



Instituto Federal de Educação, Ciência e
Tecnologia de São Paulo
Campus Boituva

Especificação dos Requisitos, Análise e Projeto do Software Malba

Versão 0.2

Professores Orientadores: Giorjety L. Dias, Marcelo F. Polido e Bernardo M. Zabadal.

Autores:
Rafael Juliani Diehl¹

Email: rafael.diehl@aluno.ifsp.edu.br

BOITUVA
Junho de 2022

Autores:
Rafael Juliani Diehl

ESPECIFICAÇÃO DOS REQUISITOS, ANÁLISE E PROJETO DO SOFTWARE MALBA

Trabalho apresentado como requisito de avaliação do projeto interdisciplinar que contempla as disciplinas Análise Orientada a Objetos, Banco de Dados II e Programação Web, do curso de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de São Paulo, Campus Boituva, sob a orientação dos professores Giorjety Licorini Dias, Marcelo Figueiredo Polido e Bernardo Moreira Zabadal.

BOITUVA
Junho de 2022

Histórico das Revisões

Data	Versão	Descrição	Autor
04/04/2022	0.1.1	Revisão da versão passada, novo levantamento de requisitos e confecção dos diagramas de casos de uso.	Rafael Juliani Diehl
09/05/2022	0.1.2	Criação da descrição geral do produto, incluindo estudo de viabilidade.	Rafael Juliani Diehl
06/06/2022	0.1.3	Criação dos diagramas de casos de classes.	Rafael Juliani Diehl
13/06/2022	0.1.4	Criação do manual do sistema e finalização do protótipo interativo.	Rafael Juliani Diehl
19/06/2022	0.2	Criação dos documentos de implantação e componentes, e revisão dos outros diagramas.	Rafael Juliani Diehl

SUMÁRIO

1 INTRODUÇÃO	5
1.1 Objetivo	5
1.2 Escopo do Produto	5
1.2.1 Componentes Principais	5
1.2.2 Missão do Produto	5
1.3 Definições, Siglas e Abreviações	6
1.4 Visão Geral	6
2 DESCRIÇÃO GERAL DO PRODUTO	7
2.1 Estudo de Viabilidade	7
2.2 Perspectiva do produto	7
2.3 Características do Usuário	7
2.4 Técnica(s) utilizada(s) para levantamento de requisitos	7
2.5 Requisitos do Sistema	8
2.5.1 Requisitos funcionais	9
2.5.2 Requisitos não-funcionais	12
2.6 Usuários e sistemas relacionados	16
3 PROJETO DE SOFTWARE	17
3.1 Diagrama de Caso de Uso	17
3.2 Diagrama de Classes	18
3.3 Diagrama de Componentes	20
3.4. Diagrama de Implantação	21
4 MODELAGEM E PROJETO DO BANCO DE DADOS	22
5 MANUAL DO SISTEMA	25
APÊNDICE A	38
REFERÊNCIAS	39

1 INTRODUÇÃO

Este documento especifica os requisitos que o sistema Malba deve atender, fornecendo aos desenvolvedores as informações necessárias para o projeto de implementação, realização dos testes e homologação do sistema.

1.1 Objetivo

Todo ser humano, ao longo de sua vida, adquire habilidades distintas que são aperfeiçoadas com o passar do tempo de acordo com seu desenvolvimento e prática. Algumas pessoas, entretanto, não conseguem obter proficiência nessas possíveis habilidades por meio de práticas convencionais, pois cada indivíduo aprende de uma forma diferente.

Neste contexto, o objetivo do software é auxiliar indivíduos com dificuldade de aprendizagem da matemática básica e proporcioná-los um ambiente de ensino lúdico e acessível, democratizando assim o acesso ao conhecimento.

Público Alvo: alunos que possuem baixo desempenho na disciplina de matemática, entre outras pessoas que tem interesse em se ligar mais com a área ou experimentar novas metodologias de ensino.

1.2 Escopo do Produto

1.2.1 Componentes principais

- Malba (componente único)

1.2.2 Missão do produto

A missão do Malba é tornar o aprendizado divertido e acessível a todos, usando a tecnologia para auxiliar os usuários de forma direta na compreensão dos assuntos abordados e mantê-los motivados e consistentes em suas trajetórias.

Acreditamos que a educação só se exerce plenamente num ambiente democrático, onde as diferentes necessidades dos indivíduos são consideradas.

1.3 Definições, Siglas e Abreviações

Sigla	Definição
API	Interface de Programação de Aplicações.
Axios	Cliente HTTP usado para consumir dados de APIs.
CSS	Linguagem de estilização, usada em conjunto com HTML.
HTTP	Protocolo de rede usado para comunicação na Web.
HTML	Linguagem de marcação, bloco mais básico da Web.
Git	Sistema de Controle de Versões Distribuído.
Github	Plataforma que armazena código-fonte com Controle de Versão.
JavaScript	Linguagem de programação usada ao lado do cliente.
Laravel	Framework PHP que usa o padrão MVC.
MySQL	Sistema Gerenciador de Banco de Dados SQL.
PHP	Linguagem de programação usada ao lado do servidor.
REST	Modelo de arquitetura para comunicação cliente-servidor.
Sass	Pré-processador CSS.
SPA	Aplicativo de Página Única, técnica usada em aplicações web.
Vue.js	Framework progressivo JavaScript, para construção de interfaces.

Tabela 1: Siglas e Definições

1.4 Visão Geral

Este documento contém a especificação de requisitos, análise e projeto do software Malba. Conta com os diagramas UML: diagrama de casos de uso, diagrama de classes, diagramas de sequência (para cada caso de uso), diagrama de componentes e diagrama de implantação. O banco de dados está representado de acordo com os modelos conceitual, lógico e físico. No fim do documento, há um link para o protótipo e um manual do sistema, com a explicação do uso de cada tela disponível.

2 DESCRIÇÃO GERAL DO PRODUTO

Esta seção apresenta uma descrição do produto, explicitando o estudo de viabilidade, a perspectiva do produto, as características dos usuários do sistema, as técnicas de levantamento de requisitos e os requisitos obtidos (classificados em funcionais e não-funcionais, e separados de acordo com sua prioridade).

2.1 Estudo de Viabilidade

Apesar do modelo do sistema ser relativamente popular, tendo como precursor a plataforma Duolingo (voltada para o ensino de idiomas), o ensino de matemática através de navegadores web é extremamente escasso, concentrando-se mais em aplicativos para dispositivos móveis. Pensando nisso, Malba foi criado a fim de atingir um grande número de usuários que usam a web e não encontram uma plataforma de ensino focada em séries de lições, viável dentre as opções disponíveis.

2.2 Perspectiva do produto

O sistema relaciona-se diretamente com os usuários, que o acessam em seus computadores pessoais através da Internet. Além disso, o sistema se comunica com um servidor de e-mails, responsável pelo envio e recebimento da confirmação de autenticação do usuário.

O sistema pode ser acessado localmente, através dos servidores de desenvolvimento concedidos pelas ferramentas Vue CLI e Laravel e espera-se, no futuro, hospedá-los no provedor web Hostinger.

2.3 Características do Usuário

Os usuários não precisam de qualquer conhecimento prévio sobre o sistema para poder utilizá-lo. O uso é bastante simples, prezando por uma interface minimalista e intuitiva, permitindo que usuários com pouco conhecimento de informática possam sentir-se à vontade.

2.4 Técnica(s) utilizada(s) para levantamento de requisitos

Nesta seção estão descritas as técnicas usadas para levantar os requisitos do sistema e, posteriormente, realizar a validação dos mesmos.

2.4.1 Brainstorming

A etapa inicial foi o brainstorming, através de uma reunião entre os desenvolvedores do projeto. Durante essa fase, os participantes pesquisaram sobre modelos semelhantes já em alta no mercado, expuseram suas ideias e tentaram chegar num consenso, enriquecendo as soluções para o sistema como um todo. Após a reunião, foi feita uma análise das ideias coletadas e houve o prosseguimento para as próximas etapas.

2.4.2 Questionário

Como o software contempla um público-alvo amplo, um questionário que levantou dados sobre o público, como suas possíveis dificuldades em matemática e interesse em determinadas funcionalidades, foi feito anteriormente numa fase beta do sistema. Esta foi uma fase necessária para delimitar o público-alvo e consolidar o estudo de viabilidade. O gerenciamento de pesquisas foi realizado através do serviço Google Forms.

2.4.3 Prototipagem

O protótipo do sistema serviu para validar os requisitos obtidos anteriormente. Já que o front-end do sistema é baseado em componentes, foi escolhido a representação de protótipo de alta fidelidade, permitindo grande controle sobre espaçamento, cores e tipografia. O protótipo foi feito no editor gráfico Figma, onde foi possível construir uma série de telas sequenciais que permitiram uma visão mais ampla do layout e das funcionalidades do sistema. O editor é colaborativo, portanto foi possível ter feedbacks.

O protótipo está descrito detalhadamente no [Apêndice A](#), no final deste documento.

2.5 Requisitos do Sistema

Os requisitos funcionais e não funcionais, foram identificados pela sua prioridade.

Para estabelecer a prioridade dos requisitos, foram adotadas as denominações: essencial, importante e desejável. Abaixo temos a descrição de significado de cada uma dessas denominações:

Essencial	É o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.
Importante	É o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.
Desejável	É o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

Tabela 2: Prioridade dos requisitos

2.5.1 Requisitos funcionais

Os requisitos funcionais podem ser definidos com as funções que os clientes querem ou precisam que o software ofereça, isto é, requisitos relacionados diretamente à funcionalidade do software.

- RF01 — Gerenciar contas

Prioridade: Essencial.

Descrição: Os clientes podem criar uma conta, alterar seu nome, avatar e suas preferências e encerrar sua conta a qualquer momento. Para criar a conta, ele deve acessar a página de cadastro e inserir seu nome, um nome de usuário (que é verificado quanto a disponibilidade), e-mail, senha (que é validada de acordo com sua complexidade) e o tipo de plano.

O administrador único do sistema, tem sua conta previamente cadastrada no banco de dados, com seu email e senha, e também pode alterar seus próprios dados.

- RF002 — Selecionar subtópico disponível

Prioridade: Essencial.

Descrição: Em seu dashboard, o cliente tem acesso a subtópicos disponíveis. Cada tópico possui quatro partições que são completadas em caso de sucesso.

Ao clicar num tópico, o usuário realiza uma série de questões.

- RF003 — Resolver questões

Prioridade: Essencial.

Descrição: O usuário realiza séries de dez questões (seis de dificuldade fácil, três de dificuldade média e uma de dificuldade difícil). Ele deve selecionar uma das alternativas disponíveis ou digitar num input, e, em seguida, clicar no botão usado para prosseguir entre as questões.

O cliente ganha pontos de experiência caso acerte questões. A quantidade de pontos varia de acordo com a dificuldade das questões.

- RF004 — Visualizar resultado

Prioridade: Essencial.

Descrição: Após realizar a série de questões, o usuário vê seu resultado. O resultado mostra as questões que ele acertou e quais ele errou, permitindo que ele revise seus erros.

- RF005 — Perder vidas

Prioridade: Desejável.

Descrição: O cliente perde vidas ao atingir menos de 60% de acertos numa série de questões. As vidas podem ser regeneradas, mas isto leva tempo, permitindo maior comprometimento do cliente ao realizar as questões.

- RF006 — Completar etapa

Prioridade: Importante.

Descrição: As etapas servem para delimitar o conhecimento que o cliente deve ter como pré-requisito. Cada subtópico possui quatro marcações de etapas que são completadas quando o cliente atinge mais de 60% de acertos numa série de questões. Ao completar todas as marcações, os próximos subtópicos são liberados.

Por exemplo, para acessar os subtópicos “Multiplicação” e “Divisão”, o cliente deve antes, completar os subtópicos “Adição” e “Subtração”.

- RF007 — Visualizar ranking

Prioridade: Desejável.

Descrição: o usuário pode visualizar um ranking com todos os participantes do sistema, vendo suas respectivas pontuações e posições. Posições altas concedem recompensas.

- RF008 — Gerenciar tópicos

Prioridade: Essencial.

Descrição: O administrador pode gerenciar os tópicos de questões. Cada tópico representa uma modalidade da Matemática e contém subtópicos. Quando o cliente preenche todas as etapas dos subtópicos de um tópico em específico, ele libera o próximo tópico. A página de gerenciamento contém funcionalidade de *drag and drop*, permitindo maior facilidade para ordenar os tópicos.

- RF009 — Gerenciar subtópicos

Prioridade: Essencial.

Descrição: O administrador pode gerenciar os subtópicos de questões. Cada subtópico representa uma modalidade dentro de um tópico. Por conta disso, a página deve ter um menu de opções com os tópicos, ao cadastrar um subtópico.

- RF010 — Gerenciar questões

Prioridade: Essencial.

Descrição: O administrador pode gerenciar questões do sistema através de seu dashboard. Cada questão possui um número identificador, uma imagem

ilustrativa, uma descrição, uma dificuldade (fácil, médio ou difícil) e um tipo (alternativa, selecione, dissertativa textual, dissertativa numérica, e verdadeiro-ou-falso). Os tipos estão mais detalhados abaixo:

- Dificuldade
 - a. Fácil: questões elementares, que podem ser resolvidas rapidamente. Concedem 2 XP.
 - b. Médio: questões que exigem mais raciocínio. Concedem 3 XP.
 - c. Difícil: questões difíceis, última de cada série. Concedem 4 XP.
- Tipo
 - a. Alternativa: questões que possuem quatro alternativas. Somente uma é a certa.
 - b. Selecione: questões que possuem quatro alternativas, porém é possível selecionar mais de uma.
 - c. Dissertativa textual: questões que possuem um input do tipo texto para o usuário preencher.
 - d. Dissertativa numérica: questões que possuem um input do tipo numérico para o usuário preencher.
 - e. Verdadeiro-ou-falso: questões com duas alternativas, verdadeiro ou falso.

Cada questão possui um tópico e cada tópico pode possuir várias sessões, por conta disso, a página deve ter um menu de opções com os tópicos, ao cadastrar uma nova questão.

2.5.2 Requisitos não funcionais

Os requisitos não funcionais são definidos de forma indireta, expressando restrições que o software deve cumprir. São fatores de qualidade e a satisfação deles pode afetar vários componentes do sistema.

I. Segurança

Descreve os requisitos associados à integridade dos dados, privacidade, como o sistema trata de informação confidencial, liberação de acesso aos usuários do sistema.

- RNF01 — Segurança no login

Prioridade: Essencial.

Descrição: Ao tentar se identificar, sempre ocorre uma validação da senha do usuário. Seus dados são armazenados no banco de dados e ficam acessíveis somente por ele mesmo e pelo administrador. A senha será armazenada de forma criptografada, usando as funções de hash SHA-2.

- RNF02 — Logout automático

Prioridade: Essencial.

Descrição: Logout se refere ao encerramento da sessão do usuário no navegador. Para garantir a preservação das informações e evitar vazamentos, assim que o navegador é fechado ou o usuário passa por um período de inatividade de 30 minutos, o logout ocorre automaticamente.

II. Desempenho

Descreve o tempo de resposta do sistema durante o uso dos recursos disponibilizados.

- RNF03 — Processamento de dados

Prioridade: Desejável.

Descrição: O sistema será um aplicativo de página única (SPA), concentrando toda sua funcionalidade numa única página. Ao invés de recarregar as páginas ou redirecionar, o conteúdo principal é modificado de forma assíncrona, garantindo alto desempenho e proporcionando uma experiência semelhante a de um aplicativo desktop.

III. Usabilidade

Descreve os requisitos não-funcionais associados à facilidade de uso do sistema.

- RNF04 — Interface minimalista e intuitiva

Prioridade: Importante.

Descrição: O design do layout será minimalista e padronizado, usando ícones e cores que representam as ideias e funcionalidades. O texto será objetivo, de fácil compreensão e usará fonte Inter no tamanho 16. Os menus serão posicionados na parte superior do sistema e permanecerão fixos, para facilitar o uso. Haverá opção de selecionar entre tema claro e escuro, para que o usuário possa adequar o plano de fundo de acordo com o seu ambiente e evitar problemas visuais futuros.

IV. Confiabilidade

Descreve os requisitos não funcionais associados à frequência de falha, e a robustez do sistema na recuperação destas falhas.

- RNF05 — Backup

Prioridade: Importante.

O sistema fará um backup (cópia de segurança) dos dados a cada 24 horas. As cópias são feitas a partir dos registros acessados pelo SGBD e são armazenadas em diretórios separados do sistema.

V. Padrões

Descreve quais os padrões e normas a serem seguidas ao desenvolvimento do sistema.

- RNF06 — Arquitetura REST

Prioridade: Desejável.

Descrição: REST (Transferência de Dados Representacionais) é um conjunto de restrições de arquitetura que permite a interação com serviços web RESTful. O Malba usa uma API REST própria que envia os dados para o front-end do sistema, que, por sua vez, os consome e transforma-os para adequar-se à interface.

VI. Hardware e Software

Descreve qual o hardware e software que será utilizado pelo sistema.

- RNF07 — Acesso web

Prioridade: Essencial.

Descrição: O sistema será acessado pela web, via internet, sendo necessário possuir um navegador instalado na máquina. Será compatível com os navegadores mais populares, usando recursos de layout estáveis que evitam quebras devido a incompatibilidade com versões mais antigas.

- RNF08 — Banco de dados

Prioridade: Desejável.

Descrição: Bancos de dados são conjuntos de arquivos relacionados entre na forma de registros. Para acessar um banco de dados, normalmente usa-se um Sistema Gerenciador de Banco de Dados (SGBD), que permite gerenciar seus registros. O SGBD usado no Malba será o MySQL da Oracle Corporation.

- RNF09 — Front-end

Prioridade: Desejável.

Descrição: O frontend será uma SPA (Single Page Application) desenvolvida usando Vue.js, um framework progressivo JavaScript. A estilização será combinada com Sass, um pré-processador CSS, e Tailwind, um framework que será processado pelo PostCSS. Para consumir os dados da API, será usado o cliente HTTP Axios.

- RNF10 — Back-end

Prioridade: Desejável.

Descrição: O back-end refere-se ao que está por trás das aplicações, isto é, oferece ações para o usuário da máquina. Malba usará uma API construída usando a linguagem PHP e o framework Laravel.

2.6 Usuários e sistemas relacionados

Nesta subseção estão definidos os atores do sistema Malba, e seus respectivos números de identificação.

Número de ordem	Ator	Definição
1	Cliente	Qualquer pessoa cadastrada sem poder administrativo, que terá acesso.
2	Administrador	Pessoa de nível administrativo superior que terá acesso a todas as funções de gerenciamento do sistema.

Tabela 3: Atores do sistema

3 PROJETO DE SOFTWARE

Esta seção especifica os principais diagramas UML necessários ao projeto, fornecendo múltiplas visões do sistema.

3.1 Diagrama de Casos de Uso

O diagrama de casos de uso tem por objetivo apresentar uma visão externa geral das funcionalidades que o sistema deverá oferecer aos usuários, sem se preocupar muito com a questão de implementação. (Guedes, 2009)

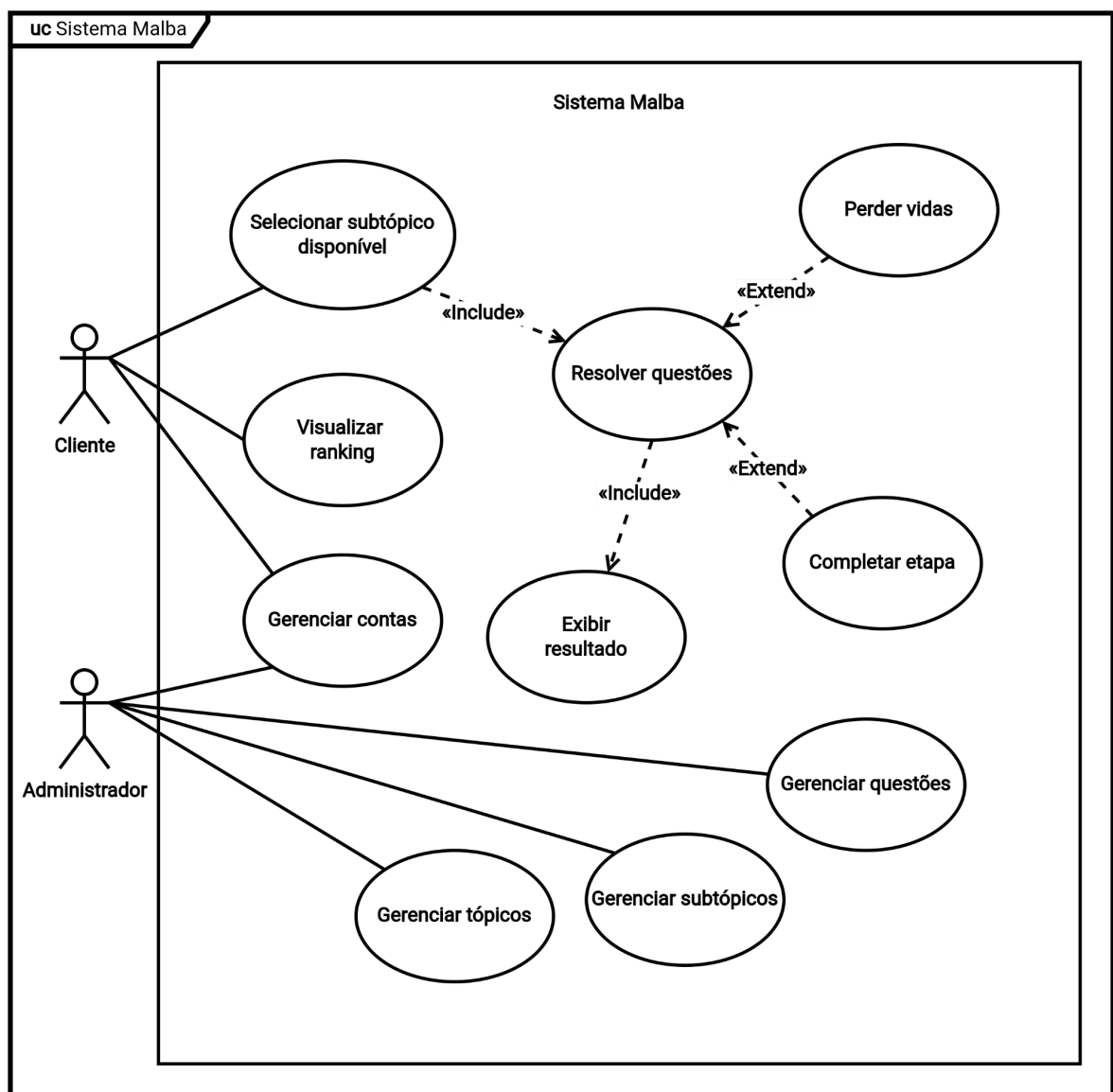


Figura 1: Diagrama de Casos de Uso

3.1.1 Identificação dos requisitos

Para facilitar a identificação dos casos de uso junto aos atores e contextos, abaixo está uma tabela contendo a relação entre eles. A sigla dos requisitos refere-se a [subseção 2.5.1](#).

Sigla do Requisito	Identificação do Ator	Caso de Uso
RF01	Cliente, Administrador	UC01
RF02	Cliente	UC02
RF03	Cliente	UC03
RF04	Cliente	UC04
RF05	Cliente	UC05
RF06	Cliente	UC06
RF07	Cliente	UC07
RF08	Administrador	UC08
RF09	Administrador	UC09
RF10	Administrador	UC10

Tabela 4: Identificação dos requisitos

3.2 Diagrama de Classes

O principal enfoque do diagrama de classes está em permitir a visualização das classes que compõem o sistema com seus respectivos atributos e métodos, bem como demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si. Esse diagrama apresenta uma visão estática de como as classes estão organizadas. (Guedes, 2009)

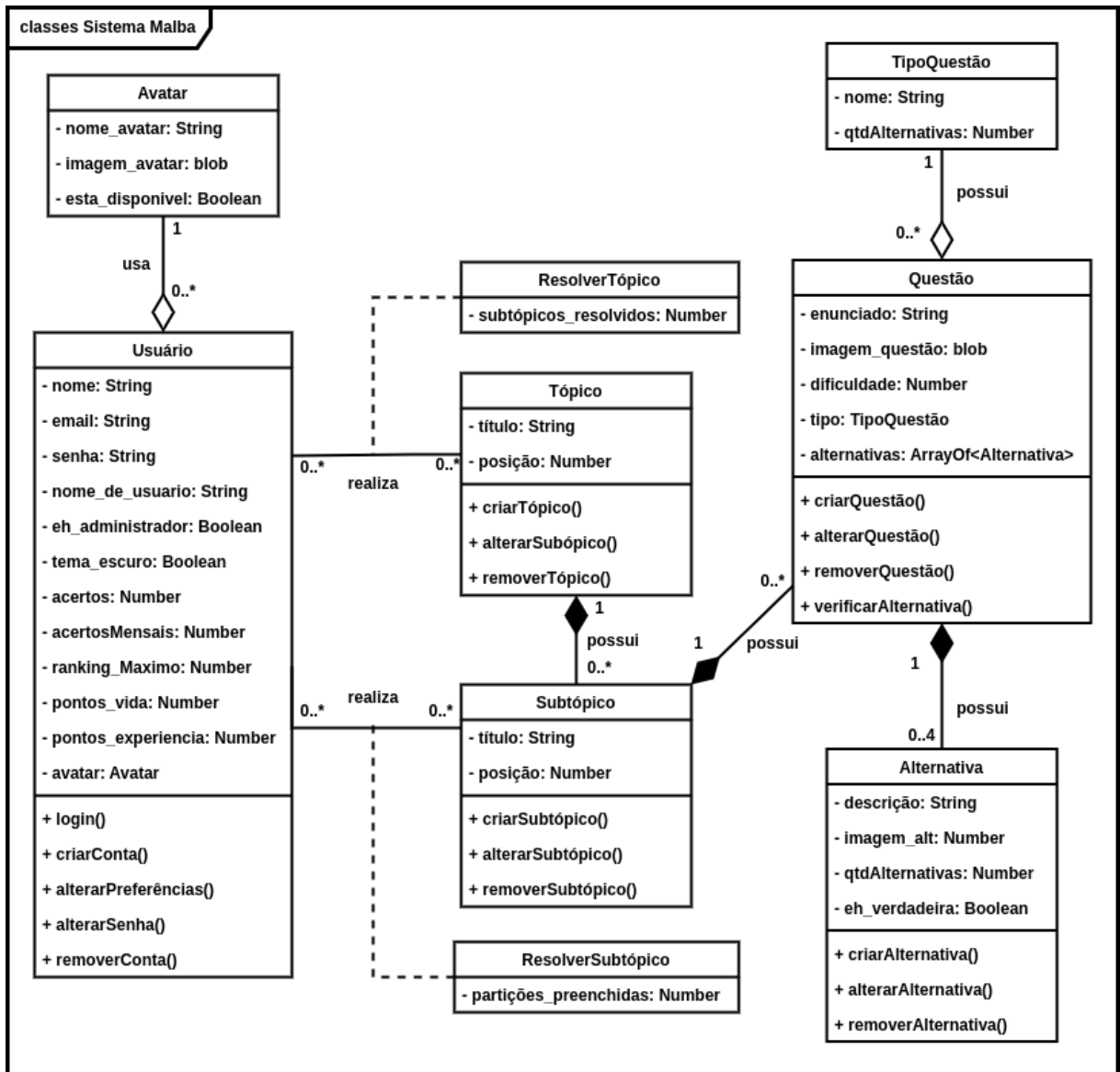


Figura 2: Diagrama de Classes

3.3 Diagrama de Componentes

O diagrama de componentes, como seu próprio nome indica, identifica os componentes que fazem parte de um sistema, um subsistema ou mesmo os componentes ou classes internas de um componente individual. (Guedes, 2009)

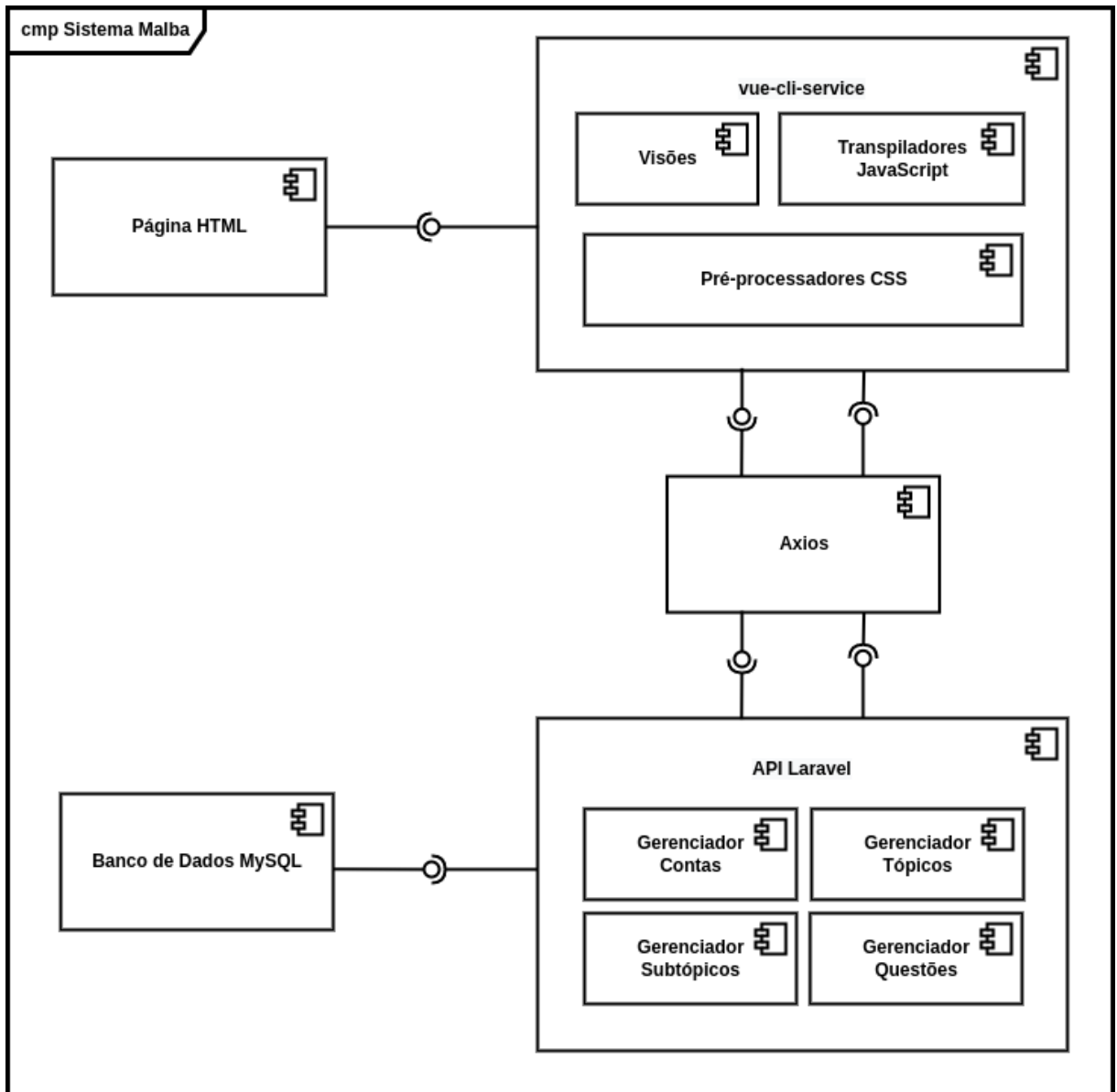


Figura 3: Diagrama de Componentes

3.4 Diagrama de Implantação

O diagrama de implantação determina as necessidades de hardware do sistema, como servidores, estações, topologias e protocolos de comunicação, ou seja, todo o aparato físico sobre o qual o sistema deverá ser executado. Esse diagrama permite demonstrar também como se dará a distribuição dos módulos do sistema, em situações em que estes forem executados em mais de um servidor. A figura 1.11 apresenta um exemplo desse diagrama. (Guedes, 2009)

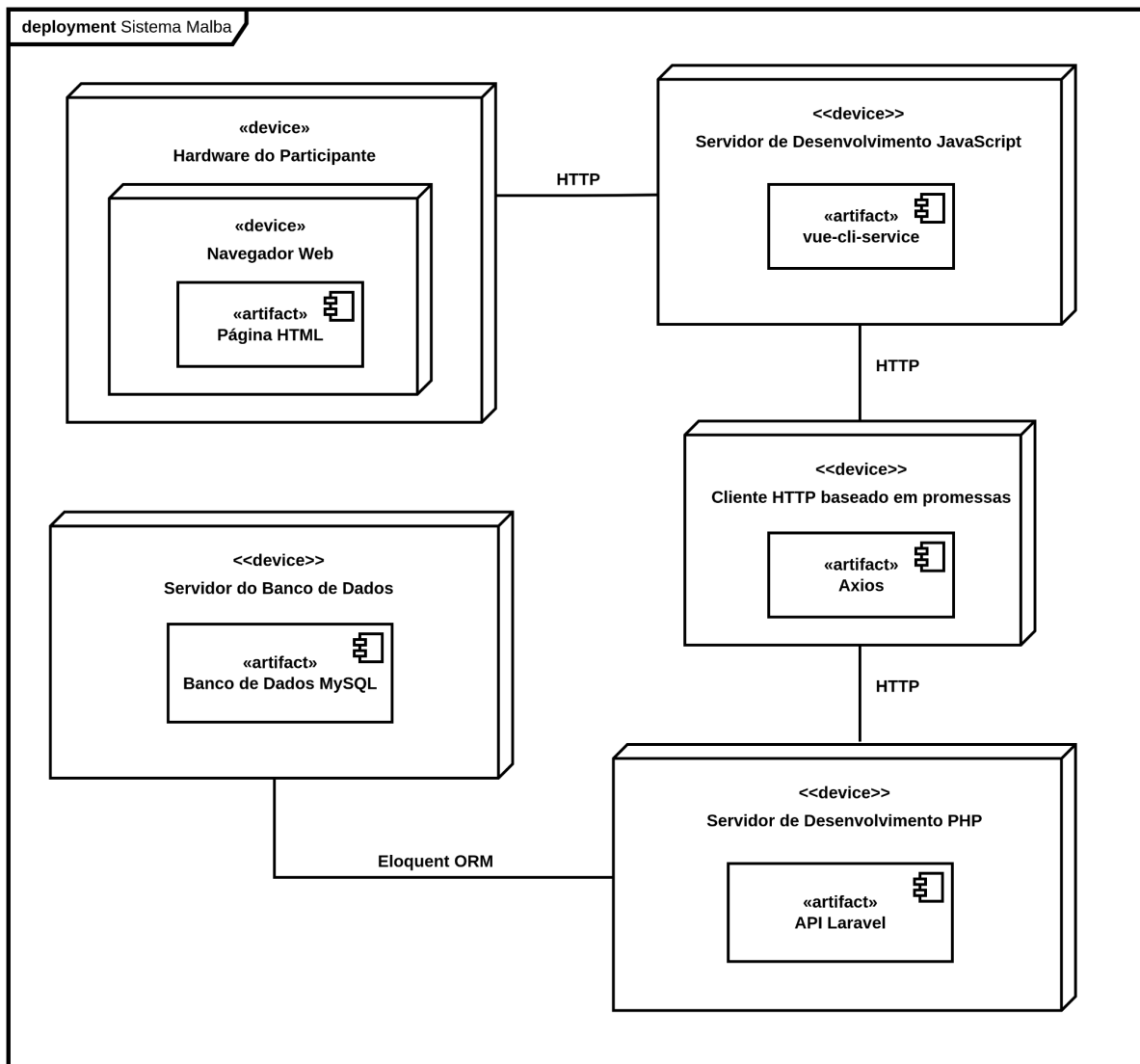


Figura 4: Diagrama de Implantação

4 MODELAGEM E PROJETO DO BANCO DE DADOS

Nesta seção apresenta-se o modelo conceitual do banco de dados do Malba, mostrando de forma gráfica as entidades, relacionamentos e cardinalidades presentes nele.

Cada usuário tem um nome, e-mail, senha, nome de usuário, uma condição que verifica se ele é administrador, uma condição que verifica qual tema ele prefere. Pode ter dados do jogo, caso não seja administrador, como: acertos totais, acertos mensais, ranking máximo obtido, pontos de vida, pontos de experiência. O usuário também escolhe um avatar dentro da lista de avatares disponíveis.

Um avatar possui uma imagem, um título e uma condição que verifica se ele está disponível no momento ou não.

Além disso, cada usuário pode completar tópicos e subtópicos. Cada tópico tem um título e uma posição (quanto menor, mais simples, quanto maior, mais exige conhecimentos prévios). Cada subtópico possui um título e uma posição, igual ao tópico. Um tópico pode ter vários subtópicos e cada subtópico tem um tópico relacionado.

Cada questão tem um enunciado, uma imagem, uma dificuldade (1 para fácil, 2 para média e 3 para difícil). Um subtópico tem várias questões e cada questão tem um sub tópico relacionado. As questões também possuem um tipo e alternativas. O tipo da questão tem um nome e a quantidade de alternativas que ele suporta. Por exemplo, o tipo "múltipla escolha" suporta quatro alternativas, enquanto o tipo "dissertativa" não suporta nenhuma (apesar de ter uma alternativa cadastrada para validar se o que o usuário digitou está correto). Cada alternativa pode ter uma descrição, uma imagem e a condição que verifica se ela é verdadeira. Questões podem ter uma alternativa verdadeira ou várias.

Abaixo, estão representados de forma gráfica os modelos conceitual, lógico e físico, respectivamente.

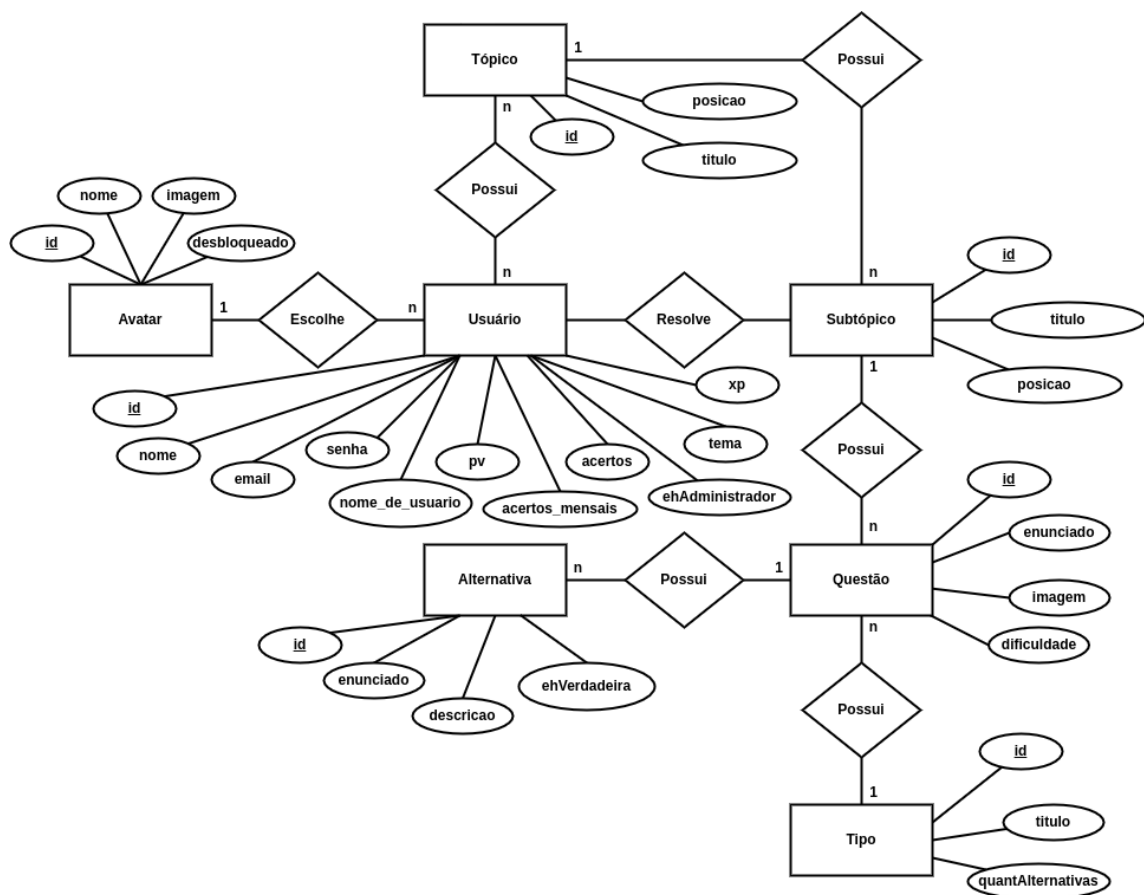


Figura 5: Modelo Conceitual

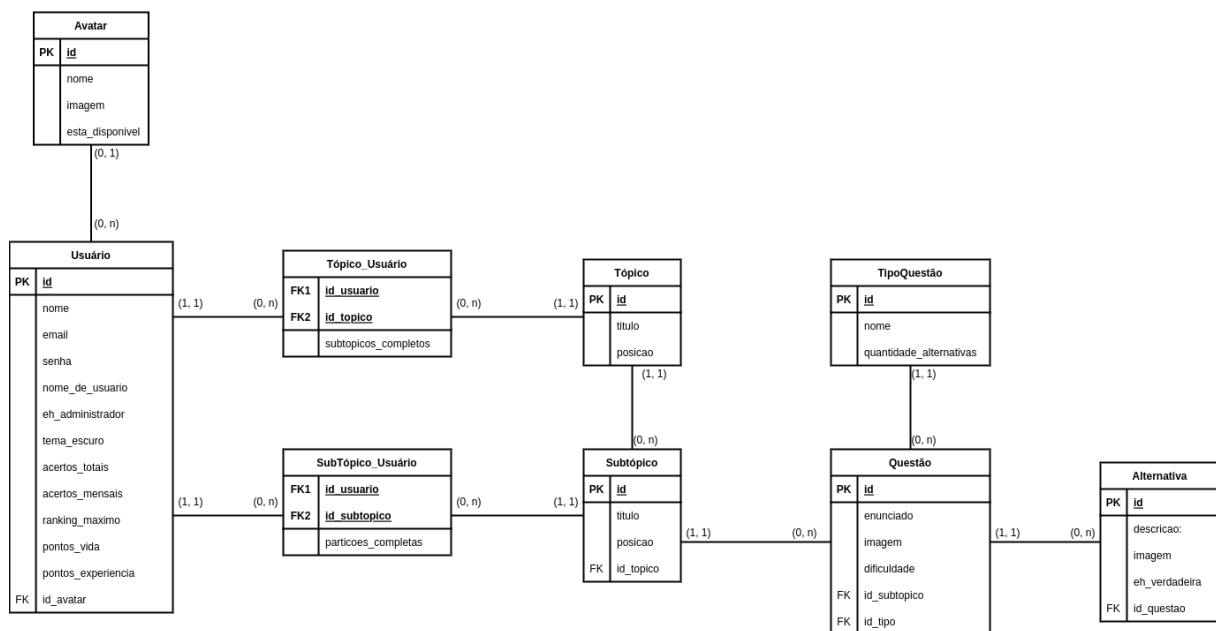


Figura 6: Modelo Lógico

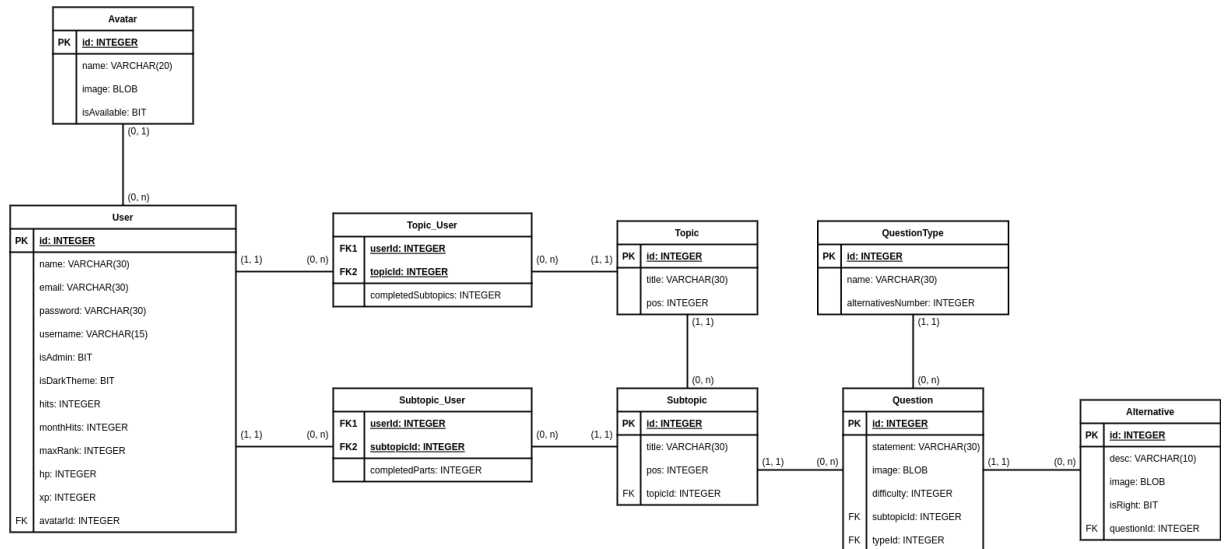


Figura 7: Modelo Físico

O sistema foi desenvolvido usando Laravel, que possui uma ORM que possibilita que sejam usados comandos do próprio PHP ao invés de SQL.

Abaixo, algumas conversões dos comandos usados no sistema.

Criação de Tabelas

```

CREATE TABLE avatars (
    id INT NOT NULL AUTO_INCREMENT,
    name TEXT NOT NULL,
    imageUrl TEXT NOT NULL,
    isAvailable TINYINT(1),
    PRIMARY KEY (id)
);
  
```

```

CREATE TABLE users (
    id INT NOT NULL AUTO_INCREMENT,
    name VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL,
    username VARCHAR(255),
    isAdmin TINYINT(1) NOT NULL,
    isDarkTheme TINYINT(1) NOT NULL,
    hits INT,
  
```



```
    monthHits INT,  
    maxRank INT,  
    xp INT,  
    hp INT,  
    avatar INT NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (avatar) REFERENCES avatars(id)  
);
```

```
CREATE TABLE topics (  
    id INT NOT NULL AUTO_INCREMENT,  
    title TEXT NOT NULL,  
    pos INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE sub_topics (  
    id INT NOT NULL AUTO_INCREMENT,  
    title TEXT NOT NULL,  
    pos INT NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (topic) REFERENCES topics(id)  
);
```

```
CREATE TABLE types (  
    id INT NOT NULL AUTO_INCRMEENT,  
    name TEXT NOT NULL,  
    alternativesNumber INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

```
CREATE TABLE questions (  
    id INT NOT NULL AUTO_INCREMENT,  
    statement TEXT NOT NULL,
```

```
    image BLOB,  
    difficulty INT NOT NULL,  
    type INT NOT NULL,  
    sub_topic INT NOT NULL,  
    PRIMARY KEY (id),  
    FOREIGN KEY (type) REFERENCES types(id),  
    FOREIGN KEY (sub_topic) REFERENCES sub_topics(id)  
);
```

```
CREATE TABLE alternatives (  
    id INT NOT NULL AUTO_INCREMENT,  
    name TEXT NOT NULL,  
    image BLOB,  
    isRight TINYINT(1) NOT NULL,  
    PRIMARY KEY (id)  
);
```

Comandos usados nas rotas da API, de acordo com o método HTTP

GET /api/types

```
SELECT * FROM types;
```

GET /api/types/:id

```
SELECT * FROM types WHERE id = (:id);
```

GET /api/avatars

```
SELECT * FROM avatars;
```

GET /api/avatars/:id

```
SELECT * FROM avatars WHERE id = (:id);
```

GET /api/topics

```
SELECT * FROM topics ORDER BY hits ASC;
```

SELECT title, pos FROM sub_topics WHERE topic = (:topic.id) -- *foi adicionado através do comando for em PHP*

GET /api/topics/:id

SELECT * FROM topics WHERE id = (:id);

POST /api/topics

INSERT INTO topics (title, pos) VALUES (:request.title, :request.pos);

PUT /api/topics/:id

SELECT * FROM topics WHERE id = (:id); – *verifica se existe*

UPDATE topics SET title = (:request.title); – *insere separado pois verifica se o valor da requisição é nulo no PHP*

UPDATE topics SET pos = (:request.pos);

DELETE /api/topics/:id

SELECT * FROM topics WHERE id = (:id); – *verifica se existe*

DELETE FROM topics WHERE id = (:id);

GET /api/subtopics

SELECT * FROM sub_topics;

GET /api/subtopics/:id

SELECT * FROM sub_topics WHERE id = (:id);

POST /api/subtopics

SELECT * FROM topics WHERE topic = (:request.topic); – *verifica se o tópico existe*

INSERT INTO sub_topics (title, pos, topic) VALUES (:request.title, :request.pos, :request.topic);

PUT /api/subtopics/:id

SELECT * FROM sub_topics WHERE id = (:id); – *verifica se existe*

UPDATE sub_topics SET title = (:request.title); – *insere separado pois verifica se o valor da requisição é nulo no PHP*

```
UPDATE sub_topics SET pos = (:request.pos);  
UPDATE sub_topics SET topic = (:request.topic);
```

DELETE /api/subtopics/:id

```
SELECT * FROM sub_topics WHERE id = (:id); – verifica se esiste  
DELETE FROM sub_topics WHERE id = (:id);
```

GET /api/users

```
SELECT id, name, email, username, isAdmin, isDarkTheme, hits, monthHits,  
maxRank, xp, hp, avatar FROM users WHERE isAdmin = 0;
```

GET /api/users/admin

```
SELECT id, name, email, username, isAdmin, isDarkTheme, hits, monthHits,  
maxRank, xp, hp, avatar FROM users WHERE isAdmin = 1;
```

GET /api/users/hits/asc

```
SELECT id, name, email, username, isAdmin, isDarkTheme, hits, monthHits,  
maxRank, xp, hp, avatar FROM users WHERE isAdmin = 0 ORDER BY hits ASC;
```

GET /api/users/hits/desc

```
SELECT id, name, email, username, isAdmin, isDarkTheme, hits, monthHits,  
maxRank, xp, hp, avatar FROM users WHERE isAdmin = 0 ORDER BY hits DESC;
```

GET /api/users/:id

```
SELECT id, name, email, username, isAdmin, isDarkTheme, hits, monthHits,  
maxRank, xp, hp, avatar FROM users WHERE id = (:id);
```

POST /api/register

```
INSERT INTO users (name, email, password, isAdmin, isDarkTheme, avatar)  
VALUES (:request.name, :request.email, :request.password, 0, 0, 1);
```

POST /api/login

```
SELECT * FROM users WHERE email = (:request.email) AND password =  
(:request.password);
```

PUT /api/users/:id

SELECT * FROM users WHERE id = (:id); – *verifica se existe*

UPDATE user SET name = (:request.name); – *insere separado pois verifica se o valor da requisição é nulo no PHP*

UPDATE users SET username = (:request.username);

UPDATE users SET isDarkTheme = (:request.isDarkTheme);

UPDATE users SET avatar = (:request.avatar);

UPDATE users SET hits = (:request.hits);

UPDATE users SET monthHits = (:request.monthHits);

UPDATE users SET maxRank = (:request.maxRank);

UPDATE users SET hp = (:request.hp);

UPDATE users SET xp = (:request.xp);

DELETE /api/users/:id

SELECT * FROM topics WHERE id = (:id); – *verifica se existe*

DELETE FROM topics WHERE id = (:id);

5 MANUAL DO SISTEMA

A seguir estão a descrição e imagem de cada tela do sistema, seguida da explicação de cada campo.

- **Tela inicial**

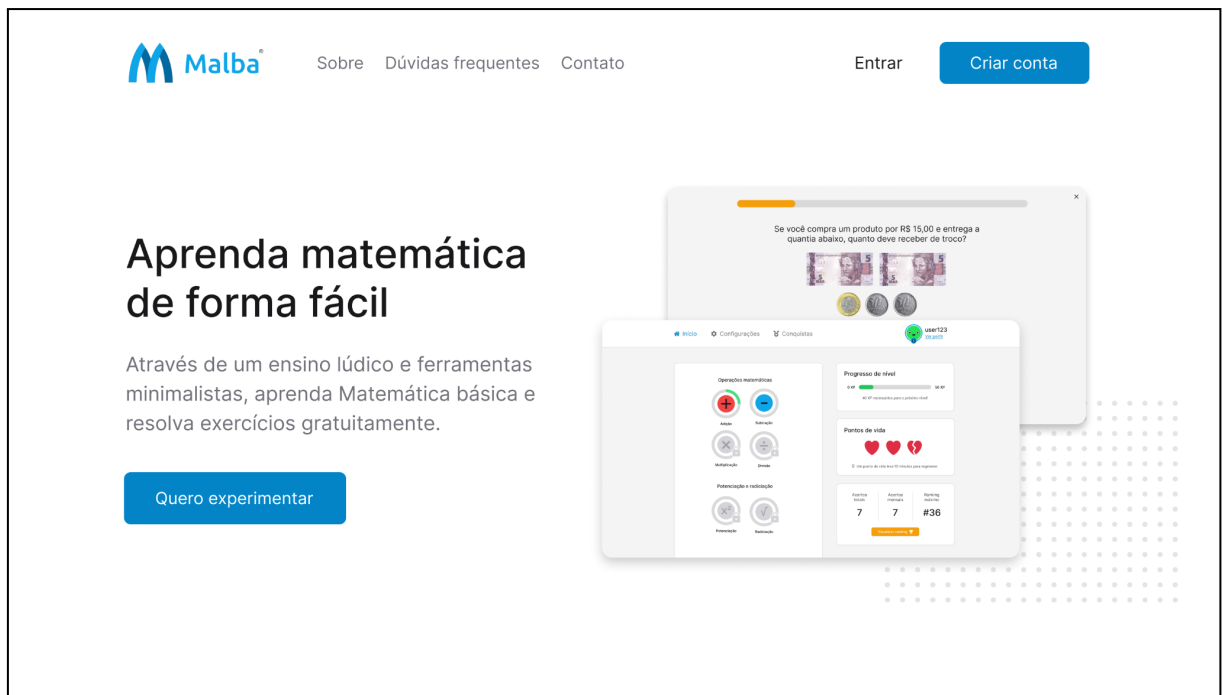


Figura 8: Tela Inicial

A primeira tela do sistema contém botões do tipo *call-to-action*, estimulando-o a criar uma conta ou entrar no sistema. Também há a exposição da identidade visual e algumas telas que mostram o funcionamento do sistema.

- **Tela de Cadastro**

←

Criar conta

Registre-se agora mesmo com seu e-mail.

Nome

Rafael Juliani Diehl

E-mail

Informe seu e-mail

Senha

Crie uma senha

Senha

Digite sua senha novamente

☐ Li e concordo com os [Termos de Uso](#) e a [Política de Privacidade](#).

Criar conta

Já tem uma conta? [Faça login](#).

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip.

Figura 9: Tela de Cadastro

A tela de cadastro contém os campos nome, e-mail, senha (que deve ser validada duas vezes por fins de segurança) e um aviso para o usuário ler os termos de uso e política de privacidade. Além disso, há um carroussel com informações sobre o sistema, ao lado, a fim de captar a atenção do usuário e tornar a página visualmente mais agradável e um link abaixo do formulário para redirecionar o usuário para a tela de login, caso ele já possua uma conta.

←

Criar conta

Registre-se agora mesmo com seu e-mail.

Nome

Rafael Juliani Diehl

E-mail

Informe seu e-mail

Senha

Crie uma senha

Senha

Digite sua senha novamente

☐ Li e concordo com os [Termos de Uso](#) e a [Política de Privacidade](#).

Criar conta

Já tem uma conta? [Faça login](#).

Link de verificação enviado

Um link para verificação da conta foi enviado para seu e-mail.
Caso não o encontre, verifique a caixa de spam.

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip.

Figura 10: Modal de confirmação

Após realizar a conta, é exibido um modal que pede para que o usuário vá até seu e-mail para receber um link de verificação e dar seus próximos passos.

- **Tela de Login**

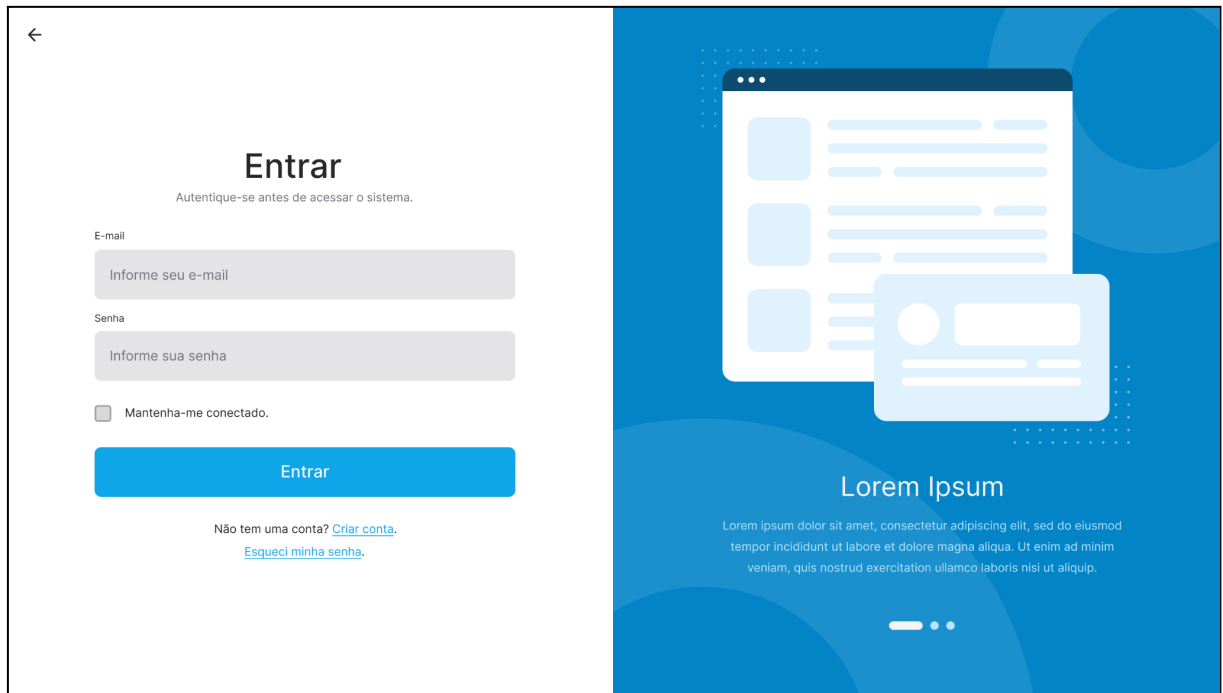
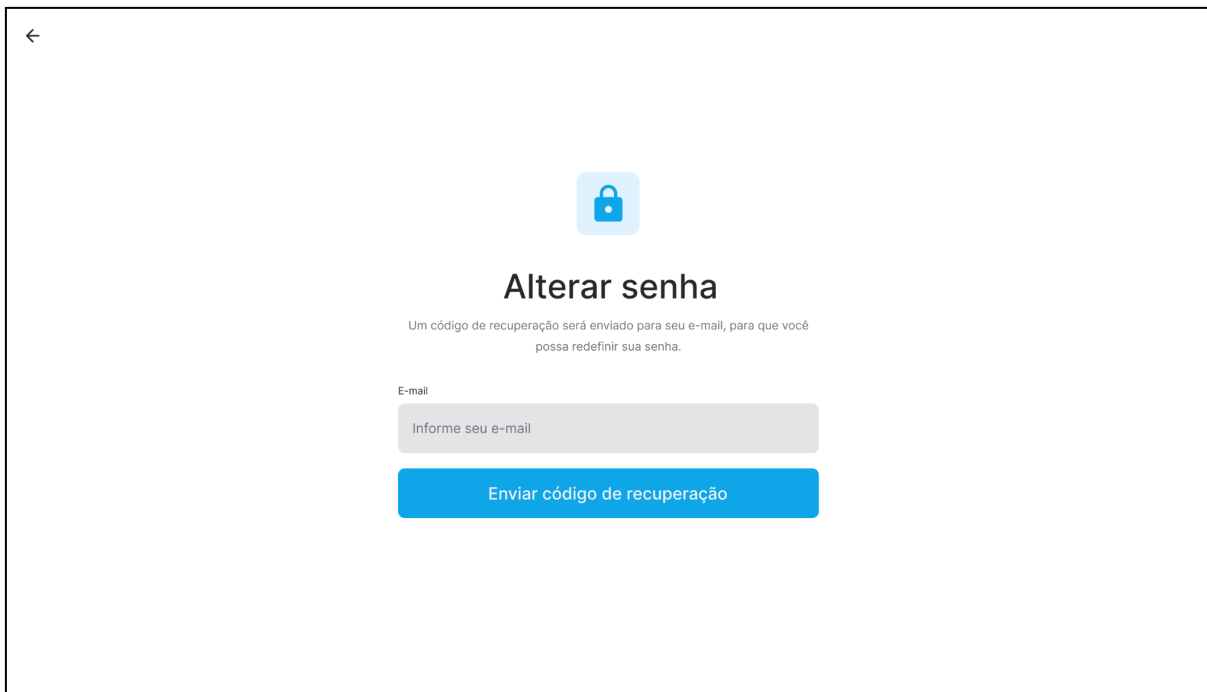



Figura 11: Tela de Login

A tela de login é extremamente semelhante a tela de cadastro, porém tem seus campos alterados. Para entrar, o usuário precisa informar seu e-mail e senha, e tem a opção de manter-se conectado. Também há links de redirecionamento para a tela de cadastro, caso o usuário não possua uma conta, e para a tela de recuperação de senha, caso ele tenha esquecido sua senha.

- **Tela de Recuperação de Senha**



←



Alterar senha

Um código de recuperação será enviado para seu e-mail, para que você possa redefinir sua senha.

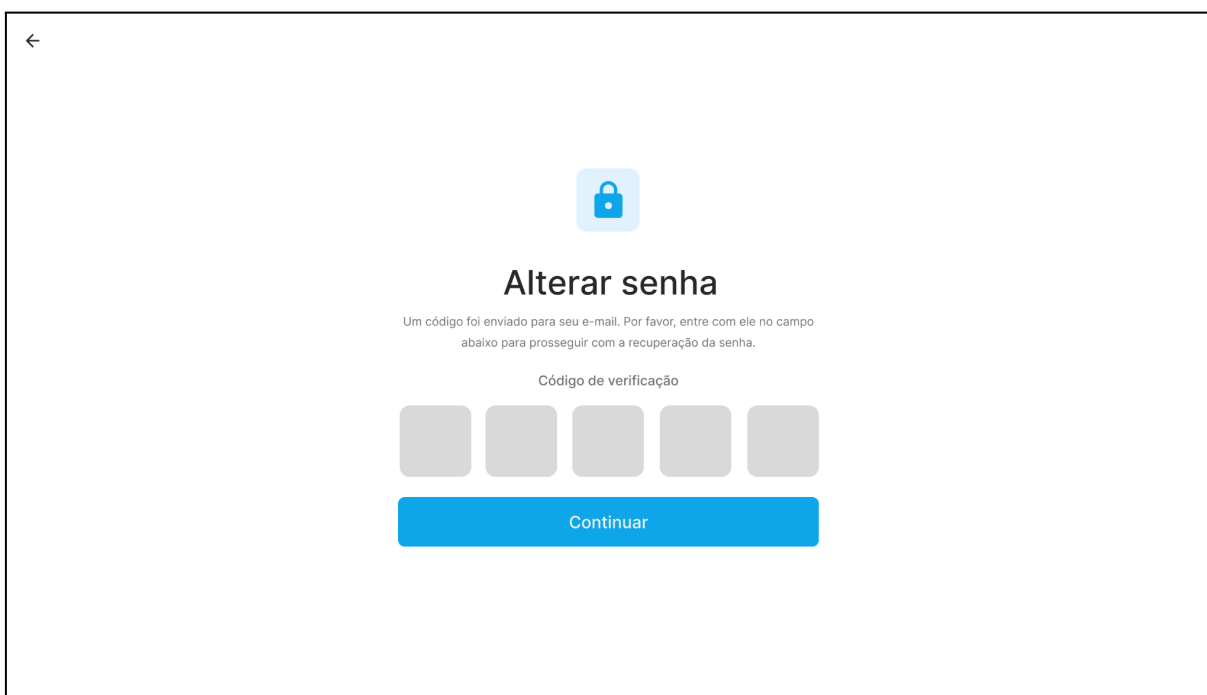
E-mail

Enviar código de recuperação


Figura 12: Tela de Recuperação de Senha

A tela de recuperação de senha contém um único campo, para o usuário preencher seu e-mail. Lá, ele deverá receber um código inicial e inseri-lo na página seguinte.

- **Tela de Recuperação de Senha (inserir código)**



←



Alterar senha

Um código foi enviado para seu e-mail. Por favor, entre com ele no campo abaixo para prosseguir com a recuperação da senha.

Código de verificação

Continuar

Figura 13: Tela de Recuperação de Senha (inserir código)

Após receber o e-mail, o usuário deve digitar o código e redefinir sua senha.

- Tela de “primeiros passos”

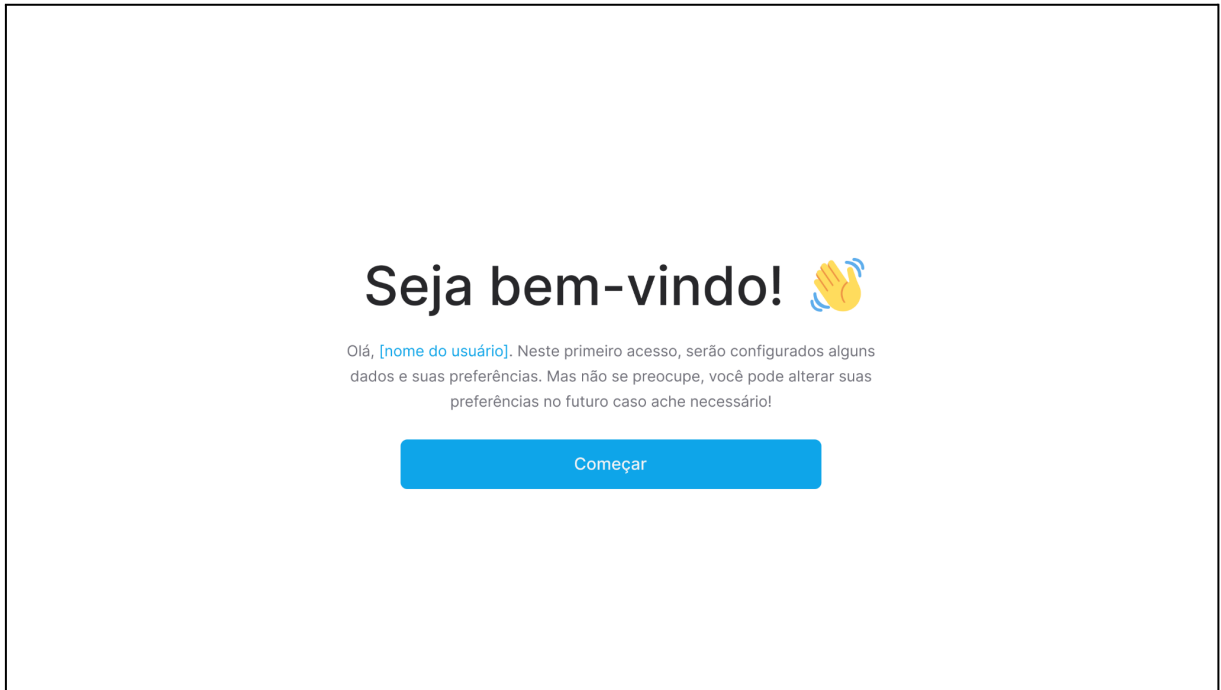


Figura 14: Tela de Primeiros Passos

Na primeira vez que o usuário faz login, ele é redirecionado para uma tela amigável e minimalista para configurar suas preferências, como: nome de usuário, tema de cor e avatar. Em seguida, ele deve confirmar as preferências ou redefini-las até ficar satisfeito.

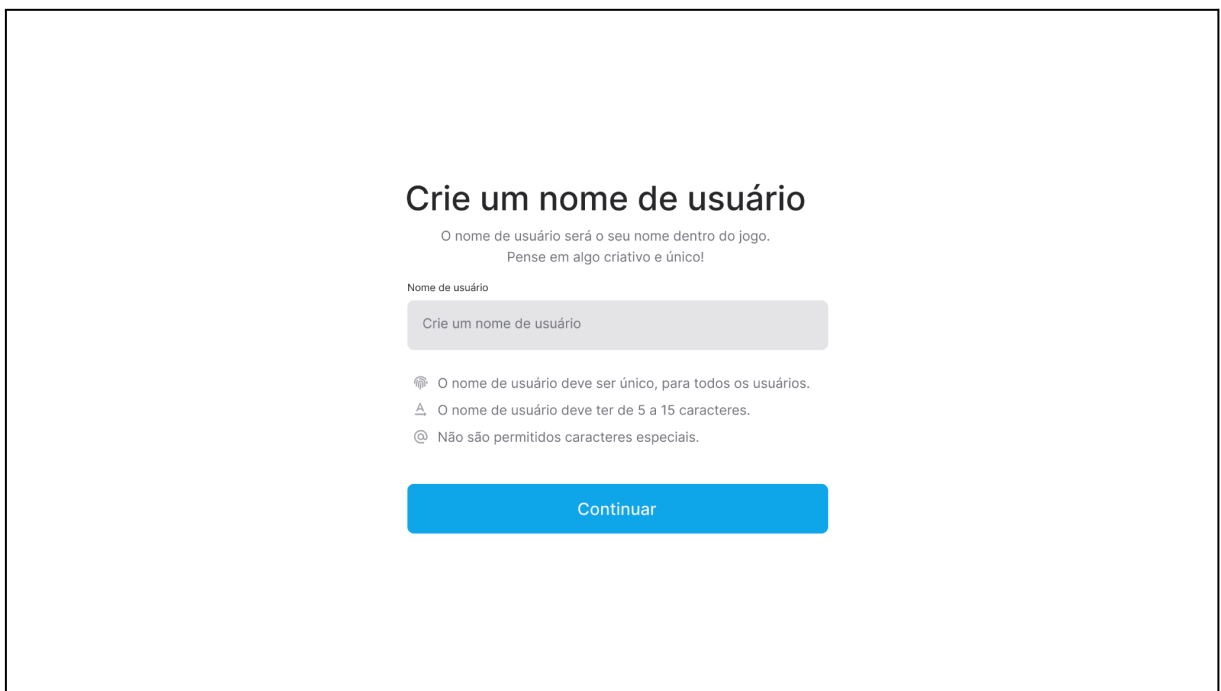



Figura 15: Tela de Primeiros Passos (nome de usuário)


Crie um nome de usuário


O nome de usuário será o seu nome dentro do jogo.
Pense em algo criativo e único!

Nome de usuário

raf@el123

 O nome de usuário deve ser único, para todos os usuários.

 O nome de usuário deve ter de 5 a 15 caracteres.

 Não são permitidos caracteres especiais.

Continuar


Figura 16: Tela de Primeiros Passos (nome de usuário, validação do campo)

Selecione o tema de cor

Clique nos botões abaixo para selecionar a sua preferência de cores do sistema. Quando estiver decidido, clique em continuar.

Aa


Lorem ipsum dolor



Tema claro

Aa

Lorem ipsum dolor



Tema escuro

Continuar

Figura 17: Tela de Primeiros Passos (tema)

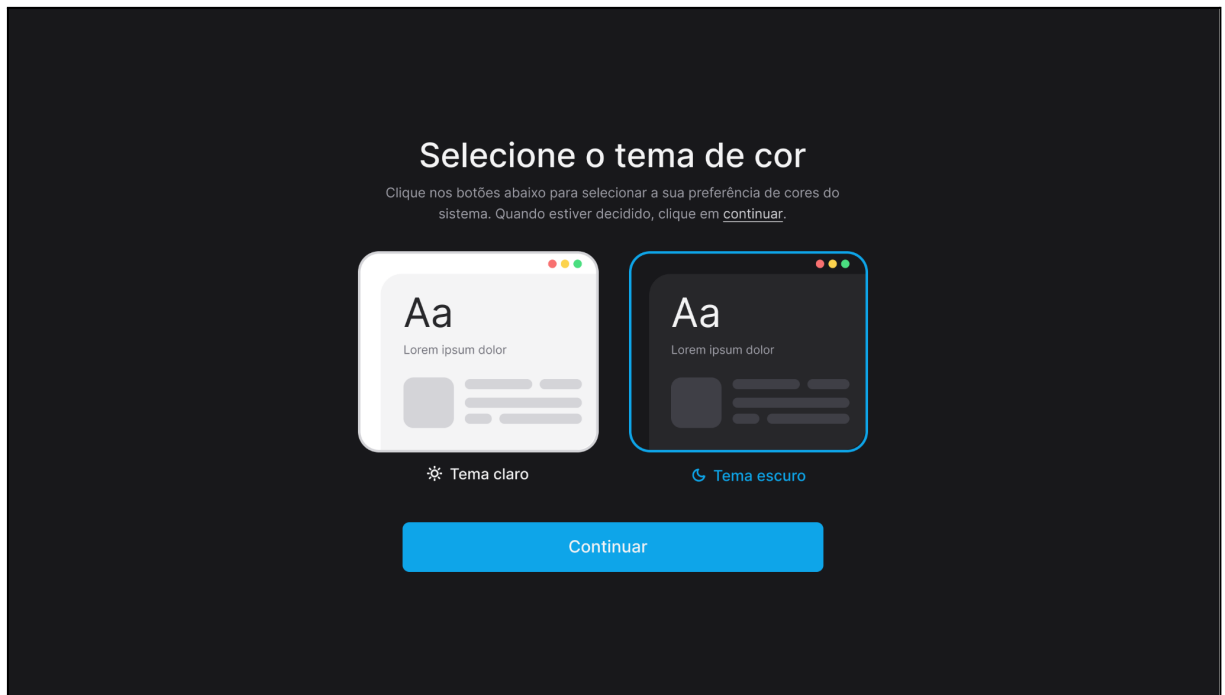


Figura 18: Tela de Primeiros Passos (tema escuro)

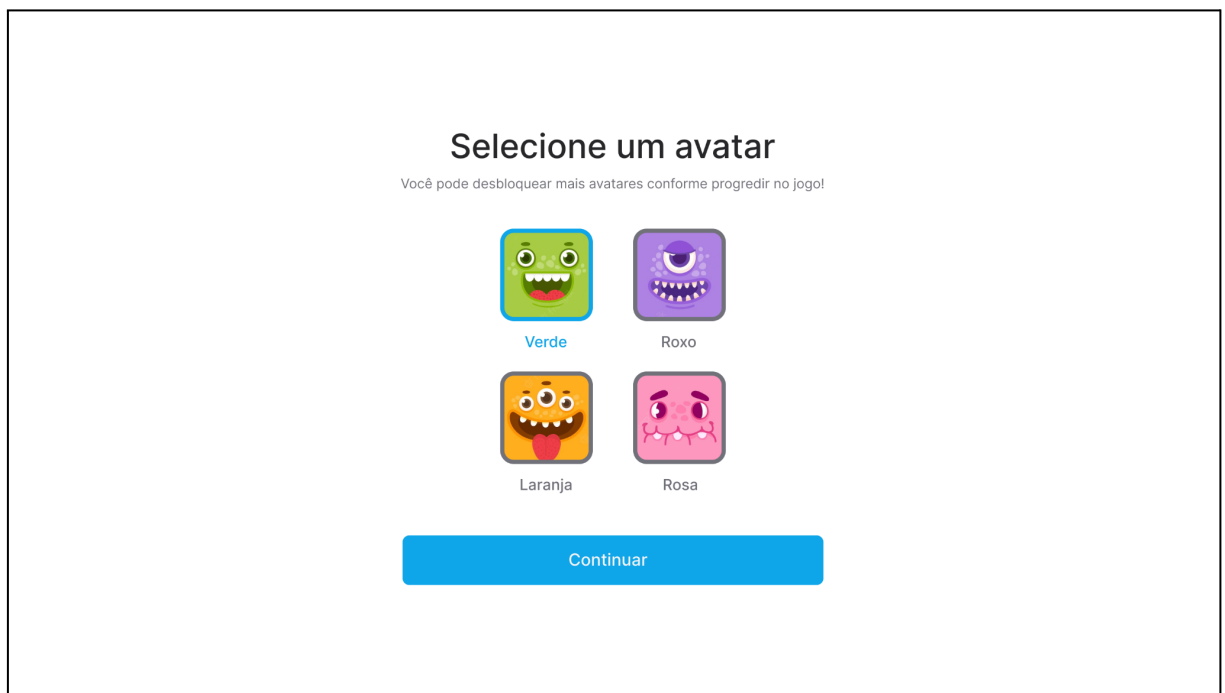


Figura 19: Tela de Primeiros Passos (avatar)

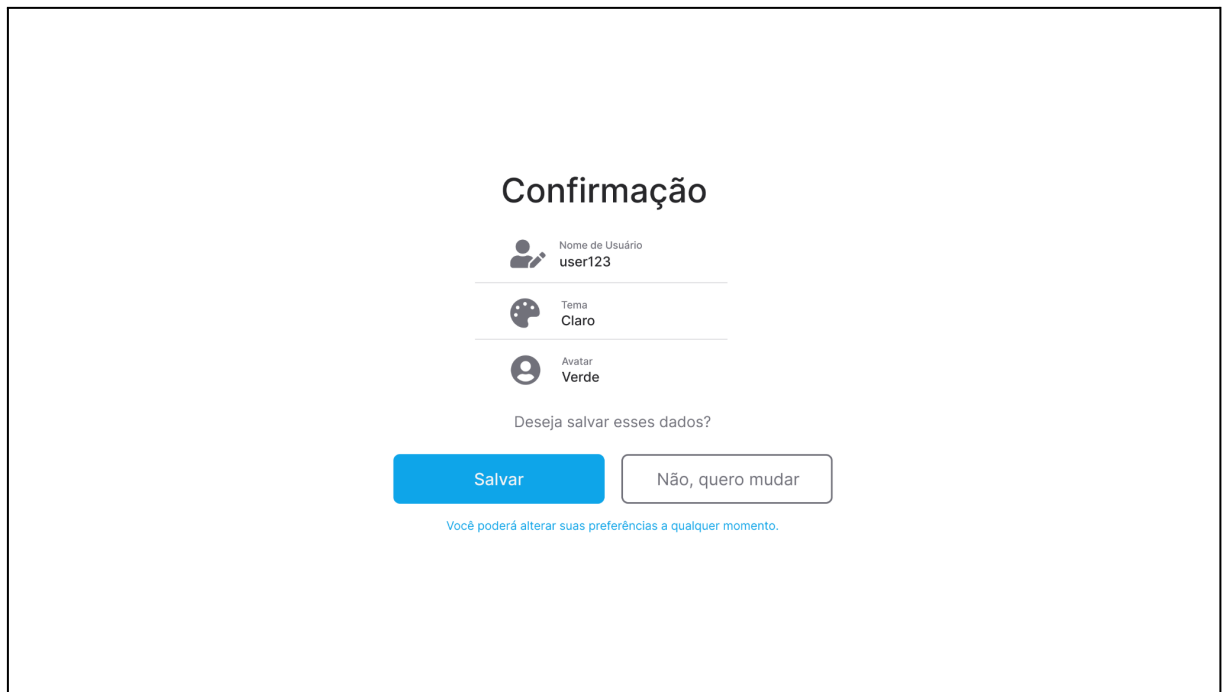


Figura 20: Tela de Primeiros Passos (confirmação)

- **Dashboard usuário**

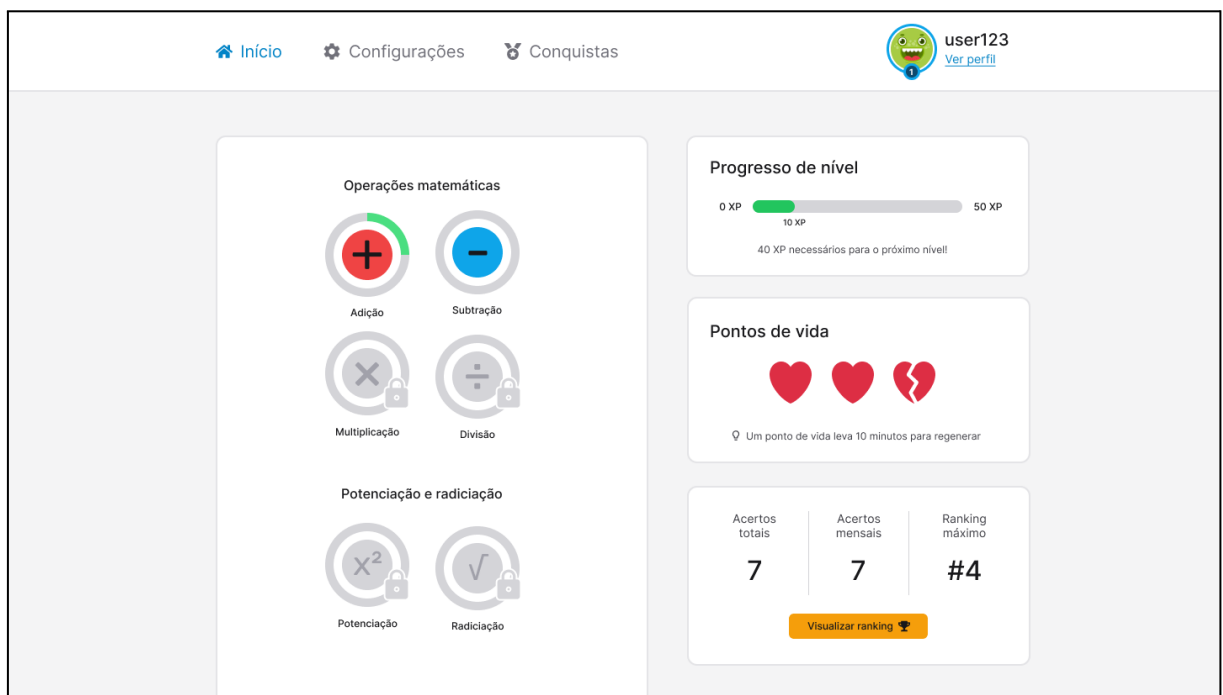


Figura 21: Dashboard usuário (tela inicial)

Na tela de dashboard, o usuário tem uma visão geral de suas informações (como progresso de nível, pontos de vida e pontuações), um painel com os subtópicos disponíveis para selecionar e links para suas configurações, conquistas e visualização de perfil de perfil.

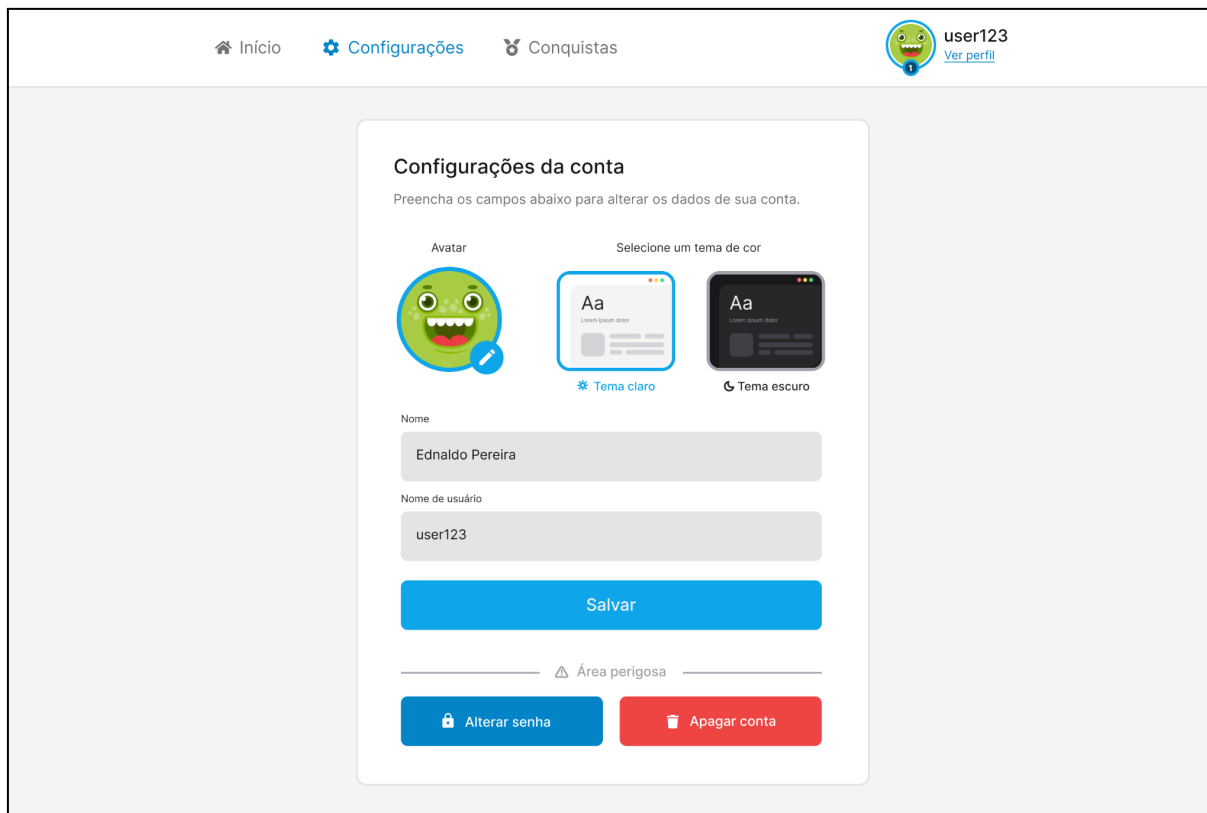


Figura 22: Dashboard usuário (configurações da conta)

Nas configurações da conta, o usuário pode alterar seu nome, nome de usuário, tema e avatar. Dados mais importantes como a senha, e a exclusão da conta, são reservados a uma sessão separada, onde ele é redirecionado para a tela de recuperação de senha e um modal de exclusão, respectivamente.

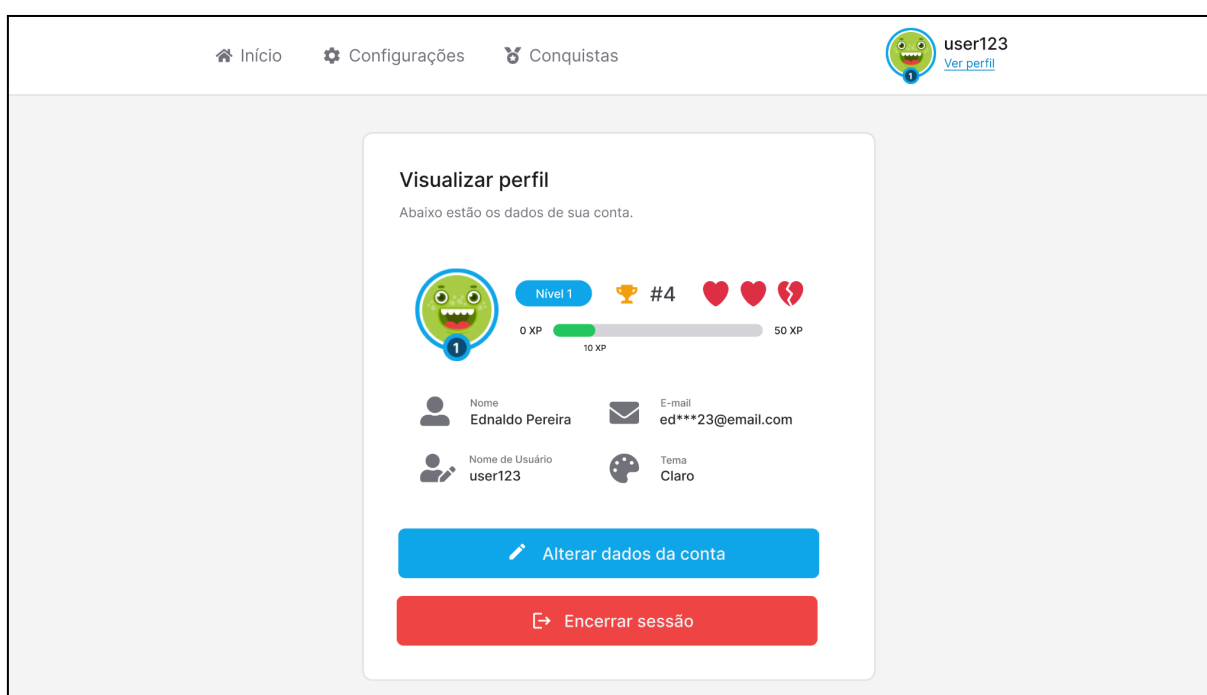


Figura 23: Dashboard usuário (ver perfil)

Nesta tela, o usuário pode visualizar seu perfil público e tem acesso a links para alterar seus dados e encerrar sessão.



Figura 24: Dashboard do usuário (ranking)

Nesta tela, o usuário pode visualizar um ranking com as maiores pontuações e seus respectivos usuários.

- **Tela de Questões**

Se você compra um produto por R\$ 15,00 e entrega a quantia abaixo, quanto deve receber de troco?

Two 5 Real banknotes and three 1 Real coins are shown as payment.

Options for the amount paid:

- R\$ 2,50
- R\$ 3,00** (selected)
- R\$ 3,50
- R\$ 4,00

Continuar

Figura 25: Tela de Questões

Ao clicar num subtópico, o usuário é redirecionado para uma série de questões, com dez perguntas. Para cada questão, ele pode selecionar um input, e, ao final, ele tem exibido seu resultado na tela.

- **Dashboard administrador**

Navigation links: [Início](#), [Tópicos](#), [Subtópicos](#), [Questões](#)

User profile: [admin](#) [Editar perfil](#)

Painel do administrador

Clique nos botões abaixo para gerenciar as tabelas.

- Gerenciar tópicos
- Gerenciar subtópicos
- Gerenciar questões

Figura 26: Dashboard administrador (tela inicial)

O usuário do tipo administrador tem acesso a um dashboard especial, onde ele pode exercer suas funções gerenciais através de botões.

The screenshot displays the 'Configurações da conta' (Account Settings) page for an administrator. The page features a navigation bar at the top with links to 'Início', 'Tópicos', 'Subtópicos', and 'Questões', along with a user profile icon and the name 'admin' with a link to 'Editar perfil'. The main content area is titled 'Configurações da conta' and includes a sub-header 'Preencha os campos abaixo para alterar os dados de sua conta.' Below this, there are sections for 'Avatar' (showing a purple alien head icon), 'Seleção um tema de cor' (with options for 'Tema claro' and 'Tema escuro'), and input fields for 'Nome' (containing 'Administrador') and 'Nome de usuário' (containing 'admin'). A large blue 'Salvar' button is positioned at the bottom of the form.

Figura 27: Dashboard administrador (configurações)

Assim como o cliente, o administrador pode alterar suas preferências, mas funções mais avançadas não são permitidas pelo navegador, cabendo direto ao administrador do banco de dados.

The screenshot displays the 'Gerenciamento de tópicos' (Topic Management) page for an administrator. The page features the same navigation bar as Figure 27. The main content area is titled 'Gerenciamento de tópicos' and includes a sub-header 'Preencha os campos abaixo para alterar os dados de sua conta.' Below this, there is a table with columns 'Tópico' and 'Posição'. The table lists four topics: 'Operações básicas' (position 1), 'Potenciação e radiação' (position 2), 'Expressões numéricas' (position 3), and 'Polinômios' (position 5). Each row has edit and delete icons. Below the table, there are input fields for 'Nome' (containing 'Notação científica') and 'Posição' (containing '7'), followed by a green '+' button to add a new topic.

Tópico	Posição
Operações básicas	1
Potenciação e radiação	2
Expressões numéricas	3
Polinômios	5

Nome: Notação científica Posição: 7

Figura 28: Dashboard administrador (gerenciamento)

Ao clicar num botão de gerenciamento, o administrador é redirecionado a uma tela com vários campos onde pode fazer operações *CRUD* (create, read, update e delete).

APÊNDICE A: Protótipo do Sistema

O protótipo pode ser acessado publicamente através do link abaixo:
<https://www.figma.com/file/FMZNLPG7E6dZjupipOozBw/Malba?node-id=0%3A1>

Para navegar entre as seções do design, basta clicar no ícone do Figma no canto superior esquerdo e selecionar a opção desejada.

Também há um protótipo executável, bastando clicar no botão de “play”, no canto superior direito. Alguns recursos como modelos de componentes e visualização de componentes em determinados eventos, estão indisponíveis na execução do protótipo, servindo apenas como base para o desenvolvedor seguir fielmente ao protótipo.

REFERÊNCIAS

GUEDES, G. T. A. **UML 2: Uma abordagem prática**. São Paulo: Novatec Editora, 2009. p 76.

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO CAMPUS PRESIDENTE EPITÁCIO. **Regulamento do Trabalho de Conclusão de Curso: Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas**. Disponível em: <http://pep.ifsp.edu.br/wp-content/uploads/2014/10/Regulamento-TCC-ADS.pdf>.

Acesso em: 30 de mai. 2016.

Object Management Group. **OMG Unified Modeling Language (OMG UML): Version 2.5.1**. OMG Unified Modeling Language, 2017. Disponível em: <https://www.omg.org/spec/UML/2.5.1/PDF>. Acesso em: 5 jun. 2022.