

**Documentação de um**

**Produto de Software**

**CodeQuiz**

**Nome dos Alunos:**

|  |  |
| --- | --- |
| **RA** | **Nome** |
| **23.00292-0** | **Eduarda Portel** |
| **23.01524-0** | **Gustavo Araújo** |
| **23.01330-3** | **Matheus Yukio** |
| **23.01001-0** | **Murilo Molina** |
| **23.00591-2** | **Vinícius Parelho** |

**2023**

ÍNDICE DETALHADO

1. Levantamento dos Requisitos do Sistema de Software 4

1.1. Extração de Requisitos 4

1.2. Análise da Coleta de Requisitos 4

2. Especificação dos Requisitos do Sistema de Software 7

2.1. Requisitos Funcionais 7

2.2. Requisitos Não-Funcionais 10

3. Análise/Projeto 10

3.1. Diagrama de Classes 10

3.2. Diagrama de Sequência 10

3.3. Modelo de Banco de Dados 10

3.4. Diagrama de Atividades (opcional) 10

3.5. Diagrama de estados (opcional) 11

4. Implementação 11

5. Testes 11

6. Resultados e Considerações 11

Apêndice I 11

**Descrição/Resumo do Projeto** :

-O jogo será um quiz com progressão em níveis.

-Será desenvolvido em Python.

-O jogo será sobre Python e Java.

- O projeto se baseia num aplicativo versátil que aborda a programação em uma maneira leve e interativa, num jogo de perguntas e respostas.

A medida que o usuário acerta perguntas acumula experiencia(xp) para aumentar o seu nível, desbloqueando novas perguntas explorando novas habilidades e desafios. O xp libera novas opções do jogo que expandira o conhecimento do jogador.

-Será necessário a biblioteca pygame para desenvolver o jogo.

1. Levantamento dos Requisitos do Sistema de Software
   1. Extração de Requisitos

Para a extração de requisitos utilizamos um formulário do google para coletar a opinião das pessoas.

*https://docs.google.com/forms/d/e/1FAIpQLSdyDLZWpgmeNWXP0dKvbVf3kAMMUNPBpXchmDKYxUCu6HpFMw/viewform?usp=pp\_url*

* 1. Análise da Coleta de Requisitos

Chart, bubble chart

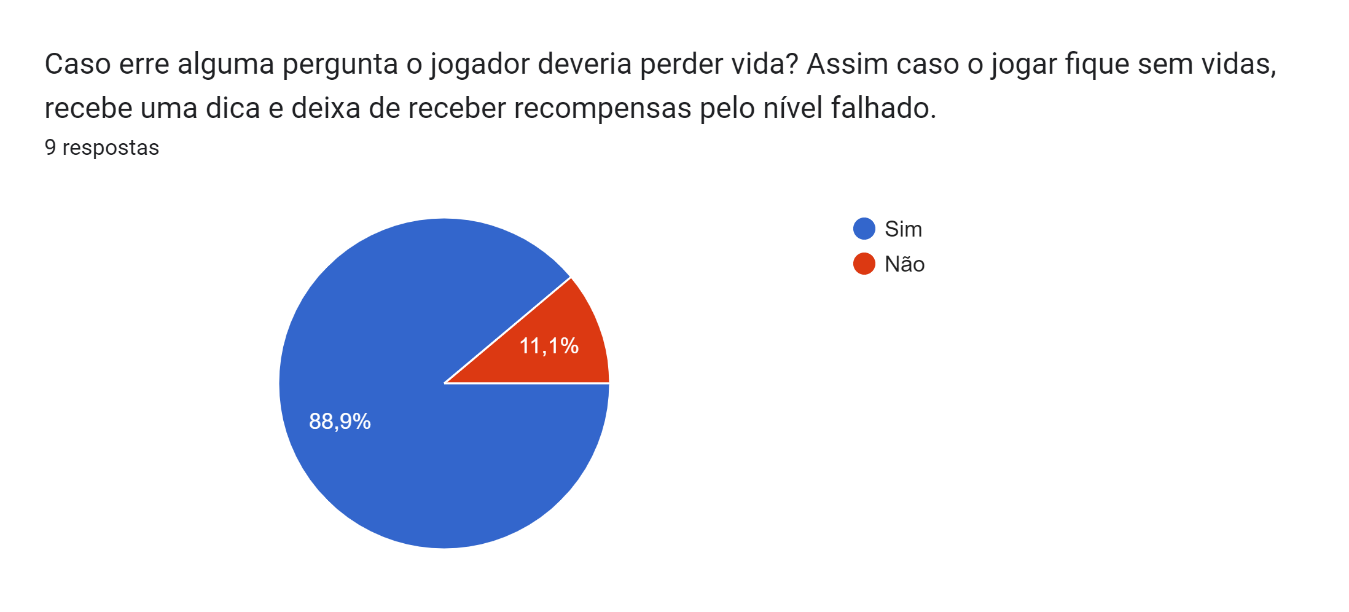
Description automatically generated

O gráfico indica que todos os participantes da pesquisa estão mais interessados em um jogo com um sistema de recompensas ao completar níveis.

Chart, bubble chart

Description automatically generated

O gráfico indica que todas os participantes da pesquisa se interessaram na ideia de vender “power-ups” na loja do jogo.



O gráfico indica que a maioria dos participantes se interessaram na ideia em dar dicas aos jogadores após errarem uma certa quantidade de respostas.

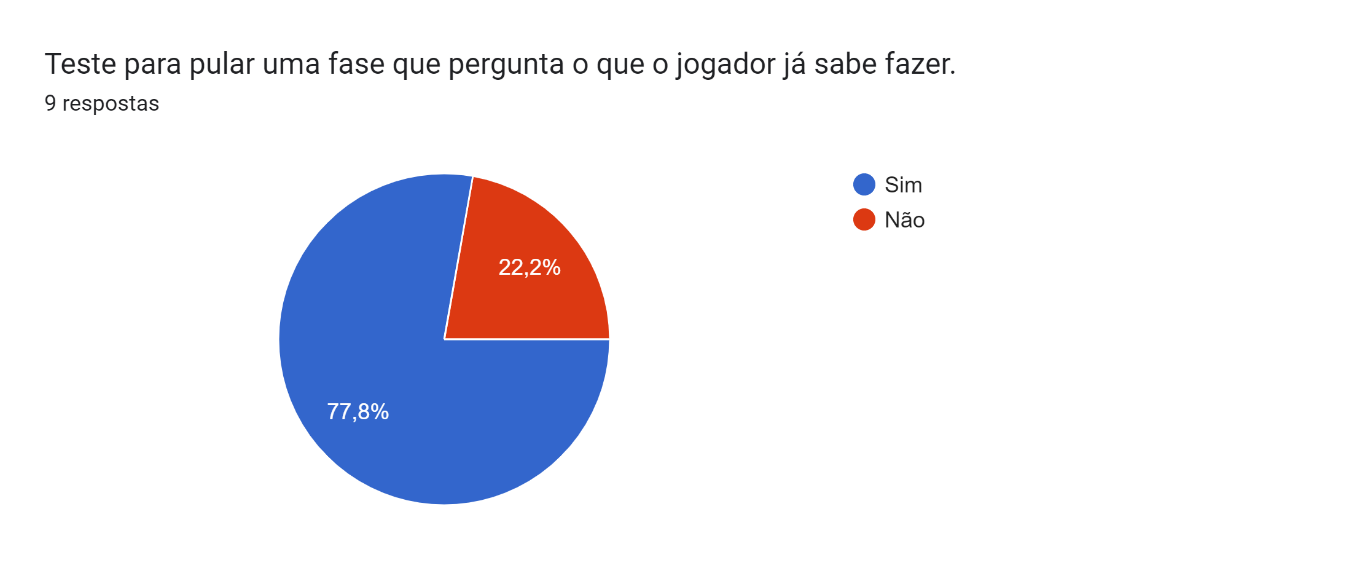
Chart, pie chart, bubble chart

Description automatically generated

O gráfico indica que todos os participantes da pesquisa se interessaram na ideia de um sistema de “combos” para recompensar o jogador com respostas corretas consecutivas.



O gráfico mostra que os participantes da pesquisa gostaram da ideia de permitir os jogadores a utilizarem as suas moedas para congelar o seu “streak” para não perder o bônus caso erre uma pergunta.



O gráfico indica que a maioria dos participantes querem poder pular uma fase que ensine algo que já sabem.

Chart, pie chart

Description automatically generated

O gráfico mostra que os jogadores gostaram mais da ideia de um mascote misturando os mascotes do Java e Python.

1. Especificação dos Requisitos do Sistema de Software
   1. Requisitos Funcionais

RF1-SISTEMA DE LOGIN O usuário poderá se cadastrar e fazer login para entrar na sua conta.

RF2- MENU PRINCIPAL No lado esquerdo ficará o mascote, configurações, a loja. No meio ficará os níveis. No lado direito ficará o total de vidas, moedas, etc.

I-MASCOTE Ao abrir o jogo ou clicar no mascote, o mascote falará alguma frase(exemplo: bom dia)

II-OPÇÕES Mudar cor de fundo(modo claro ou escuro), deslogar usuário.

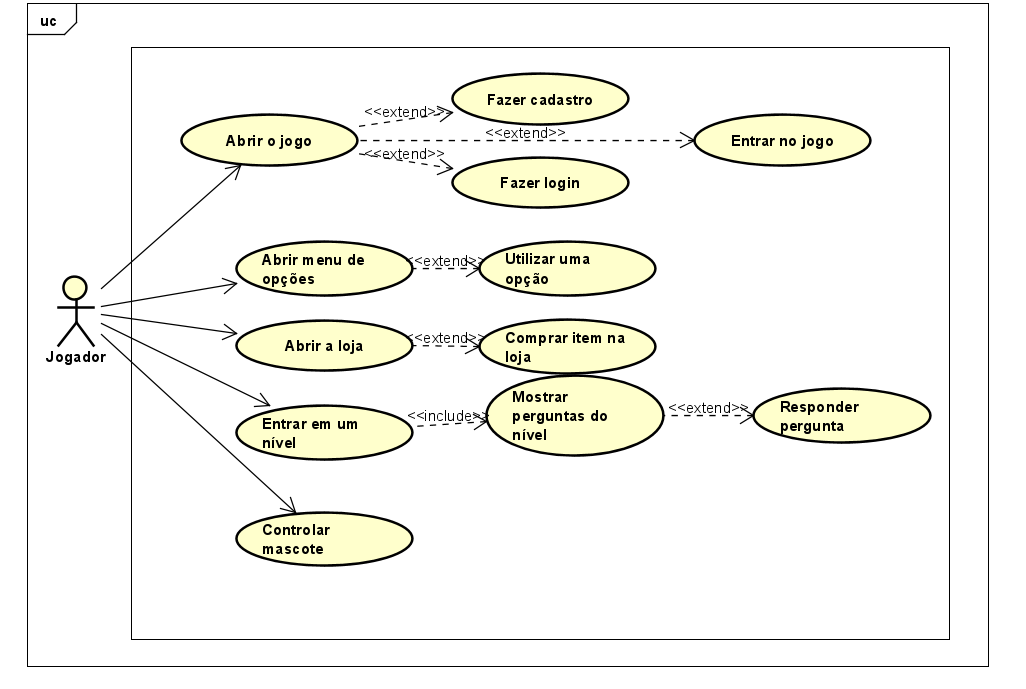
III-LOJA Comprar powerups(Não perder vida, congelar streak, etc.), comprar cosméticos para o mascote.

IV-NIVEIS Lista de níveis que são desbloqueados ao jogador subir de nível. Caso não esteja disponível, ficara com o fundo escuro e terá ao lado escrito(Nível necessário)

V-CONTADOR DE XP E MOEDAS Ao lado direito do menu principal terá um ícone mostrando o total de níveis e experiencia faltando para subir de nível, mais um ícone mostrando o total de moedas que o jogador possuí.

RF3-DENTRO DOS NÍVEIS. Dentro de cada nível terá uma caixa com a pergunta dentro, e abaixo terão 4 alternativas para serem selecionadas, apenas uma sendo a correta. Caso o jogador erre a resposta aparecera uma mensagem na tela falando que errou a resposta e para tentar novamente, caso erre 2 vezes o mascote no canto da tela vai dar uma dica ao jogador.

**Casos de uso**



# Catálogo de Atores

|  |  |
| --- | --- |
| Nome | Descrição |
| Jogador | Interage com o jogo e entrega comandos para o jogo interpretar. |

# CSU01 - Abrir o jogo

## Breve Descrição

Ao executar o jogo, o jogador será levado a uma tela para iniciar o jogo, depois poderá fazer login, cadastro ou abrir o menu principal.

# Fluxo de Eventos

## Fluxo Básico

1. Menu com o nome do jogo e um botão “Entrar” é mostrado ao jogador
2. Jogador clica no botão “Entrar”
3. Jogador é dado as opções de fazer login ou cadastrar um novo usuário

# CSU02 - Fazer cadastro

## Breve Descrição

Neste caso de uso o jogador pode criar um novo usuário para fazer login no jogo.

# Fluxo de Eventos

## Fluxo Básico

1. Jogador clica em registrar um novo usuário
2. Campo para inserir um nome de usuário e senha são mostrados ao jogador
3. Jogador informa um usuário e uma senha[FA1]
4. Usuário é cadastrado no sistema

## Fluxos Alternativos

# Jogador insere um usuário já cadastrado no sistema

1. Enviar ao jogador a mensagem: Usuário já cadastrado

**Pós-condições**

Jogador é registrado no banco de dados.

# CSU03 - Fazer login

## Breve Descrição

Neste caso de uso o jogador pode fazer login para acessar o jogo

# Fluxo de Eventos

## Fluxo Básico

1. Jogador escolhe a opção de fazer login
2. Campo para inserir um nome de usuário e senha são mostrados ao jogador
3. Jogador informa seu usuário e sua senha[FA1]
4. Jogador entra no menu principal

## Fluxos Alternativos

# Jogador insere um usuário inexistente ou uma senha errada

1. Enviar ao jogador a mensagem: Usuário ou senha errada

Pré-Condições

1. Nenhum usuário já estar logado no jogo

**Pós-condições**

Jogador é dado acesso ao jogo.

* 1. Requisitos Não-Funcionais
  + Jogo feito em Python utilizando pygame.
  + Jogo tem que funcionar em qualquer tipo de computador sem perda de performance.
  + Interface simples e fácil de entender.
  + Jogo deve ser leve.
  + Informações de cadastro são confidenciais

1. Análise/Projeto
   1. Diagrama de Classes

Neste item deve ser apresentado o modelo do domínio, visão de negócio, que representa um primeiro modelo conceitual do diagrama de classes.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamentos entre elas.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. Diagrama de Sequência

Neste item deve ser apresentado os diagramas de sequência com maior valor de negócio ao sistema. A escolha é realizada por caso de uso. Não há necessidade de realizar o diagrama de sequência para <<crud>>.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. Modelo de Banco de Dados

Neste item deve ser apresentado o modelo lógico relacional de banco de dados como proposta de solução.

Também deve ser representado o link de acesso ou repositório dos scripts físicos de banco de dados.

*Disciplina de Apoio: Banco de Dados Relacional.*

* 1. Diagrama de Atividades (opcional)

O diagrama de atividades representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema, geralmente utilizado para os métodos que contém regras de negócio.

*Esse diagrama deverá ser elaborado se houver necessidade e agregar valor ao projeto.*

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. Diagrama de estados (opcional)

O diagrama de estados especifica as sequências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

1. Implementação

Neste item indicar o link de acesso ou repositório de toda implementação de programação orientada a objetos Java e Python (opcional).

*Disciplina de Apoio: Programação Orientada a Objetos, Lógica de Programação e Banco de Dados Relacional.*

1. Testes

Neste item indicar o link de acesso ou repositório de todas as evidências de testes unitários realizados no projeto, de acordo, com os casos de uso especificados.

*Disciplina de Apoio: Programação Orientada a Objetos.*

1. Resultados e Considerações

Neste item devem ser apresentados os principais “prints” das telas do sistema de software desenvolvido, com uma breve explicação de cada tela e ao final as considerações gerais do projeto, sob o ponto de vista dos requisitos que foram implementados e os resultados obtidos.

Apêndice I

Neste item deve ser anexado o roteiro de entrevista ou questionário respondido.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*