

**Documentação de um**

**Produto de Software**

**CodeQuiz**

**Nome dos Alunos:**

|  |  |
| --- | --- |
| **RA** | **Nome** |
| **23.00292-0** | **Eduarda Portel** |
| **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.

- O projeto se baseia num aplicativo versátil que aborda a programação em uma maneira simples 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 sobre detalhes que seriam adicionados ao jogo. Além de procurar diversos aplicativos que possuem sistema de quiz(Duolingo, Kahoot, Codewars, etc) para ter uma ideia do que fazer para criar o jogo e perguntar a professores sobre como melhor desenvolver certos sistemas do jogo.

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

* 1. Análise da Coleta de Requisitos

Chart, bubble chart

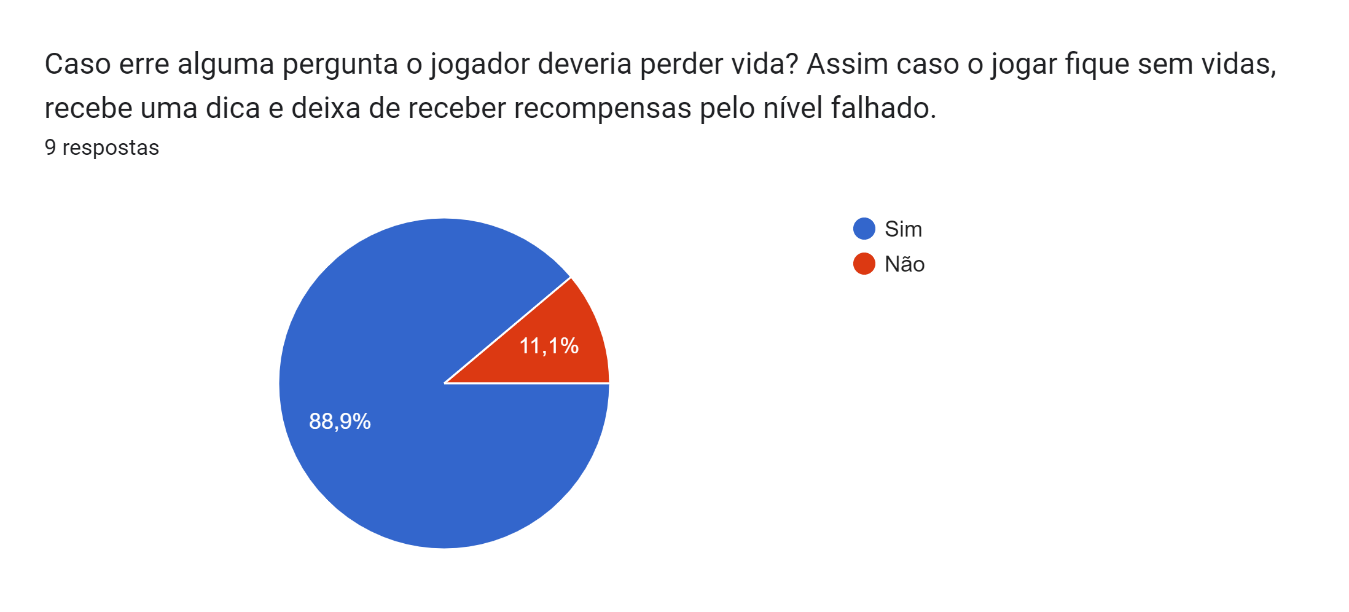
Description automatically generated

A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar moedas, e ao olhar outros sistemas de quiz, também ficou aparente que seria bom adicionar um sistema de pontos.

Chart, bubble chart

Description automatically generated

A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um item na loja que pode aumentar a quantidade de recompensa de experiencia. Ao perguntar para professores sobre esse bônus, sugeriram entregar o bônus aos pontos, e ai recompensar xp e moedas baseado na quantidade de pontos.



A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um sistema de vida ao jogo, porém ao experimentar com jogos que possuem sistemas parecidos, ficou evidente que era um mecanismo frustrante feito para incentivar os jogadores a comprarem mais vidas com seu dinheiro ao invés de incentivar os jogadores a não errarem. No final foi decidido que era um sistema que diminuiria a qualidade do jogo e não foram adicionadas vidas ao jogo

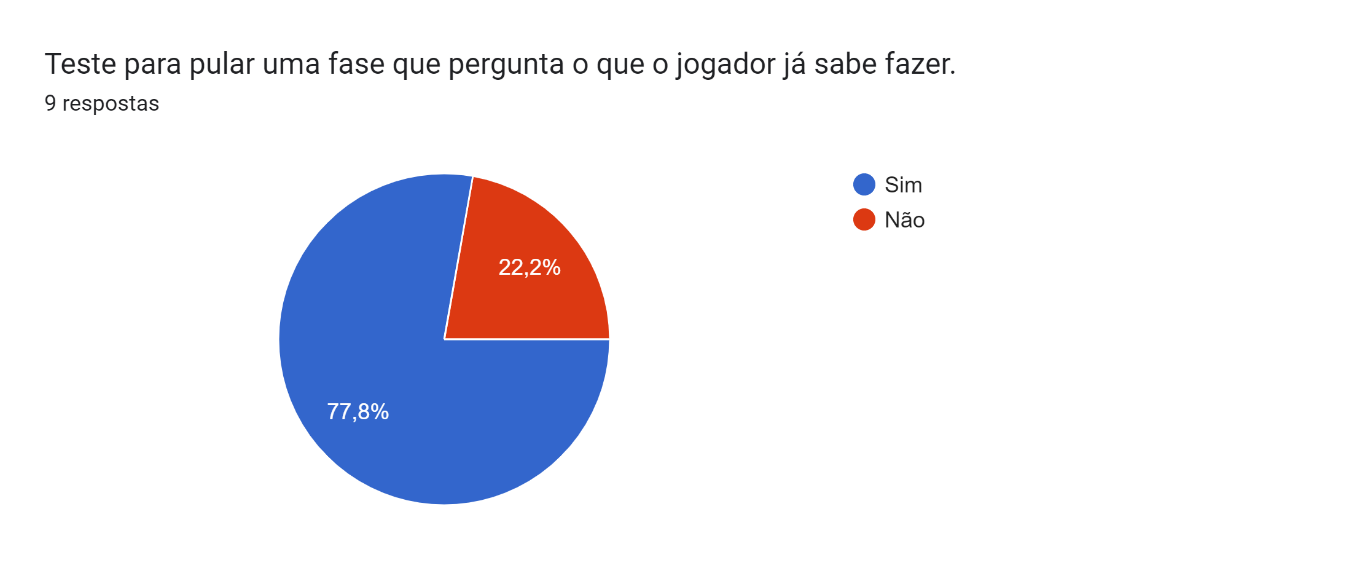
Chart, pie chart, bubble chart

Description automatically generated

A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um sistema de streak ao jogo. Ao observar outros jogos, a maioria também possuía um sistema parecido.



A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um item para salvar o streak caso o jogador erre uma pergunta. Ao observar outros jogos, poucos tinham um sistema parecido, porem como ajuda bastante a aumentar os pontos o streak como no Duolingo acaba sendo um sistema bom para o CodeQuiz.



A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um teste para pular um nível que já tenham conhecimento sobre. Porém ao desenvolver o jogo, pular os níveis acabaria atrapalhando o balanceamento do jogo e como fica fácil de subir de nível acumulando um streak alto seria fácil passar pelos níveis que já tem conhecimento sobre, acabou sendo melhor remover esse recurso.

Chart, pie chart

Description automatically generated

A maioria dos alunos que responderam ao questionário e os membros do grupo decidiram que o mascote deveria ser uma xícara com uma cobra dentro dela, parecidos com a logo do Python e Java.

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**

Diagrama

Descrição gerada automaticamente

# Catálogo de Atores

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

# CSU01 - Abrir o menu de login

## Breve Descrição

Este caso de uso descreve como o jogador pode abrir o menu de login para escolher se vai fazer cadastro ou login em sua conta.

# 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[FA1]
3. Jogador informa um usuário e uma senha
4. Usuário é cadastrado no sistema

## Fluxos Alternativos

[FA1] Jogador insere um usuário já cadastrado no banco de dados

# Jogador insere um usuário já cadastrado no sistema

1. Mostra uma mensagem ao jogador falando que a senha ou usuário são inválidos

**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[FA1]
3. Jogador informa seu usuário e sua senha
4. Jogador entra no menu principal

## Fluxos Alternativos

[FA1] Jogador insere um usuário ou senha não cadastrado no banco de dados

# Jogador insere um usuário inexistente ou uma senha errada

1. Mostra uma mensagem ao jogador falando que a senha ou usuário estão errados

Pré-Condições

1. Nenhum usuário já estar logado no jogo

**Pós-condições**

Jogador é dado acesso ao jogo.

# CSU04 – Fazer logout

## Breve Descrição

Neste caso de uso o jogador pode fazer logout para sair do jogo.

# Fluxo de Eventos

## Fluxo Básico

1. No menu principal o jogador clica no botão da engrenagem (configurações)
2. Ao entrar nas configurações irá aparecer o botão de logout no canto inferior esquerdo

3. Ao clicar no botão de logout será realizado o encerramento de sua sessão no jogo, posteriormente será encaminhado para tela de abertura do jogo.

Pré-Condições

1. Jogador estar logado no jogo

# CSU05 – Mostrar perguntas

## Breve Descrição

Neste caso de uso o jogador terá acesso as perguntas de cada nível do jogo.

# Fluxo de Eventos

## Fluxo Básico

1. Jogador abre um nível
2. Uma das possíveis perguntas é mostrada ao jogador

**Pós-condições**

Jogador conseguira responder as perguntas e usufruir do jogo.

# CSU06 –Responder perguntas

## Breve Descrição

Neste caso de uso o jogador terá a possibilidade de escolher as respostas das perguntas de seu respectivo nível.

# Fluxo de Eventos

## Fluxo Básico

1. Uma pergunta é exibida na tela
2. Jogador responde corretamente à pergunta[FA1]
3. Jogador tem seu streak aumentado por 1
4. Nova pergunta é mostrada para o jogador

## Fluxos Alternativos

[FA1] Jogador erra a resposta da pergunta

1. Jogador perde seu streak
2. Pergunta se mantém a mesma até o jogador acertar

**Pós-condições**

1. Moedas e XP serão adicionados a conta do jogador

# CSU07 – Sair do jogo

## Breve Descrição

# Este caso de uso descreve o processo de saída do jogo por parte do usuário. Ao optar por sair, o jogador encerra a sessão atual e retorna ao ambiente de onde iniciou o jogo.

# Fluxo de Eventos

## Fluxo Básico

1. O usuário decide sair do jogo[FA1]
2. O usuário clica no botão de finalizar sessão
3. O sistema exibe uma confirmação para garantir que o usuário deseja finalizar a sessão e sair do jogo
4. O usuário confirma a saída
5. O sistema encerra a sessão atual do jogador

## Fluxos Alternativos

[FA1]O usuário deseja finalizar a sessão por meio de fechar a aba.

1. O usuário clica no botão do seu computador de fechar abas
2. O sistema fecha o jogo

**Pós-condições**

1. O jogo é encerrado.

2. O progresso do jogo é salvo.

# CSU08 – Abrir o jogo

## Breve Descrição

# Este caso de uso descreve como o jogador abre o jogo.

# Fluxo de Eventos

## Fluxo Básico

1. O jogador abre o arquivo executável
2. Jogo mostra uma tela de boas-vindas

# CSU09 – Abrir o menu principal

## Breve Descrição

# Este caso de uso descreve como o jogador abre o menu principal para poder jogar

# Fluxo de Eventos

## Fluxo Básico

1. O jogador efetua o seu login
2. O jogo mostra o menu principal, com o mascote opções, loja e suas moedas e xp

Pré-Condições

1. Estar logado no jogo, ter uma sessão ativa

**Pós-condições**

1. O jogador tem acesso ao jogo

# CSU10 – Abrir o menu de opções

## Breve Descrição

# Este caso de uso descreve como o jogador abre o menu de opções para deslogar ou trocar de tema

# Fluxo de Eventos

## Fluxo Básico

1. O jogador clica na engrenagem do menu principal
2. Jogo mostra ao jogador os possíveis temas que pode escolher e o botão para deslogar

**Pós-condições**

1. O jogador tem acesso as opções do jogo

# CSU11 – Trocar de tema

## Breve Descrição

# Este caso de uso descreve como o jogador pode trocar o tema de fundo do seu jogo

# Fluxo de Eventos

## Fluxo Básico

1. O jogador abre o menu de opções
2. O jogador clica em uma das cores no menu de opções
3. O jogo muda a cor de fundo para a respectiva cor que o jogador escolheu

**Pós-condições**

1. A cor de fundo do jogo muda para a cor desejada

# CSU12 – Abrir a loja

## Breve Descrição

# Este caso de uso descreve como o jogador abre a loja para comprar cosméticos ou powerups

# Fluxo de Eventos

## Fluxo Básico

1. O jogador clica no carrinho de compras do menu principal
2. Jogo mostra ao jogador os powerups e os cosméticos do mascote

**Pós-condições**

1. O jogador tem acesso as opções do jogo

# CSU13 – Comprar item na loja

## Breve Descrição

# Este caso de uso descreve como o jogador compra cosméticos e powerups

# Fluxo de Eventos

## Fluxo Básico

1. O jogador clica em algum powerup ou cosmético
2. É verificado se o jogador possui moedas o suficiente para comprar o que deseja[FA1]
3. Compra é efetuada
4. Moedas são retiradas da conta do jogador

## Fluxos Alternativos

[FA1]O usuário não possui moedas o suficiente para comprar o que deseja

1. O jogo mostra ao jogador que não possui moedas o suficiente

**Pós-condições**

1. O jogador tem acesso aos cosméticos ou powerups que comprou

# CSU14 – Abrir nível

## Breve Descrição

# Este caso de uso descreve como o jogador pode abrir um nível para jogar

# Fluxo de Eventos

## Fluxo Básico

1. O jogador abre o menu principal
2. O jogador clica em algum dos níveis[FA1]
3. O jogo abre o nível que o jogador deseja

## Fluxos Alternativos

[FA1]O usuário não possui xp o suficiente para abrir o nível que deseja

1. O nível que o jogador não pode acessar é escondido por um cadeado
2. Botão do nível é bloqueado até o jogador ter xp o suficiente
   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 rodar em computador Windows.
   * Informações de cadastro são confidenciais
3. Análise/Projeto
   1. Diagrama de Classes

Diagrama

Descrição gerada automaticamente

* 1. Diagrama

     Descrição gerada automaticamenteDiagrama de Sequências

Diagrama

Descrição gerada automaticamenteDiagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

Diagrama

Descrição gerada automaticamente

* 1. Uma imagem contendo Diagrama

     Descrição gerada automaticamenteModelo de Banco de Dados

.

1. Implementação

O repositório do jogo pode ser encontrado no GitHub pelo link a seguir:

<https://github.com/Parelho/PI>

1. Testes

O projeto foi testado pelos membros do grupo, com ajuda de professores que ajudaram a encontrar e resolver alguns bugs que apareceram ao longo dos testes.

1. Resultados e Considerações

Levando em consideração os requisitos levantados pelo formulário do google respondido pelos alunos do laboratório 3 e sugestões dos professores, foi cumprido todos os requisitos essenciais para o funcionamento do jogo que foi planejado, faltando apenas recursos de qualidade de vida.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança média

Gráfico de radar

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente com confiança média

Gráfico

Descrição gerada automaticamente com confiança média

Uma imagem contendo Texto

Descrição gerada automaticamente

Forma, Retângulo

Descrição gerada automaticamente

Uma imagem contendo Texto

Descrição gerada automaticamente

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

Uma imagem contendo Gráfico

Descrição gerada automaticamente

Apêndice I

1. Completar os níveis deveria recompensar com algum tipo de moeda para comprar cosméticos para o mascote?
2. Caso uma moeda seja implementada, a loja deveria ter multiplicadores que aumentam o tamanho da recompensa de xp?
3. Caso erre alguma pergunta o jogador deveria perder vida? Assim caso o jogar fique sem vidas, recebe uma dica e deixa de receber recompensas pelo nível falhado.
4. Um sistema de "Streak" seria interessante? Ex: Cada nível feito sem errar nenhuma pergunta, recompensara o jogador com um multiplicador de 25% para cada nível perfeito feito em seguida.
5. Caso uma moeda seja implementada, a loja deveria ter um item que salva o seu streak de níveis perfeitos?
6. Teste para pular uma fase que pergunta o que o jogador já sabe fazer.
7. Qual deveria ser o mascote do jogo?

Respostas:

1. A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar moedas, e ao olhar outros sistemas de quiz, também ficou aparente que seria bom adicionar um sistema de pontos.
2. A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um item na loja que pode aumentar a quantidade de recompensa de experiencia. Ao perguntar para professores sobre esse bônus, sugeriram entregar o bônus aos pontos, e ai recompensar xp e moedas baseado na quantidade de pontos.
3. A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um sistema de vida ao jogo, porém ao experimentar com jogos que possuem sistemas parecidos, ficou evidente que era um mecanismo frustrante feito para incentivar os jogadores a comprarem mais vidas com seu dinheiro ao invés de incentivar os jogadores a não errarem. No final foi decidido que era um sistema que diminuiria a qualidade do jogo e não foram adicionadas vidas ao jogo.
4. A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um sistema de streak ao jogo. Ao observar outros jogos, a maioria também possuía um sistema parecido.
5. A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um item para salvar o streak caso o jogador erre uma pergunta. Ao observar outros jogos, poucos tinham um sistema parecido, porem como ajuda bastante a aumentar os pontos o streak como no Duolingo acaba sendo um sistema bom para o CodeQuiz.
6. A maioria dos alunos que responderam ao questionário decidiram que seria uma boa ideia adicionar um teste para pular um nível que já tenham conhecimento sobre. Porém ao desenvolver o jogo, pular os níveis acabaria atrapalhando o balanceamento do jogo e como fica fácil de subir de nível acumulando um streak alto seria fácil passar pelos níveis que já possui conhecimento sobre, acabou sendo melhor remover esse recurso.
7. A maioria dos alunos que responderam ao questionário e os membros do grupo decidiram que o mascote deveria ser uma xícara com uma cobra dentro dela, parecidos com a logo do Python e Java.