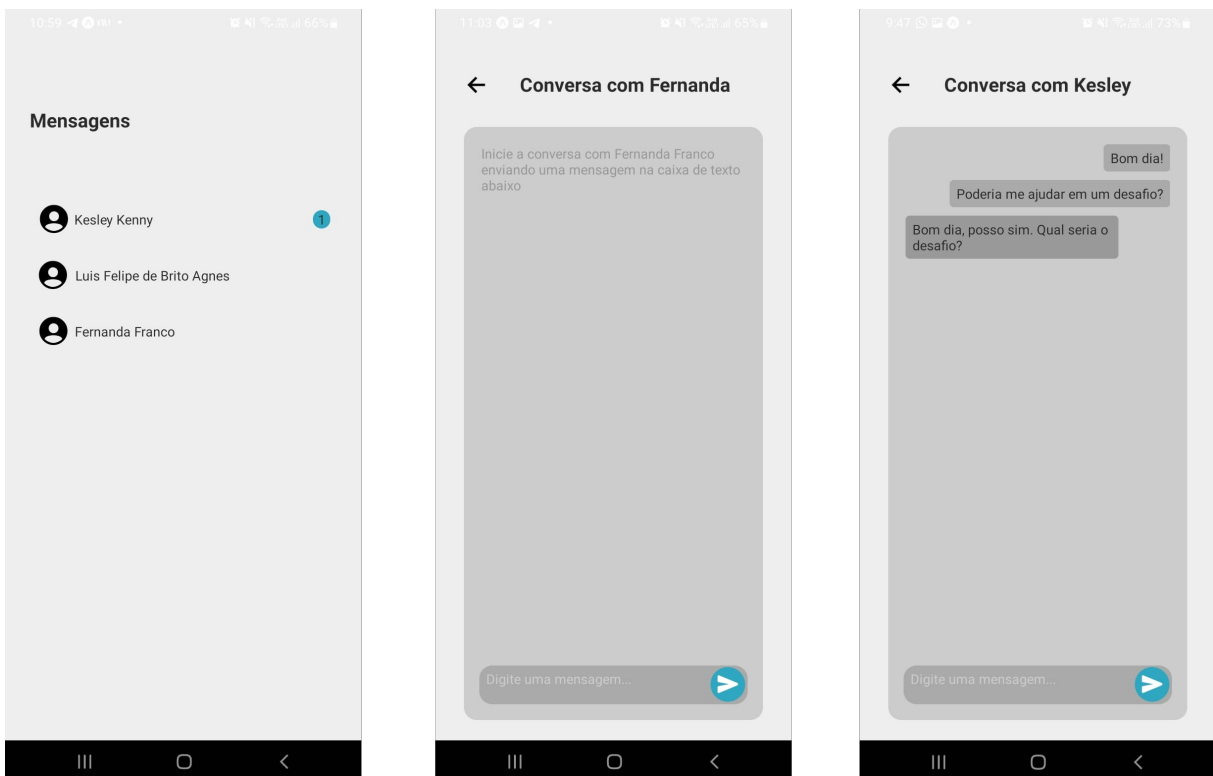


Figura 4.5: Telas para a funcionalidade de *Chat - Front Office*.

A segunda das telas listadas sobre a funcionalidade de *Chat* mostra uma conversa recém-iniciada, sem nenhuma mensagem. Já a última tela mostrada, inclui uma mensagem de cada uma das partes, demonstrando o *layout* e funcionamento das conversas no aplicativo.

4.1.6 Tópicos e Desafios

A funcionalidade de desafios pode ser acessada por meio dos tópicos cadastrados em um time. Cada tópico possui seus próprios conteúdos que podem ser acessados para que o usuário resolva os desafios.

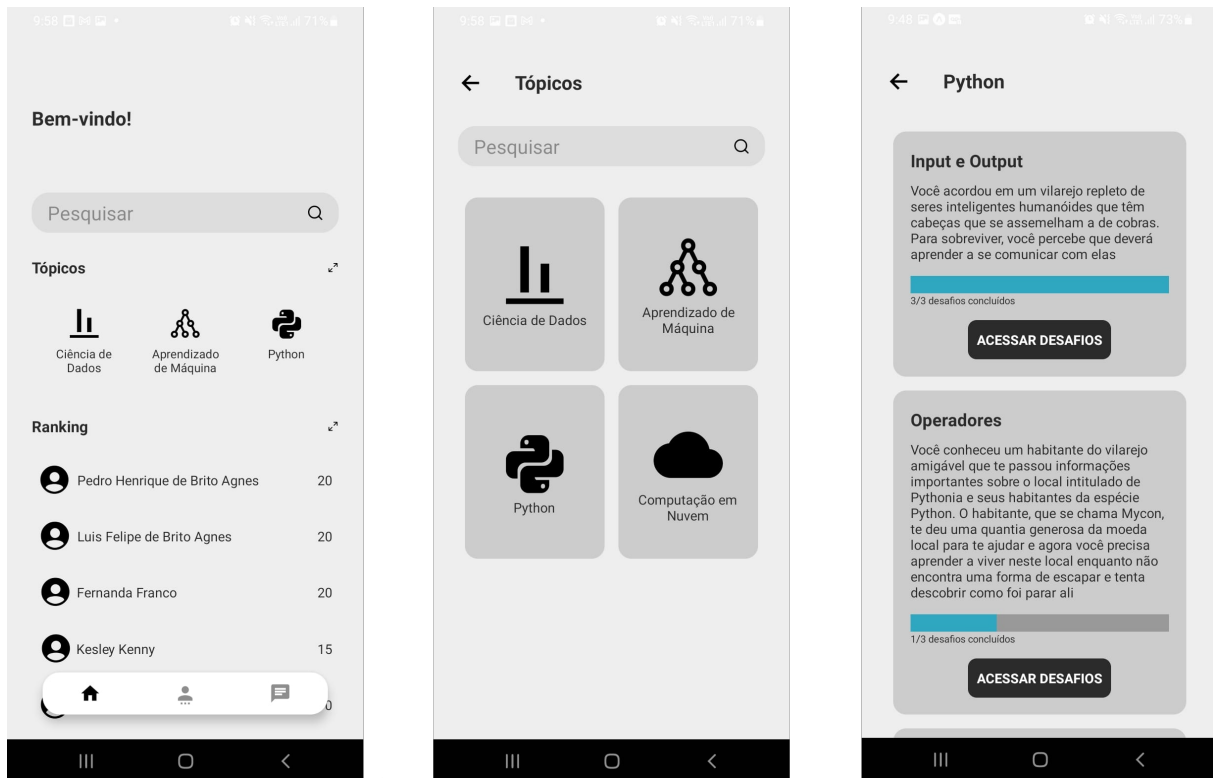


Figura 4.6: Telas relacionadas à funcionalidade de Tópicos - *Front Office*.

Pela tela *Home*, ao pressionar o subtítulo "Tópicos", o usuário será redirecionado para uma tela que mostrará todos os tópicos cadastrados do time, permitindo a filtragem por meio da barra de pesquisa ao topo. Ao acessar um tópico pela tela *Home* ou pela tela de Tópicos, o aluno poderá encontrar subtópicos em uma tela dedicada ao tópico selecionado. Cada um dos subtópicos possuem uma quantidade de desafios definida pelo professor, que podem ser acessados por meio do botão presente em cada item e o progresso de resolução é mostrado na barra de progresso acima do botão. Uma vez que o estudante resolve acessar os desafios, ele será direcionado para a tela de desafio, onde deverá responder a primeira questão do subtópico ou passar para o próximo e deixar o problema para depois.

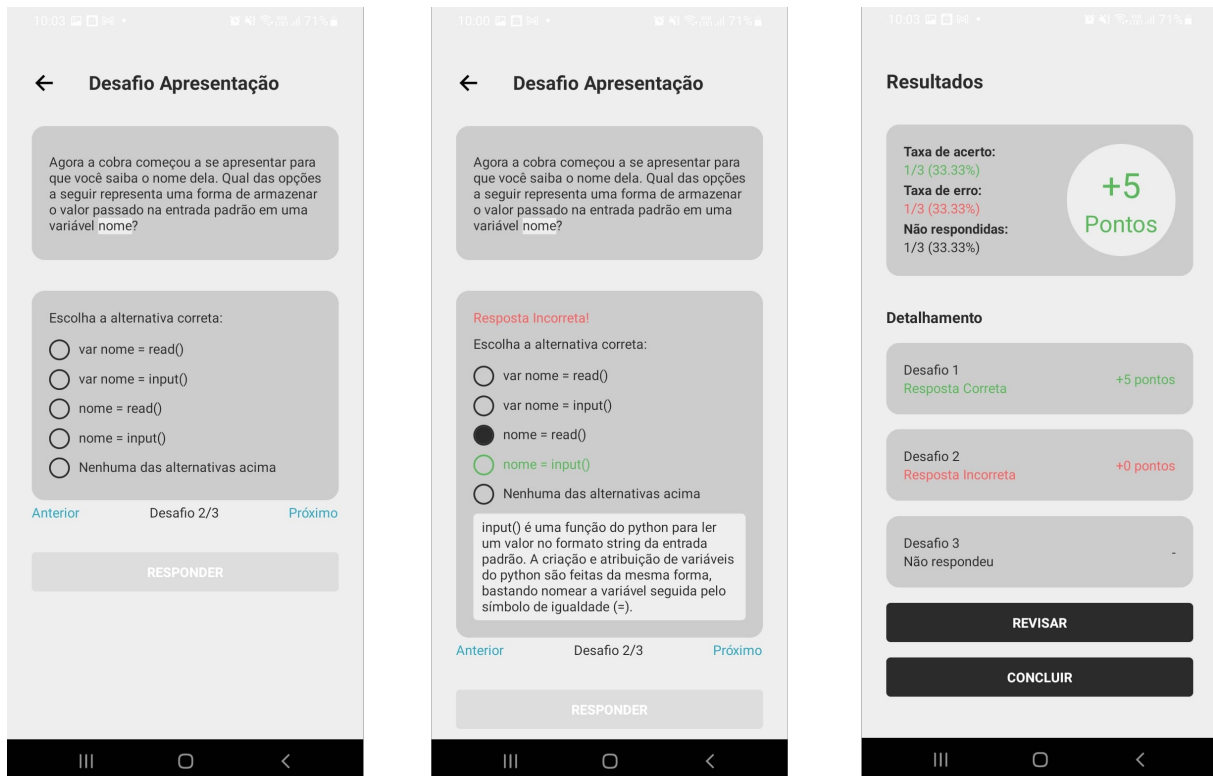


Figura 4.7: Telas de Desafio e *Feedback* - *Front Office*.

O estudante poderá navegar entre os desafios do subtópico para resolver na ordem que achar pertinente e, ao responder ou pular a última questão, ele será direcionado para o *Feedback* deste desafio, onde verá um resumo do seu desempenho, demonstrando o elemento de *design* de jogos que diz respeito à *performance*, que permite ao aluno acompanhar a sua evolução pessoal ao fazer comparações de seu desempenho em relação a desafios anteriores. Nos resultados, o estudante será capaz de visualizar a sua taxa de acerto e erro e a quantidade de questões em branco, que ele poderá resolver depois. Também serão mostrados dados de pontuação obtida em cada desafio abaixo do resumo geral, que contém a quantidade total de pontos obtida.

A qualquer momento, o usuário é capaz de acessar os desafios já resolvidos e visualizar a resposta que marcou e qual seria a resposta correta, porém, não será possível alterar as respostas uma vez que elas foram submetidas. Assim que o desafio é finalizado, o *ranking* é atualizado com a nova pontuação do usuário, podendo alterar as colocações dos estudantes, sendo capaz de promover a competição entre os colegas, o que faz parte da gamificação [13].

4.2 Back Office

Versão *Web* do Algoritmia, possui as seguintes características:

- **Público-alvo:** Professores que optarem por utilizar o Algoritmia como recurso tecnológico em suas práticas docentes;
- **Plataforma:** Navegadores de Internet;
- **Objetivo geral:** Cadastrar e gerenciar equipes, alunos, desafios e cursos na plataforma Algoritmia.

A seguir, estão listadas as diferentes telas e funcionalidades do *Back Office* do Algoritmia.

4.2.1 Login e Registro

Ao acessar a aplicação *Web*, o usuário se deparará com a tela de *Login*, componente fundamental para certificar que o usuário é um membro válido da plataforma.

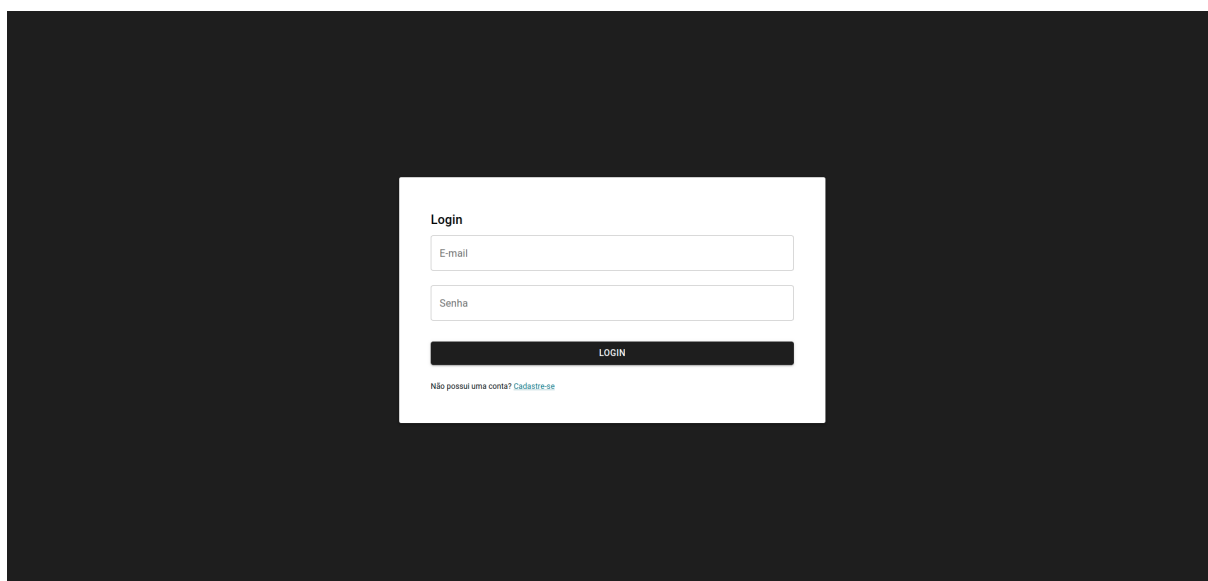


Figura 4.8: Tela de Login - *Back Office*.

Na tela de *Login*, o usuário informará o seu *E-mail* e *Senha* para que sejam feitas as validações necessárias dos seus dados. Caso os dados informados sejam válidos, o usuário será direcionado à tela Home da aplicação, conforme Figura 4.10 e Figura 4.11. Se as informações inseridas não condizem com os dados cadastrados na base de dados, será exibido um alerta de erro.

Abaixo do botão de *Login*, encontra-se o botão que direciona o usuário à tela de registro, conforme a figura Figura 4.9.

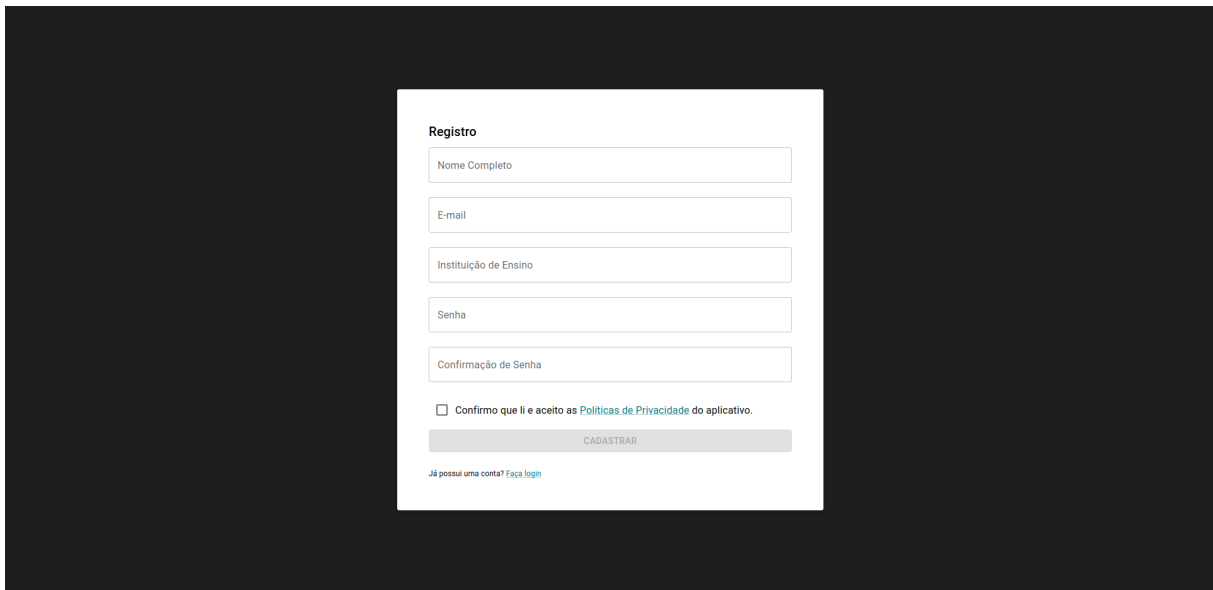


Figura 4.9: Tela de Cadastro - *Back Office*.

Na tela de Registro, o usuário encontrará o formulário para cadastro de conta, necessitando preencher os dados: nome completo, *e-mail*, instituição de ensino, senha e confirmação de senha. Caso o usuário deseje prosseguir com o cadastro, deverá aceitar as políticas de privacidade, que podem ser acessadas pelo *link* presente na *checkbox* da parte inferior do formulário. Esta medida serve para deixar o software em conformidade com a Lei Geral da Proteção de Dados (LGPD) [42]. O modelo utilizado para a criação da política de privacidade foi disponibilizado gratuitamente na plataforma Jusbrasil pelo usuário Pensador Jurídico [43]

Após clicar no botão de Cadastro, os dados informados pelo usuário serão validados. Caso os dados apresentados estejam inválidos, será exibido um alerta de erro. Caso contrário, o usuário será redirecionado para a tela de *Login*, podendo acessar a aplicação utilizando os dados recém-cadastrados.

4.2.2 *Home* e Navegação

A tela *Home* é a primeira tela acessada pelo usuário após o *login*, contendo uma visão geral sobre as atividades do professor na plataforma: quantidade de times cadastrados, participantes, tópicos, subtópicos e desafios. Além disso, a tela possui informações do desempenho geral dos estudantes, representadas por meio de um gráfico, na resolução dos desafios cadastrados pelo professor.

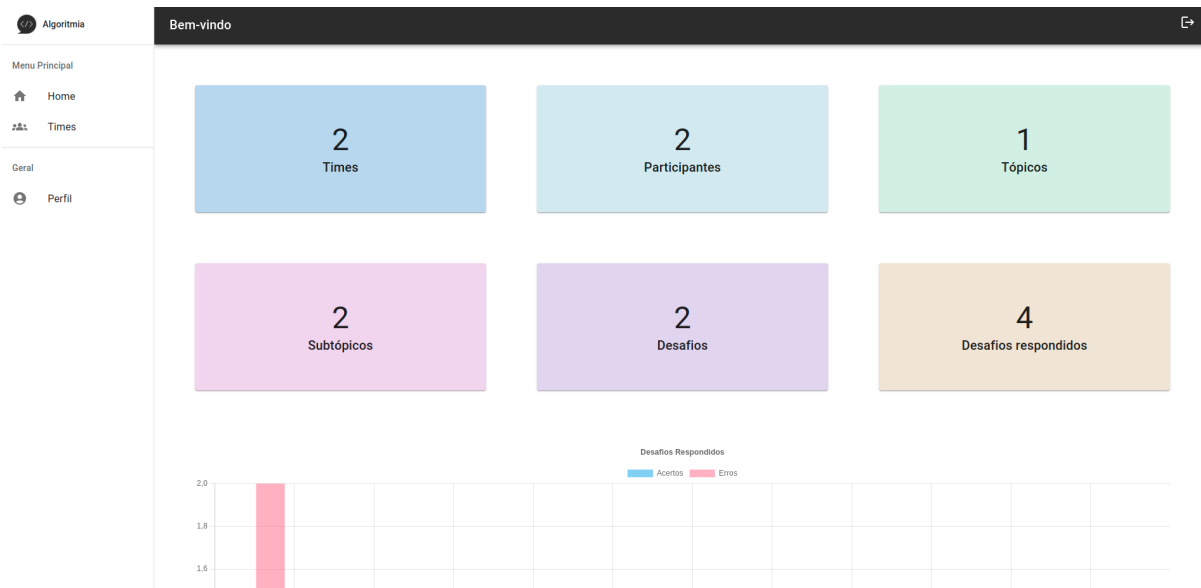


Figura 4.10: Tela *Home* - *Back Office*.

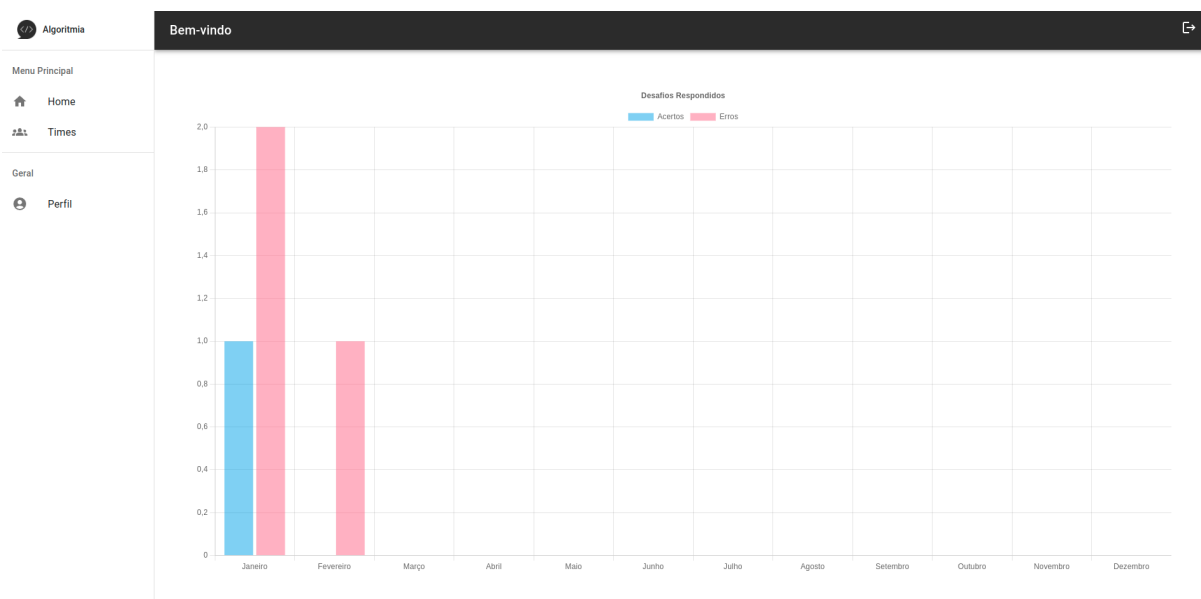


Figura 4.11: Tela *Home* - *Back Office*.

Pela Figura 4.10, é possível visualizar que, além do conteúdo da tela *Home*, a aplicação também conta com uma barra de navegação lateral e uma barra superior com uma mensagem de boas-vindas e um botão para realizar o *logout* ao final dela. Estes recursos são reutilizados em todas as telas, permitindo uma navegação de acesso mais fácil na aplicação *Web* para chegar à *Home*, tela de gerenciamento de times e à tela de gerenciamento de perfil.

4.2.3 Perfil

A tela de Perfil, acessada pela barra de navegação agrega os dados de usuário, permitindo a edição dos dados pessoais e da senha. Apenas o *e-mail* não pode ser editado, pois é o principal identificador de um usuário, conforme descrito nas políticas de privacidade da plataforma, onde é informado que alterar o *e-mail* ou apagar uma conta só é possível mediante o contato direto com um administrador da plataforma.

The image displays two screenshots of a web application's user profile management interface. The top screenshot, titled "Perfil do Usuário", shows a form with three input fields: "Nome Completo" containing "Pedro Henrique de Brito Agnes", "E-mail" containing "pedenite@gmail.com", and "Instituição de Ensino". Below the fields is a dark button labeled "ALTERAR DADOS". The bottom screenshot, titled "Editar Senha", shows a form with three input fields: "Senha Atual", "Nova Senha", and "Confirmação de Senha". Below these fields is a dark button labeled "ALTERAR SENHA".

Figura 4.12: Tela de gerenciamento de Perfil - *Back Office*.

Na Figura 4.12, é possível visualizar dois formulários, sendo um para alterar o nome completo e a instituição de ensino do usuário e o outro para alterar a senha. Para realizar a alteração de senha, é necessário validar a senha anteriormente cadastrada e digitar a nova senha duas vezes, trazendo uma conformidade à LGPD [42].

4.2.4 Times

A tela de Times pode ser acessada pela barra de navegação lateral, e é onde o professor é capaz de visualizar os seus times, assim como cadastrar novos. Para cada um dos times cadastrados, é possível acessar a tela específica dele para visualizar mais informações sobre ele.

Bem-vindo

Inserir Novo Time

Nome do time
Turma de APC 2023/1

Descrição
Aqui você encontra os tópicos relacionados à turma de APC do primeiro semestre de 2023

ADICIONAR

Times cadastrados

Time	Participantes	Tópicos
------	---------------	---------

Figura 4.13: Tela de Times - *Back Office*.

Na Figura 4.13, apenas o conteúdo da tela específica foi incluído, porém a barra de navegação lateral continua presente. É possível visualizar que o usuário está cadastrando um novo time chamado "Turma de APC 2023/1", bastando pressionar o botão de adicionar para realizar esta inclusão.

Bem-vindo

Time cadastrado com sucesso

Inserir Novo Time

Nome do time

Descrição

ADICIONAR

Times cadastrados

Time	Participantes	Tópicos
Turma de APC 2023/1	0	0

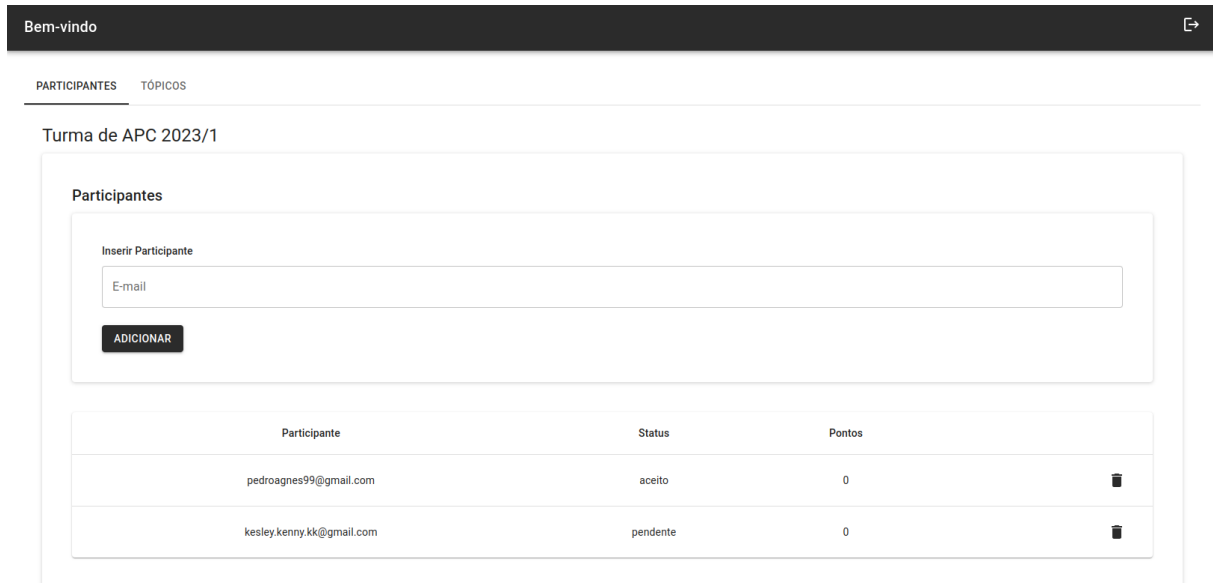
Figura 4.14: Tela de Times - Novo time cadastrado - *Back Office*.

Ao cadastrar um time, o usuário vai visualizar uma mensagem que serve como *feedback* e vai ver o time cadastrado na tabela abaixo dos campos do formulário, como mostrado

na Figura 4.14. Ao pressionar a lupa em um dos times listados, o usuário é redirecionado à tela deste time.

4.2.5 Participantes e Tópicos

Aqui o professor poderá incluir participantes e tópicos para o seu time por meio de um formulário presente em cada uma das duas abas mostradas na Figura 4.15 e na Figura 4.16



The screenshot displays the 'Participantes' (Participants) section for the 'Turma de APC 2023/1'. It features a form to add a new participant and a table listing existing participants.

Formulário de Adição:

Inserir Participante

E-mail

ADICIONAR

Tabela de Participantes:

Participante	Status	Pontos	
pedroagnes99@gmail.com	aceito	0	🗑️
kesley.kenny.kk@gmail.com	pendente	0	🗑️

Figura 4.15: Tela do Time selecionado - Participantes - *Back Office*.

Para cadastrar participantes, o professor deverá digitar o e-mail de um usuário existente na plataforma. Na tabela que lista os participantes, o professor verá o *Ranking* dos alunos cadastrados no time, servindo como uma forma de avaliar o desempenho de cada. Se por algum motivo, o professor precise desativar um usuário que adicionou anteriormente, poderá fazê-lo por meio do ícone de lata de lixo à direita.

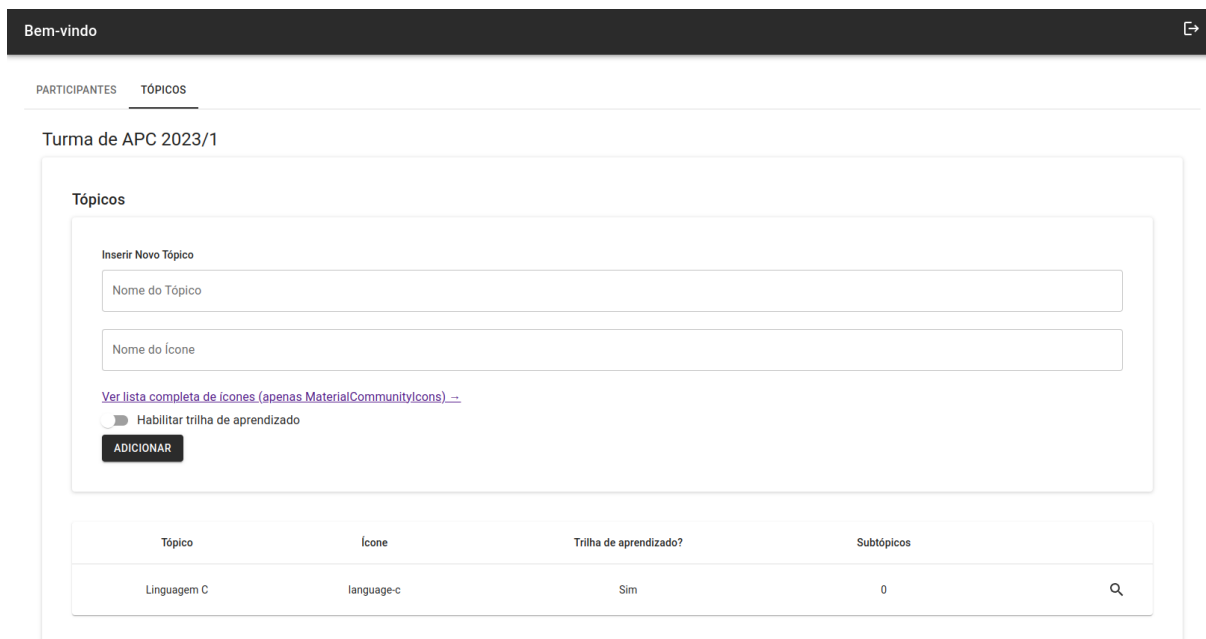


Figura 4.16: Tela do Time selecionado - Tópicos - *Back Office*.

No cadastro de tópico, o professor deverá informar um nome, um ícone, seguindo o padrão disponibilizado e definir se o curso terá uma trilha de aprendizado ou não. A trilha de aprendizado ativada exige que o aluno realize os exercícios disponibilizados em uma ordem definida pelo professor. Caso não esteja ativada, o aluno é livre para realizar os desafios propostos na ordem que preferir. Com o tópico cadastrado, o ícone de lupa permite a navegação para a tela de gerenciamento deste tópico.

As telas presentes em cada uma das duas abas funcionam da mesma forma que o formulário de adicionar times. Ao realizar o cadastro, o novo item vai aparecer na tabela presente no final da página.

4.2.6 Subtópicos

Ao selecionar um tópico cadastrado, o professor poderá explorar o conteúdo deste tópico, que são cadastrados como subtópicos. Da mesma forma que os formulários anteriores, o professor poderá cadastrar um novo subtópico incluindo um nome e uma descrição.

Bem-vindo

Turma de APC 2023/1

Subtópico

Inserir Novo Subtópico

Nome do Subtópico

Descrição do subtópico

ADICIONAR

Subtópico	Desafios
Exercícios Básicos	0

Figura 4.17: Tela do Tópico selecionado - Subtópicos - *Back Office*.

A Figura 4.17 mostra a organização do formulário e a tabela abaixo com os subtópicos existentes. No momento do cadastro, o professor poderá opcionalmente, iniciar uma história na descrição do subtópico, que irá aparecer para o aluno, de forma a deixar a experiência de aprendizado mais expressiva [1].

4.2.7 Desafios

Os desafios são cadastrados dentro de um subtópico, sendo o principal pilar da gamificação do *Algoritmia*. Para realizar o cadastro de um desafio, o professor deverá preencher diversos campos que estão listados a seguir.

- **Nome:** Um identificador para o desafio.
- **Pontos:** A quantidade de pontos que o estudante vai obter ao acertar o desafio.
- **Descrição:** Serve como o corpo do desafio, sendo o elemento mais importante para o correto entendimento por parte do estudante do que o desafio pede. O campo foi configurado para permitir a sintaxe da linguagem Markdown [24], porém com botões de formatação para facilitar a escrita para usuário que não possuam conhecimento da linguagem. Caso o professor esteja elaborando uma história para o seu curso para uma melhor aplicação da gamificação [1], é neste campo que ele deverá continuar a história que foi escrita nos desafios anteriores e descrições do subtópico.
- **Modelo da Resposta:** Define se o usuário deverá digitar a sua resposta, se deverá marcar uma opção de múltipla escolha ou se poderá de marcar mais de uma opção para chegar à resposta correta.

- **Feedback:** Campo opcional, que serve como uma justificativa da resposta correta para um melhor entendimento do aluno.

Bem-vindo [→]

Turma de APC 2023/1

Desafios

Inserir Novo Desafio

Nome do Desafio: Pontos do Desafio:

Descrição do Desafio

Complete o seguinte código de forma a obter o "output" mostrado abaixo.

```
...
__(&quot;Hello World&quot;);
...
```

Output desejado:
-> Hello World

Output desejado:
Hello World

Modelo da Resposta

input
 radio
 checkbox

Resposta Resposta do desafio

Feedback do Desafio

ADICIONAR

Desafio	Pontos	Tipo	
Input e Output	10	input	🔍

Figura 4.18: Tela do Subtópico selecionado - Desafios - *Back Office*.

Assim como nas telas anteriores, abaixo do formulário, se encontra uma tabela com os desafios já cadastrados. Ao *clique* na lupa à direita, o professor será capaz de visualizar um relatório geral sobre o desafio, visualizando os estudantes que acertaram e os que erraram este desafio em específico, assim sendo capaz de observar em quais assuntos os alunos demonstram maior dificuldade, logo tendo os dados para realizar uma possível melhoria no seu curso e estando em concordância com o elemento de *design* de jogos de gráficos e performance por meio do relatório.

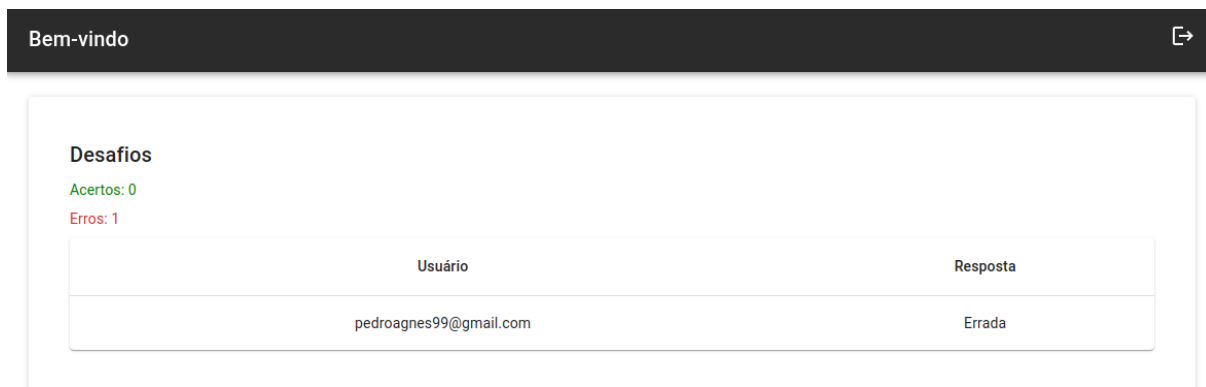


Figura 4.19: Tela do Desafio selecionado - *Back Office*.

Com todas estas telas, o professor tem acesso a todas as ferramentas do **Algoritmia** para realizar a criação do seu curso de forma gamificada, porém alguns pontos devem ser observados pelo próprio professor como a história [1], cabendo à sua decisão de implementar ou não. A pontuação atribuída a cada desafio deve ser observada com cuidado também para não deixar uma diferença muito grande de pontos entre um desafio e outro.

Por meio da visualização do *Ranking*, o professor também pode incluir prêmios para os alunos que se destacaram de forma a tentar aumentar a motivação dos estudantes [12], mas ao mesmo tempo, não deve excluir os usuários que estão no fundo da classificação para não gerar uma desmotivação destes. A ideia seria procurar ajudá-los e incentivar que os alunos se comuniquem entre si para promover a ajuda coletiva e a interação, melhorando o engajamento de forma geral.

4.3 Testes Preliminares

A fim de avaliar a eficiência do **Algoritmia** como um instrumento de ensino e aprendizagem gamificada de computação, o aplicativo móvel passou por um processo de validação por testes exploratórios feitos por uma vasta gama de pessoas, incluindo pessoas sem conhecimento de computação, pessoas que estão ingressando na área, pessoas que estão no processo de graduação, e pessoas que já têm uma formação na área ou até uma pós-graduação. O teste consiste do usuário explorar o aplicativo em seu dispositivo, acessando as funcionalidades e responder um questionário para informar a sua opinião sobre ele. Primeiramente, tem uma pergunta sobre o seu nível de conhecimento de computação e depois perguntas sobre a usabilidade, efetividade no ensino e aprendizagem de computação e sugestões sobre melhorias ou até correções de possíveis erros encontrados.

O questionário foi criado usando o Google Forms² [44], onde foi gerado uma planilha³ que pode ser acessada remotamente com as respostas. O aplicativo foi testado por um total de oito pessoas. Dentre as perguntas mais relevantes que foram realizadas, estão:

4.3.1 Nível de ensino em computação

Pergunta feita para realizar uma pesquisa demográfica dos testadores, foi obtida a seguinte porcentagem para cada nível de ensino:

- 12,5% não possui conhecimento algum de computação;
- 12,5% está iniciando uma graduação em alguma área da computação
- 25% está próximo de concluir a graduação em computação;
- 37,5% já possui uma graduação na área de computação;
- 12,5% possui uma pós-graduação na área de computação.

4.3.2 Nota para a experiência de uso do aplicativo

Para a nota, o usuário pode atribuir qualquer valor inteiro entre 0 e 10. A distribuição das notas foi feita conforme as porcentagens a seguir:

- 50% avaliou com a nota 10;
- 12,5% avaliou com a nota 9;
- 25% avaliou com a nota 8;
- 12,5% avaliou com a nota 7.

4.3.3 O aplicativo possui uma interface intuitiva?

- 87,5% achou intuitiva e extremamente fácil de utilizar;
- 12,5% achou intuitiva, porém se perdeu em alguns momentos enquanto utilizava o aplicativo.

Todos responderam que acreditam que a plataforma tem a capacidade de auxiliar os estudantes e professores no ensino de computação e a maioria respondeu "Sim" sobre acreditar que o formato dos exercícios e *ranking* pode servir como fatores motivadores

²<https://forms.gle/DH92rb5BbDZyWCJJA>

³https://docs.google.com/spreadsheets/d/17K7eX7XIzmfCAD77se4NNyVmgom-Bp3m6KPL_c35uTM/edit?usp=sharing

durante a resolução dos exercícios, porém duas pessoas responderam "Talvez". A média das notas que os usuários atribuíram para o aplicativo de acordo com as suas experiências na plataforma foi 9 de um total de 10.

4.3.4 Você acredita que a plataforma é capaz de auxiliar os estudantes e professores no ensino e aprendizagem de computação?

As opções disponíveis para a pergunta eram "Sim", "Não" e "Talvez". Todos responderam "Sim" para esta pergunta.

4.3.5 Você acha que o formato dos exercícios e do ranking serviram como fatores motivadores durante a resolução dos desafios?

Nesta pergunta, 75% dos participantes responderam "Sim", enquanto os outros 25% responderam "Talvez". Nenhum participante respondeu "Não".

4.3.6 Pontos positivos e negativos e sugestões

Esta pergunta foi dividida em duas, sendo uma para os pontos positivos e negativos e a outra para sugestões opcionais. Algumas das respostas mais relevantes dos participantes estão listadas a seguir:

- Pontos positivos:
 - Modo de *storytelling* para mostrar os cenários;
 - União da tecnologia ao ambiente de aprendizagem;
 - Facilidade de interação e aprendizagem através do aplicativo;
 - Gamificação e visualização da performance.
- Pontos negativos ou defeitos encontrados:
 - Algumas opções de resposta não apareceram na tela corretamente;
 - Uma conversa é iniciada com outro usuário antes de mandar a primeira mensagem.
- Sugestões:
 - Adicionar opções de alteração de tema do aplicativo;

- Iniciar uma conversa com outro usuário somente quando a primeira mensagem for enviada;
- Inclusão de parte teórica antes dos desafios por meio de textos ou vídeos para tornar mais didático.

Capítulo 5

Considerações Finais

Tendo em vista os objetivos específicos deste trabalho, sendo o primeiro deles, o desenvolvimento de um aplicativo que integre elementos de gamificação em um ambiente de ensino de computação, foi desenvolvido com êxito o *Front Office* do *Algoritmia*, e o *feedback* dos usuários que validaram o aplicativo foi em geral, muito positivo. Já o segundo objetivo, de fornecer uma ferramenta para que os professores usem a tecnologia de maneira efetiva e integrem-na em sua prática docente, foi cumprido por meio do *Back Office*, e foi testado com a inclusão de um curso para os testes exploratórios pelo usuários, que em maioria acreditam que o formato dos exercícios e *ranking* pode gerar uma maior motivação por parte dos alunos, melhorando a aprendizagem. Para o terceiro objetivo, de oferecer um mecanismo de *feedback* imediato e eficaz para os alunos, ao concluir um desafio no *Front Office*, o aluno tem imediatamente acesso à sua resposta, assim como as taxas de acerto e erro e uma justificativa sobre as respostas corretas. Por fim, o *Back Office*, o professor é capaz de visualizar os relatórios de desempenho dos alunos, sendo capaz de visualizar as taxas de acerto e erro para cada desafio criado, satisfazendo o quarto e último objetivo, para permitir o acompanhamento do progresso dos alunos e fornecer relatórios detalhados aos professores.

O ensino de computação tem a capacidade de abrir muitas portas para o futuro dos profissionais da área e até para o Brasil como um todo. O setor de TIC tem se tornado, cada vez mais, uma das áreas mais importantes do desenvolvimento do país, pois trata da aplicação da tecnologia na sociedade e é uma área que está precisando de cada vez mais profissionais qualificados. Apesar disso, as taxas de evasão dos cursos relacionados à computação se mostram altas, que pode ter como causa, diversos fatores. O *Algoritmia* é um aplicativo móvel que procura aproximar mais pessoas do ensino de computação por meio da gamificação, disponibilizando ferramentas para os professores adequarem as suas aulas ao modelo de forma a tentar aumentar a motivação e o engajamento dos alunos, conseqüentemente, tendo a capacidade de diminuir as taxas de evasão elevadas.

Seguindo a metodologia Scrum, ao final de cada *Sprint*, foi desenvolvido um protótipo funcional para avaliação do produto atual. A versão atual representa uma versão funcional da ideia do projeto, que ainda tem muito a evoluir com novas *Sprints* a serem desenvolvidas para realizar melhorias dos mais diversos aspectos como alguns dos apontamentos obtidos nos testes exploratórios para tornar a experiência do usuário mais agradável e potencializar o fator da gamificação. O processo de desenvolvimento foi feito utilizando conceitos de metodologias ágeis, produzindo as documentações necessárias para o projeto e que facilitam a manutenção e compreensão de cada módulo desenvolvido. Os testes automatizados [45] são uma ótima fonte para tentar garantir que todas as funcionalidades continuarão funcionando independente de quaisquer ações evolutivas ou de manutenção. Também foram criados e utilizados diversos componentes por meio do React para modularizar as aplicações, assim como promover o reaproveitamento.

5.1 Trabalhos Futuros

Como trabalhos futuros, pretendemos melhorar principalmente a questão da gamificação e experiência de usuário para que o *Algoritmia* se torne uma ferramenta completa de ensino e aprendizagem de computação. O uso de técnicas de UX [46] e a inclusão de mais elementos de gamificação [1] como uma loja no aplicativo para gastar os pontos comprando avatares ou outros itens, trazem a possibilidade de aumentar a motivação do estudante. No *Back Office*, a inclusão de mais opções para o professor monitorar o desempenho dos alunos e gerar relatórios mais específicos poderia ajudar nas metodologias do professor. Por fim, o teste da plataforma em sala de aula por professores com uma boa experiência na área que estejam abertos às ideias de ensino inclusivo por meio da gamificação ajudaria na obtenção de resultados mais concretos e *feedbacks* de alunos e professores sobre possíveis melhorias mais diretas a implementar tanto no *Front Office* quanto no *Back Office*.

Referências

- [1] Sailer, Michael, Jan Ulrich Hense, Sarah Katharina Mayr e Heinz Mandl: *How gamification motivates: An experimental study of the effects of specific game design elements on psychological need satisfaction*. Computers in Human Behavior, 69:371–380, 2017, ISSN 0747-5632. <https://www.sciencedirect.com/science/article/pii/S074756321630855X>. 1, 5, 7, 31, 46, 48, 53
- [2] Roumen Vesselinov, John Grego: *Duolingo effectiveness study*. 2012. 3
- [3] Briana B. Morrison, Betsy DiSalvo: *Khan academy gamifies computer science*. ACM technical symposium on Computer science education, 2014. 3
- [4] Kishimoto, Tizuko M.: *Jogo, Brinquedo, Brincadeira e a Educação*, volume 1. ABDR, 2017. 5
- [5] Brougère, Gilles: *Brinquedo e Cultura*, volume 2. Cortez editora, 1997. 5
- [6] Henriot, Jacques: *Le Jeu*, volume 1. Presses Universitaires de France, 1969. 5
- [7] ESA: *2022 essential facts about the video game industry*. <https://www.theesa.com/resource/2022-essential-facts-about-the-video-game-industry/>. acesso em 14/01/2023. 6
- [8] Marchand, André e Thorsten Hennig-Thurau: *Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities*. Journal of Interactive Marketing, 27(3):141–157, 2013, ISSN 1094-9968. <https://www.sciencedirect.com/science/article/pii/S1094996813000170>. 6
- [9] apptamin: *The rise of mobile gaming in 2021*. <https://www.apptamin.com/blog/the-rise-of-mobile-gaming-in-2021/>, 2022. acesso em 30/01/2023. 6
- [10] IBGE: *Pesquisa nacional por amostra de domicílios contínua*. Diretoria de Pesquisas, 2019. acesso em 30/01/2023. 6
- [11] Almeida, Laiane Lima dos Santos Cantuária, Joana Corrêa Goulart Elaine Vieira de: *Os avanços tecnológicos no século xxi: desafios para os professores na sala de aula*. REEDUC, 7(2), 2021. 6
- [12] K. Werbach, D. Hunter: *For the win: How game thinking can revolutionize your business*. Wharton Digital Press, Philadelphia, 2012. 7, 8, 48

- [13] Burguillo, J.C.: *Using game theory and competition-based learning to stimulate student motivation and performance*. Computers & Education, 2(55):566–575, 2010. 7, 38
- [14] Nicholson, S.: *A recipe for meaningful gamification*. Springer, New York, 2015. 7
- [15] Kapp, Karl M.: *The gamification of learning and instruction: Game-based methods and strategies for training and education*. Pfeiffer and ASTD, 2012. 8
- [16] Brasscom: *Relatório setorial de tic 2021*. <https://brasscom.org.br/pdfs/relatorio-setorial-de-tic/>. acesso em 14/01/2023. 8
- [17] Araújo, Antoniel de Souza Alves, Ilda Neta Silva de Almeida David Pereira de: *As tics e a evasão escolar nos cursos superiores da Área de computação: Um estudo de caso*. Multi Debates ITOP, 5(1), 2021. 8
- [18] Hoed, Raphael Magalhães: *Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de computação*. UnB, 2016. 8
- [19] Aléssio, Simone Cristina, Neli Miglioli Sabadin e Pedro Sidnei Zanchett: *Processos de Software*, volume 2017. UNIASSELVI, 2017. 11
- [20] Schwaber, Ken e Jeff Sutherland: *The Definitive Guide to Scrum: The Rules of the Game*. Ken Schwaber and Jeff Sutherland, 2020. 11
- [21] Pekka Abrahamsson, Outi Salo, Jussi Ronkainen e Juhani Warsta: *Agile software development methods: Review and analysis*. VTT Technical Research Centre of Finland, 2002(478), 2002. 11
- [22] Connell JL, Shafer LB.: *Structured rapid prototyping*. Prentice Hall International, Yourdon Press, 1989(1989), 1989. 13
- [23] SAS, Material UI: *Material ui*. <https://mui.com/>. acesso em 30/01/2023. 13, 27
- [24] Cone, Matt: *Markdown guide*. <https://www.markdownguide.org/>. acesso em 28/01/2023. 17, 46
- [25] Grady Booch, James Rumbaugh, Ivar Jacobson: *The Unified Modeling Language User Guide*. Addison-Wesley, 2ª edição, 2005. 21
- [26] Loeliger, Jon e Matthew McCullough: *Version Control With Git*. O’Reilly, 2020. 23
- [27] GitHub: *Github pages*. <https://pages.github.com/>. acesso em 28/01/2023. 24
- [28] StatCounter: *Mobile operating system market share worldwide*. <https://gs.statcounter.com/os-market-share/mobile/worldwide>. acesso em 31/12/2022. 25
- [29] Google: *Android*. <https://developer.android.com/guide/platform>. acesso em 31/12/2022. 25

- [30] Microsoft: *Typescript*. <https://www.typescriptlang.org/>. acesso em 31/12/2022. 26
- [31] Mozilla: *Javascript*. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. acesso em 31/12/2022. 26
- [32] Meta: *React*. <https://reactjs.org/>. acesso em 31/12/2022. 27
- [33] Meta: *React native*. <https://reactnative.dev>. acesso em 31/12/2022. 27
- [34] Expo: *Expo*. <https://expo.dev/>. acesso em 23/01/2022. 28
- [35] Fewster, Mark e Dorothy Graham: *Software Test Automation*. ACM Pres, 1999. 28
- [36] Meta: *Jest*. <https://jestjs.io/pt-BR/>. acesso em 14/01/2023. 28
- [37] Microsoft: *Visual studio code*. <https://code.visualstudio.com/>. acesso em 31/12/2022. 29
- [38] Microsoft: *Visual studio code intellisense*. <https://code.visualstudio.com/docs/editor/intellisense>. acesso em 31/12/2022. 29
- [39] Google: *Firebase*. <https://firebase.google.com/>. acesso em 31/12/2022. 29
- [40] Google: *Cloud firestore*. <https://firebase.google.com/docs/firestore>. acesso em 31/12/2022. 29
- [41] choosealicense: *Mit license*. <https://choosealicense.com/licenses/mit/>. acesso em 09/01/2023. 30
- [42] Secretaria-Geral da república: *Lei nº 13.709, de 14 de agosto de 2018 (lgpd)*. https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/113709.htm. acesso em 07/01/2023. 32, 40, 42
- [43] Pensador Jurídico: *[modelo] política de privacidade*. <https://pensadorjuridico.jusbrasil.com.br/modelos-pecas/784935103/modelo-politica-de-privacidade>. acesso em 05/01/2023. 32, 40
- [44] Google: *Forms*. <https://www.google.com/forms/about/>. acesso em 03/02/2023. 49
- [45] Beck, Kent: *Test-Driven Development by Example*. Addison-Wesley, 2000. 53
- [46] Hassenzahl, Marc: *User experience (ux): towards an experiential perspective on product quality*. Conference on l'Interaction Homme-Machine, 2008. 53