**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC DA ZONA LESTE**

**Novotec Desenvolvimento de Sistemas**

**João Enrique Barbosa Santos Alves**

**Júlia da Silva Dias Gonçalves**

**Sarah Jandozza Laurindo**

**SCIENCE AR: Sistema educacional de Ciências da natureza utilizando Realidade Aumentada**

**São Paulo**

**2023**

**João Enrique Barbosa Santos Alves**

**Júlia da Silva Dias Gonçalves**

**Sarah Jandozza Laurindo**

**SCIENCE AR: Sistema educacional de Ciências da natureza utilizando Realidade Aumentada**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Desenvolvimento de Sistemas da Etec Zona Leste, orientado pelo Prof. Esp. Jeferson Roberto de Lima, como requisito parcial para obtenção do título de técnico em Desenvolvimento de Sistemas.

**São Paulo**

**2023**

**DEDICATÓRIA**

Dedicamos esta monografia a todas as pessoas que estiveram presentes durante a realização deste trabalho. Aos meus companheiros de trabalho, enfrentamos desafios ao longo do caminho, mas também compartilhamos momentos alegres. Agradecemos por estarmos juntos, superando as diferenças e apoiando uns aos outros. Nossa união nos tornou mais fortes e motivados.

**AGRADECIMENTOS**

Gostaríamos de expressar nossos sinceros agradecimentos a todas as pessoas que contribuíram para a realização desta monografia.

Aos meus companheiros de trabalho, que estiveram juntos durante todo o processo, enfrentando desafios e celebrando conquistas juntos. Nossa união e apoio mútuo foram fundamentais para o sucesso deste trabalho.

Aos nossos familiares, que sempre estiveram presentes, oferecendo seu apoio incondicional e palavras de incentivo, somos imensamente gratos. Vocês foram nossa fonte de motivação durante toda essa jornada.

Gostaríamos também de expressar nossa profunda gratidão aos nossos orientadores e professores. Sem o seu conhecimento, orientação e dedicação, este trabalho não seria possível. Suas contribuições foram essenciais para o aprimoramento desta monografia.

Que este agradecimento represente o quanto cada um de vocês foi fundamental em nossa jornada. Estamos verdadeiramente honrados e gratos por termos tido a oportunidade de aprender e crescer ao lado de pessoas tão especiais. Muito obrigado a todos!

**EPÍGRAFE**

“Temos que continuar aprendendo. Temos que estar abertos. E temos que estar prontos para espalhar nosso conhecimento a fim de chegar a uma compreensão mais elevada da realidade”.

**Thich Nhat Hanh**

**Resumo**

Um dos problemas do ensino tradicional, é a falta de compreensão do aluno em relação ao aprendizado teórico, onde muitas das vezes não possuem estímulos visuais.  Com base nos dados coletados, foi desenvolvida a ideia de um para a solução dos problemas encontrados. O que pode ser feito para que os alunos aprendam de uma forma mais simples e objetiva?  Tendo em vista deste problema, a plataforma Science AR através da realidade aumentada, busca elevar a taxa de sucesso no aprendizado dos alunos na disciplina e engajar o nível de aprendizado.  A plataforma disponibilizará matérias de apoio para os alunos estudarem e terá questionários sobre o tema estudado, onde o aluno poderá aplicar o seu conhecimento na matéria.

**Palavras-chave:** Estímulo visual. Realidade aumentada. Taxa de sucesso.

**Abstract**

One of the problems with traditional education is the lack of understanding by students in relation to theoretical learning, often lacking visual stimulation. Based on the collected data, an idea was developed to solves these problems. What can be done to make learning simpler and more objective for students? Considering this issue, the Science AR platform, through augmented reality, aims to increase the success rate of student learning in the subject and enhance the level of engagement. The platform will provide support materials for students to study and will have quizzes on the studied topics, allowing students to apply their knowledge in the subject.

**Keywords:** Visual stimulation. Augmented reality. Success rate.

**Lista de ilustrações**

[Figura 1 – Estrutura básica HTML 18](#_Toc143088308)

[Figura 2 - Exemplo de Login com HTML puro. 19](#_Toc143088309)

[Figura 3 - Execução do código em HTML puro. 19](#_Toc143088310)

[Figura 4 – Exemplo de site no computador 20](#_Toc143088311)

[Figura 5 – Linguagens de programação mais populares 21](#_Toc143088312)

[Figura 6 – Exemplo da Função no JavaScript 23](#_Toc143088313)

[Figura 7 – 1ª Execução do Exemplo 23](#_Toc143088314)

[Figura 8 – 2ª Execução do Exemplo 24](#_Toc143088315)

[Figura 9 – Exemplo de variável no PHP 25](#_Toc143088316)

[Figura 10 – Exemplo de constante no PHP 26](#_Toc143088317)

[Figura 11 – Exemplo da estrutura de controle IF no PHP 29](#_Toc143088318)

[Figura 12 – Exemplo da estrutura de controle WHILE no PHP 29](#_Toc143088319)

[Figura 13 – Exemplo de uso da estrutura FOR 30](#_Toc143088320)

[Figura 14 – Exemplo de tabela SQL 32](#_Toc143088321)

[Figura 15 – Exemplo de tabela de clientes no SQL 32](#_Toc143088322)

[Figura 16 – Exemplo de estrutura SQL 33](#_Toc143088323)

[Figura 17 – Exemplo de SQL PHP com MYSQL 34](#_Toc143088324)

[Figura 18 – Estrutura do Laravel 36](#_Toc143088325)

[Figura 19 – Laravel com Banco de Dados 36](#_Toc143088326)

[Figura 20 – Exemplo de Rotas 37](#_Toc143088327)

[Figura 21 – Exemplo de Model 38](#_Toc143088328)

[Figura 22 – Exemplo de View 38](#_Toc143088329)

[Figura 23 – Exemplo de Controller 39](#_Toc143088330)

[Figura 24 – Primeiro monitor usado na cabeça 39](#_Toc143088331)

[Figura 25 – Exemplo de Realidade Aumentada 40](#_Toc143088332)

[Figura 26 – Interface gráfico do Unity3D 41](#_Toc143088333)

[Figura 27 – Exemplo da aplicação de texturas e cores em uma modelagem 3D 42](#_Toc143088334)

[Figura 28 – Exemplo de estrutura do C# 43](#_Toc143088335)

[Figura 29 – Exemplo de estrutura do C# no Unity 44](#_Toc143088336)

[Figura 30 – Indicando o índice para visualizar o nome 46](#_Toc143088337)

[Figura 31 – Alterando o valor de uma chave-valor 46](#_Toc143088338)

[Figura 32 – Diagrama de Caso de Uso 47](#_Toc143088339)

[Figura 33 – Modelo Entidade Relacionamento 57](#_Toc143088340)

[Figura 34 – Diagrama Entidade Relacionamento 58](#_Toc143088341)

[Figura 35– Diagrama de atividade do Realizar Login 59](#_Toc143088342)

[Figura 36 – Diagrama de atividade do Criar Cadastro 60](#_Toc143088343)

[Figura 37 – Diagrama de atividade do Visualizar Cadastro 61](#_Toc143088344)

[Figura 38 – Diagrama de atividade do Editar Cadastro 62](#_Toc143088345)

[Figura 39 – Diagrama de atividade do Deletar Cadastro 63](#_Toc143088346)

[Figura 40 – Diagrama de atividade do Criar Quiz 64](#_Toc143088347)

[Figura 41 – Diagrama de atividade do Visualizar Quiz 65](#_Toc143088348)

[Figura 42 – Diagrama de atividade do Editar Quiz 66](#_Toc143088349)

[Figura 44 – Diagrama de atividade do Deletar Quiz 67](#_Toc143088350)

[Figura 45 – Diagrama de atividade do Realizar Quiz 68](#_Toc143088351)

[Figura 46 – Diagrama de atividade do Visualizar Resultado do Quiz 69](#_Toc143088352)

[Figura 47 – Diagrama de atividade do Visualizar Modelo 3D 70](#_Toc143088353)

[Figura 48 – Diagrama de sequência do Criar Cadastro 71](#_Toc143088354)

[Figura 49 – Diagrama de sequência do Visualizar Cadastro 72](#_Toc143088355)

[Figura 50 – Diagrama de sequência do Atualizar Cadastro 73](#_Toc143088356)

[Figura 51 – Diagrama de sequência do Deletar Cadastro 74](#_Toc143088357)

[Figura 52 – Diagrama de sequência do Criar Quiz 75](#_Toc143088358)

[Figura 53 – Diagrama de sequência do Visualizar Quiz 76](#_Toc143088359)

[Figura 54 – Diagrama de sequência do Editar Quiz 77](#_Toc143088360)

[Figura 55 – Diagrama de sequência do Deletar Quiz 78](#_Toc143088361)

[Figura 56 – Diagrama de sequência do Realizar Quiz 79](#_Toc143088362)

[Figura 57 – Diagrama de sequência do Realizar Login 80](#_Toc143088363)

[Figura 58 – Diagrama de Classe 81](#_Toc143088364)

[Figura 59 - Wireframe de baixa fidelidade “Página Inicial” 82](#_Toc143088365)

[Figura 60 – Wireframe de baixa fidelidade “Realizar Cadastro do Aluno” 83](#_Toc143088366)

[Figura 61 – Wireframe de baixa fidelidade “Realizar Cadastro do Administrador” 84](#_Toc143088367)

[Figura 62 – Wireframe de baixa fidelidade “Realizar Login” 85](#_Toc143088368)

[Figura 63 – Wireframe de baixa fidelidade “Visualizar Perfil” 86](#_Toc143088369)

[Figura 64 – Wireframe de baixa fidelidade “Visualizar Aluno” 87](#_Toc143088370)

[Figura 65 – Wireframe de baixa fidelidade “Visualizar Administrador” 88](#_Toc143088371)

[Figura 66 – Wireframe de baixa fidelidade “Sobre Nós” 88](#_Toc143088372)

[Figura 67 – Wireframe de baixa fidelidade “Realidade Aumentada” 89](#_Toc143088373)

[Figura 68 – Wireframe de baixa fidelidade “Visualizar Materiais” 90](#_Toc143088374)

[Figura 69 – Wireframe de baixa fidelidade “Realizar Aula” 91](#_Toc143088375)

[Figura 70 – Wireframe de baixa fidelidade “Visualizar Questionários” 92](#_Toc143088376)

[Figura 71 – Wireframe de baixa fidelidade “Realizar Questionário” 93](#_Toc143088377)

[Figura 72 – Wireframe de baixa fidelidade “Resultado Questionário (positivo)” 94](#_Toc143088378)

[Figura 73– Wireframe de baixa fidelidade “Resultado Questionário (negativo)” 95](#_Toc143088379)

[Figura 74 – Wireframe de baixa fidelidade “Visualizar Progresso” 96](#_Toc143088380)

[Figura 75 – Wireframe de baixa fidelidade “Alterar Questionários” 97](#_Toc143088381)

[Figura 76 – Wireframe de baixa fidelidade “Criar Questionário” 98](#_Toc143088382)

[Figura 77 – Wireframe de baixa fidelidade “Adicionar Questões” 99](#_Toc143088383)

[Figura 78 - Página Inicial 100](#_Toc143088384)

[Figura 79 – Cadastro do Aluno 101](#_Toc143088385)

[Figura 80 – Cadastro do Administrador 102](#_Toc143088386)

[Figura 81 – Realizar Login 102](#_Toc143088387)

[Figura 82 – Visualizar Perfil 103](#_Toc143088388)

[Figura 83 - Visualizar Aluno 104](#_Toc143088389)

[Figura 84 - Visualizar Administrador 104](#_Toc143088390)

[Figura 85 – Sobre Nós 105](#_Toc143088391)

[Figura 86 – Realidade Aumentada 106](#_Toc143088392)

[Figura 87 – Realidade Aumentada (segunda parte) 106](#_Toc143088393)

[Figura 88 – Visualizar Materiais 107](#_Toc143088394)

[Figura 89 – Realizar Aula 108](#_Toc143088395)

[Figura 90 – QR Code 108](#_Toc143088396)

[Figura 91 – QR Code escaneado 109](#_Toc143088397)

[Figura 92 – Visualizar Questionários 110](#_Toc143088398)

[Figura 93 – Realizar Questionários 110](#_Toc143088399)

[Figura 94 – Resultado Questionário (Positivo) 111](#_Toc143088400)

[Figura 95 – Resultado Questionário (Negativo) 112](#_Toc143088401)

[Figura 97 – Visualizar Progresso 112](#_Toc143088402)

[Figura 98 – Alterar Questionários 113](#_Toc143088403)

[Figura 99 – Criar Questionários 114](#_Toc143088404)

[Figura 100 – Adicionar Questões 114](#_Toc143088405)

**Lista de Quadros**

[Quadro 1 – Tipos de variáveis e suas determinadas funções 25](#_Toc143070270)

[Quadro 2 – Tipos de operadores lógicos e seus respectivos símbolos e funções 27](#_Toc143070271)

[Quadro 3 – Tipos de operadores aritméticos e suas respectivas funções 27](#_Toc143070272)

[Quadro 4 – Tipos de operadores relacionais e suas respectivas funções 28](#_Toc143070273)

[Quadro 5 – Descrição do caso de uso “Criar Cadastro” 49](#_Toc143070274)

[Quadro 6 – Descrição do caso de uso “Manter Cadastro” 50](#_Toc143070275)

[Quadro 7 – Descrição do caso de uso “Realizar Login” 51](#_Toc143070276)

[Quadro 8 – Descrição do caso de uso “Manter Quiz” 52](#_Toc143070277)

[Quadro 9 – Descrição do caso de uso “Realizar Quiz” 53](#_Toc143070278)

[Quadro 10 – Descrição do caso de uso “Resultado Quiz” 54](#_Toc143070279)

[Quadro 11 – Descrição do caso de uso “Visualizar Modelo 3D” 55](#_Toc143070280)

[Quadro 12 – Descrição do caso de uso “Escanear QR Code” 56](#_Toc143070281)

**Lista de abreviaturas e siglas**

C Sharp (C#)

Cascading Style Sheets (CSS)

HyperText Markup Language (HTML)

Hypertext Preprocessor (PHP)

International Business Machines Corporation (IBM)

Modelo Entidade Relacionamento (MER)

My Structured Query Language (MYSQL)

Network (.NET)

Quick Responde Code (QR Code)

Realidade Aumentada (RA)

Sistema de Gerenciamento de Banco de Dados (SGBD)

Structured Query Language (SQL)

Three-dimensional (3D)

Two-dimensional (2D)

Unified Modeling Language (UML)

Uniform Resource Locator (URL)

**SUMÁRIO**

[1. INTRODUÇÃO 14](#_Toc144454393)

[2. REFERENCIAL TEÓRICO 15](#_Toc144454394)

[2.1 Tecnologias Utilizadas 15](#_Toc144454395)

[2.1.1 Figma 15](#_Toc144454396)

[2.1.2 UML 15](#_Toc144454397)

[2.1.2.1 Diagramas comportamentais 16](#_Toc144454398)

[2.1.2.2 Diagramas estruturais 17](#_Toc144454399)

[2.1.3 HTML 17](#_Toc144454400)

[2.1.4 CSS 20](#_Toc144454401)

[2.1.5 JavaScript 21](#_Toc144454402)

[2.1.5.1 Tag Script 21](#_Toc144454403)

[2.1.5.2 Variáveis 22](#_Toc144454404)

[2.1.5.3 Função 22](#_Toc144454405)

[2.1.6 PHP 24](#_Toc144454406)

[2.1.6.1 Variáveis 24](#_Toc144454407)

[2.1.6.2 Constantes 26](#_Toc144454408)

[2.1.6.3 Operadores 26](#_Toc144454409)

[2.1.6.4 Estruturas de controle 28](#_Toc144454410)

[2.1.7 Bootstrap 30](#_Toc144454411)

[2.1.8 Banco De Dados 30](#_Toc144454412)

[2.1.8.1 MySQL 32](#_Toc144454413)

[2.1.8.2 SQL 33](#_Toc144454414)

[2.1.8.3 MySQL com PHP 34](#_Toc144454415)

[2.1.9 O que é o MER 34](#_Toc144454416)

[2.1.10 Laravel 35](#_Toc144454417)

[2.1.10.1 Laravel com Banco de Dados 36](#_Toc144454418)

[2.1.10.2 MVC no Laravel 37](#_Toc144454419)

[2.1.10.3 Rotas 37](#_Toc144454420)

[2.1.10.4 Model 37](#_Toc144454421)

[2.1.10.5 View 38](#_Toc144454422)

[2.1.10.6 Controller 38](#_Toc144454423)

[2.1.11 Realidade Aumentada 39](#_Toc144454424)

[2.1.12 Unity 40](#_Toc144454425)

[2.1.13 C# 42](#_Toc144454426)

[2.1.13.1 .NET 42](#_Toc144454427)

[2.1.13.2 C# na Unity 43](#_Toc144454428)

[2.1.14 Python 44](#_Toc144454429)

[2.1.14.1 Listas 45](#_Toc144454430)

[2.1.14.2 Dicionário 46](#_Toc144454431)

[3. DESENVOLVIMENTO 47](#_Toc144454432)

[3.1 Diagrama de Caso de Uso 47](#_Toc144454433)

[3.1.1 Documentação dos Casos de Uso 48](#_Toc144454434)

[3.2 Modelo Entidade-Relacionamento (MER) 57](#_Toc144454435)

[3.3 Diagrama Entidade-Relacionamento (DER) 57](#_Toc144454436)

[3.4 Diagrama de Atividade 58](#_Toc144454437)

[3.5 Diagramas de Sequência 71](#_Toc144454438)

[3.6 Diagrama de Classe 81](#_Toc144454439)

[4. DESENVOLVIMENTO DA APLICAÇÃO 81](#_Toc144454440)

[4.1 Wireframes de Baixa Qualidade 81](#_Toc144454441)

[4.2 Prototipação de Interface 99](#_Toc144454442)

[5. CONSIDERAÇÕES FINAIS 115](#_Toc144454443)

[REFERÊNCIAS 14](#_Toc144454444)

1. INTRODUÇÃO

Dentre muitos problemas encontrados no ensino tradicional, é possível observar a falta de compreensão dos alunos quanto ao aprendizado teórico, muitas vezes sem estímulos visuais. Segundo Piaget (1973), a evolução da cognição de imagens mentais inicia por volta dos dois anos segue em constante evolução, a capacidade cognitiva de resolução passa a ser desenvolvida através de imagens.

O processo de aprendizado pode ser prejudicado muitas vezes por falta de professores capacitados, estímulos visuais ou até mesmo por materiais didáticos que não trazem interesse ao aluno, tornando o estudo improdutivo, maçante e cansativo. Como disse Senhoras (2020), com a pandemia de COVID-19, os estudantes ficaram em confinamento, impedindo-os de irem à escola, dificultando o ensino e trazendo a necessidade do uso de tecnologias na educação.

Segundo o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP, 2021), é necessário acertar no mínimo 25 questões em ciências da natureza para tirar 700 pontos no Exame Nacional do Ensino Médio.

Segundo o ENEM (2021), as Ciências da Natureza foram as áreas do ENEM com a menor proficiência média em 2021, contendo matérias que são lecionadas no 7º ano.

Com base nos dados que foram coletados durante a pesquisa, foi desenvolvida a ideia de um projeto para a solução dos problemas e dificuldades encontradas na área da educação.

Esse trabalho trata-se de uma pesquisa bibliográfica, do tipo exploratório que busca caracterizar os trabalhos sobre o uso de novas tecnologias no ensino de ciências para estudantes do 7º ano. Para Sousa, Moita e Carvalho (2011), a escola precisa se reinventar e o professor deve se apropriar da presença de tecnologias, pois é favorável a seu trabalho se utilizada com uma boa abordagem. Com o objetivo de desenvolver uma aplicação de um Sistema Web utilizando a Realidade Aumentada (RA), com o intuito de contribuir na melhoria do aprendizado de Ciências da Natureza, tornando-o interativo e objetivo.

2. REFERENCIAL TEÓRICO

Neste capítulo, para o entendimento do trabalho, é fornecido o embasamento teórico, abordando tópicos envolvidos, apresentando as tecnologias/linguagens necessárias para o desenvolvimento do projeto.

2.1 Tecnologias Utilizadas

As seguintes tecnologias foram utilizadas para o desenvolvimento e planejamento da aplicação.

2.1.1 Figma

Segundo sua documentação (2023) o figma é um framework para criar interfaces (prototipação), onde as pessoas podem compartilhar seu projeto com outras.

De acordo com Staiano (2022) pode-se construir brainstorm e wireframes, disponibilizando também o código CSS para facilitar o trabalho dos desenvolvedores, o figma permite que crie ilustrações vetoriais, e atividades de formação em equipes.

Em sua documentação (2023) cita que o figma pode ser usado no navegador, funcionando no Mac, Windows e Linux, e todo projeto é salvo na nuvem.

Segundo Staiano (2022), os wireframes auxiliam a equipe a definir os recursos do site de maneira rápida e barata, possibilitando a realizações de testes, a facilitação do feedback, com o objetivo principal de tornar a interface acessível ao usuário.

2.1.2 UML

Diversas linguagens são utilizadas durante o gerenciamento e execução de um projeto, para que ele cumpra o seu objetivo com qualidade e precisão. Dentre essas linguagens a Unified Modeling Language (UML) é considerada uma linguagem de notação de extrema importância para o uso em projeto de sistemas.

Segundo Guedes (2011), a UML é uma linguagem utilizada para a modelagem de softwares que utilizam a orientação de objetos, podendo ser aplicada a todos os domínios. Com o passar dos anos está linguagem tornou-se a linguagem-padrão internacionalmente na engenharia de software.

A linguagem é composta por dois grupos: os diagramas estruturais e diagramas comportamentais. De acordo com Melo (2004), os diagramas estruturais são responsáveis por mostrar características estruturais, já os comportamentais mostram como o sistema reage às requisições propostas.

A UML é uma linguagem para visualização, especificação e para documentação, contendo diferentes diagramas dentro desses dois grupos que exercem inúmeras funções, para entender a importância dessa linguagem dentro de um projeto é necessário conhecer alguns desses diagramas e suas funcionalidades.

2.1.2.1 Diagramas comportamentais

Os diagramas comportamentais são um dos grupos que compõem a linguagem UML, são organizados através de diferentes maneiras para a realização da modelagem do sistema. De acordo com Booch (2012), esses diagramas têm como função a documentação, construção e especificação dos aspectos de um sistema dinâmico, eles são:

* Caso de Uso: é um dos diagramas comportamentais mais conhecidos, normalmente utilizados para a visualização do comportamento do sistema de maneira estática, através da interação dos atores com o mesmo.
* Sequência: O Diagrama de Sequência, também conhecido como Diagrama de Interação, é utilizado para apresentar aspectos dinâmicos, permitindo a visualização de interações entre os objetos a partir do encadeamento de informações, dando ênfase à organização estrutural de objetos, determinando como o processo deverá ser efetuado.
* Atividade: O Diagrama de Atividade é essencial para a visualização do fluxo de atividades do sistema, demonstrando suas atividades e as decisões das mesmas que o usuário/sistema pode percorrer. Estes diagramas em sua maioria são utilizados para a modelagem do fluxo do controle de uma operação.

2.1.2.2 Diagramas estruturais

Como dito anteriormente e ressaltado por Melo (2004), os diagramas estruturais ou estáticos são aqueles que apresentam características do sistema que não mudaram com o tempo, para facilitar a compreensão da estrutura. Para a construção desta aplicação, o diagrama estrutural utilizado foi o seguinte:

* Classe: Sendo um dos diagramas mais utilizados para a modelagem de um sistema e sendo base para alguns diagramas, os Diagramas de Classe possibilitam uma visão estática da estrutura, exibindo um conjunto de classes, interfaces e seus relacionamentos. Os diagramas de classes são importantes para construção de sistemas executáveis através da engenharia direta, além de permitir sua documentação, visualização e documentação.

2.1.3 HTML

HTML (HiperText Markup Language) é uma linguagem de marcação utilizada para inserir, textos, imagens vídeos e áudios na criação de páginas web.

Segundo Castro e Hyslop (2012), HTML teve seu início no começo dos anos 90, onde começou como um simples documento que detalhava elementos para uma página web. Atualmente está sendo utilizada a versão HTML5.

Segundo Ferreira e Eis (2012) O HTML é a linguagem de marcação de maior importância, porque é ele quem definirá cada informação que será inserida e que levará para as páginas web, possuindo diferentes elementos, eles são:

* Hipertextos: Todo texto que é inserido em uma página web pode ser interligado à outras páginas web do mesmo projeto ou de diferentes páginas web com o uso dos links, são chamados de hipertextos.
* Marcação: os meios de comunicação (smartphone, tablet, browser ou outro sistema) não conseguem diferenciar, um título de um parágrafo. Por isso, o HTML é baseado por marcação.

Figura 1 – Estrutura básica HTML



Fonte: Do Próprio Autor, 2023.

* DOCTYPE: O DOCTYPE informa para o navegador qual versão do HTML está sendo utilizada e o tipo de documento HTML, ajudando-o a interpretar o código corretamente e exibir o conteúdo da página corretamente.
* Tag: é o elemento raiz de todo o documento HTML, portanto, todo conteúdo que conterá na página deve estar dentro do elemento HTML. E dentro da tag HTML, há um atributo nomeadoLANG.
* Meta charset: Atualmente utilizando a tabela Unicode, no HTML5 o elemento usado é da seguinte forma: *<meta charset = “UTF-8”>.* Sua função é garantir que o navegador interprete corretamente os caracteres especiais e acentos do conteúdo textual da página, sendo legível para todos.

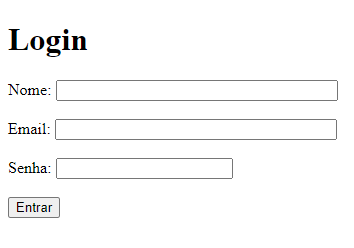
Figura 2 - Exemplo de Login com HTML puro.



Fonte: Do próprio autor, 2023.

A figura 2 mostra um pequeno exemplo de código HTML, onde é criado um formulário que pode ser utilizado para a realização de um login por meio do campo nome, e-mail e senha.

Figura 3 - Execução do código em HTML puro.



Fonte: Do próprio autor, 2023.

2.1.4 CSS

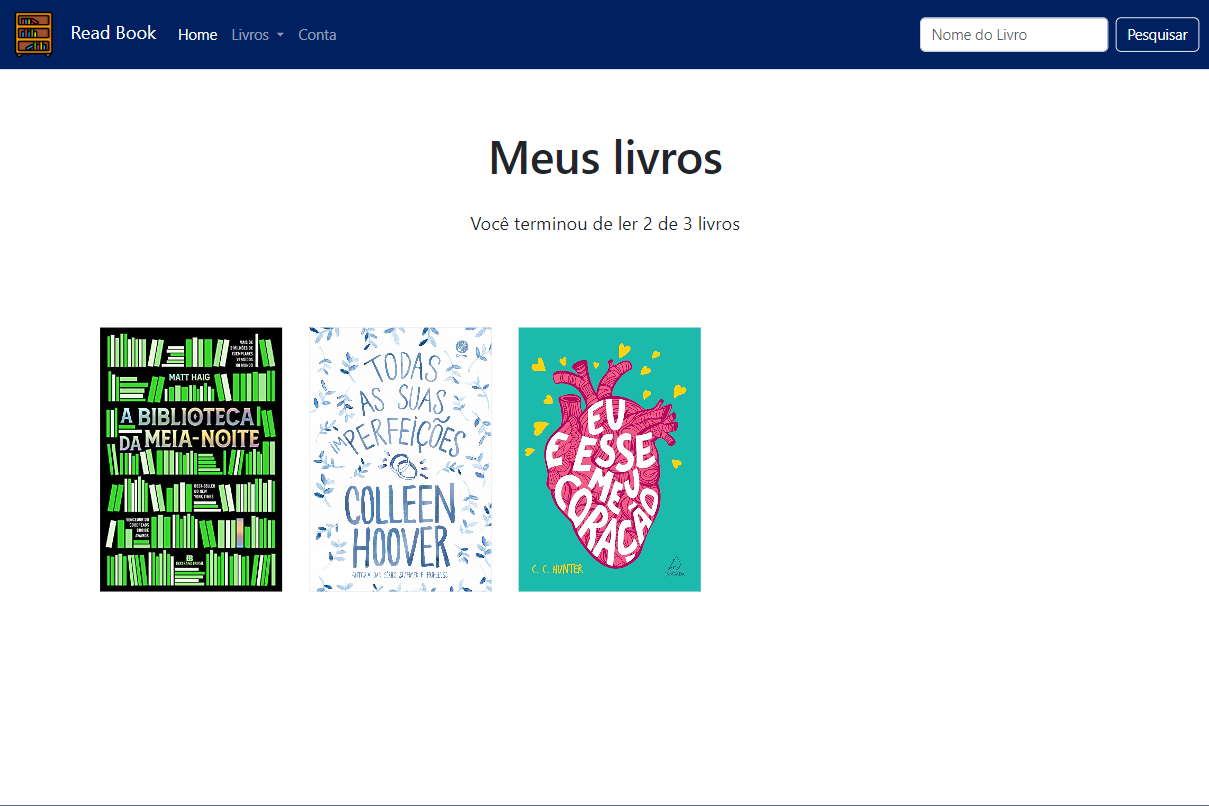
Segundo Zemel (2015), o CSS é a sigla para Cascading Style Sheets, ou seja, Folhas de Estilo em Cascatas, ela é uma linguagem que dá estilo ao código criado por algumas linguagens, como o HTML.

Conforme afirmou Eis e Ferreira (2012), o CSS formata a informação do HTML antes de chegar ao navegador. Essa informação estilizada pode ser um texto, imagem, vídeo ou qualquer elemento que será introduzido dentro do site.

Com isso, utilizando o CSS, é possível formatar características visuais, como cores, fontes, margens, botões e a posição de um elemento do HTML, o CSS trata do design do site, podendo deixar a página elegante e responsiva, enquanto o HTML inserir cada elemento.

Segundo Zemel (2015), um site com design responsivo pode ser acessado de um computador, notebook, smartphone, tablet ou de uma televisão. Qualquer dispositivo com acesso à internet pode abrir um site responsivo e apresentar suas informações de forma adequada. Um site que não é responsivo não se adapta a formatação a cada dispositivo e acaba não sendo acessível.

Figura 4 – Exemplo de site no computador



Fonte: Do Próprio Autor, 2023.

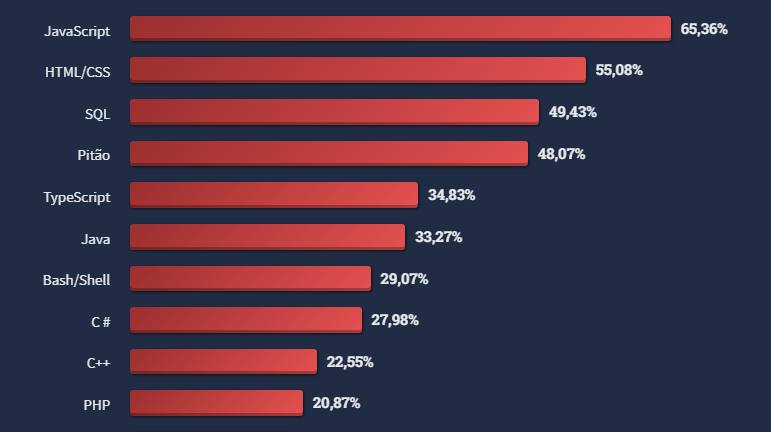
2.1.5 JavaScript

De acordo com Silva (2010) o Javascript foi criado pela Netscape junto com a Sun Microsystems, com o intuito de dar mais interatividade a uma página web, sua primeira versão foi lançada em 1995.

Segundo Iepsen (2018) a linguagem JavaScript é utilizada para definir o comportamento dos elementos da página, os códigos escritos em JavaScript, são interpretados diretamente pelos navegadores web.

Ou seja, é uma linguagem de programação de comportamento, que permite a criação de conteúdos interativos, dando interatividade as páginas web como, por exemplo, através de formulários onde realiza a validação dos dados, animações, apresentar mensagens ao usuário, etc.

Figura 5 – Linguagens de programação mais populares

Fonte: (Stack Overflow Developer Survey, 2022).

2.1.5.1 Tag Script

De acordo com Powers (2010) a linguagem é usada com frequência dentro de outra linguagem como, a linguagem de marcação HTML. Para se iniciar um código Javascript é necessário abrir o elemento <script>. Dentro do elemento <script> há um atributo type onde define o tipo de script que será utilizado, sendo utilizado o text/javascript, mas por ser uma linguagem padrão nos navegadores, não é necessário usar o atributo type, pois o navegador já interpretará como um script.

2.1.5.2 Variáveis

De acordo com Silva (2010, p.55) “Variável é um nome qualquer ao qual se atribui um valor ou dado. Uma variável pode conter uma string, um número inteiro, um número fracionário, um array, um booleano, uma função, um objeto etc”. Sendo assim, pode dar um nome à variável que se sinta confortável ou que seja fácil para não se esquecer. Segundo Iepsen (2018) “As variáveis declaradas em um programa devem possuir um nome, seguindo algumas regras de nomenclatura”. Sendo, elas:

* Não pode conter espaços e números;
* Não pode conter caracteres especiais, como $, #, -, +, \*, (,), {,};
* E não pode conter palavras-chaves como return, function, for, new, por fazer parte da sintaxe da linguagem do Javascript.

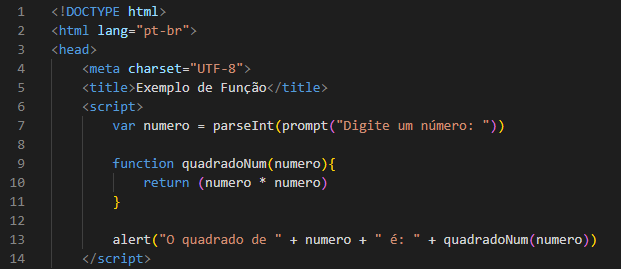
2.1.5.3 Função

Segundo Flanagan (2011) função é um bloco de código definido uma vez, onde pode ser chamado para executar uma ação diversas vezes. Em Javascript as funções incluem identificadores (parâmetros) que agem como variáveis locais.

Em Javascript, as funções são objetos, podendo atribuir funções a variáveis e passa-las para outras funções. Também, pode definir propriedades e chamar métodos. Para definir uma função é usada a palavra-chave *function*, a seguir dela pode dar um nome qualquer a função que for criar. Para a função, segue a definição:

* Dar um nome identificador à função. Sendo obrigatória para a declaração da função.
* Deve possuir parênteses, em seguida da função. E dentro colocar parâmetros. Quando for mais de um, é separado por vírgula.
* Em seguida, deve conter chaves, para colocar dentro o código que dirá como tem que rodar o código. Depois, chamará a função para realizar o que o código propõe.

Figura 6 – Exemplo da Função no JavaScript



Fonte: Do próprio autor, 2023.

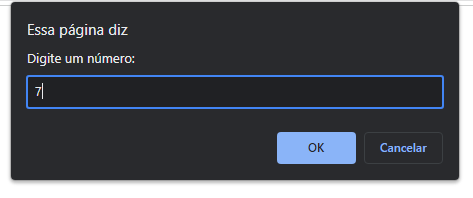
Na figura acima, foi criado uma variável que será armazenado o número digitado, a variável será transformada em número inteiro através da função *parseInt*.

Para retornar o valor desejado, foi criado uma função chamada *quadradoNum* com o argumento *número,* essa função retornará o quadrado do número que foi digitado.

Para exibir o resultado usa o método *alert()*, onde está sendo concatenado o número digitado e o resultado do quadrado, sendo assim, será exibido tudo junto.

Nas figuras abaixo, mostrará a execução do código.

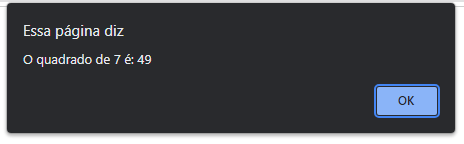
Figura 7 – 1ª Execução do Exemplo



Fonte: Do próprio autor, 2023.

Na imagem acima é apresentado um campo desenvolvido com Javascript, permitindo com que o usuário digite um número para o programa calcular o quadrado do número selecionado. Com isso é mostrado o resultado na imagem a seguir.

Figura 8 – 2ª Execução do Exemplo



Fonte: Do próprio autor, 2023.

2.1.6 PHP

O PHP (Hypertext Prepocessor) é uma linguagem de programação interpretada de desenvolvimento web, desenvolvida pelo programador dinamarquês Rasmus Lerdorf por volta de 1994, com o objetivo inicial de fazer a ligação lógica entre dois sistemas.

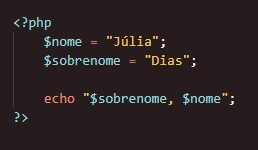
Segundo Converse e Park (2003), o PHP possibilita a incorporação de fragmentos de código em páginas de HTML, interpretado conforme as páginas são apresentadas, facilitando a conexão com o banco de dados.

Atualmente sendo uma das linguagens mais utilizadas no mundo, o PHP é uma linguagem utilizada para a criação de páginas web, permitindo a criação de páginas dinâmicas, manipulação de dados, interação com o banco de dados e entre muitas funções. Segundo Bento (2021) o PHP torna o pré-processamento de páginas, permitindo que a linguagem altere os elementos de uma página antes que a mesma seja enviada ao navegador.

2.1.6.1 Variáveis

No PHP a principal maneira de armazenar informações é através de variáveis sendo precedida pelo sinal de cifrão ($) na frente. Segundo Dall’Oglio (2018, p. 24) “Variáveis são identificadores utilizados para representar valores mutáveis e voláteis que só existem durante a execução do programa”.

Figura 9 – Exemplo de variável no PHP



Fonte: Do Próprio Autor, 2023.

No exemplo acima é possível identificar duas variáveis tipo String e seus valores atribuídos, após o recebimento dos valores o comando *echo* é responsável por enviar para a saída qualquer informação como texto, números ou variáveis (no exemplo o *echo* será responsável por enviar os valores atribuídos nas variáveis até a saída).

De acordo com Dall’Oglio (2018) existem diferentes tipos de variáveis que armazenam diferentes tipos de atribuições, elas são do tipo booleano, numérico, String, Array, Objeto, Recurso, Misto e NULL.

Quadro 1 – Tipos de variáveis e suas determinadas funções

|  |  |
| --- | --- |
| Tipo da variável | Armazena |
| Booleano | Valor lógico podendo ser verdadeiro (True) ou falso (False) |
| Numérico | Números em notação decimal, hexadecimal ou octal. |
| String | Cadeia de caracteres alfanuméricos. |
| Array | Lista de valores armazenados, podendo ser de tipos diferentes. |
| Objeto | Entidade que tem seu comportamento definido por métodos e propriedades. |
| Recurso | Variável especial que tem a capacidade de manter uma referência de recurso externo. |
| Misto | Mais de um tipo de dado dentro do mesmo parâmetro. |
| NULL | A variável não tem um valor. |

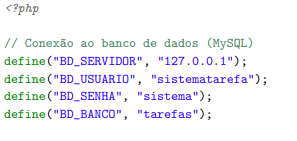
Fonte: (Dall’Oglio, 2018).

**2.1.6.2** Constantes

Segundo exemplifica o PHP (2023) em sua documentação, “uma constante é um identificador (nome) para um valor simples”. Diferente das variáveis que existem apenas durante a execução do programa, as constantes são valores que não podem ser modificados durante a execução. As constantes são armazenadas em valores escalares e que são acessíveis em toda parte dos scripts. Para a definição de uma constante é necessário utilizar a função define( ). Segundo Bento (2021), a função define( ) é responsável por receber o nome da constante e seus respectivo valor.

O exemplo abaixo demonstra as configurações das constantes para a conexão com o banco de dados, contendo seus dados que não poderão ser esquecidos após o fim da execução do programa.

Figura 10 – Exemplo de constante no PHP



Fonte: (Bento, 2021)

2.1.6.3 Operadores

Os operadores são ferramentas utilizadas para comparar valores, segundo PHP (2023), no mesmo existem três principais tipos de operadores, eles são: operadores lógicos, aritméticos e relacionais.

Operadores lógicos: Segundo Converse e Park (2003), os operadores lógicos utilizam outros valores lógicos para a produção de novos valores booleanos

Quadro 2 – Tipos de operadores lógicos e seus respectivos símbolos e funções

|  |  |  |
| --- | --- | --- |
| Operador | Símbolo | Função |
| and | && | Usado para combinar múltiplas condições e retornar verdadeiro caso todas as condições forem verdadeiras. |
| Ou | | | | É verdadeiro se um ou ambos os argumentos forem verdadeiros |
| ! | Não possui | Caso um argumento seja falso, seu argumento retornará verdadeiro. |
| Xor | Não possui | É verdadeiro se qualquer um dos argumentos, (mas não ambos) forem verdadeiros. |

Fonte: (Converse e Park, 2003).

Operadores aritméticos: No dizer de Dall’Oglio (2018), os operadores aritméticos são aqueles que são utilizados para realização de cálculos.

Quadro 3 – Tipos de operadores aritméticos e suas respectivas funções

|  |  |
| --- | --- |
| Operadores | Funções |
| + | Adição |
| - | Subtração |
| \* | Multiplicação |
| / | Divisão |
| % | Resto da divisão (módulo) |

Fonte: (Dall’Oglio, 2018)

Operadores relacionais: Os operadores relacionais fazem parte dos três principais tipos de operadores encontrados no PHP. Como caracteriza PHP (2023), os operadores relacionais são aqueles que são utilizados para a realização de comparações entre valores e/ou expressões.

Quadro 4 – Tipos de operadores relacionais e suas respectivas funções

|  |  |
| --- | --- |
| Operadores | Funções |
| $a && $b | Retorna verdadeiro se ambos os argumentos forem verdadeiros. |
| $a or $b | Retorna verdadeiro se um dos argumentos for verdadeiro. |
| $a xor $b | Retorna verdadeiro se ambos argumentos forem verdadeiros, de forma exclusiva. |
| ! $a | Retorna verdadeiro se o argumento for falso. |
| $a | | $b | Retorna verdadeiro se um dos argumentos for verdadeiro. |

Fonte: (PHP, 2023).

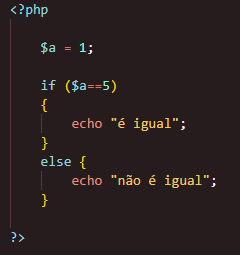
**2.1.6.4** Estruturas de controle

Uma estrutura de controle é um bloco de programação que é capaz de analisar variáveis e de escolher uma direção baseados nos parâmetros pré-definidos. Como caracteriza Converse e Park (2003), para que um sistema possua diferentes possibilidades é necessária uma estrutura de controle que indicam diferentes situações e como a execução dos códigos acontecerá com as mesmas.

Existem diversas estruturas de controle no PHP, mas para melhor entendimento será abordado apenas três estruturas, IF, WHILE e FOR.

* Estrutura IF: o *IF* é uma estrutura de controle que introduz um desvio condicional, ou seja, um desvio na execução natural do programa. O exemplo abaixo exemplifica o entendimento do autor, caso a condição dada para expressão condiz como verdadeira o bloco de comandos será executado.

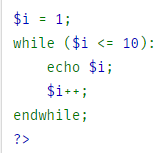
Figura 11 – Exemplo da estrutura de controle IF no PHP



Fonte: Do Próprio Autor, 2023.

* Estrutura WHILE: A estrutura de controle while exerce a função de loop, sendo um dos tipos mais simples. Esta estrutura tem como função executar as funções instruídas repetidamente tendo a condição dos argumentos verificada a cada início de laço. No exemplo apresentado na documentação da linguagem, é possível entender a funcionalidade dessa estrutura, o comando while avalia o valor atribuído a variável (caso o mesmo seja 3) para a iniciação do laço. A variável inicia-se com um valor atribuído equivalente a 1, há como condição que a variável ($i) deve ser menor ou igual a 10, o código será repetido até o contador atingir o número indicado, o echo tem como função apresentar o avanço do contador.

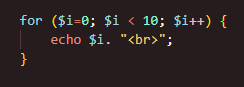
Figura 12 – Exemplo da estrutura de controle WHILE no PHP



Fonte: (PHP, 2023).

* Estrutura FOR: A última estrutura de controle é o FOR, assim como o while, o FOR também é uma estrutura de repetição, porém um pouco mais complexa. Essa estrutura apresenta um laço de repetição baseado em um contador, sendo controlado por um bloco de três comandos. No exemplo abaixo, é possível conhecer um pouco mais sobre a estrutura, composta por três expressões separadas por ponto e vírgula. A expressão 1, será executada somente uma vez, ao iniciar o loop; a expressão 2, é uma expressão booleana validada antes de cada iteração do loop; A expressão 3 é executada ao final de cada iteração

Figura 13 – Exemplo de uso da estrutura FOR



Fonte: Do Próprio Autor, 2023.

2.1.7 Bootstrap

Segundo sua documentação (2023), o Bootstrap é um framework front-end para o desenvolvimento de sites responsivos, onde possui códigos com CSS e Javascript. Desenvolvido por um designer e desenvolvedor no Twitter.

Segundo Spurlock (2013) o Bootstrap é um site de código aberto, onde qualquer pessoa conseguirá ter acesso aos códigos presentes e usá-los em seu projeto livremente. Foi desenvolvido por Mark Otto e Jacob Thorton, funcionários do Twitter.

De acordo com sua documentação (2023), o Bootstrap foi criado no Twitter em meados de 2010 e lançado em 2011, tendo vinte versões. Sendo uma delas com a atualização de funcionalidade responsiva opcional.

Um dos destaques do Bootstrap, é que o desenvolvedor pode utilizar e personalizar os códigos disponíveis para se adequar a seu projeto, escolhendo quais recursos CSS ou Javascript deseja incluir no projeto e também a responsividade, onde através do CSS permite que o site seja adaptado para qualquer tamanho de tela.

2.1.8 Banco De Dados

Para a realização do projeto, será utilizado um banco de dados relacional. Segundo Date (2003), um banco de dados é um sistema cuja finalidade geral é armazenar informações e permitir que os usuários busquem e atualizem essas informações quando solicitar. Um exemplo de como pode ser utilizado um banco de dados, é com cadastro de clientes, onde as informações de login estão salvas em uma tabela.

Segundo Taylor (2015), os bancos de dados modernos conseguem armazenar dados rápido e fácil, são seguros e todos os dados são armazenados nas nuvens, além de recuperar os dados de maneira rápida.

De acordo com Date (2003), há muitos motivos para a utilização de um banco de dados. Algumas delas são:

1. A quantidade de registros que podem ser salvos, já que não tem a necessidade de arquivos de papel e tudo é salvo em um servidor.
2. Os dados podem ser alterados com facilidade a qualquer momento e serão atualizados em todos os dispositivos.
3. A rapidez que a máquina pode obter e modificar dados, deixando o processo de análise mais rápido.
4. Os dados estão mais seguros, sendo protegidos de perda não intencional e podem ser criptografados para o acesso ser apenas por pessoas autorizadas.

A imagem abaixo mostra um exemplo de tabela do banco de dados onde é armazenado as informações necessárias para o cadastro de um produto em um mercado e assim ter um controle de estoque.

Figura 14 – Exemplo de tabela SQL



Fonte: Do próprio autor, 2023.

A figura a seguir apresenta a tabela de usuário do mesmo banco de dados dos produtos, com essa tabela o usuário poderia realizar o login dentro do sistema e realizar a compra de um produto, por exemplo.

Figura 15 – Exemplo de tabela de clientes no SQL



Fonte: Do próprio autor, 2023.

2.1.8.1 MySQL

O MYSQL é o Sistema de Gerenciamento de Banco de Dados (SGBD) que foi escolhido para o desenvolvimento do projeto. Como afirma Luckow e Melo (2010), o MySQL é o banco de dados de código-fonte aberto mais popular do mundo, tendo mais de 70 milhões de instalação no mundo.

Segundo Taylor (2015), um SGBD é um conjunto de programas utilizados para construir a estrutura e operar os dados contidos dentro do banco de dados. Com isso, um SGBD define, administra e processa um banco de dados.

2.1.8.2 SQL

Structured Query Language (SQL), é uma linguagem de consulta estruturada e foi especialmente projetada para criar banco de dados, adicionar, modificar dados a este, e recuperar partes selecionadas dos dados, de acordo com Taylor (2015).

Essa linguagem consiste em um número limitado de instruções que executam três funções de manipulação de dados: definir dados, manipular dados e controlar dados. Ela é utilizada no SGBD para gerenciar um banco de dados. Como dito por Taylor (2015), o SQL é uma linguagem que pode ser usada de várias maneiras. É a ferramenta mais utilizada para a comunicação em um banco de dados relacional.

A figura abaixo é um exemplo de código em SQL que faz a criação de um banco de dados para um mercado. Esse banco possui duas tabelas, uma sendo para armazenar informações dos produtos e a outra para salvar dados dos clientes. Após a criação da estrutura, é inserido alguns dados nas duas tabelas.

Figura 16 – Exemplo de estrutura SQL



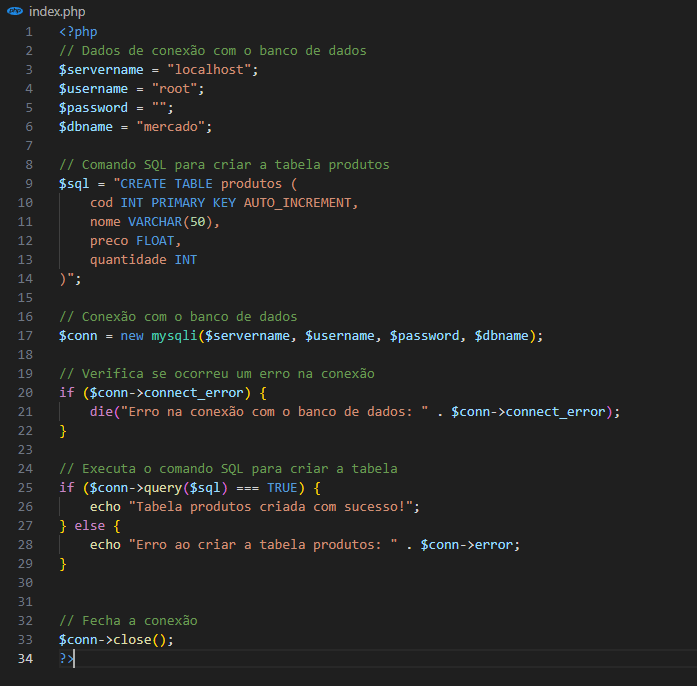
Fonte: Do próprio autor, 2023.

2.1.8.3 MySQL com PHP

Na programação de sistemas com interface, não é utilizado apenas o MYSQL para o gerenciamento de banco de dados, é preciso de uma linguagem que faça essa integração do projeto com a base de dados. Para isso, é utilizado o PHP.

Segundo Bento (2021), ele é uma ferramenta que possibilita o pré-processamento de páginas HTML, conseguindo editar o conteúdo de um site antes de enviá-la para o navegador. Abaixo é mostrado um exemplo de estrutura em PHP que conecta em um banco de dados utilizando SQL e cria uma tabela com o nome de “produtos”.

Figura 17 – Exemplo de SQL PHP com MYSQL



Fonte: Do próprio autor, 2023.

2.1.9 O que é o MER

Como disse Elmasri e Navathe (2011), uma fase importante no projeto de uma aplicação que faz uso de um banco de dados, é a modelagem conceitual. A técnica de modelagem mais conhecida é o MER (Modelo de Entidade-Relacionamento). Segundo Heuser (2009), um modelo conceitual é uma descrição do banco de dados sem utilizar um SGBD. O modelo conceitual apresenta que dados podem aparecer ao banco de dados, mas não informa como esses dados são armazenados com o SGBD. Assim o modelo ER permite a representação dos atributos e entidades de um banco de dados.

* Entidades: Uma entidade é algo do mundo real com uma existência independente. Um objeto pode ser um objeto físico como, por exemplo, uma pessoa e um carro, ou até mesmo com existência conceitual, como uma empresa ou um curso. Cada entidade possui seus próprios atributos.
* Atributos: são as propriedades específicas que descrevem cada entidade, em sua maioria possuindo um valor único como, por exemplo, idade. Atributos são os dados de uma entidade, portanto idade pode ser um atributo da entidade pessoa.

2.1.10 Laravel

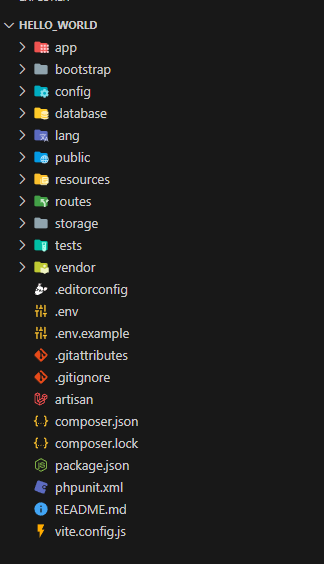
Segundo Silva (2022), o Laravel é o framework open-source que ajuda na criação de sites em PHP que está em constante desenvolvimento e tem uma grande comunidade e uma boa documentação. Ele foi desenvolvido por Taylor Otwell.

Independente da linguagem ou tecnologia que será usada, é preciso pensar no conceito de infraestrutura, porém boa parte da estrutura de um software é igual. Como afirma Turini (2015) é utilizado um framework, que ajuda agilizando o processo de desenvolvimento de forma organizada.

Como descrito por Gabardo (2017), os frameworks servem para diversos propósitos, como eliminar a necessidade de reescrever um pedaço de código por meio da reutilização de métodos, classes e funções.

A imagem abaixo mostra como é a estrutura do Laravel após criar um projeto vazio, o Laravel cria um conjunto de arquivos e pastas para sua configuração.

Figura 18 – Estrutura do Laravel



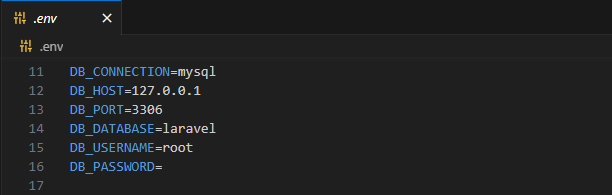
Fonte: Do próprio autor, 2023.

2.1.10.1 Laravel com Banco de Dados

O Laravel facilita a conexão com o banco de dados por meio do arquivo “.env” (DotEnv). Segundo Silva (2015), esse arquivo tem o objetivo de guardar todas as configurações de ambiente, como usuário e senha do banco de dados.

Abaixo é mostrado um exemplo de código do .env que faz a conexão com um banco de dados.

Figura 19 – Laravel com Banco de Dados



Fonte: Do próprio autor, 2023.

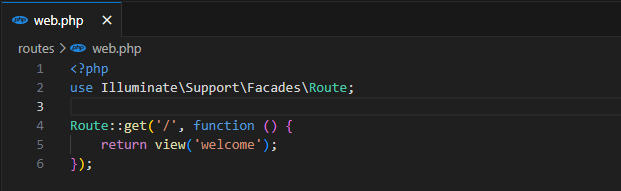
2.1.10.2 MVC no Laravel

O Laravel utiliza o padrão MVC, como disse Gabardo, MVC é o acrônimo para Model, View e Controller. Em português é o Model, Visualização e Controlador. Esse é um padrão de design de projetos que separa as camadas do projeto como forma de organização. O Laravel também utiliza o sistema de rotas para mapear requisições.

2.1.10.3 Rotas

As rotas mapeiam requisições e encaminham para o Controller. Não é uma camada do modelo MVC, mas redireciona o usuário para uma View. A imagem a seguir mostra um exemplo de rota que redireciona o usuário para uma View que se chama “welcome” assim que usuário acessar a url com o caminho “/” no navegador.

Figura 20 – Exemplo de Rotas

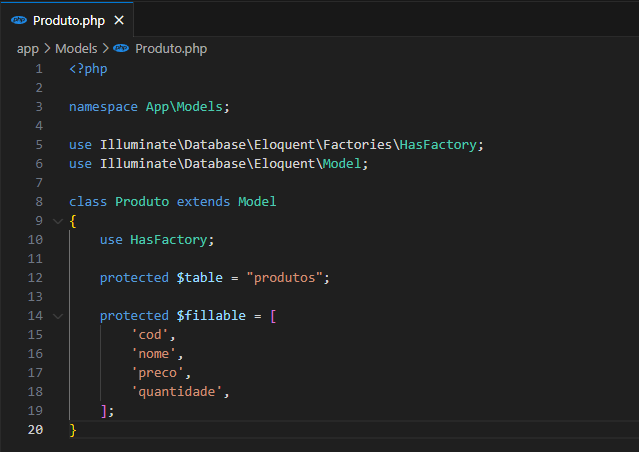


Fonte: Do próprio autor, 2023.

2.1.10.4 Model

Os Models são responsáveis por modelar os objetos utilizados na aplicação, define quais os tipos e os campos de um objeto. Ele faz conexão com uma tabela do banco de dados, conseguindo gerenciar seus dados. A seguir é mostrado um exemplo de model chamado “Produto” que possui alguns atributos e uma tabela do banco de dados.

Figura 21 – Exemplo de Model

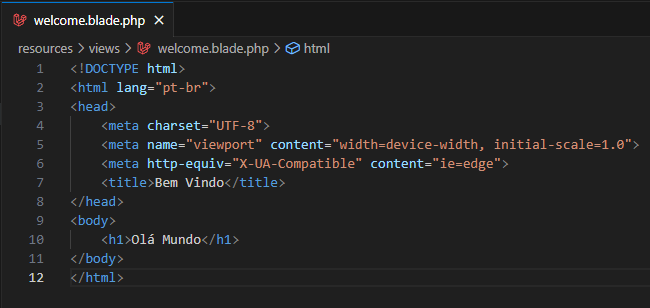


Fonte: Do próprio autor, 2023.

2.1.10.5 View

Views são os arquivos de exibição do projeto, ou seja, as telas da aplicação. Logo abaixo é mostrado um exemplo de view, um site simples em HTML com um texto.

Figura 22 – Exemplo de View



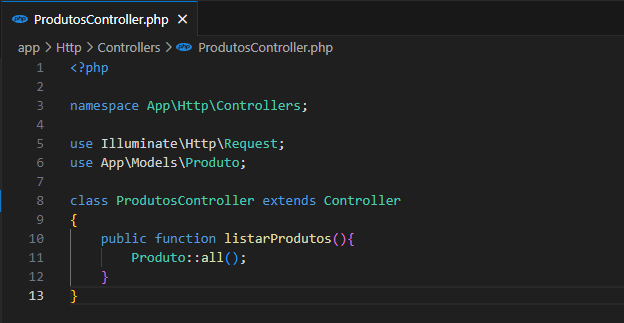
Fonte: Do próprio autor, 2023.

2.1.10.6 Controller

Já os Controllers, recebem requisições e as processam de acordo com a lógica implementada, por fim, os Controllers enviam os dados para a Model para gravar em

um banco de dados e recebem os dados para apresentar na View. A próxima imagem mostra uma função dentro do Controller de produtos que lista todos os produtos do banco de dados por meio de uma requisição do Model.

Figura 23 – Exemplo de Controller

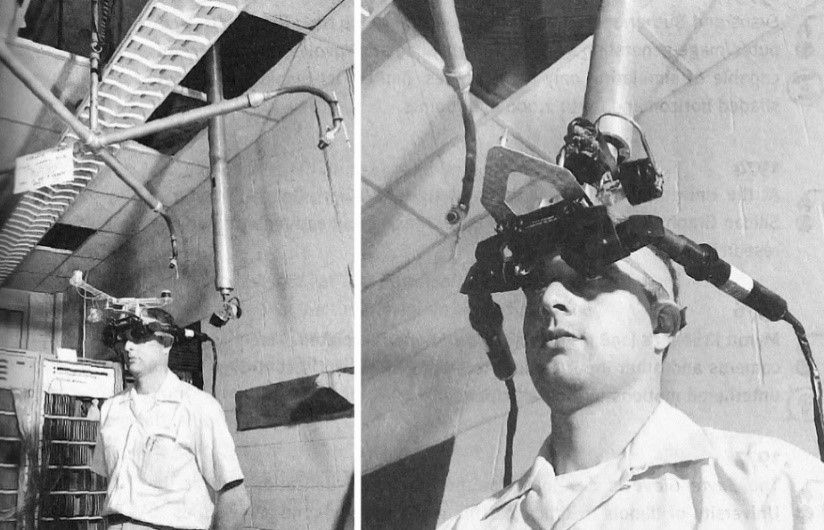


Fonte: Do próprio autor, 2023.

2.1.11 Realidade Aumentada

A realidade aumentada surgiu em 1963, nos Estados Unidos, por Ivan Sutherland, onde inventou o primeiro monitor usado na cabeça, o capacete de visão ótica.

Figura 24 – Primeiro monitor usado na cabeça

****

Fonte: (TecMundo, 2017).

De acordo com Siscoutto, Kirner e Tori (2006, p.22) “a Realidade Aumentada (RA) transporta o ambiente virtual para o espaço do usuário, permitindo que o usuário no mundo real tenha interação com o mundo virtual”.

Figura 25 – Exemplo de Realidade Aumentada

Fonte: (Scurra, 2018).

Nesse ambiente de realidade misturada, o RA ocorre quando os objetos virtuais são colocados no mundo real. Sendo assim, a possível interação do usuário como o objeto virtual. Para a Microsoft (2023), a realidade aumentada cria uma experiência de imersão para todos que a utilizam. As formas mais comuns de RA são por óculos ou lente de câmeras.

2.1.12 Unity

A criação de jogos 3D tem crescido cada vez mais com o avanço de tecnologias que permitiram o acesso a recursos mais aprimorados. Plataformas como a Unity são capazes de oferecer diversas ferramentas para a criação de jogos 3D, criando cenários, personagens e efeitos visuais com diferentes cores, texturas e profundidades.

Conforme Viana (2009), o Unity3D é um programa de desenvolvimento para jogos com diversas funcionalidades para a criação dos mesmos, a plataforma permite que os jogos sejam publicados em uma página web ou para Windows. O Unity3D é usado por algumas empresas como ferramenta para criação dos seus projetos, como a Ubisoft. Esta plataforma possui uma interface com Barras de ferramentas, janelas Hierárquicas, visualização do jogo, a visualização da cena, as sobreposições, a janela Inspetor, a janela do projeto e status.

Figura 26 – Interface gráfico do Unity3D

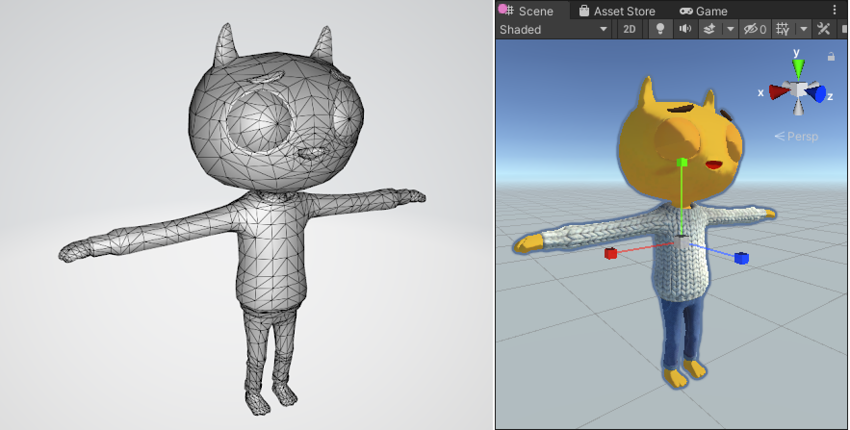


Fonte: (Unity, 2023)

Para o desenvolvimento de jogos 3D em plataformas como a Unity3D são utilizados planos cartesianos, conceito matemático aplicado na construção dessas interfaces, definindo dois eixos sendo eles x e y é possível dividir estes planos em quatro quadrantes. A representação 3D normalmente é realizada através do uso de polígonos como os triângulos para representarem uma montanha. Como descrito por Viana (2009), ao combinar diversos polígonos e as ferramentas propostas pela plataforma, é possível realizar a construção de formas mais complexas dando realismo e profundidade a cena.

Como dito anteriormente e reforçado pelo Unity3D (2021) a construção de modelagens 3D são realizadas através de polígonos. Para garantir que a sensação de hiper-realidade seja ainda maior é possível a implementação de texturas, cores e sombras em qualquer objeto da cena, utilizando o componente Renderizador de Malha é possível implementar suas texturas para a visualização final da aplicação mesclada em um ambiente real através da câmera de um dispositivo móvel.

Figura 27 – Exemplo da aplicação de texturas e cores em uma modelagem 3D



Fonte: (Unity, 2021).

2.1.13 C#

O C# (pronuncia-se "C Sharp") é uma linguagem de programação moderna, altamente versátil, orientada a objetos e fortemente tipada, criada pela Microsoft. De acordo com a própria Microsoft (2023), o objetivo do C# é possibilitar que os desenvolvedores criem uma ampla variedade de sistemas, desde aplicativos para analisar dados, programas web, jogos 2D e 3D, além de aplicativos móveis.

Para Trigo e Henriques (2023), programar é dizer ao computador o que é que deve ser feito, ou seja, dar instruções ao computador de uma forma lógica, o que resulta da programação é um sistema. A Microsoft (2023) enfatiza que, todos os aplicativos são compostos por muitas linhas de código que trabalham em conjunto para alcançar uma tarefa.

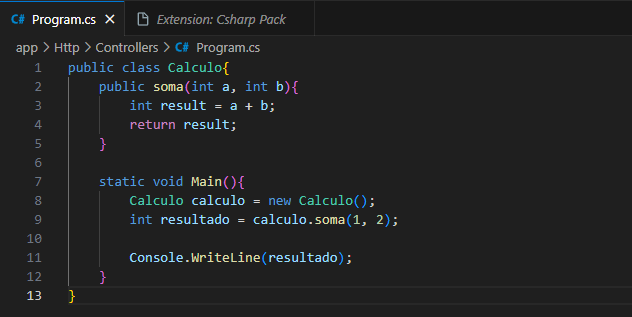
É nesse contexto que a escolha da linguagem de programação se torna importante. No caso do projeto em questão, a Unity foi selecionada como plataforma de desenvolvimento, que utiliza a linguagem C# para criar jogos e aplicativos.

2.1.13.1 .NET

É importante mencionar que os programas desenvolvidos em C# são executados no .NET (lê-se doNet), que é uma plataforma gratuita para os desenvolvedores, que permite criar vários tipos de aplicativos, de acordo com a Microsoft (2023).

O C# é uma linguagem de programação poderosa e versátil, amplamente utilizada para criar uma variedade de sistemas e aplicativos. Sua integração com a plataforma .NET torna o desenvolvimento mais acessível e oferece um conjunto abrangente de recursos para os desenvolvedores.

Figura 28 – Exemplo de estrutura do C#

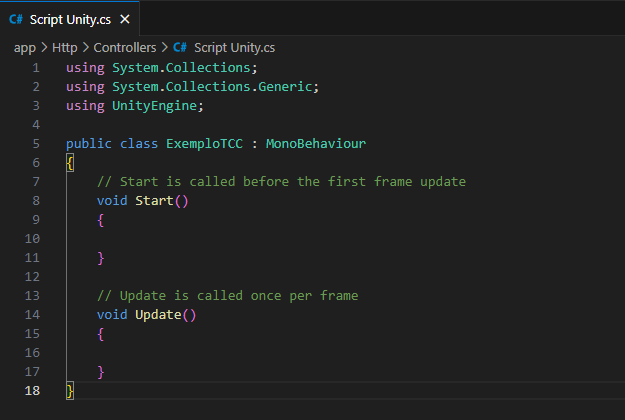


Fonte: Do próprio autor, 2023.

2.1.13.2 C# na Unity

A estrutura do C# muda dentro da Unity, já que ela possui suas próprias práticas e técnicas para acessar o mecanismo de script. Segundo a Unity (2021), um Script permite que você adicione eventos no jogo, modifique propriedades dos componentes ao longo do tempo e responda a entrada do usuário.

Figura 29 – Exemplo de estrutura do C# no Unity



Fonte: Do próprio autor, 2023.

A imagem mostra a estrutura do C# dentro do Unity. Ela possui duas funções, a Start e a Update. A Start é executada assim que o jogo inicia e a Update é chamada a cada frame que se passa.

2.1.14 Python

Segundo Borges (2010, p.14), “Python foi criado em 1990 na Holanda, por Guido Van Rossum, criado originalmente para ser usado por físicos e engenheiros”. Python é uma linguagem de programação de nível alto e fortemente “tipada”, interpretada por script.

Em sua documentação (2023), diz que a linguagem possui módulos, classes, exceções, tipos de dados de alto nível e tipagem dinâmica, podendo ser usada como linguagem de extensão para programas desenvolvidos em outras linguagens.

De acordo com Menezes (2014) a linguagem é conhecida por sua simplicidade e clareza, sendo considerada uma linguagem poderosa, afinal pode ser usada em diversos programas, também é um software livre, sendo usado gratuitamente.

O Python também não possui marcações como o ponto e vírgula (;) no final da linha de código, assim facilitando o desenvolvimento do projeto, esse detalhe diferencia essa linguagem das outras.

De acordo com sua documentação (2023) a linguagem tem recursos, como:

* Possui uma sintaxe elegante, sendo ela fácil de ler e fácil de usar;
* Possui uma biblioteca que permite fazer tarefas comuns de programação como, ler e modificar arquivos;
* É um software livre (como dito anteriormente). Podendo ser modificado e redistribuído, porque está sob licença de código aberto, etc;
* A linguagem oferece suporte ao levantamento e captura de exceções, resultando em um tratamento de erros mais limpo etc;
* Possui tipagem dinâmica, ou seja, o interpretador define o tipo de variável em tempo de execução, sendo assim, não precisa informar o tipo da variável ao declará-la.

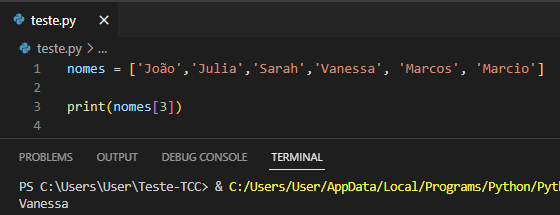
2.1.14.1 Listas

Segundo Downey (2016) listas são uma sequência de valores, podendo ser qualquer tipo, estes valores são chamados de elementos ou itens.

De acordo com Menezes (2014) a lista pode conter zero (0) ou mais elementos dentro dela, até mesmo outra lista.

Downey diz (2016) “listas são mutáveis. Quando o operador de colchete aparece do lado esquerdo de uma atribuição, ele identifica o elemento da lista”. Os valores que contém na lista podem ser alterados a qualquer momento, por isso são mutáveis

Figura 30 – Indicando o índice para visualizar o nome



Fonte: Do Próprio Autor, 2023.

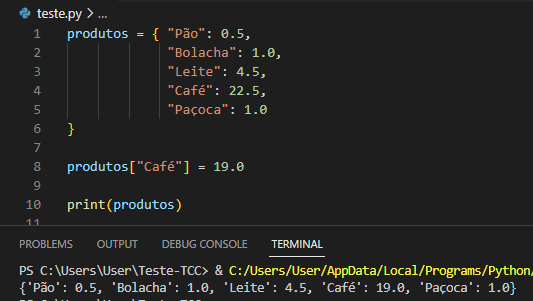
2.1.14.2 Dicionário

De acordo com Menezes (2014) os dicionários são parecidos com as listas como, serem mutáveis, porém com propriedades de acesso diferente. Sua estrutura possui chaves e valores, sendo chaves a índice.

Segundo Matthes (2016) o dicionário é uma coleção de chave-valor, cada chave é conectada com um valor, podendo-a usar para acessar o valor, chave-valor é um conjunto de valores associados um ao outro.

Pode-se usar os valores, inteiros, string, listas ou outro dicionário.

Figura 31 – Alterando o valor de uma chave-valor



Fonte: Do Próprio Autor, 2023.

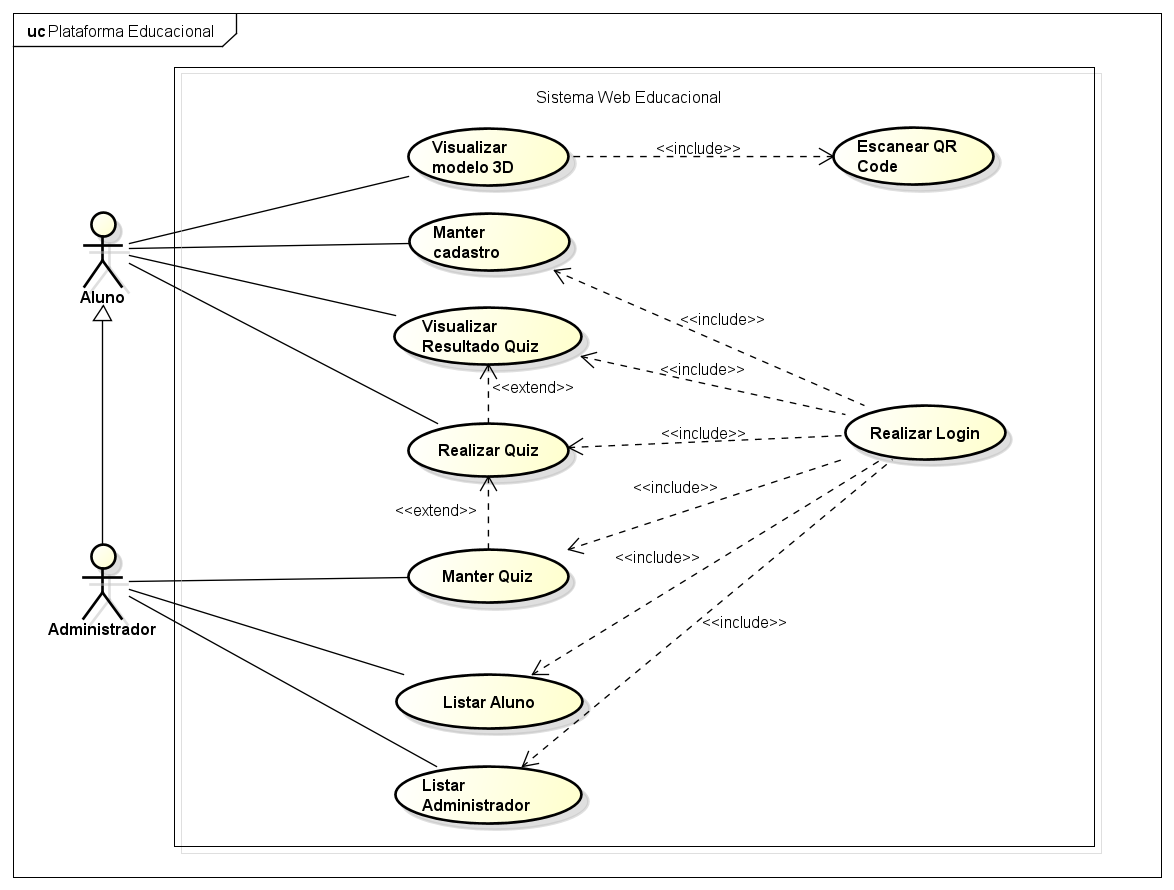
3. DESENVOLVIMENTO

Neste capítulo, é apresentado o processo de criação do Science AR, documentando o desenvolvimento por meio da linguagem UML, exemplificando com trechos de código e mostrando ilustrações das telas da aplicação.

3.1 Diagrama de Caso de Uso

Através da figura abaixo, é possível analisar o caso de uso do sistema, demonstrando como o aluno e o administrador interagem com o sistema, além das ações que os dois atores podem realizar.

Figura 32 – Diagrama de Caso de Uso

’

Fonte: Do Próprio Autor, 2023.

3.1.1 Documentação dos Casos de Uso

De acordo com os módulos UML, a seguir a documentação dos casos de uso de uso visto anteriormente.

Requisitos funcionais Aluno:

* RF01 – O sistema deverá permitir que o aluno efetue o login.
* RF02 – O aluno poderá manter seu cadastro.
* RF03 – O sistema deverá permitir que o aluno efetue o questionário caso o mesmo esteja logado/cadastrado.
* RF04 – O aluno poderá visualizar o resultado do questionário efetuado.
* RF05 – O aluno poderá escanear o QR Code.
* RF06 – O sistema deverá permitir que o aluno visualize o modelo 3D.

Requisitos funcionais Administrador:

* RF01 – O sistema deverá permitir que o aluno efetue o login.
* RF02 – O aluno poderá manter seu cadastro.
* RF03 – O sistema deverá permitir que o aluno efetue o questionário caso o mesmo esteja logado/cadastrado.
* RF04 – O aluno poderá visualizar o resultado do questionário efetuado.
* RF05 – O aluno poderá escanear o QR Code.
* RF06 – O sistema deverá permitir que o aluno visualize o modelo 3D.
* RF07 – O sistema deverá permitir que o administrador mantenha os questionários.

Requisitos não-funcionais da aplicação geral:

* RNF01 - O sistema deverá ser seguro.
* RNF02 - O sistema deverá ser intuitivo e acessível.
* RNF03 – O sistema deverá ser responsivo, sendo executado em qualquer dispositivo.
* RNF04 – O sistema deverá fornecer a visualização dos modelos 3D de uma forma rápida e dinâmica.

Quadros descritivos dos casos de uso:

Quadro 5 – Descrição do caso de uso “Criar Cadastro”

|  |  |
| --- | --- |
| Nome do caso de uso | Criar cadastro |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas para o aluno realizar o cadastro. |
| Pré-condições |  |
| Pós-condições | O aluno poderá realizar o login. |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. Informa os dados cadastrais: nome, e-mail, senha. |  |
|  | 2. Armazenar dados inseridos e validar cadastro. |

Fonte: Do Próprio Autor, 2023.

Quadro 6 – Descrição do caso de uso “Manter Cadastro”

|  |  |
| --- | --- |
| Nome do caso de uso | Manter Cadastro |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas pelo aluno/administrador para editar, excluir e/ou visualizar cadastro. |
| Pré-condições | O aluno precisa estar “logado” para executar uma das funções propostas. |
| Pós-condições |  |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. Informa os dados do login: e-mail e senha. |  |
|  | 2. Consultar dados do usuário. |
|  | 3. Se houver um cadastro com os dados informados, a conta será exibida. |
| 4. Se necessário alterar ou inativar os dados do usuário. |  |
|  | 5. Se necessário gravar atualizações. |
| Restrições/Validações | 1. Todos os dados precisam ser validos. |
| 2. Os campos e-mail e senha são obrigatórios. |

Fonte: Do Próprio Autor, 2023.

Quadro 7 – Descrição do caso de uso “Realizar Login”

|  |  |
| --- | --- |
| Nome do caso de uso | Realizar Login |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas para o aluno e o administrador realizarem o login. |
| Pré-condições |  |
| Pós-condições |  |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. Informa dados do login: e-mail e senha. |  |
|  | 1. Consultar dados do usuário. |
|  | 1. Se os dados forem válidos, realiza o login. |
| Restrições/Validações | 1. Os dados do login precisam ser válidos. |

Fonte: Do Próprio Autor, 2023.

Quadro 8 – Descrição do caso de uso “Manter Quiz”

|  |  |
| --- | --- |
| Nome do caso de uso | Manter Quiz |
| Caso de uso geral |  |
| Ator principal | Administrador |
| Ator secundário |  |
| Resumo | Este caso de uso descreve as etapas percorridas para o administrador criar, editar, excluir ou visualizar o questionário. |
| Pré-condições | O administrador precisa estar “logado” para a manipulação do quiz. |
| Pós-condições |  |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. Informar dados do login: e-mail e senha |  |
|  | 1. Consultar dados do administrador. |
|  | 1. Se os dados forem válidos, apresenta quiz. |
| 1. Se necessário, alterar, inativar ou inserir dados do quiz. |  |
|  | 1. Se necessário, armazenar alterações do quiz. |
| Restrições/Validações | 1. Os dados do quiz precisam ser válidos. |

Fonte: Do Próprio Autor, 2023.

Quadro 9 – Descrição do caso de uso “Realizar Quiz”

|  |  |
| --- | --- |
| Nome do caso de uso | Realizar Quiz |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas para o aluno realizar o questionário. |
| Pré-condições | O usuário precisa estar “logado” para a realização do quiz. |
| Pós-condições | Visualizar informações do quiz (Caso de Uso Visualizar Quiz). |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. Selecionar questionário. |  |
|  | 2. Apresentar questionário. |
| 3. Inserir respostas do questionário. |  |
|  | 4. Apresentar resultados questionário. |
|  |  |

Fonte: Do Próprio Autor, 2023.

Quadro 10 – Descrição do caso de uso “Resultado Quiz”

|  |  |
| --- | --- |
| Nome do caso de uso | Visualizar Resultado Quiz |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas para o aluno visualizar o resultado do quiz. |
| Pré-condições |  |
| Pós-condições |  |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. Informa dados do login: e-mail e senha. |  |
|  | 1. Consultar dados do usuário. |
|  | 1. se os dados forem válidos, apresenta informações dos questionários respondidos. |
| Restrições/Validações | 1. Os dados do login precisam ser válidos. |
| 1. O usuário precisa ter um questionário realizado para a visualização do resultado. |

Fonte: Do Próprio Autor, 2023.

Quadro 11 – Descrição do caso de uso “Visualizar Modelo 3D”

|  |  |
| --- | --- |
| Nome do caso de uso | Visualizar Modelo 3D |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas para o aluno visualizar um modelo 3D. |
| Pré-condições | O usuário precisa escanear um QR Code para visualizar um modelo 3D. |
| Pós-condições |  |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. O usuário escaneia o QR Code. |  |
|  | 1. Consultar QR Code na base de dados. |
|  | 1. Se o QR Code for válido, apresenta modelo 3D. |
| Restrições/Validações | 1. O QR Code deve constar na base de dados do sistema |

Fonte: Do Próprio Autor, 2023.

Quadro 12 – Descrição do caso de uso “Escanear QR Code”

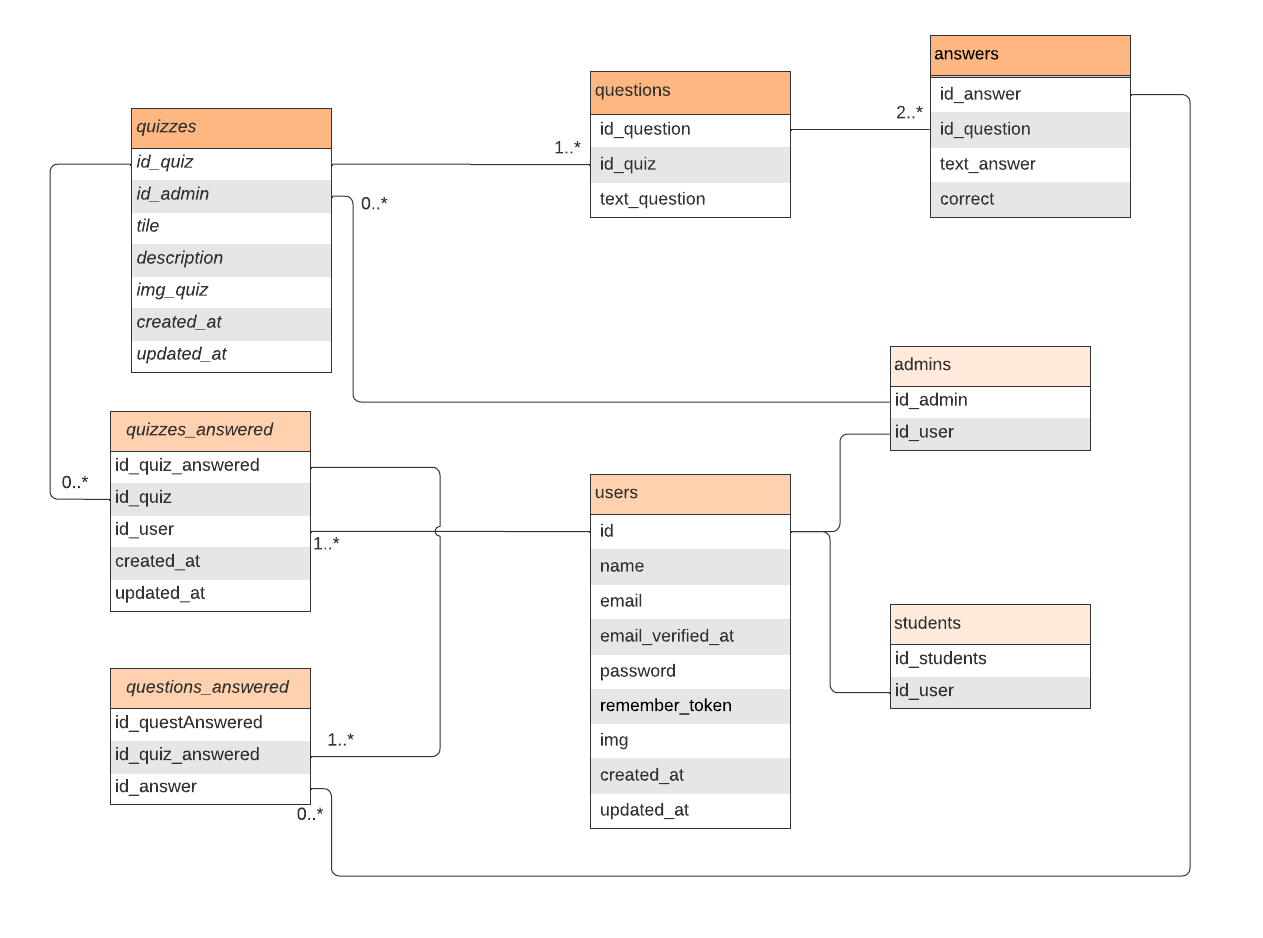
|  |  |
| --- | --- |
| Nome do caso de uso | Escanear QR Code |
| Caso de uso geral |  |
| Ator principal | Aluno |
| Ator secundário | Administrador |
| Resumo | Este caso de uso descreve as etapas percorridas para o aluno escanear QR Code. |
| Pré-condições | O usuário precisa instalar o aplicativo para escanear o QR Code. |
| Pós-condições |  |
| Fluxo principal | |
| Ações do ator | Ações do sistema |
| 1. O usuário escaneia o QR Code. |  |
|  | 1. Consultar QR Code na base de dados. |
|  | 1. Se o QR Code for válido, apresenta modelo 3D. |
| Restrições/Validações | 1. O QR Code deve constar na base de dados do sistema |

Fonte: Do Próprio Autor, 2023.

3.2 Modelo Entidade-Relacionamento (MER)

Na figura a seguir, é possível observar a estrutura do banco de dados, suas tabelas e seus respectivos atributos. O diagrama é denominado como Modelo de Entidade-Relacionamento (MER).

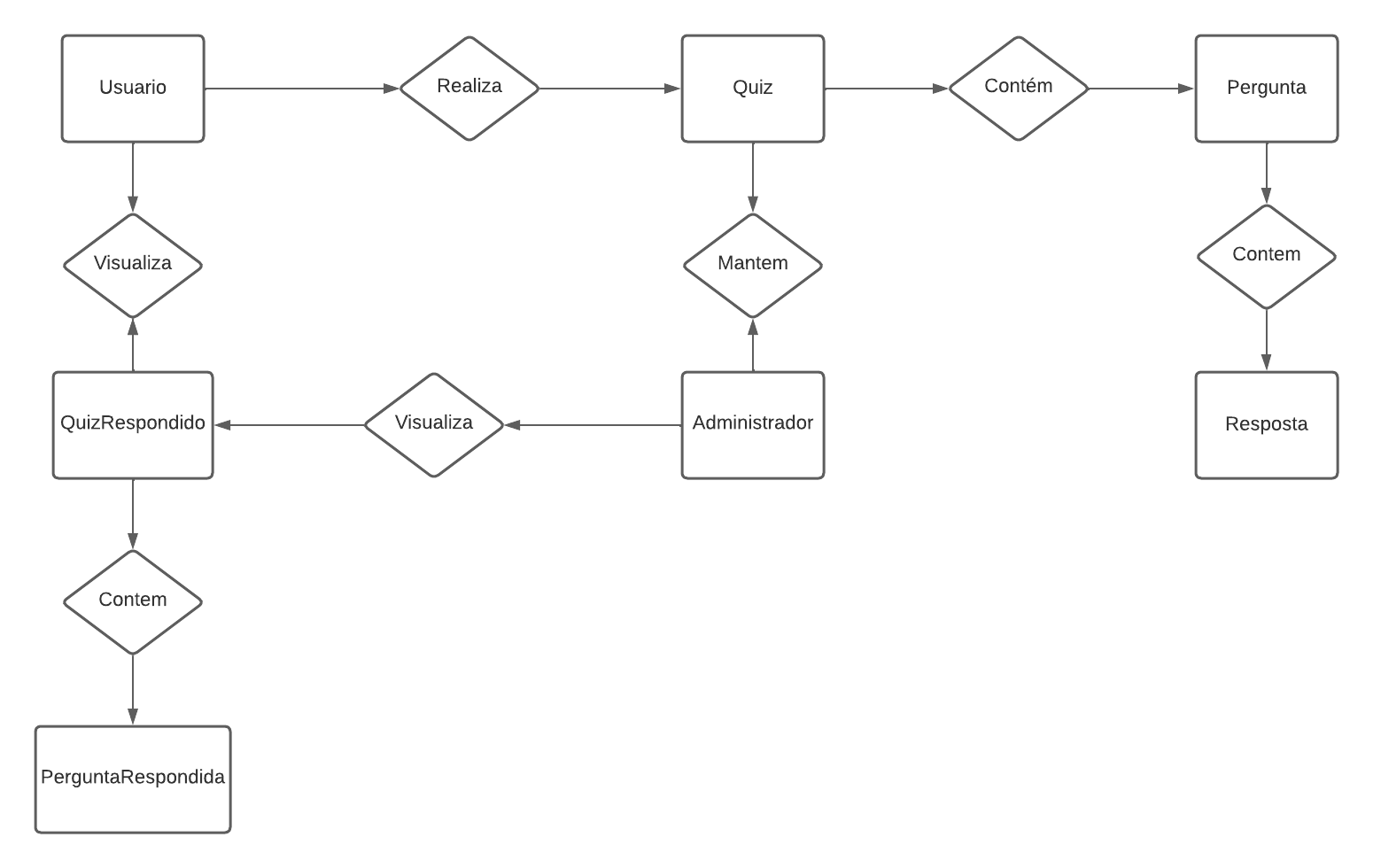
Figura 33 – Modelo Entidade Relacionamento

Fonte: Do Próprio Autor, 2023.

3.3 Diagrama Entidade-Relacionamento (DER)

Na figura a seguir, é possível observar a visão gráfica do que foi descrito no MER, através do Diagrama Entidade-Relacionamento (DER), representando o administrador, aluno e quiz. Para a visualização de como eles se relacionam.

Figura 34 – Diagrama Entidade Relacionamento

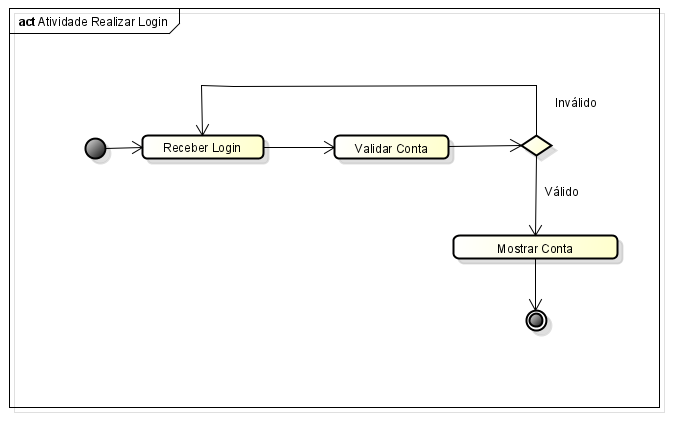


Fonte: Do Próprio Autor, 2023.

3.4 Diagrama de Atividade

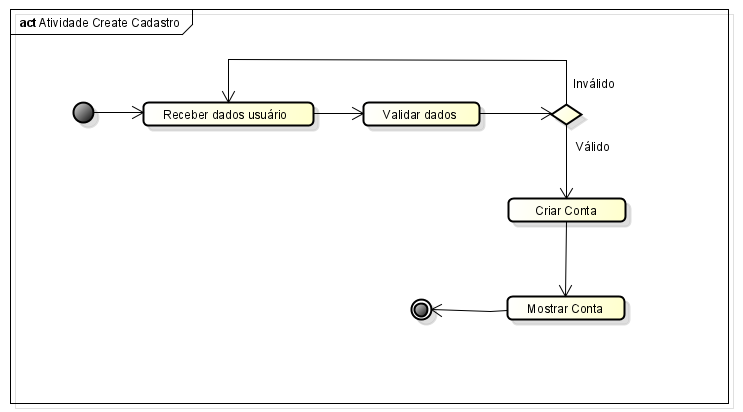
Com os diagramas a seguir, é possível analisar o fluxo de controle de cada funcionalidade.

Figura 35– Diagrama de atividade do Realizar Login

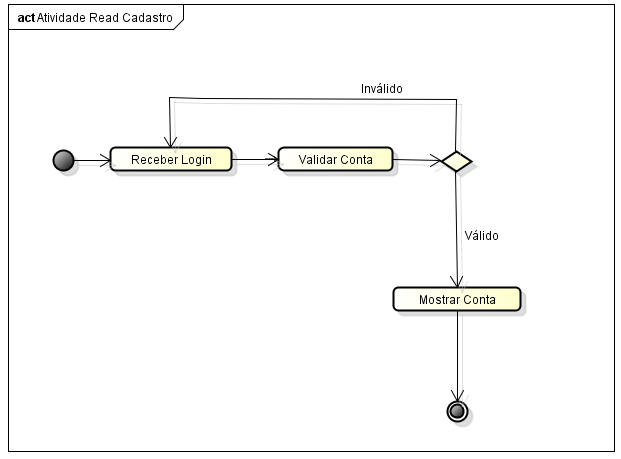


Fonte: Do Próprio Autor, 2023.

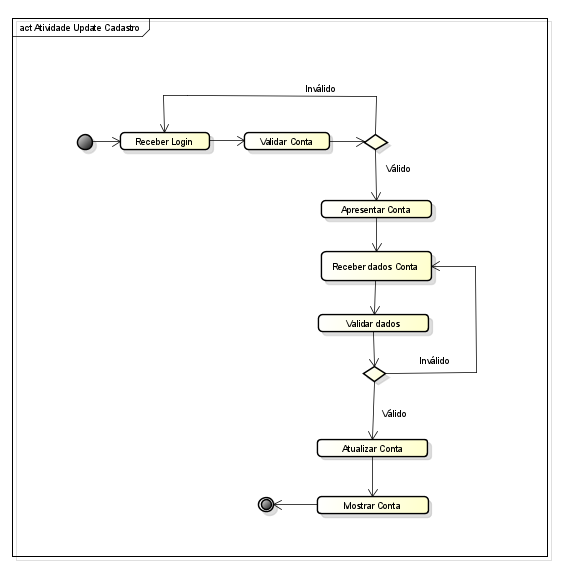
A figura 35 mostra o diagrama de atividade do login, onde o sistema precisa receber os dados de login e validar os dados. Caso os dados sejam válidos, ele mostra a conta do usuário.

Figura 36 – Diagrama de atividade do Criar CadastroFonte: Do Próprio Autor, 2023.

A figura a seguir apresenta o diagrama de atividade para criar um cadastro. O sistema precisa receber os dados pessoais do usuário e validar se as informações estão corretas, podendo assim, criar sua conta.

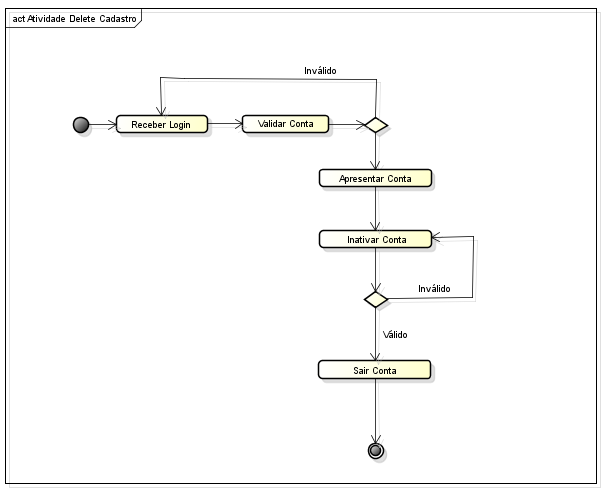
Figura 37 – Diagrama de atividade do Visualizar CadastroFonte: Do Próprio Autor, 2023.

A imagem acima mostra o diagrama de atividade para visualizar o cadastro. O usuário deve realizar o login e se os dados forem validados corretamente, ele poderá acessar sua conta.

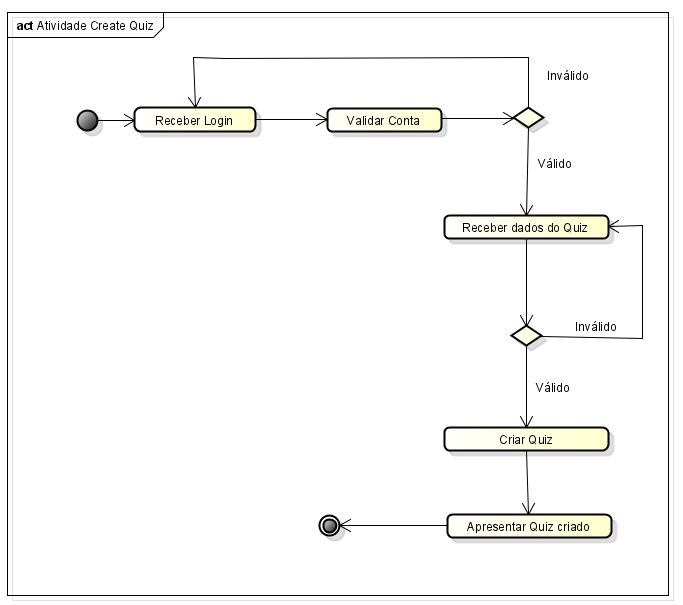
Figura 38 – Diagrama de atividade do Editar Cadastro

Fonte: Do Próprio Autor, 2023

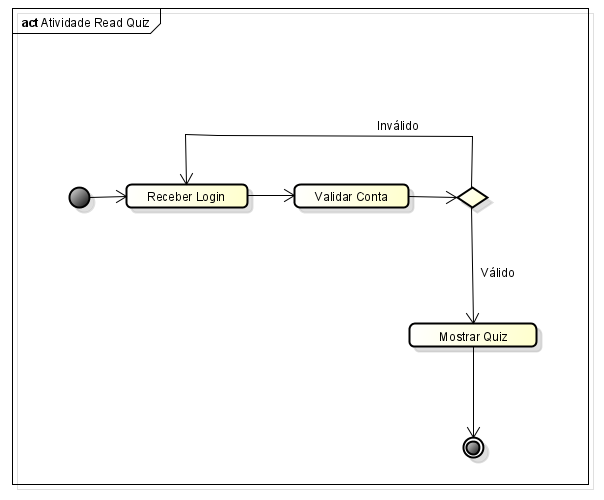
A figura 38 é o diagrama de atividade para editar o cadastro. Primeiramente, o sistema precisa receber o login e validar a conta. Se a conta for validada, ele permite que o usuário informe os dados que vão ser alterados, por fim valida os novos dados e apresenta a conta modificada.

Figura 39 – Diagrama de atividade do Deletar CadastroFonte: Do Próprio Autor, 2023.

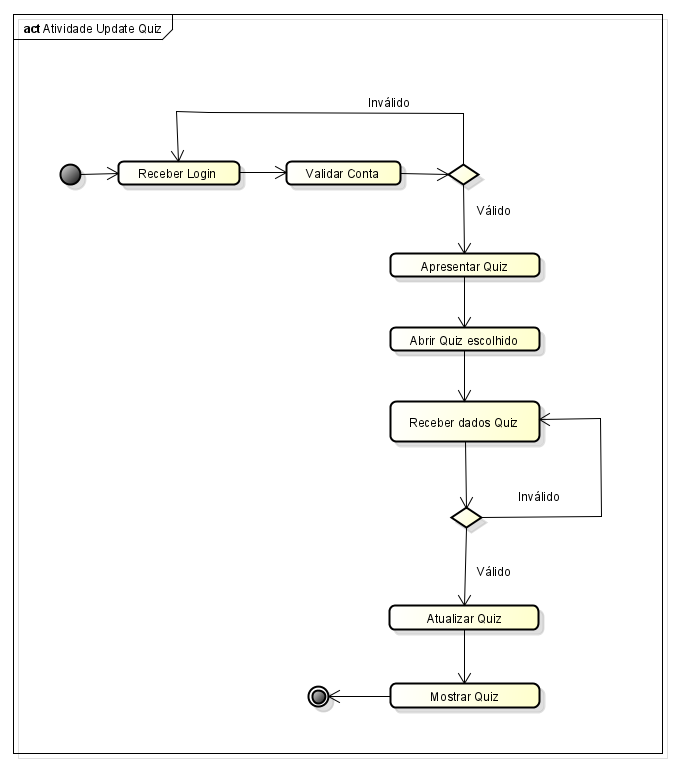
A imagem anterior mostra o diagrama de atividade para deletar o cadastro do usuário. O usuário precisa estar logado, então ele insere seu login. Após a validação do login, é apresentado a conta com a opção de exclusão. Por fim a conta é excluída caso o usuário decida saindo da conta.

Figura 40 – Diagrama de atividade do Criar QuizFonte: Do Próprio Autor, 2023.

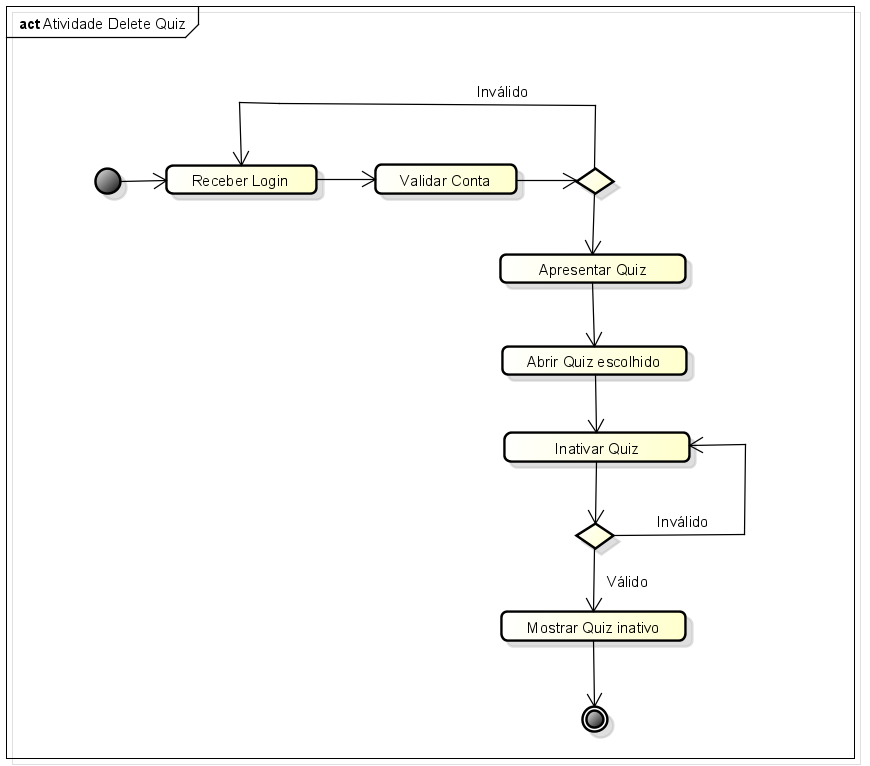
A imagem 40 apresenta o diagrama de atividade para a criação de um quiz. Caso o usuário esteja logado e seja um administrador, o sistema permite que ele cadastre um novo quiz, recebendo suas informações e mostrando o resultado no final.

Figura 41 – Diagrama de atividade do Visualizar QuizFonte: Do Próprio Autor, 2023

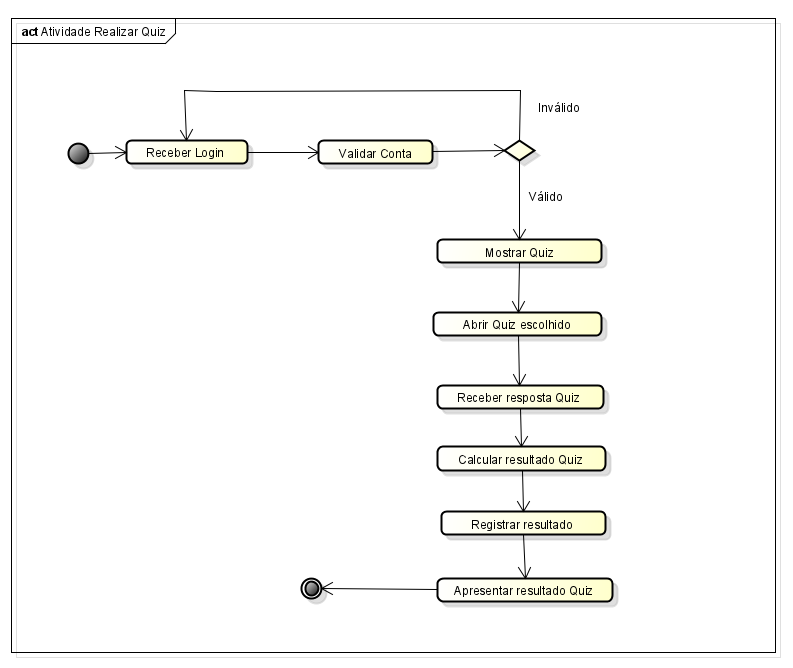
A imagem anterior apresenta o diagrama de atividade para a visualização de um quiz. O usuário deve estar logado para acessar a funcionalidade. Caso o usuário seja validado, é apresentado o quis, se o usuário não apresentar um cadastro, o mesmo é direcionado para a criação de um.

Figura 42 – Diagrama de atividade do Editar QuizFonte: Do Próprio Autor, 2023.

A imagem acima é a representação do diagrama de atividade da edição do quiz, o usuário deverá realizar seu login, selecionar um quiz para editar e atualizar o questionário, depois da atualização o administrador pode visualizar o quiz atualizado.

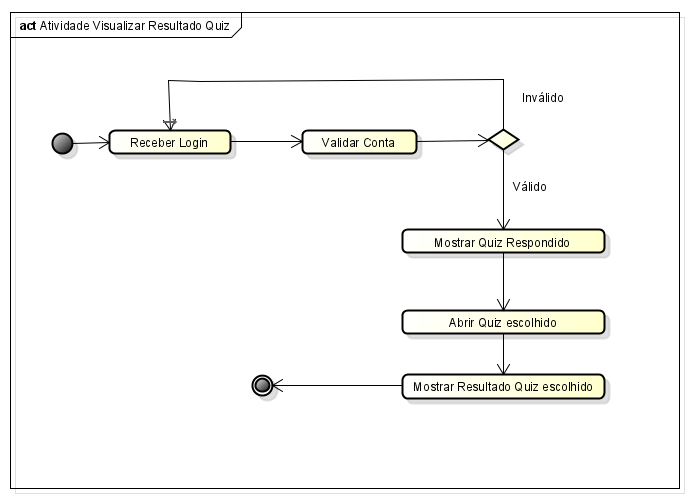
Figura 44 – Diagrama de atividade do Deletar QuizFonte: Do Próprio Autor, 2023.

A imagem 44 apresenta o diagrama de atividade do deletar quiz, onde entrando com uma conta de administrador, um usuário pode excluir um quiz que foi cadastrado. Primeiro o usuário informa seu login, sendo capaz de visualizar o quis e em seguida apagá-lo.

Figura 45 – Diagrama de atividade do Realizar QuizFonte: Do Próprio Autor, 2023.

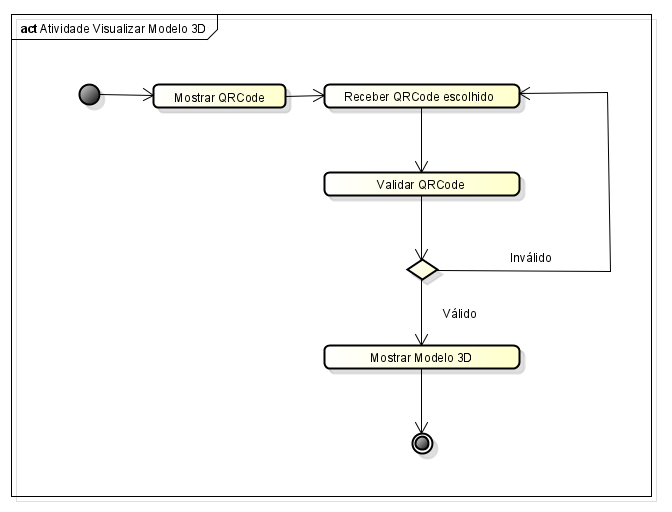
O diagrama de atividade da imagem acima representa realização do quiz, onde o usuário realiza login para responder o quis, logo após o sistema deve calcular a quantidade de acertos, mostrando o resultado.

Figura 46 – Diagrama de atividade do Visualizar Resultado do Quiz

Fonte: Do Próprio Autor, 2023.

O diagrama de atividade do visualizar o resultado do quiz é representado pela figura acima, onde o usuário logado acessa sua conta e pode consultar seu progresso de questionários que já foram realizados.

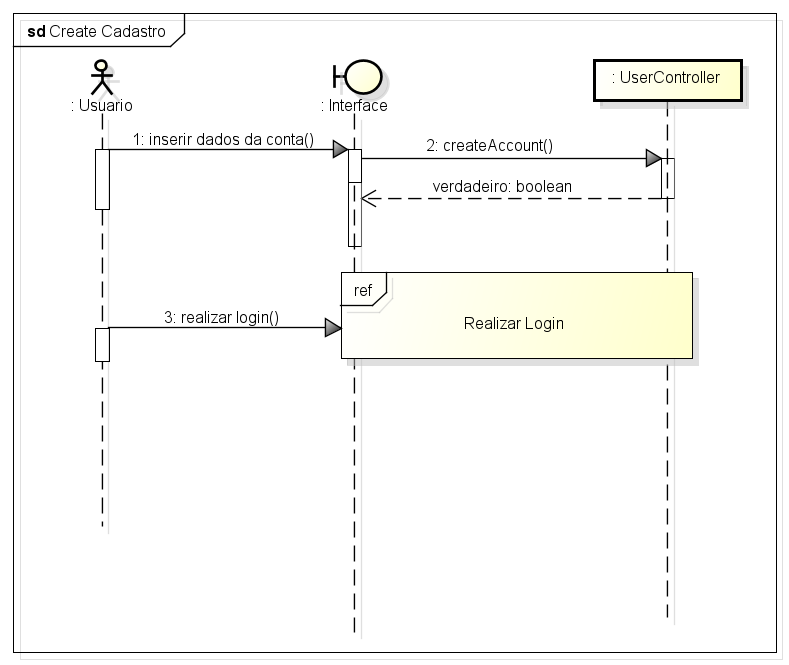
Figura 47 – Diagrama de atividade do Visualizar Modelo 3D

Fonte: Do Próprio Autor, 2023.

A imagem acima mostra o diagrama de atividade do visualizar modelo 3D dentro do aplicativo. Para acessar a funcionalidade, basta o usuário escanear o QR Code com a câmera do aplicativo e será mostrado o modelo em RA.

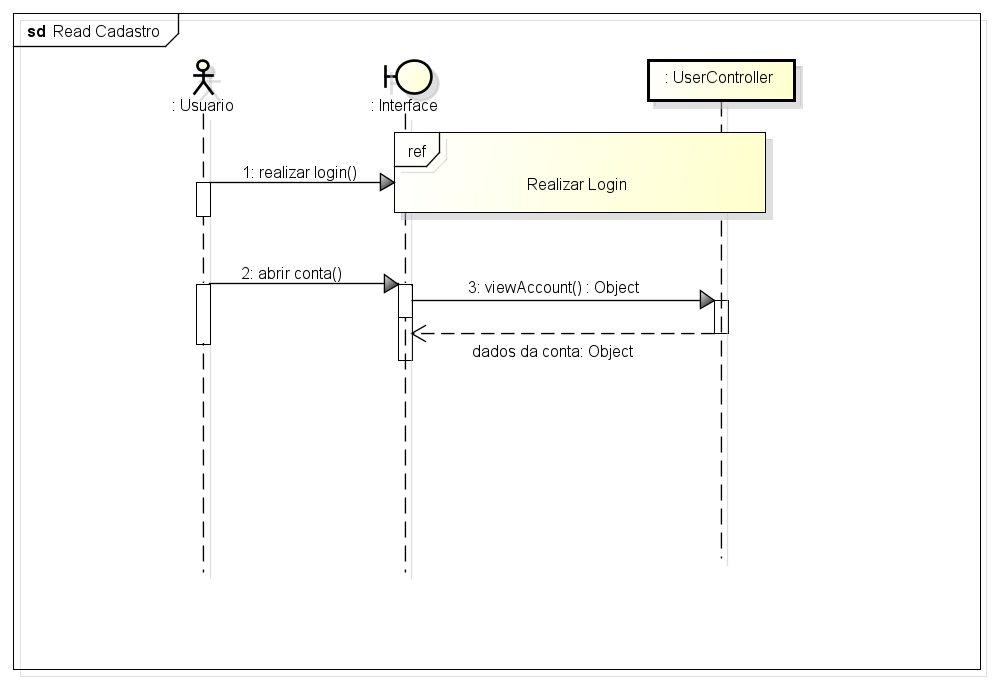
3.5 Diagramas de Sequência

Com os diagramas a seguir, é possível analisar a organização dos componentes em ordem temporal e como eles interagem entre si.

Figura 48 – Diagrama de sequência do Criar Cadastro Fonte: Do Próprio Autor, 2023.

O diagrama de sequência para a criação de cadastro é representado pela figura 48. Para criar uma conta, o usuário chama a função para criar conta, que retorna verdadeiro, caso seja executado com sucesso. Por fim é apresentado os dados da conta criada.

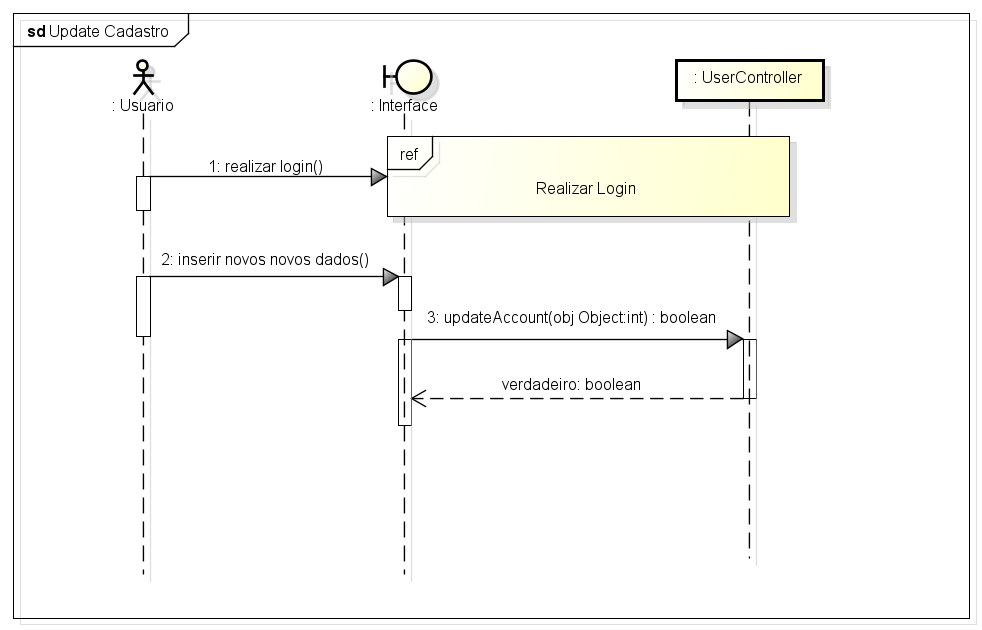
Figura 49 – Diagrama de sequência do Visualizar Cadastro



Fonte: Do Próprio Autor, 2023.

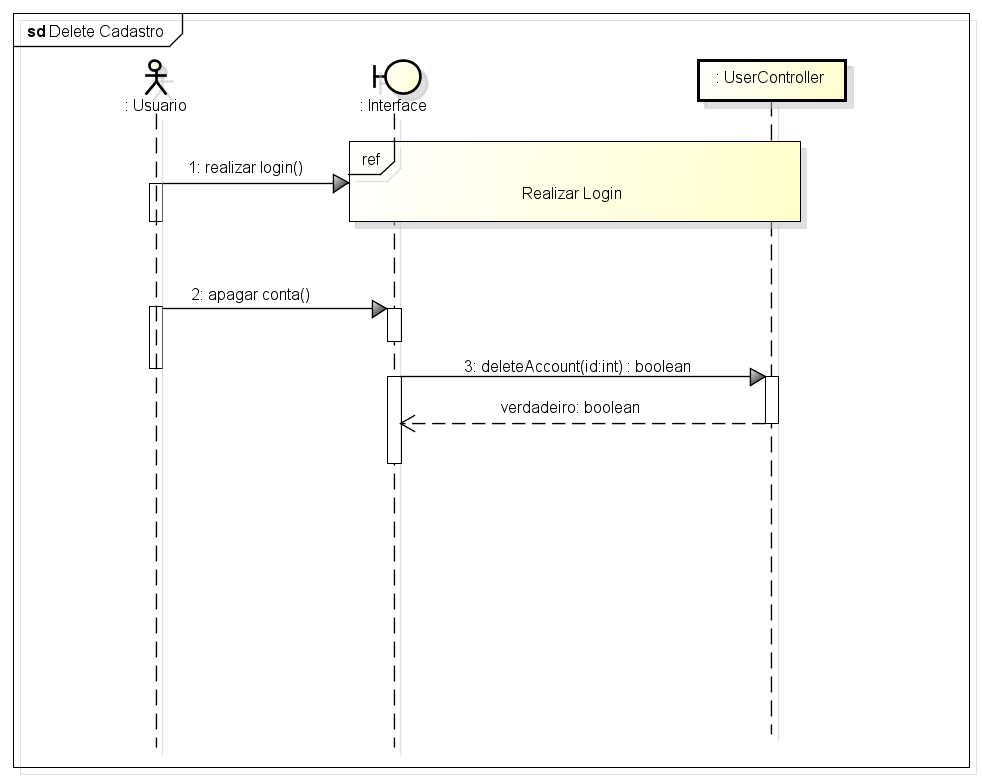
A imagem anterior mostra o diagrama de sequência para a visualização do cadastro, onde o usuário realiza o login, se o login for verdadeiro (usuário cadastrado), o sistema mostra a conta com suas respectivas informações.

Figura 50 – Diagrama de sequência do Atualizar Cadastro

Fonte: Do Próprio Autor, 2023.

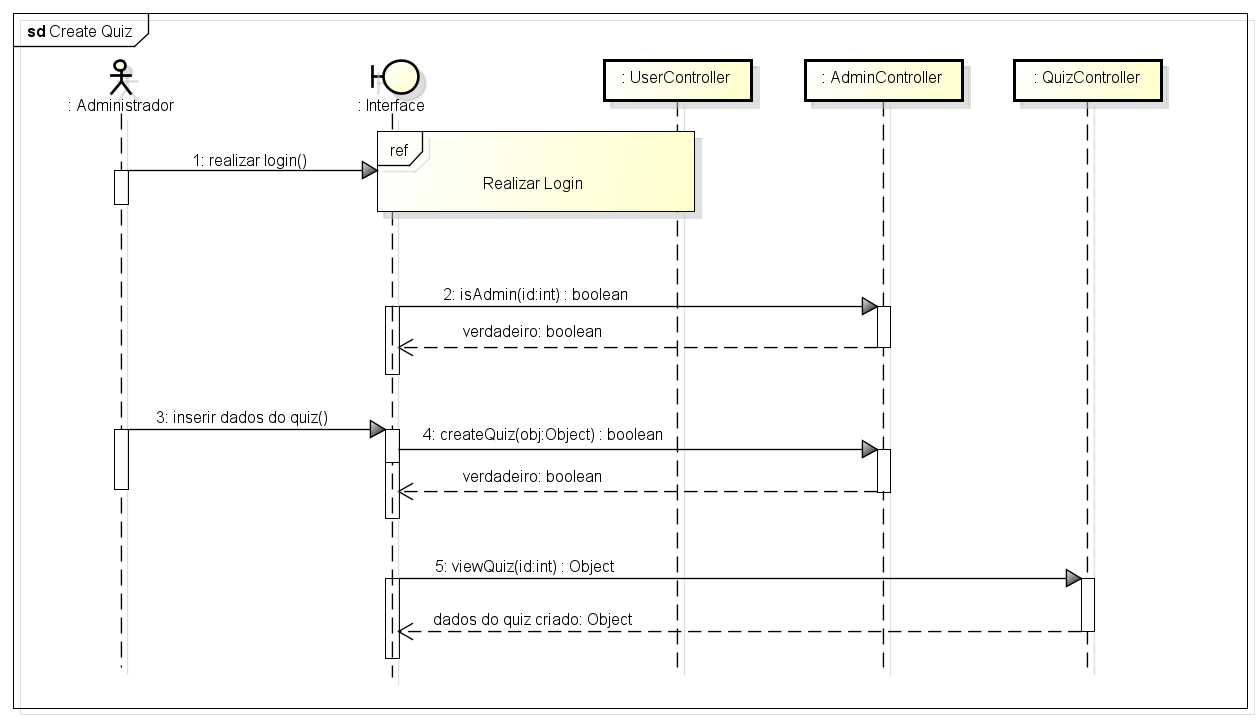
A figura acima mostra o diagrama de sequência para atualizar o cadastro. Primeiramente o usuário precisa fazer login e acessar sua conta. Por fim ele pode alterar suas informações e atualizar.

Figura 51 – Diagrama de sequência do Deletar Cadastro

Fonte: Do Próprio Autor, 2023.

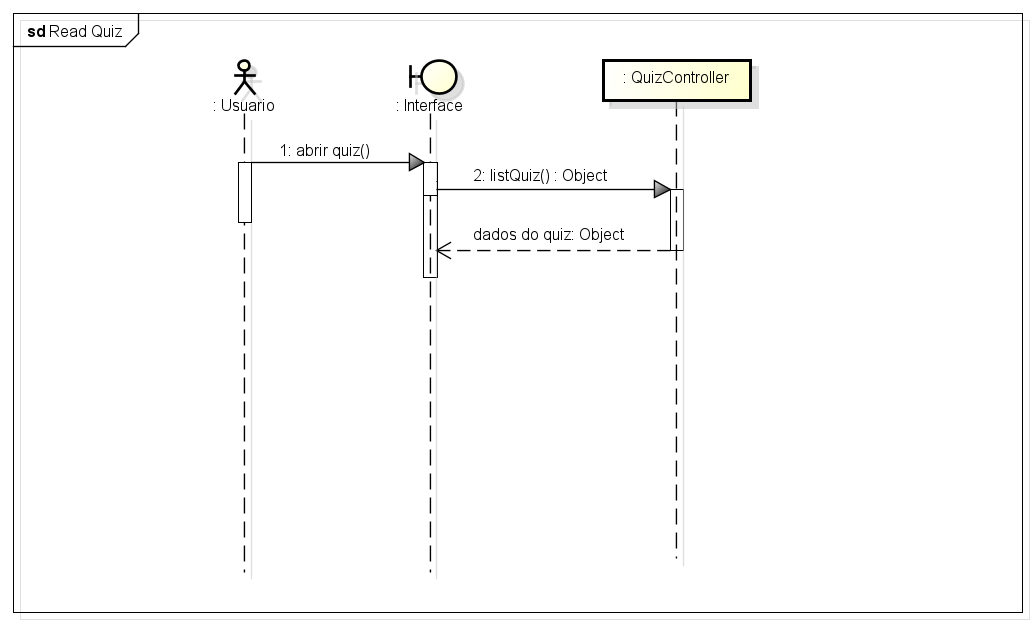
O diagrama de sequência acima mostra como será a exclusão de um cadastro. Com o usuário logado e visualizando a sua conta, ele tem a opção de apagar a conta, deletando também todo seu progresso.

Figura 52 – Diagrama de sequência do Criar Quiz

Fonte: Do Próprio Autor, 2023.

O diagrama de sequência do criar quiz é representado pela imagem acima. O sistema precisa receber o login para a verificação da conta, deve receber as informações do quiz e por fim receber as informações das questões com as respectivas respostas, apresentando o quiz criado.

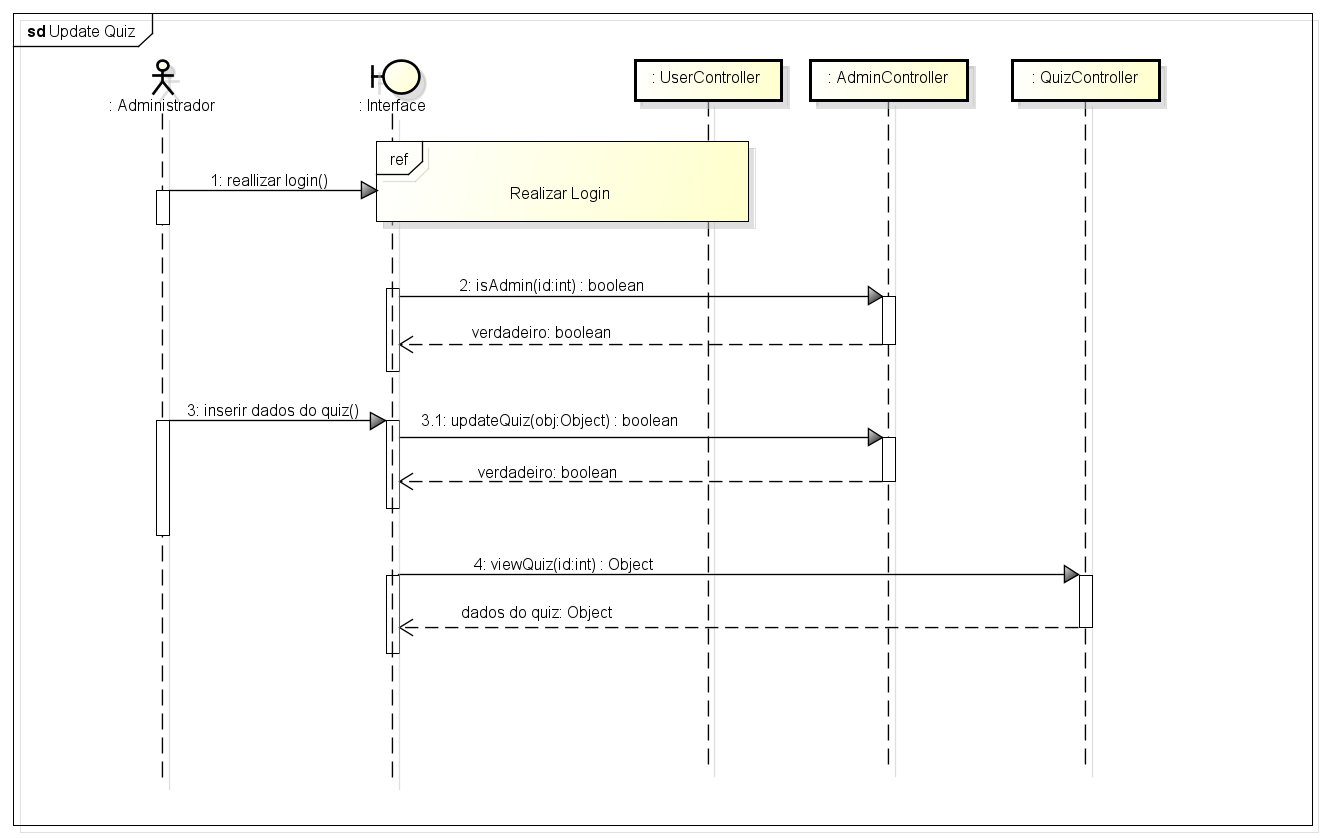
Figura 53 – Diagrama de sequência do Visualizar Quiz



Fonte: Do Próprio Autor, 2023.

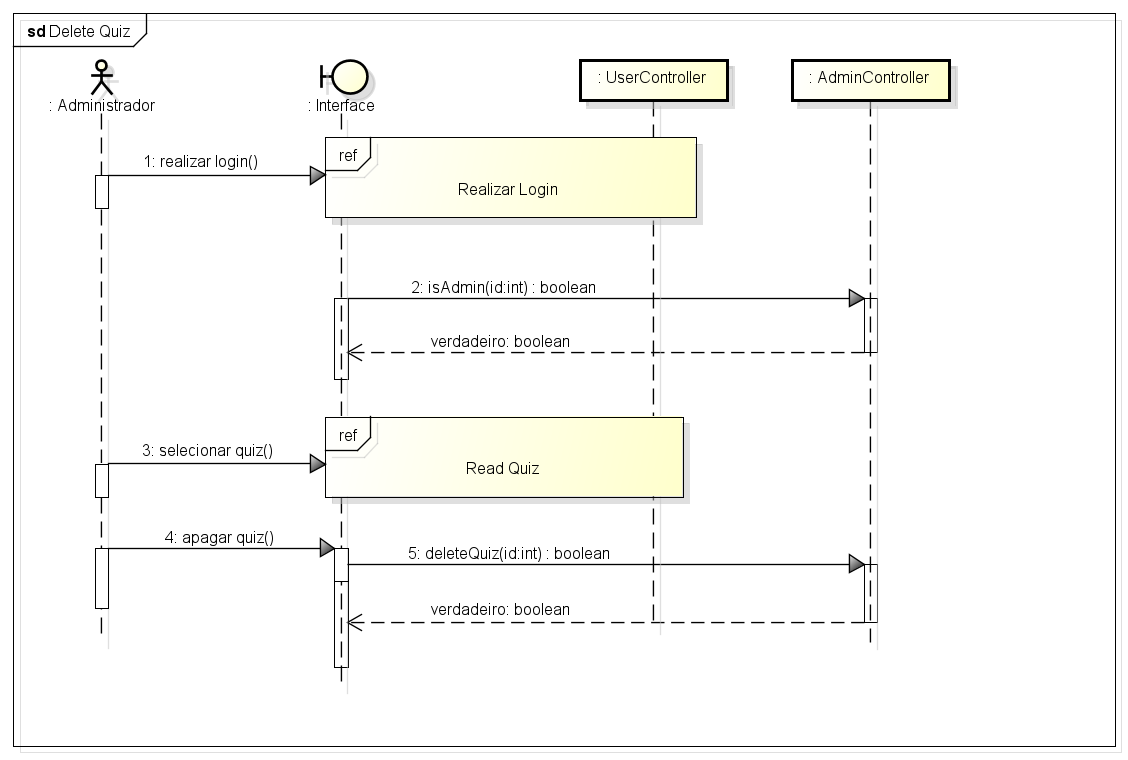
A figura acima mostra o diagrama de sequência da visualização do quiz, o usuário não precisa estar logado para listar todos os questionários que estão cadastrados no sistema.

Figura 54 – Diagrama de sequência do Editar Quiz

Fonte: Do Próprio Autor, 2023.

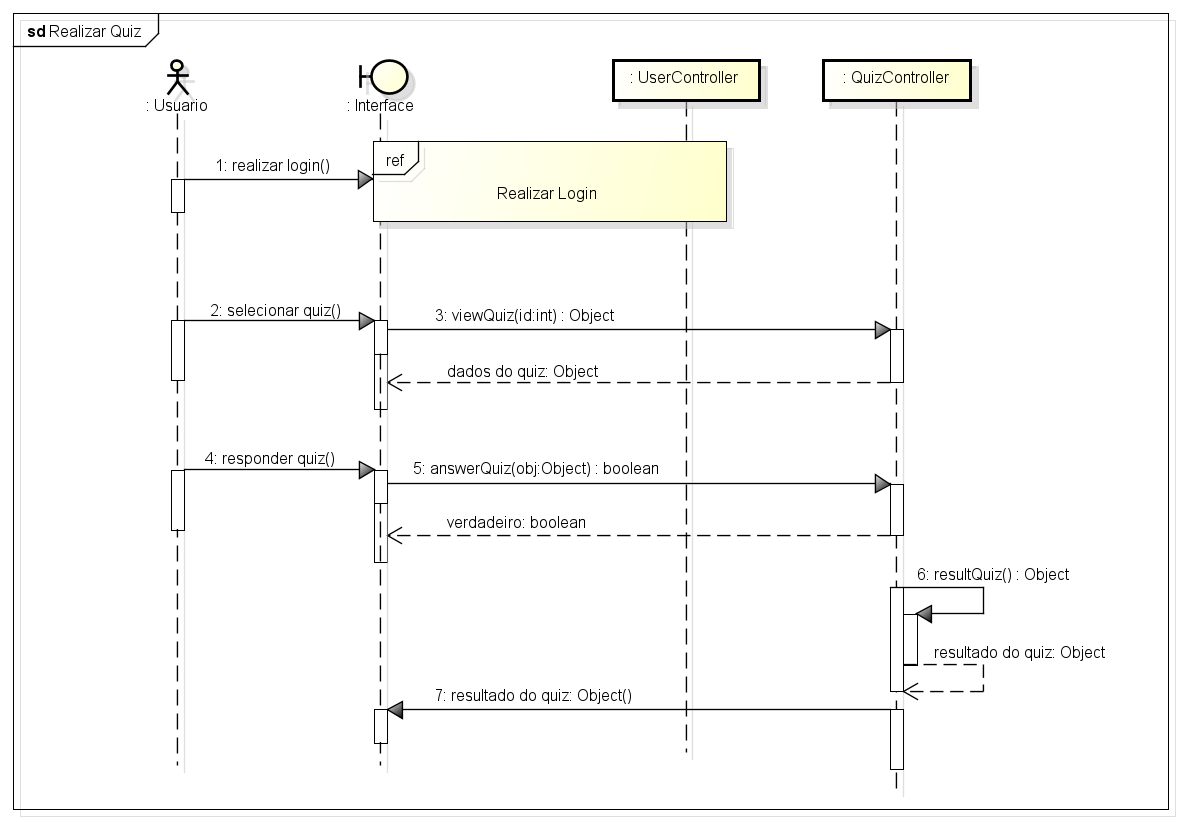
A seguir é mostrado o diagrama de sequência do editar quiz. O sistema deve receber o login do usuário, verificar se ele é um administrador e atualizar o quiz com as informações inseridas pelo administrador.

Figura 55 – Diagrama de sequência do Deletar Quiz

Fonte: Do Próprio Autor, 2023

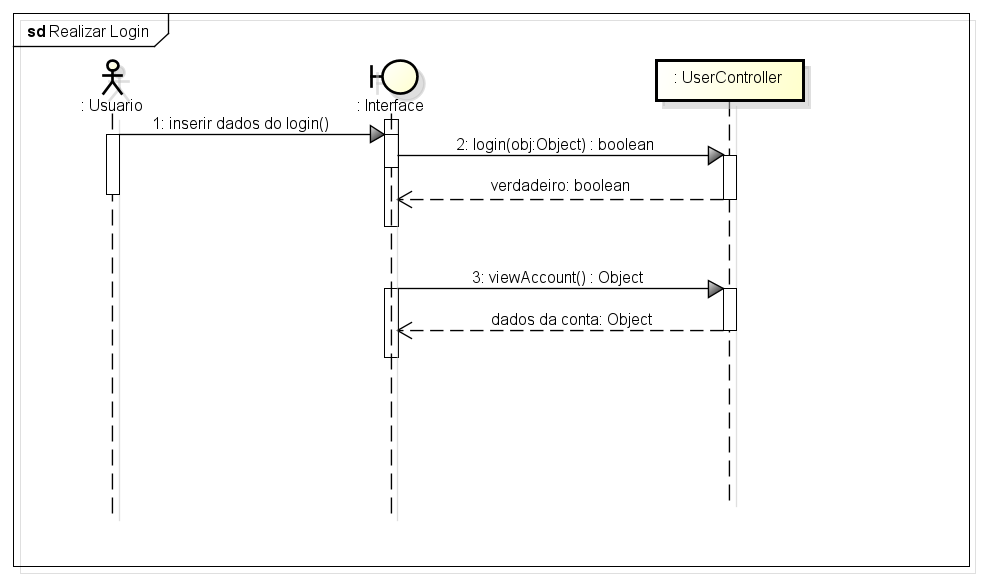
O diagrama de sequência anterior apresenta a função de deletar um quiz. O administrador deverá estar logado no sistema e poderá excluir o quiz, deletando também todo o progresso que os alunos responderam sobre esse questionário.

Figura 56 – Diagrama de sequência do Realizar Quiz

Fonte: Do Próprio Autor, 2023

Para a realização do quiz, foi criado o diagrama de sequência acima que mostra essa ação. Depois de logado, o usuário poderá responder o quiz, escolhendo uma resposta para cada pergunta e visualizar seu progresso no final.

Figura 57 – Diagrama de sequência do Realizar Login



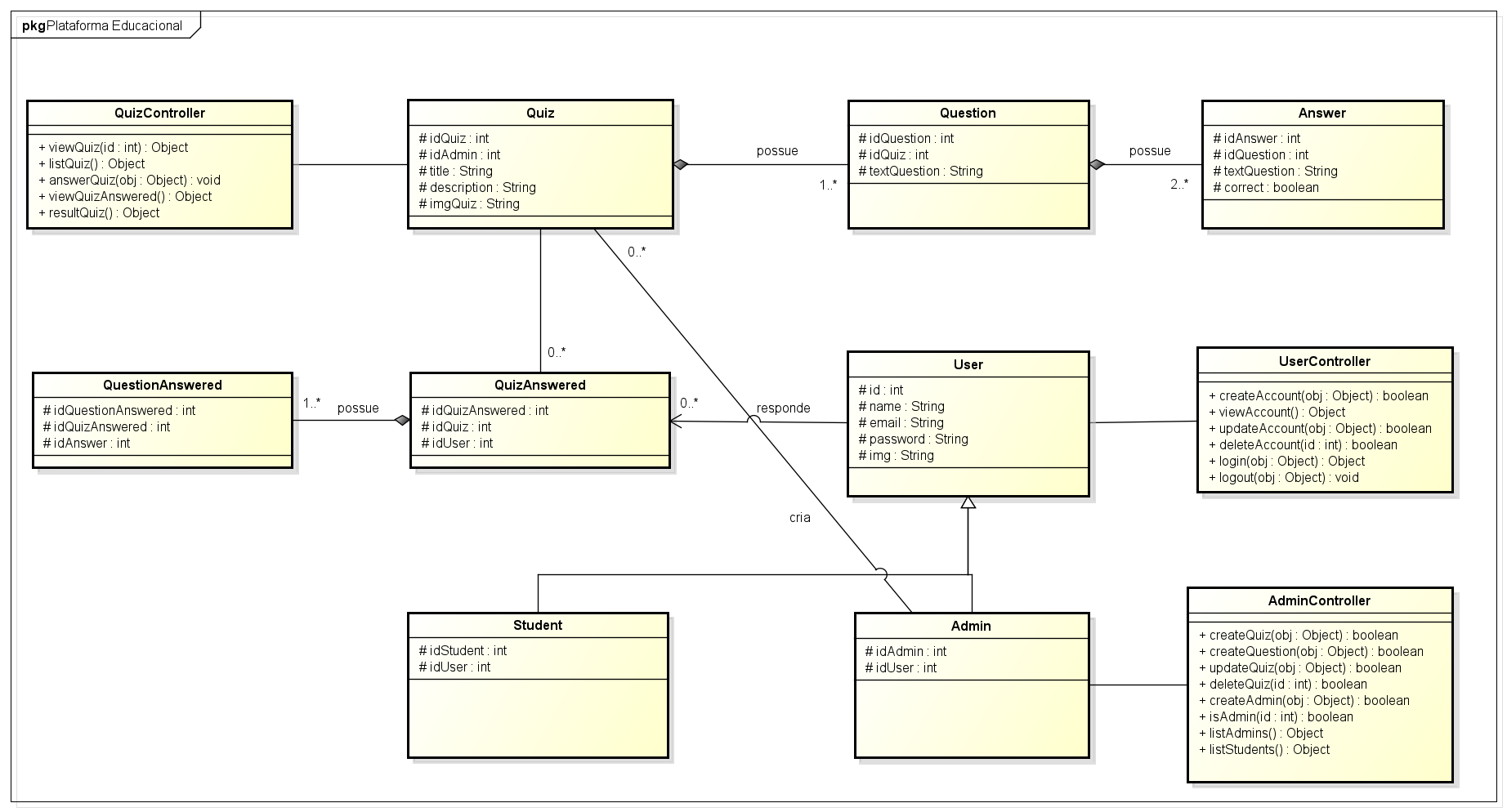
Fonte: Do Próprio Autor, 2023.

O diagrama de sequência a seguir mostra como será realizado o login dentro do site. O usuário deverá informar seu e-mail e senha e o sistema irá validar os dados pra consultar se há um cadastro.

3.6 Diagrama de Classe

Com o diagrama a seguir, é possível observar as classes que constituem os sistemas, junto de seus respectivos atributos, métodos e relacionamentos.

Figura 58 – Diagrama de Classe



Fonte: Do Próprio Autor, 2023.

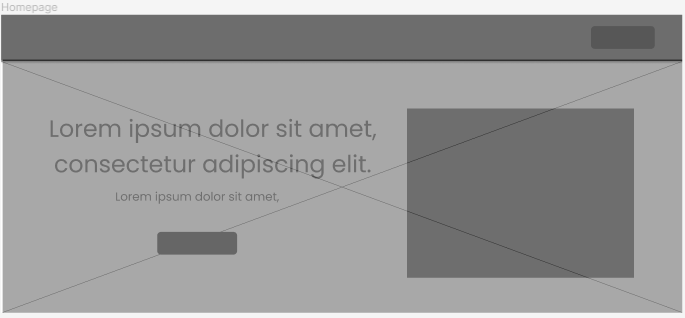
4. DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo, é apresentado o processo de criação da interface gráfica do Science AR, por meio de ferramentas que auxiliam na criação de protótipos para o desenvolvimento das telas da aplicação.

4.1 Wireframes de Baixa Qualidade

Sendo o esboço inicial da página de um site ou aplicativo, os wireframes foram essenciais para o planejamento do design e a usabilidade do projeto, tornando o desenvolvimento fluído.

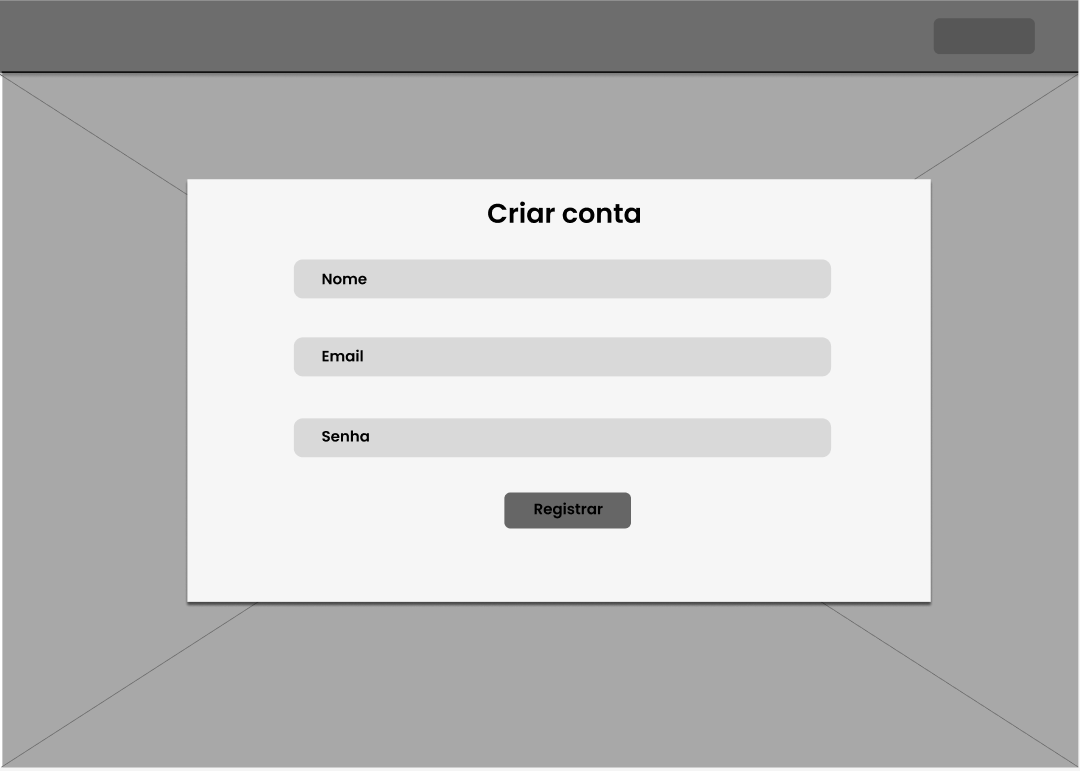
Figura 59 - Wireframe de baixa fidelidade “Página Inicial”



Fonte: Do Próprio Autor, 2023.

A construção desses wireframes envolve a utilização de ferramentas simples, como lápis e papel ou até mesmo por meio de aplicativos/sites especializados, para esboçar as posições de elementos como áreas de conteúdo.

Figura 60 – Wireframe de baixa fidelidade “Realizar Cadastro do Aluno”



Fonte: Do Próprio Autor, 2023.

Para simular o conteúdo e seu posicionamento sem detalhes textuais, os wireframes de baixa fidelidade frequentemente utilizam o "Lorem Ipsum". Essa sequência fictícia de palavras em latim preenche os espaços destinados ao texto, sem a preocupação de criar cópias de texto que poderiam mudar ao longo do desenvolvimento da aplicação.

Figura 61 – Wireframe de baixa fidelidade “Realizar Cadastro do Administrador”



Fonte: Do Próprio Autor, 2023.

No processo de montagem dos wireframes, o objetivo é capturar a arquitetura geral da interface, muitas vezes sendo desenhados em papel. Conforme o processo avança, ferramentas de design gráfico são utilizadas para criar wireframes digitais mais precisos, como nesse capítulo, onde os mesmos foram projetados através do figma.

Figura 62 – Wireframe de baixa fidelidade “Realizar Login”



Fonte: Do Próprio Autor, 2023.

Elementos como botões representam interações e ações possíveis, de acordo com sua localização, tamanho e função. Esses botões podem ser representados por formas simples como retângulos ou rótulos genéricos

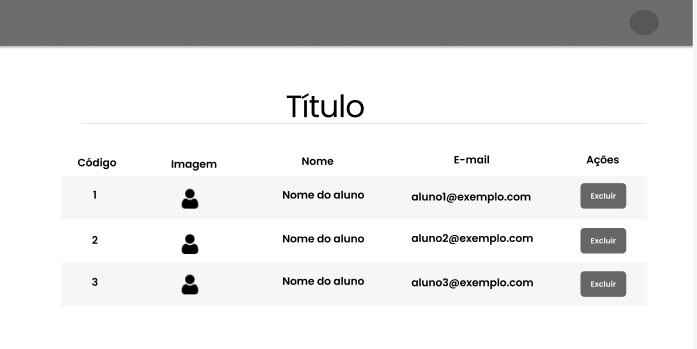
Figura 63 – Wireframe de baixa fidelidade “Visualizar Perfil”



Fonte: Do Próprio Autor, 2023.

Por exemplo, o botão "Atualizar", pode ser representado como um simples retângulo, que posteriormente, terá seu design modificado e sua função designada.

Figura 64 – Wireframe de baixa fidelidade “Visualizar Aluno”



Fonte: Do Próprio Autor, 2023.

A barra de navegação, conhecida como navbar, define a estrutura geral do site ou aplicativo, destacando as principais seções e páginas disponíveis. Sendo representada sem detalhamentos gráficos.

Figura 65 – Wireframe de baixa fidelidade “Visualizar Administrador”



Fonte: Do Próprio Autor, 2023.

Nos wireframes de baixa fidelidade as imagens foram simplificadas para manter o foco da estrutura. O posicionamento delas é representado através de retângulos.

Figura 66 – Wireframe de baixa fidelidade “Sobre Nós”



Fonte: Do Próprio Autor, 2023.

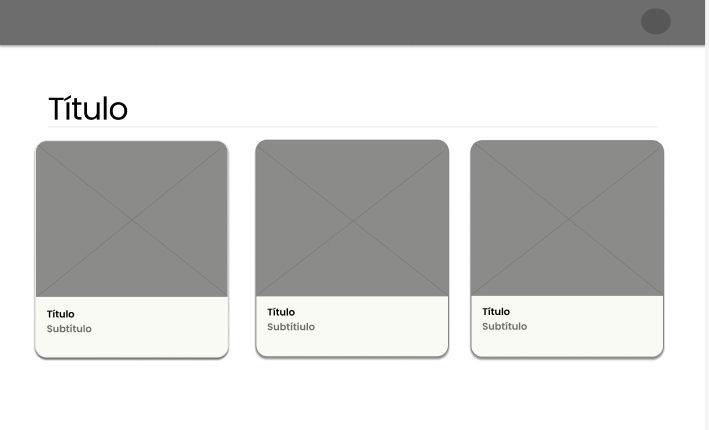
Figura 67 – Wireframe de baixa fidelidade “Realidade Aumentada”



Fonte: Do Próprio Autor, 2023.

Os wireframes de baixa fidelidade não apresentam as funcionalidades presentes no sistema, apenas representam a posição de seus elementos. Uma página pode conter mais de um wireframe, de acordo com suas funcionalidades e consequentemente suas mudanças.

Figura 68 – Wireframe de baixa fidelidade “Visualizar Materiais”



Fonte: Do Próprio Autor, 2023.

Pode-se observar que os wireframes de baixa fidelidade não apresentam cores principais do site, utilizando apenas cores neutras, como tons de cinza e branco.

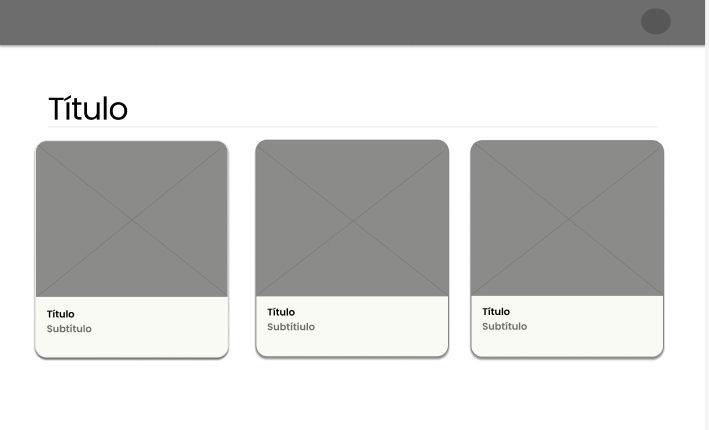
Figura 69 – Wireframe de baixa fidelidade “Realizar Aula”



Fonte: Do Próprio Autor, 2023.

Com esses esboços, é possível obter uma visão geral da navegação da aplicação, indicando o fluxo do site/aplicativo e quais rotas o mesmo pode seguir. Pode-se observar isso na figura 69.

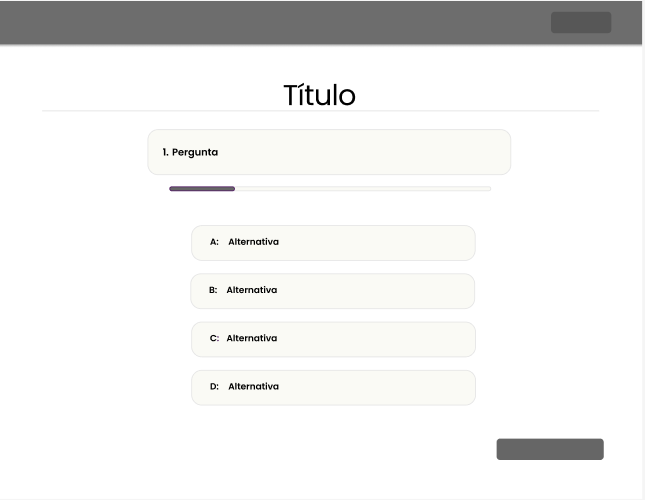
Figura 70 – Wireframe de baixa fidelidade “Visualizar Questionários”



Fonte: Do Próprio Autor, 2023.

Os cards representados na imagem acima sugerem diferentes categorias de quizzes que são apresentados no site, navegando por outras telas.

Figura 71 – Wireframe de baixa fidelidade “Realizar Questionário”



Fonte: Do Próprio Autor, 2023.

Nos wireframes, a acessibilidade do usuário deve ser priorizada, tornando os elementos textuais e visuais nítidos, melhorando a visibilidade sem cometer exageros.

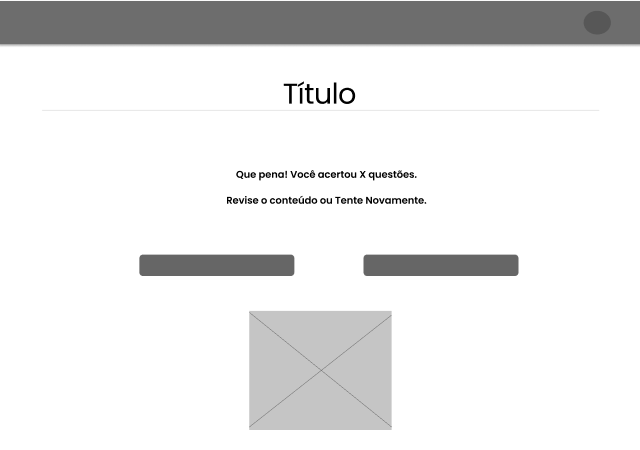
Figura 72 – Wireframe de baixa fidelidade “Resultado Questionário (positivo)”



Fonte: Do Próprio Autor, 2023.

Através das figuras 72 e 73 é possível analisar como a mesma página pode ser modificada de acordo com as navegações seguidas pelo usuário.

Figura 73– Wireframe de baixa fidelidade “Resultado Questionário (negativo)”



Fonte: Do Próprio Autor, 2023.

As mesmas podem sofrer alterações textuais, visuais e sendo direcionado para diferentes páginas, disponibilizando ao usuário diferentes rotas e diferentes experiências.

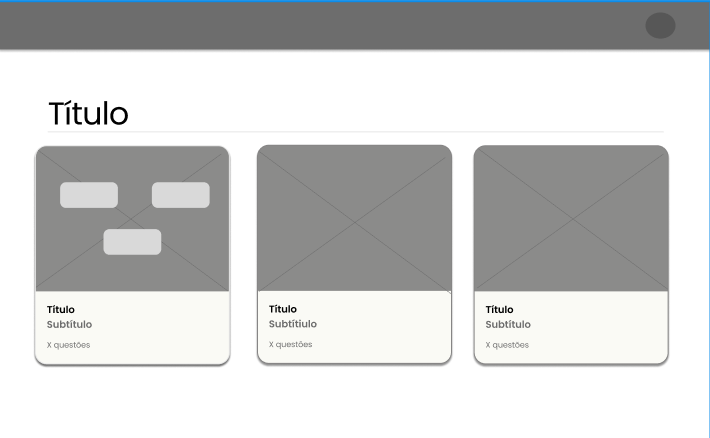
Figura 74 – Wireframe de baixa fidelidade “Visualizar Progresso”



Fonte: Do Próprio Autor, 2023.

Caixas retangulares foram utilizadas para representar um conjunto de secções/menu de contexto dentro da mesma página.

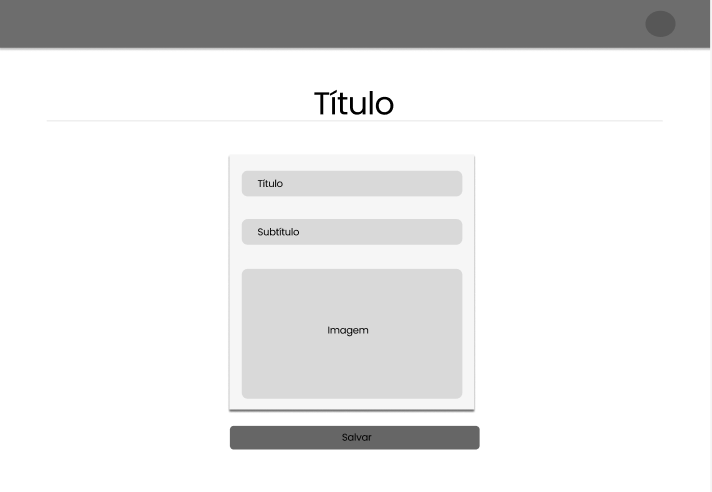
Figura 75 – Wireframe de baixa fidelidade “Alterar Questionários”



Fonte: Do Próprio Autor, 2023.

Os wireframes de baixa fidelidade também podem representar o hover através de retângulos, simbolizando a passagem do mouse no elemento.

Figura 76 – Wireframe de baixa fidelidade “Criar Questionário”



Fonte: Do Próprio Autor, 2023.

Caixas retangulares foram utilizadas para representar campos de entrada, como caixas de texto, campos de imagens, normalmente acompanhadas do placeholder, indicando ao usuário qual informação o mesmo deve inserir.

Figura 77 – Wireframe de baixa fidelidade “Adicionar Questões”



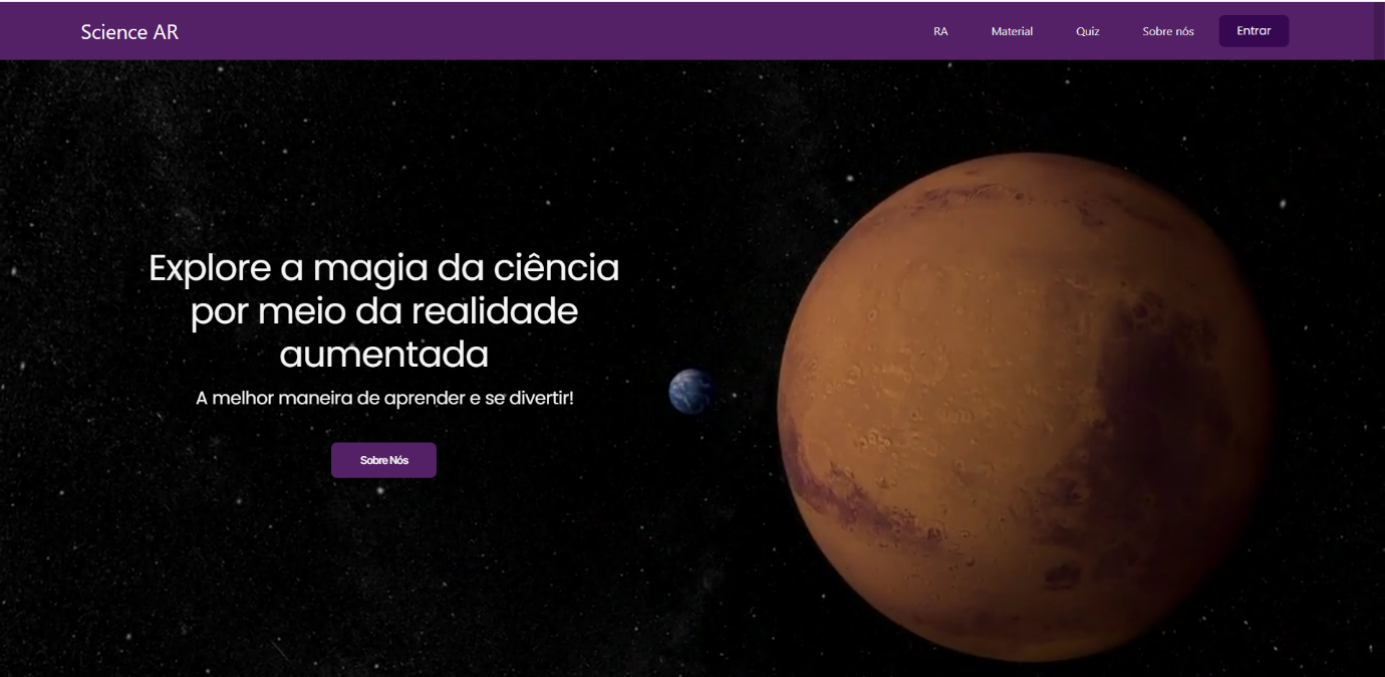
Fonte: Do Próprio Autor, 2023.

No wireframe acima, retângulos menores são utilizados para representar botões de rádio, permitindo a seleção de uma opção entre várias, geralmente acompanhadas por rótulos.

4.2 Prototipação de Interface

A página principal do Science AR possui uma animação de universo para capturar a atenção do usuário. Há um menu de navegação que facilita a exploração das demais páginas do site. E por fim, no layout é possível observar um botão onde o usuário poderá ser direcionado para descobrir mais sobre conteúdos de ciências.

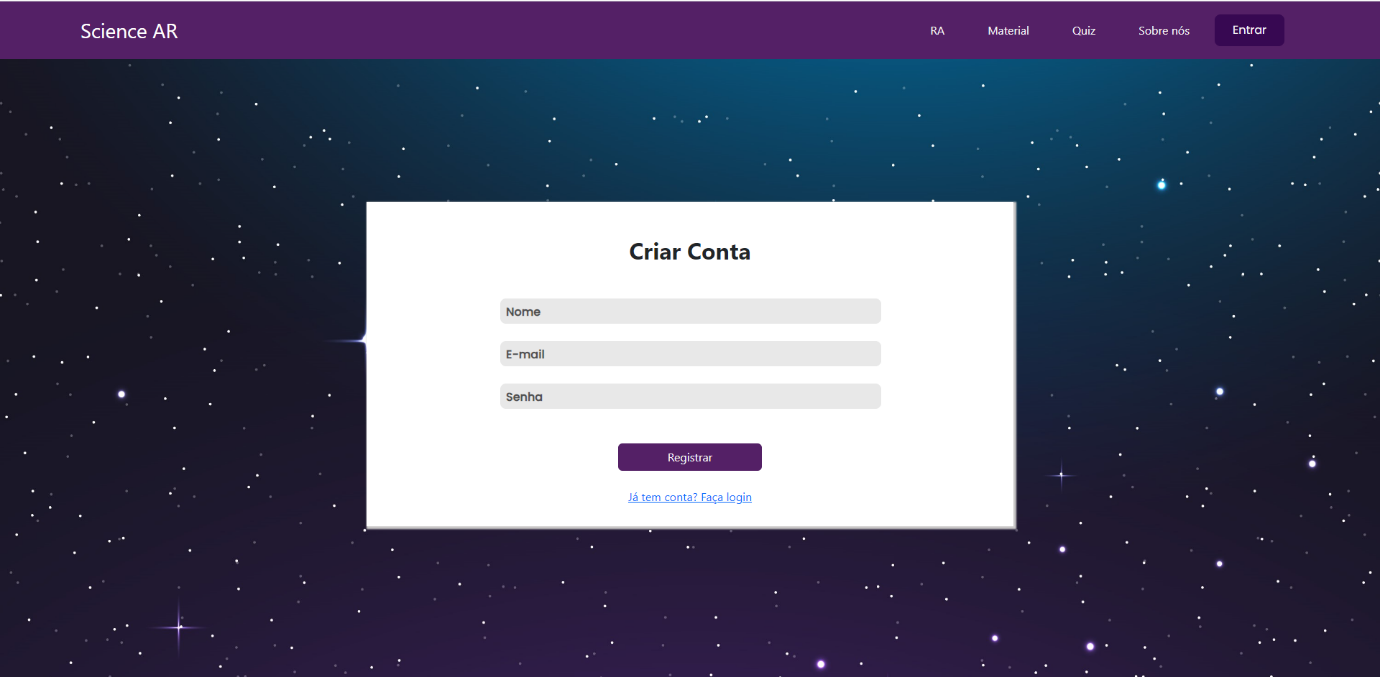
Figura 78 - Página Inicial



Fonte: Do Próprio Autor, 2023.

A interface abaixo representa o design da tela de cadastro, onde será possível realizar um cadastro, inserindo todos os dados requisitados nos campos. Após o preenchimento, a próxima etapa é salvar as informações clicando no botão “Registrar”, dando fim no procedimento.

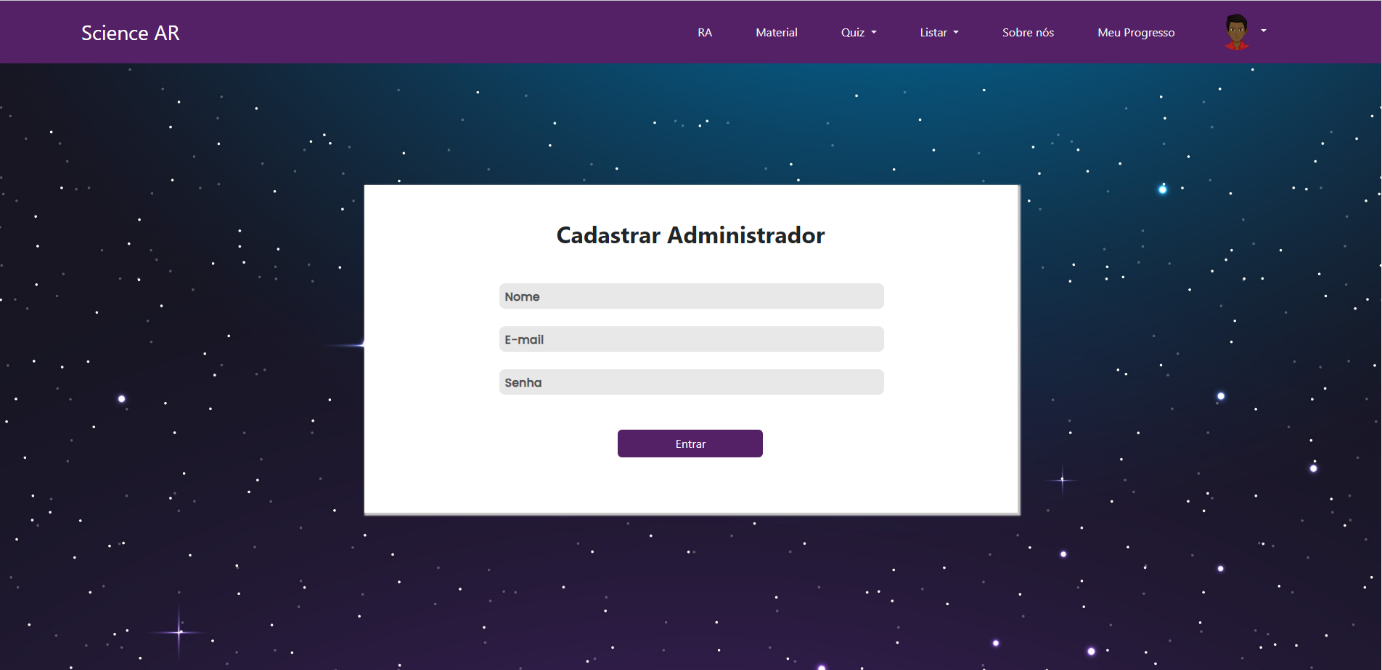
Figura 79 – Cadastro do Aluno



Fonte: Do Próprio Autor, 2023.

A interface de cadastro de administrador corresponde à tela que viabiliza a ação de um administrador ao cadastrar outros indivíduos para colaborar no gerenciamento do site de forma conjunta. Ao efetuar o cadastro de outros administradores, torna-se viável a administração dos questionários, bem como a gestão total dos usuários registrados no sistema.

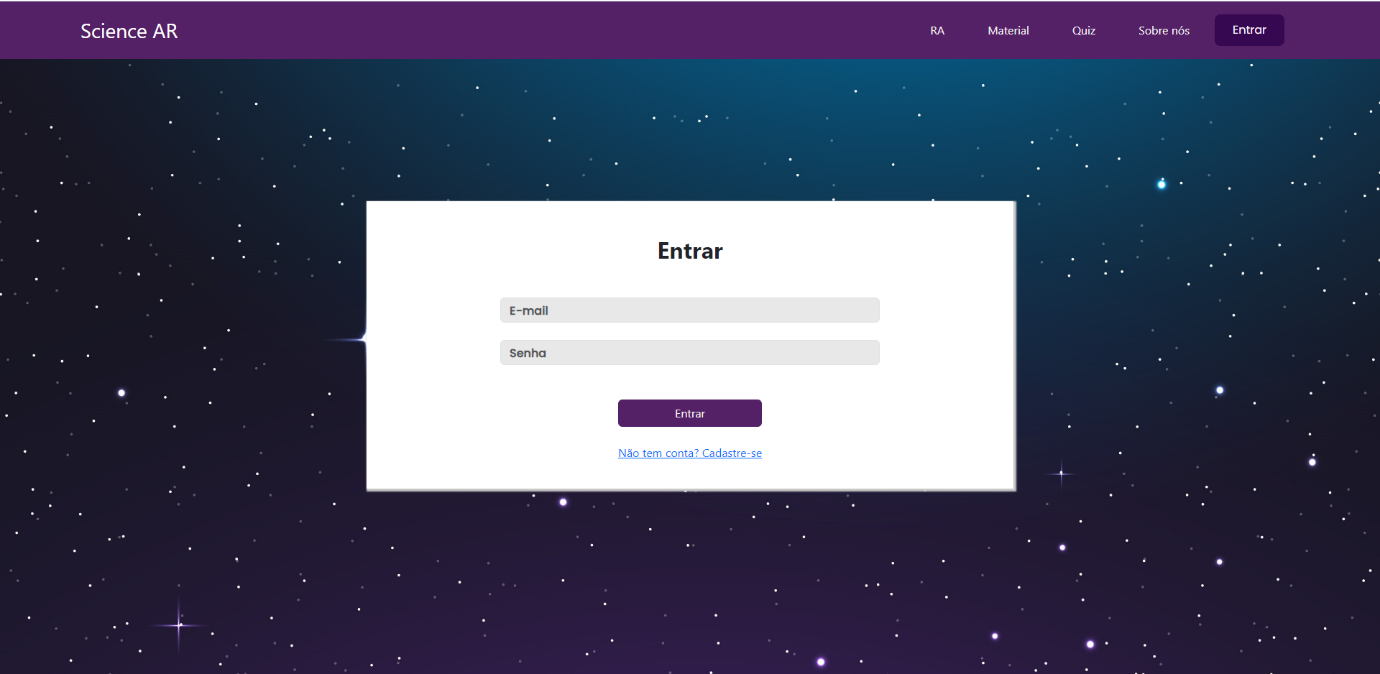
Figura 80 – Cadastro do Administrador



Fonte: Do Próprio Autor, 2023.

Na figura a seguir, está representada a interface de login, onde o usuário que já possui o registro no site, pode inserir as informações correspondentes nos campos, dando fim no processo, clicando no botão “Entrar”. No layout, o usuário encontrará um link, para que o mesmo possa criar uma conta caso ele não esteja cadastrado.

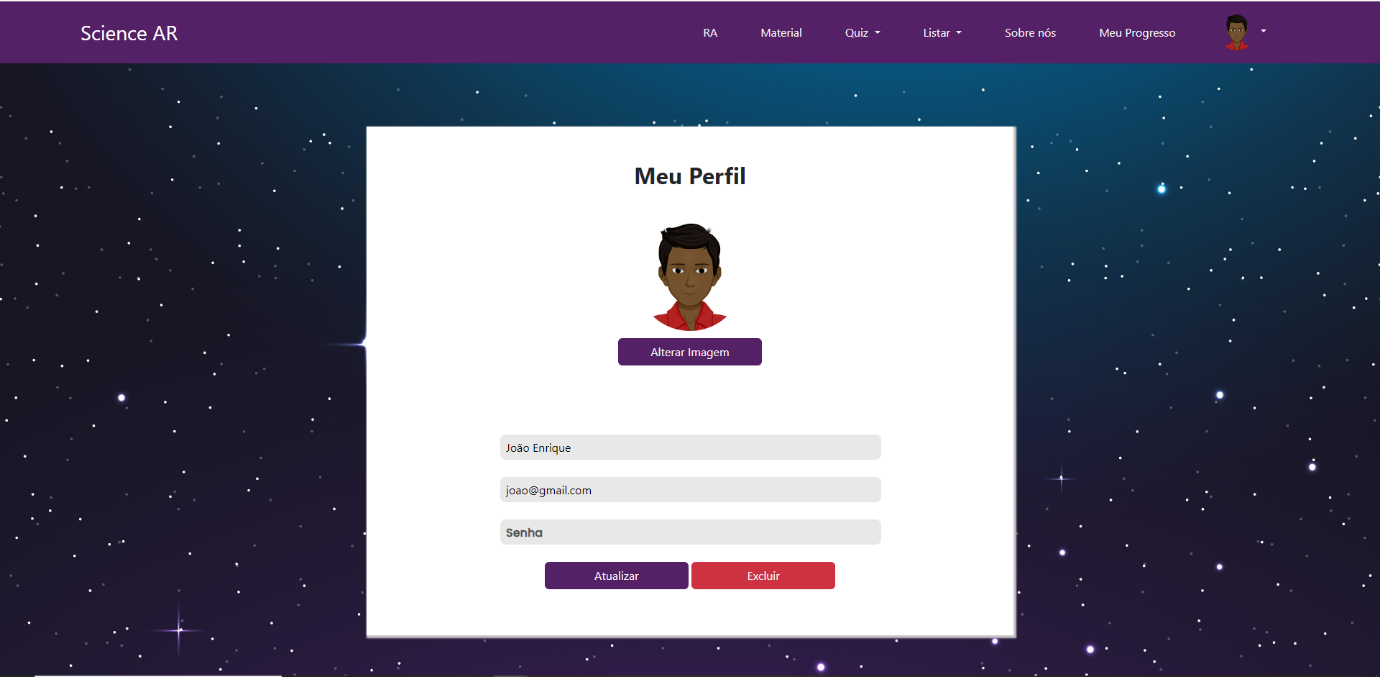
Figura 81 – Realizar Login



Fonte: Do Próprio Autor, 2023.

Na figura a seguir, os usuários, depois de logados na conta, serão capazes de visualizar seu perfil, podendo alterar suas informações. Se o usuário preferir, poderá também excluir sua conta do site, deletando todo seu progresso de questionários realizados.

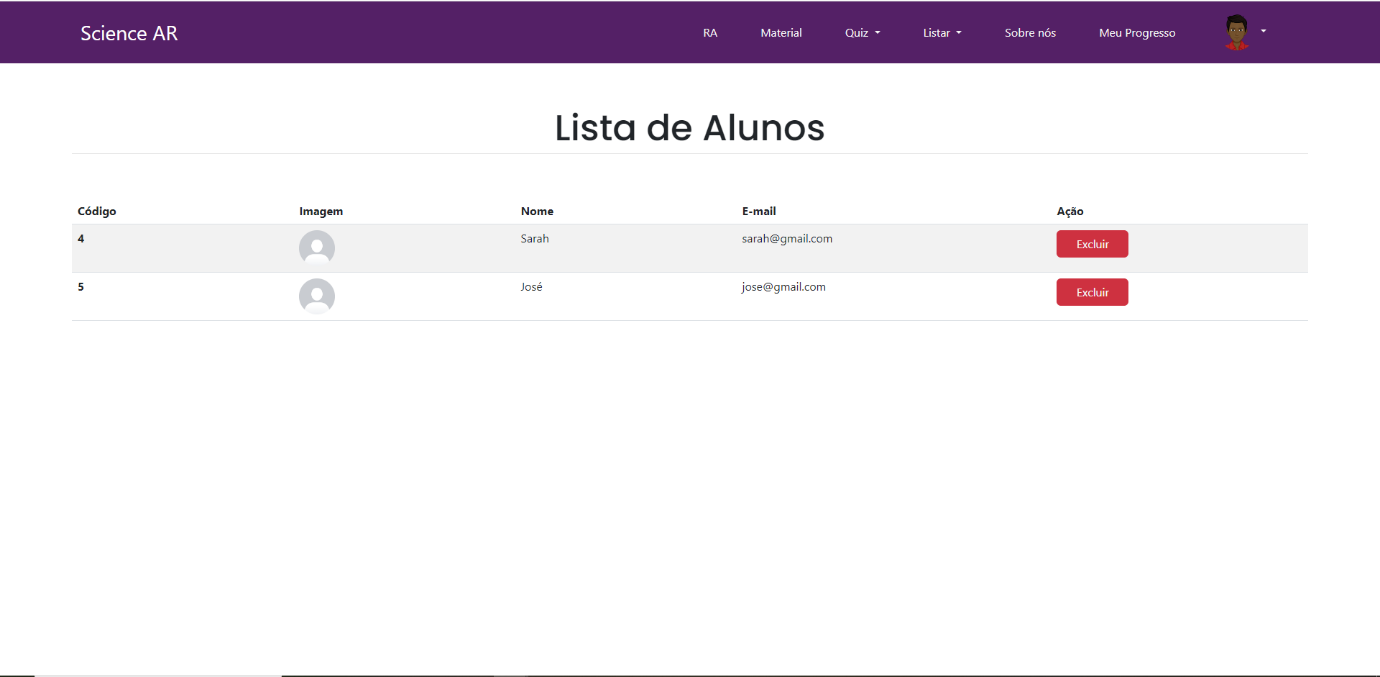
Figura 82 – Visualizar Perfil



Fonte: Do Próprio Autor, 2023.

Os administradores responsáveis pelo gerenciamento do site terão a capacidade de visualizar todos os alunos (usuários), tendo também acesso as informações de todos, com a funcionalidade de excluir uma conta, caso necessário.

Figura 83 - Visualizar Aluno



Fonte: Do Próprio Autor, 2023.

Os administradores responsáveis pelo gerenciamento do website, terão acesso a lista de administradores, onde visualizarão as informações de todos e também poderão excluir as suas contas.

Figura 84 - Visualizar Administrador



Fonte: Do Próprio Autor, 2023.

Na tela “Sobre Nós”, conterá as informações sobre a criação do site Science AR, incluindo o motivo da escolha do tema e qual o objetivo que o sistema pretende alcançar com as funcionalidades que estão disponíveis.

Figura 85 – Sobre Nós



Fonte: Do Próprio Autor, 2023.

Na tela abaixo, o usuário poderá acessar a página “Realidade Aumentada” mesmo não tendo um cadastro no sistema. Esta página explica sobre o que é realidade aumentada, facilitando assim, o aprendizado ao acessar a modelagem 3D.

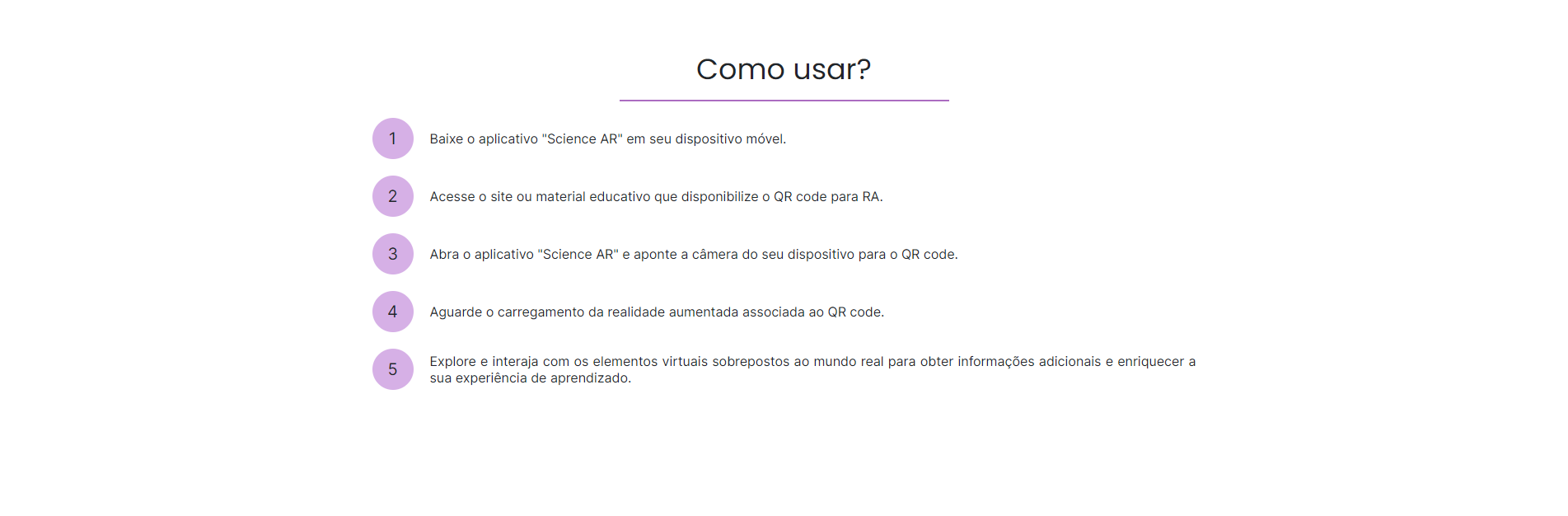
Figura 86 – Realidade Aumentada



Fonte: Do Próprio Autor, 2023.

Ainda na página “Realidade Aumentada”, contém o passo a passo de como usar o aplicativo que será fornecido, o APP será uma câmera para escanear o QR Code e abrirá um modelo 3D, onde o usuário poderá navegar por ele, aumentando seu aprendizado.

Figura 87 – Realidade Aumentada (segunda parte)



Fonte: Do Próprio Autor, 2023.

Nesta página, o usuário encontrará os cards com os materiais para estudo.  Para acessar os materiais, basta clicar em um dos cards, onde abrirá o conteúdo. Os materiais poderão ser acessados apenas com o usuário logado no site.

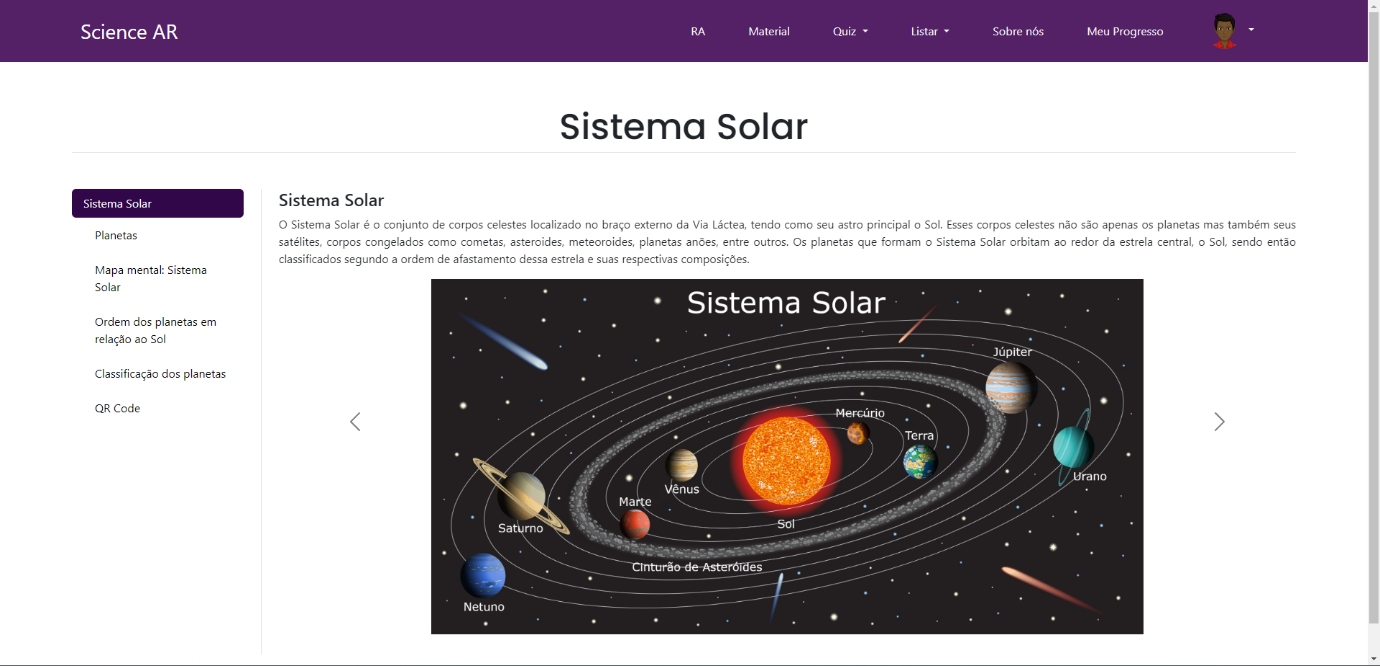
Figura 88 – Visualizar Materiais



Fonte: Do Próprio Autor, 2023.

Na figura abaixo, é apresentado o material que o usuário selecionou para estudar. A página apresentará informações e imagens sobre o tema escolhido, navegando por subtemas com mais informações, tornando a aprendizagem fluída e interativa.

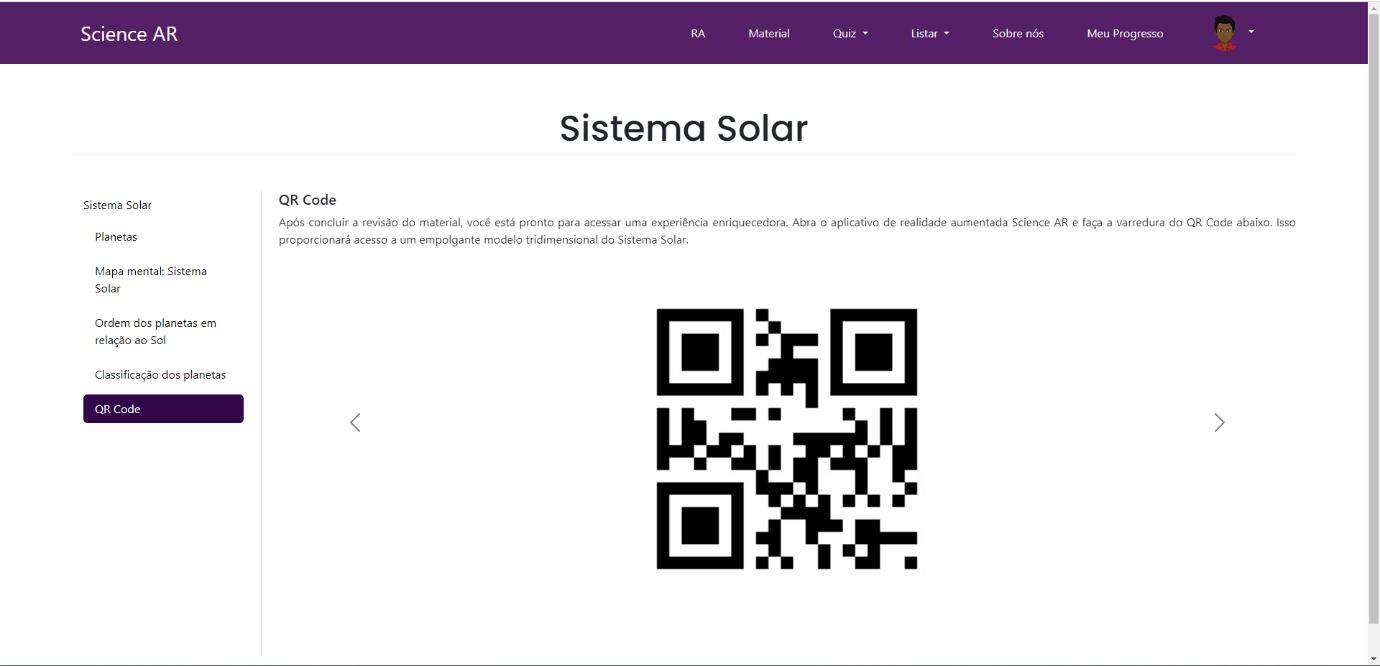
Figura 89 – Realizar Aula



Fonte: Do Próprio Autor, 2023.

Ainda no material selecionado, no final do conteúdo, será apresentado um QR Code para escanear com o aplicativo.

Figura 90 – QR Code



Fonte: Do Próprio Autor, 2023.

Um modelo 3D relacionado ao tema estudado será aberto, onde o usuário poderá navegar, assim expandindo seu conhecimento.

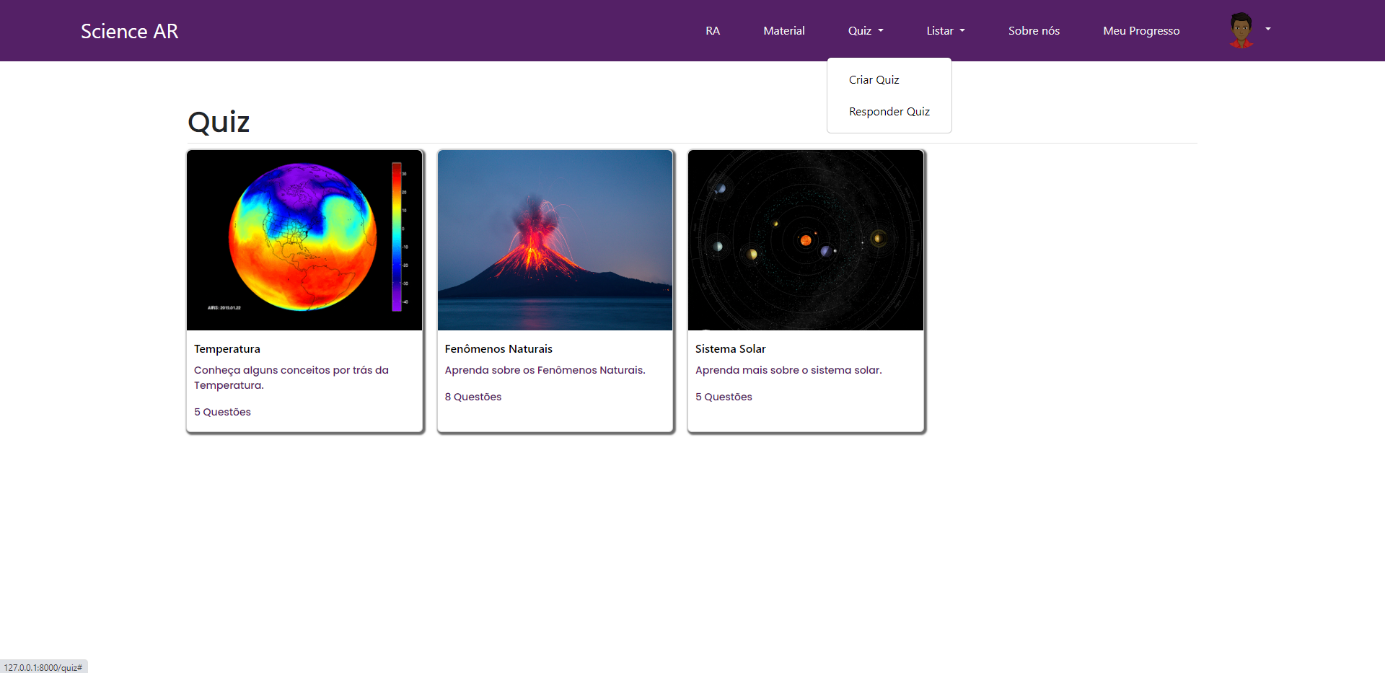
Figura 91 – QR Code escaneado



Fonte: Do Próprio Autor, 2023.

A interface a seguir permite que o usuário tenha acesso a todos os questionários que foram criados por um administrador, com isso, o aluno poderá escolher um quiz e realizá-lo, salvando suas informações no “Meu Progresso”.

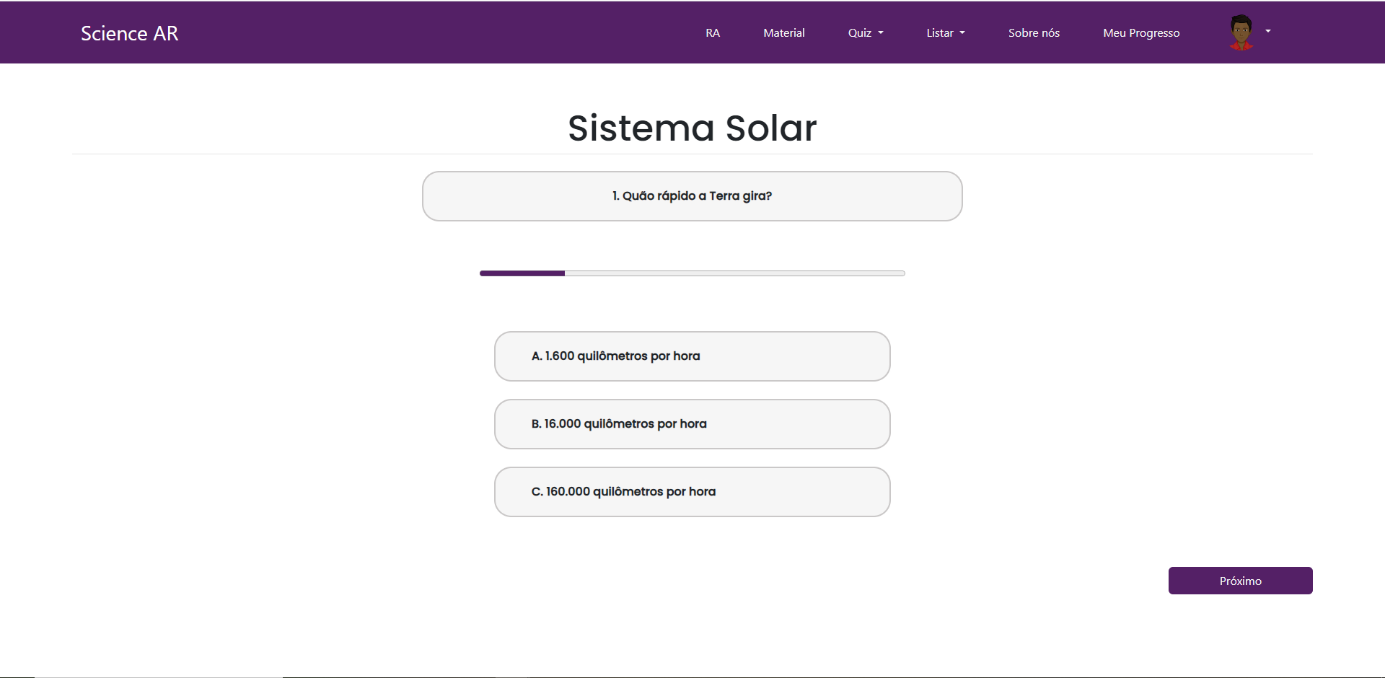
Figura 92 – Visualizar Questionários



Fonte: Do Próprio Autor, 2023.

Após selecionar um questionário, o usuário logado será redirecionado para a tela onde conterá todas as perguntas referentes ao quiz. Assim que o usuário selecionar uma resposta e clicar em “Próximo” será apresentado a próxima pergunta. Ao finalizar o questionário, o sistema mostrará o resultado das respostas.

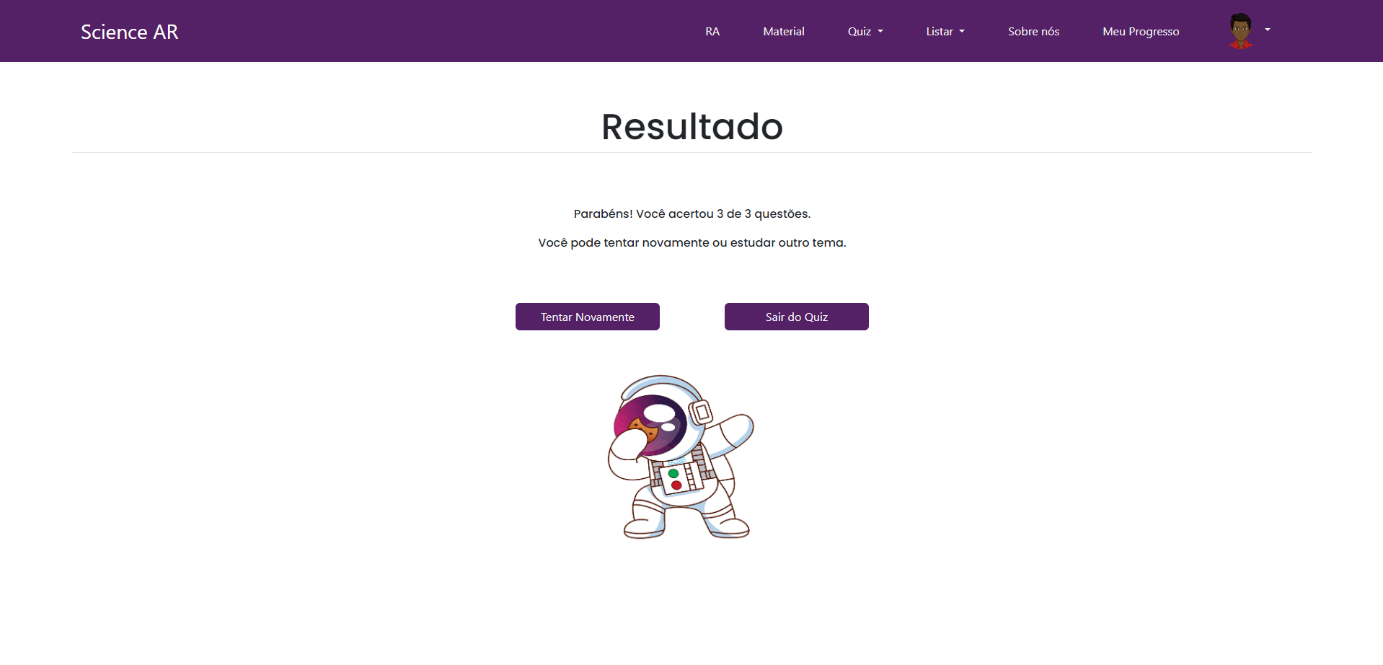
Figura 93 – Realizar Questionários



Fonte: Do Próprio Autor, 2023.

Ao finalizar o questionário, o usuário será redirecionado para a tela de resultado. Caso o índice de acertos seja mais da metade, ele mostrará a tela de resultado positivo, incentivando o usuário a realizar um outro questionário ou verificar seu progresso.

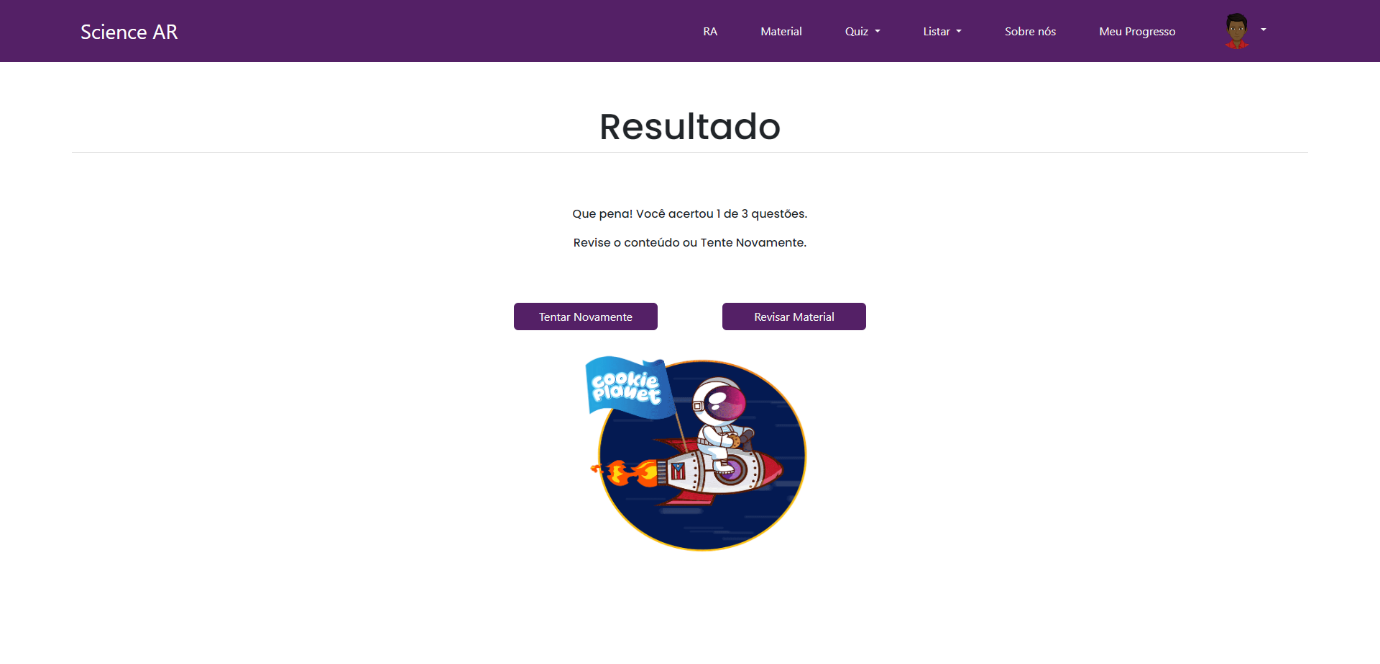
Figura 94 – Resultado Questionário (Positivo)



Fonte: Do Próprio Autor, 2023.

Após responder o questionário, o usuário verá a quantidade de acertos em relação a quantidade de perguntas. Caso o usuário acerte menos da metade, o sistema apresentará uma interface de resultado negativo, incentivando o usuário a realizar o questionário novamente.

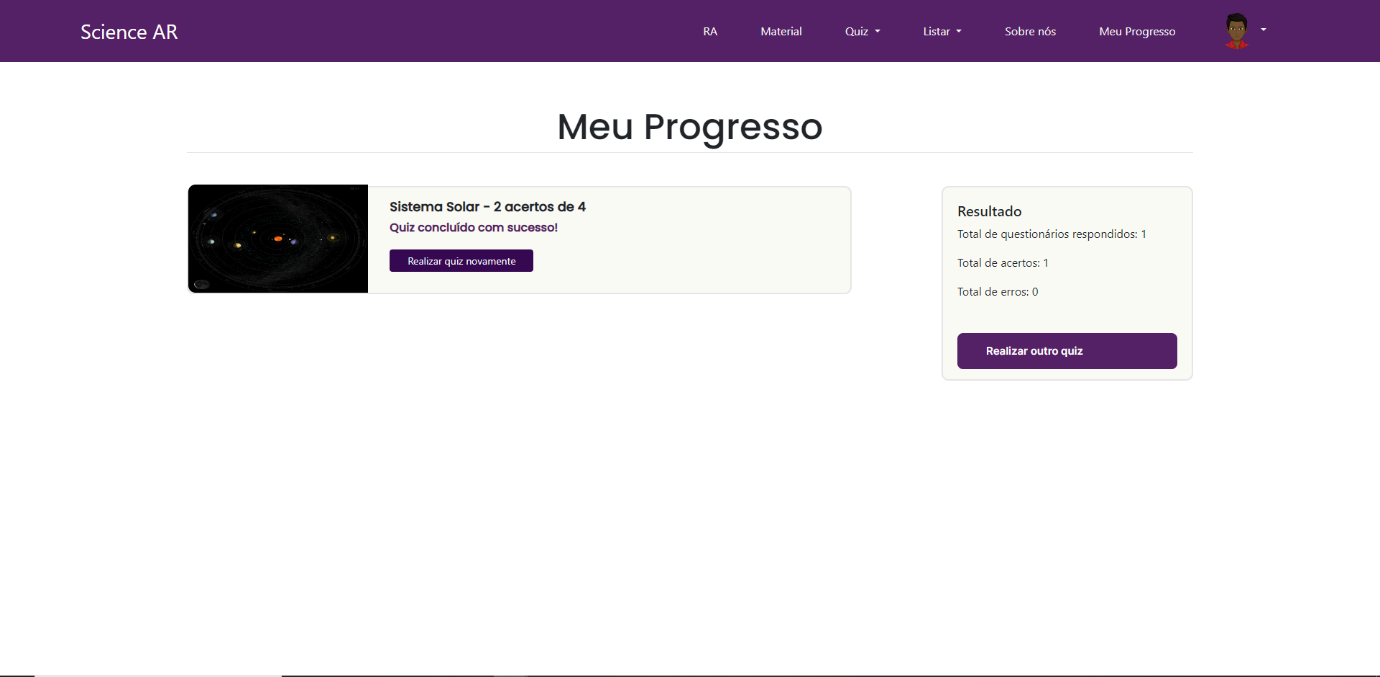
Figura 95 – Resultado Questionário (Negativo)



Fonte: Do Próprio Autor, 2023.

Na tela “Meu Progresso”, o usuário poderá ver como está indo o seu progresso, ou seja, ver os resultados dos quizzes feitos, quantos acertos e erros obteve e quantos quizzes foram realizados. Mas para acessar esta página, o usuário precisará estar logado.

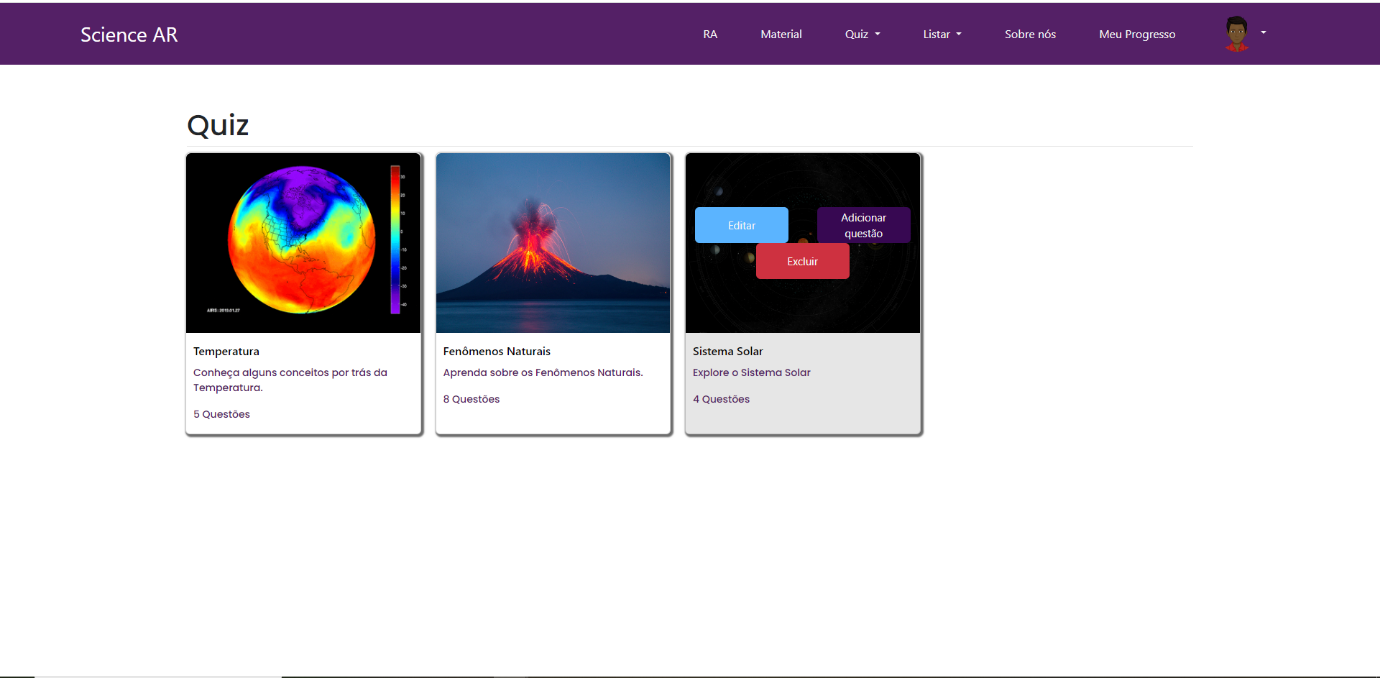
Figura 97 – Visualizar Progresso



Fonte: Do Próprio Autor, 2023.

Caso o usuário logado for um administrador, o site sofrerá mudanças na interface, permitindo que o administrador realize alterações no questionário, como adicionar questões, alterar respostas e até mesmo apagar o quiz.

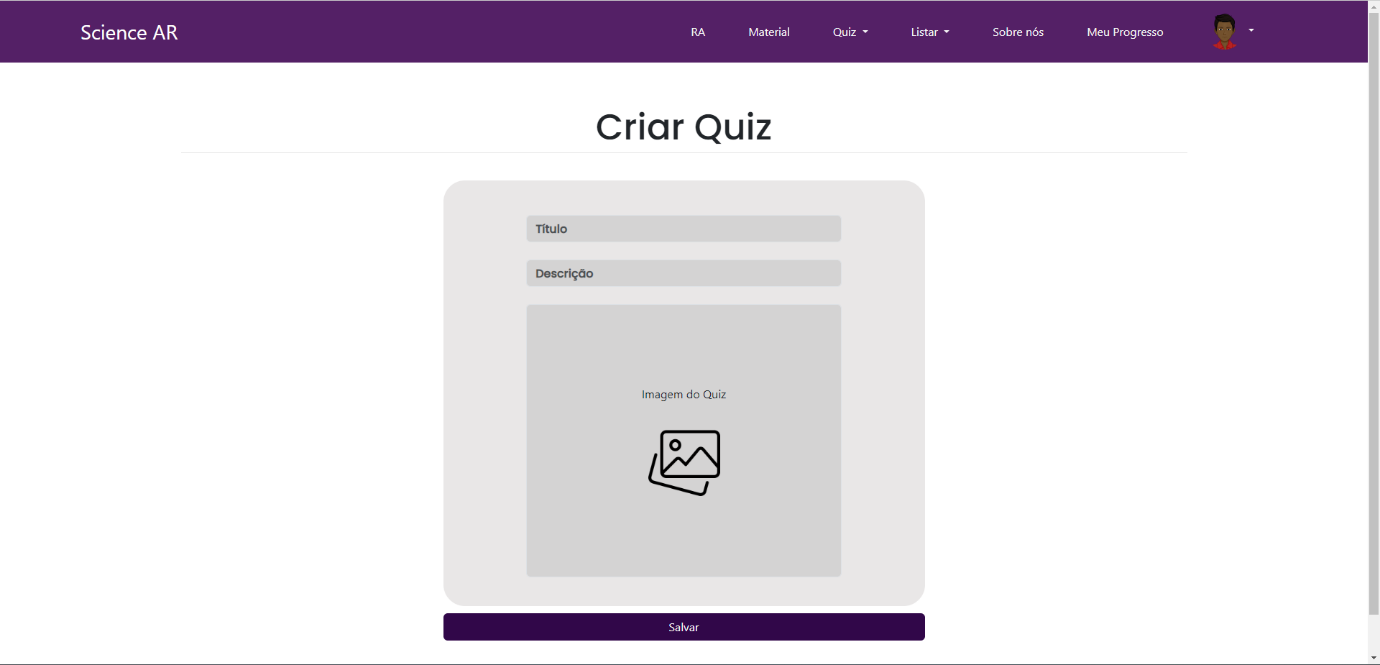
Figura 98 – Alterar Questionários



Fonte: Do Próprio Autor, 2023.

Caso o usuário que estiver logado for um administrador, o site sofrerá mudanças na interface, permitindo que o administrador crie questionários, adicionando imagens para os cards e adicionando o nome e descrição do questionário.

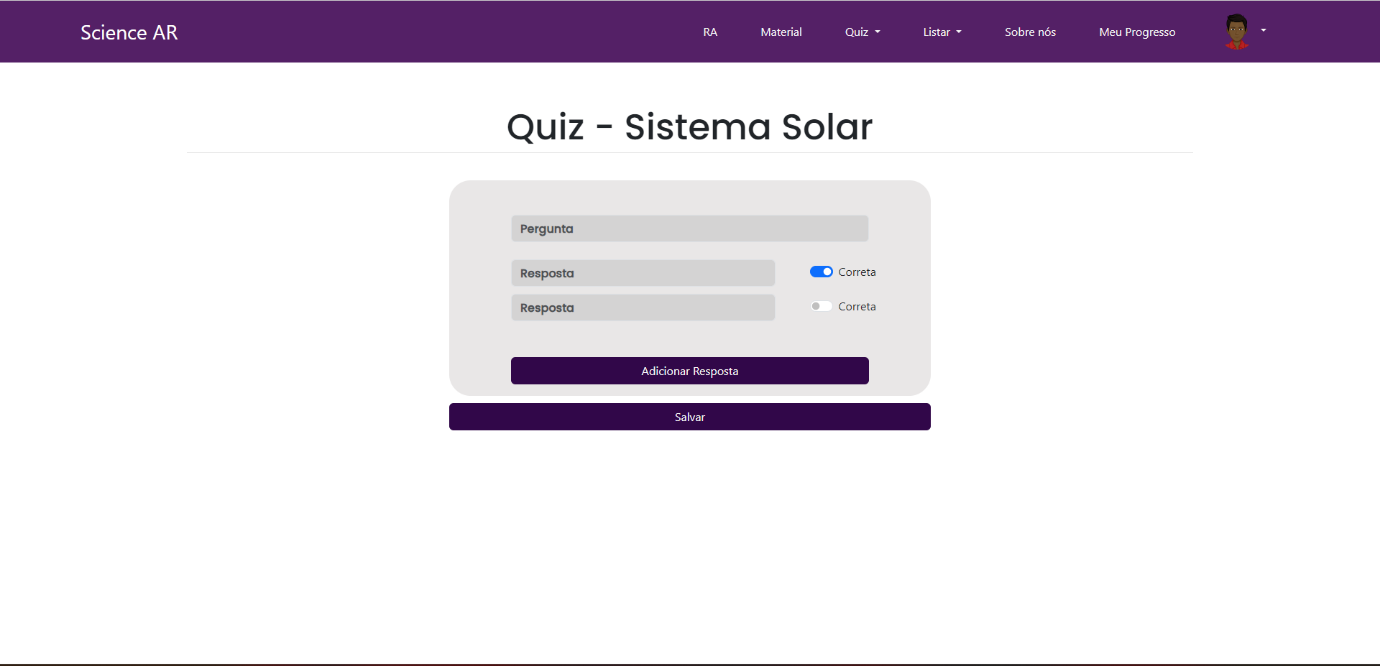
Figura 99 – Criar Questionários



Fonte: Do Próprio Autor, 2023.

Após criar um questionário, o administrador será redirecionado para a interface de criação de questões, onde o mesmo poderá adicionar as perguntas e respostas para o questionário que ele acabou de criar. O design permite que o administrador adicione e remova resposta para a pergunta e escolha qual a resposta certa.

Figura 100 – Adicionar Questões



Fonte: Do Próprio Autor, 2023.

5. CONSIDERAÇÕES FINAIS

Ao concluir este Trabalho de Conclusão de Curso (TCC) gostaríamos de fazer algumas considerações finais.

Neste estudo, buscamos explorar a aplicação da tecnologia de realidade aumentada no ensino de Ciências da Natureza para alunos do 7º ano. A proposta foi desenvolver um sistema web que utiliza essa tecnologia como uma ferramenta complementar ao ensino tradicional, visando proporcionar uma experiência de aprendizado mais dinâmica, interativa e didática.

Durante o desenvolvimento do projeto, foram identificadas diversas vantagens e benefícios da utilização da realidade aumentada no ensino de Ciências da Natureza. Através dessa tecnologia, os alunos poderão visualizar e interagir com objetos tridimensionais, simulações e informações adicionais sobre os conceitos estudados.

Além disso, a aplicação da realidade aumentada mostrou-se capaz de despertar o interesse e a motivação dos estudantes, pois proporcionou uma experiência diferenciada, aproximando-os do conteúdo de forma lúdica e dinâmica.

Embora a realidade aumentada apresente potencialidades significativas no contexto educacional, é fundamental destacar que ela não deve substituir o papel do professor, mas sim ser utilizada como uma ferramenta complementar ao ensino tradicional. O docente desempenha um papel fundamental na orientação dos conteúdos apresentados pela tecnologia.

Por fim, acreditamos que este estudo tenha contribuído para a compreensão da aplicação da realidade aumentada no ensino de Ciências da Natureza para alunos.

REFERÊNCIAS

BENTO, Evaldo Junior. **Desenvolvimento web com PHP e MySQL**. São Paulo: Casa do Código, 2021.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML**: guia do usuário. Rio de Janeiro: Elsevier, 2012.

BORGES, Luiz Eduardo. **Python para Desenvolvedores**. 2. ed. Rio de Janeiro: Novatec, 2010.

CASTRO, Elizabeth; HYSLOP, Bruce. **HTML5 and CSS3**. Berkeley: Peachpit Press, 2012.

CONVERSE, Tim; PARK, Joyce. **PHP**: a bíblia. 2. ed. Elsevier: Rio de Janeiro, 2003.

DALL'OGLIO, Pablo. **PHP:** programando com orientação a objetos. São Paulo: Novatec, 2007.

DOWNEY, Allen B. **Pense em Python**. São Paulo: Novatec, 2016.

DATE, C.J.. **Introdução a Sistemas de Banco de Dados**. 8. ed. Rio de Janeiro: Elsevier, 2003.

EIS, Diego; FERREIRA, Elcio. **HTML5 e CSS3 com farinha e pimenta**. São Paulo: Tableless, 2012.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistema de Banco de Dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2011.

FLANAGAN, David. **Javascript:** the definitive guide. [S.L]: O'Reilly Media,Inc., 2011.

GABARDO, Ademir Cristiano. **Laravel Para Ninjas**. Brasil: Novatec Editora, 2017.

GUEDES, Gilleanes T. A.. **UML2**: uma abordagem prática. 2. ed. São Paulo: Novatec, 2011.

HEUSER, Carlos Alberto. **Projeto de banco de dados**. Porto Alegre: Bookman, 2009.\

IEPSEN, Edécio Fernando. **Lógica de Programação e Algoritmos com Javascript**. São Paulo: Novatec, 2018.

Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. **Enem 2021 Resultados edição impressa, digital e PPL**. [S.l.]: [s.n.], 2021. Disponível em: https://download.inep.gov.br/enem/resultados/2021/apresentacao\_resultados\_finais.pdf. Acesso em: 20 jul. 2023.

LUCKOW, Décio Heinzelmann; MELO, Alexandre Altair de. **Programação Java para Web**. São Paulo: Novatec Editora, 2010.

MATTHES, Eric. **Curso Intensivo de Python.** São Paulo: Novatec, 2016.

MELO, Ana Carolina. **Desenvolvendo aplicações com UML 2.0**: do conceitual à implementação. 2. ed. Rio de Janeiro: Brasport, 2004.

MENEZES, Nilo Ney Coutinho. **Introdução à programação com Python**. 2. ed. São Paulo: Novatec, 2014.

Microsoft. **Um tour pela linguagem C#.** [S.l.]: [s.n.], 2023. Disponível em: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. Acesso em: 01 jun. 2023.

Microsoft. **Escrever seu primeiro código C#.** [S.l.]: [s.n.], 2023. Disponível em: <https://learn.microsoft.com/pt-br/training/modules/csharp-write-first/>. Acesso em: 01 jun. 2023.

Microsoft. **O que é o .NET?** Introdução e visão geral. [S.l.]: [s.n.], 2023. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/core/introduction>. Acesso em: 01 jun. 2023.

PIAGET, J. Main Trends in Psychology. 3.ed, Londres, UK: George Allen & Unwin,

1973.

POWERS, Shelley. **Aprendendo Javascript:** adicione brilho e vida as suas paginas web. [S.L]: Novatec, 2010.

PYTHON. **Documentação. 2022**. Disponível em: https://wiki.python.org/moin/BeginnersGuide/Overview. Acesso em: 20 jun. 2023.

SCURRA Tecnologia e Inteligência. **Realidade Virtual X Realidade Aumentada:** Entenda os Conceitos. [S.l.]: [s.n.], 2018. Disponível em: <https://www.scurra.com.br/blog/realidade-virtual-x-realidade-aumentada-entenda-os-conceitos/>. Acesso em: 19 jun. 2023.

SENHORAS, Elói Martins. **Educação, Ensino Superior E A Pandemia Da Covid-19**. Boa Vista: Ufrr, 2020.

SILVA, Delcione Lopes das. **Framework PHP Laravel 8 & AJAX**. Lisboa: Edições Sílabo, 2022.

SILVA, Mauricio Samy. **Fundamentos de HTML5 e CSS3**. São Paulo: Novatec, 2015.

SILVA, Wesley Willians Ramos da**. Laravel 5 Essencial:** Alta produtividade no mundo real. [s.l.]: School of Net, 2015.

SOUSA, Robson Pequeno de; MOITA, Filomena M. C. da S. C.; CARVALHO, Ana Beatriz Gomes. **Tecnologias Digitais na Educação**. Campina Grande: Eduepb, 2011.

SPURLOCK, Jake. **Bootstrap**. Sebastopol: O’reilly Media, Inc., 2013.

STAIANO, Fabio. **Designing and prototyping Interfaces with Figma**: learn essential ux/ui design principles by creating interactive prototypes for mobile, tablet, and desktop. Birmingham: Packt Publishing, 2022.

TAYLOR, Allen G.. **SQL Para Leigos**. 8. ed. Rio de Janeiro: Alta Books, 2015.

TORI, Romero; KIRNER, Claudio; SISCOUTTO, Robson. **Fundamentos e Tecnologia de Realidade Virtual e Aumentada**. Belém: Sbc, 2006.

TURINI, Rodrigo. **PHP e Laravel:** Crie aplicações web como um verdadeiro artesão. Brasil: Casa do Código, 2015.

TRIGO, Antonio; HENRIQUES, Jorge. **Aprenda a programar com C#**. 3. ed. Lisboa: Edições Sílabo, 2023.

Unity Technologies. **Criando e usando scripts.** [S.l.]: [s.n.], 2023. Disponível em: <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>. Acesso em: 01 jun. 2023.

Unity Technologies. **Suporte para ambiente de desenvolvimento integrado** (IDE). [S.l.]:[s.n.], 2023. Disponível em: <https://docs.unity3d.com/Manual/CreatingAndUsingScripts.html>. Acesso em: 01 jun. 2023.

ZEMEL, Tárcio. **páginas adaptáveis para todos os dispositivos**. São Paulo: Casa do Código, 2015.