

OOPPG - Object Oriented Programming Platform Game

30 de junho de 2022

Grupo	
Nome Completo	Matrícula
Alexandre Vítor Silva Braga	201965501B
Igor Correa Rodrigues	201965570C
José Ronaldo Mouro	201535011
Marcelo Ian Rezende Menezes	201965517B

Sumário

1	Introdução	3
1.1	Propósito	3
1.2	Escopo	3
1.3	Definições e Abreviações	3
1.4	Visão Geral do Documento	3
2	Descrição Geral	4
2.1	Perspectivas do Produto	4
2.2	Funções do Produto	4
2.3	Restrições	4
3	Descrição dos Requisitos Funcionais (RF)	5
4	Descrição dos Requisitos Não Funcionais (RNF)	9
5	Artes Conceituais	10
6	Diagrama de casos de uso	11
7	Diagrama de classes	12

1 Introdução

1.1 Propósito

O propósito do Documento de Especificação de Requisitos é delinear os requisitos do software a ser construído, descrevendo suas funcionalidades e características. O público alvo do documento são alunos, professores e desenvolvedores do projeto.

1.2 Escopo

O Object Oriented Programming Platform Game auxiliará o processo de aprendizado dos alunos da área de Computação na disciplina de Orientação a Objetos através de um meio descontraído e interativo de um jogo.

1.3 Definições e Abreviações

- **OOPPG**: nome dado Object Oriented Programming Platform Game;
- **RF**: requisito funcional;
- **RNF**: requisito não funcional;
- **Jogador**: usuário do sistema que tem acesso às ações básicas do personagem principal;
- **Inimigo**: personagem controlado pelo sistema com o objetivo de atrapalhar e/ou causar dano ao personagem principal;
- **Jogo**: é o próprio sistema;
- **Fase**: é um estágio do jogo onde o jogador deve completar um objetivo discreto;
- **Menu**: tela inicial do jogo;

1.4 Visão Geral do Documento

- **Seção 2 - Descrição Geral**: apresenta uma visão geral do sistema, especificando a perspectiva do produto e detalhamento do escopo do sistema através da discretização das funções do produto. Além disso, são explicitadas as características gerais dos usuários do produto e as restrições que poderão limitar as possibilidades de desenvolvimento.
- **Seção 3 - Descrição dos Requisitos Funcionais (RF)**: apresentação de todos os requisitos funcionais do sistema. Descreve as principais ações do produto, considerando a aceitação e processamento das entradas e o processamento e geração das saídas.
- **Seção 4 - Requisitos Não Funcionais**: apresentação de todos os requisitos não funcionais do sistema. Descreve todos os aspectos qualitativos do sistema, explicitando os detalhes de facilidade de uso, manutenibilidade, confiabilidade, desempenho, segurança, distribuição, adequação a padrões e requisitos de hardware e software.
- **Seção 5 - Artes Conceituais**: apresentação de algumas artes conceituais do projeto, para mostrar as ideias iniciais por trás do produto final.
- **Seção 6 - Diagrama de casos de uso**: demonstra, através de um diagrama, as diferentes formas que os atores do sistema podem interagir com o este.
- **Seção 7 - Diagrama de classes**: mapeia, através de um diagrama, a estrutura do sistema modelando suas classes, atributos, operações e relações entre objetos.

2 Descrição Geral

2.1 Perspectivas do Produto

O sistema é desenhado para ser executado em servidor Web remoto. Para que o usuário acesse o sistema, é necessário ter um computador com acesso à internet e a um navegador (ex: Chrome, Firefox, Brave, etc.). A interação com o sistema se dará por interface gráfica e o controle do personagem principal será via entrada do teclado.

2.2 Funções do Produto

O sistema tem como objetivo ensinar os tópicos principais da disciplina de Orientação a Objetos aos alunos da área de Computação através de *mini-games* e questionários acerca do tema. Dessa forma, procura-se utilizar uma maneira descontraída para auxiliar nesse aprendizado.

2.3 Restrições

O sistema deve ser desenvolvido com os recursos disponíveis na plataforma Web. O prazo final para os protótipos devem ser aqueles especificados na Gerência de Projetos: 01/06 para o primeiro, 29/06 para o segundo e 28/07 para o terceiro e último protótipo. O sistema não deverá ter custo monetário para aquisição de softwares e equipamentos de terceiros.

3 Descrição dos Requisitos Funcionais (RF)

RF001 – Movimentação do Personagem Principal

O jogador deve ser capaz de movimentar o personagem principal pelo cenário, mediante o desenvolvimento de caminhada e através de saltos.

RF002 – Inimigo Comum

Os inimigos devem ser posicionados no cenário do jogo de forma a representarem uma ameaça à conquista ou finalização de cada fase do mesmo.

RF003 – Comportamento dos Inimigos Comuns

O jogo deve ser capaz de direcionar os inimigos comuns para colidirem com o personagem principal tão logo este ocupe a plataforma por eles dominada.

RF004 – Movimentação dos Inimigos Comuns

O jogo deve ser capaz de movimentar os inimigos comuns sobre plataformas do cenário, mediante o desenvolvimento de caminhada em direção aleatória e limitada à dimensão das plataformas.

RF005 – Ação Nociva dos Inimigos Comuns sobre o Personagem Principal

O personagem principal perderá pontos de vida mediante o contato ou colisão com os personagens inimigos.

RF006 – Ação Nociva do Personagem Principal

O jogador deve ser capaz de degradar a vida dos inimigos mediante ação de ataque.

RF007 – Obstáculos

O jogador deve ser capaz de atravessar obstáculos, em formato de *puzzle* espalhados pelo cenário. Alguns obstáculos poderão retirar pontos de vida do jogador, enquanto outros serão inofensivos, apenas atrapalhando o progresso do jogador.

RF008 – Fase

Terá no mínimo 4 fases. Cada fase vai abranger um assunto relacionado à Disciplina de Orientação a Objetos, quais sejam:

- Abstração,
- Encapsulamento,
- Herança
- Polimorfismo.

RF009 – Mapa

Cada fase do jogo será composta de mapas. Os mapas representarão as etapas das fases. Inicialmente, a fase terá uma etapa apenas de plataformas, onde o jogador pode acumular colecionáveis sem se preocupar com questões relacionadas à disciplina. A partir daí, começarão etapas de perguntas, onde o jogador deve passar por uma série de questionários em forma de *puzzles* intercalados com seções de plataformas. Por fim, o mapa final consistirá de uma luta entre o personagem principal e um inimigo, representando o domínio do jogador sobre um dos quatro assuntos da disciplina e possibilitando se alcançar próxima fase.

RF010 – Cenário

O cenário será escolhido de acordo com a cena atual. De acordo com a movimentação do personagem principal, o cenário é alterado de maneira a revelar mais pedaços da fase em questão.

RF011 – Inimigo Chefe

Terá no mínimo 5 inimigos chefes principais. Cada um presente em uma fase, abrangendo um assunto relacionado aos 4 pilares. O inimigo final, representará uma revisão total do conteúdo.

RF012 – Questionários

Cada fase terá seu próprio questionário, ou seja, um banco de questões fixo, deverão ser selecionadas ao todo 10 perguntas por pilar em sua respectiva fase e para a fase final, de forma randomizada 5 de cada pilar.

RF013 – Comportamento dos Inimigos Chefes

Cada inimigo chefe fará um mini questionário sobre seu pilar, além disso cada um terá um padrão de ataques diferente que deverá ser evitado entre questões.

RF014 – Testes Dinâmicos

O jogador terá que responder a perguntas relacionadas ao conteúdo de Orientação a Objetos para continuar as cenas. A fim de serem realizados de forma dinâmica, podendo ser ocultos em diversas formas:

- Questões de múltipla escolha: Seções com diferentes rotas para diferentes respostas, podendo levar a armadilhas quando erradas;
- Questões de Verdadeiro ou Falso: Seções de plataforma marcadas com alternativas e que desaparecem quando a resposta está errada;
- Questões múltipla escolha com multi resposta: Botões a apertar de forma a acender luzes para a resposta;
- Questões de relacionar: Mover blocos à suas resposta equivalentes.

RF015 – Colecionáveis

Durante a primeira cena, o jogador poderá percorrer o cenário em busca de colecionáveis, representado por uma caneca de café. Ao acumular todos os colecionáveis de uma fase, o jogador tem o direito de pular uma das perguntas durante a segunda cena.

RF016 – Pontos de Vida

O personagem possui uma quantidade máxima de pontos de vida. Quando essa quantidade atinge zero, ele retorna ao início da cena e deve refazer esta etapa. A única forma de restaurar seus pontos de vida é avançando para a próxima cena.

RF017 – Registro de Tempo

Cada fase é dividida em cenas, e assim como em jogos antigos o tempo será registrado. Caso o jogador não consiga concluir a fase em determinado tempo limite, será fim de jogo e deverá reiniciar a fase a partir de sua primeira cena.

RF018 – Pontos

Os pontos do jogador serão registrados de acordo com seu desempenho ao longo da fase. Para contabilizar os pontos serão levados em consideração os seguintes fatores:

- O número de inimigos derrotados
- O número de coletáveis encontrados
- O tempo para concluir uma fase
- O número de respostas certas/erradas em um teste dinâmico ou em uma luta com um Inimigo Chefe

RF019 – Quadro de Pontuações

De forma similar às máquinas de *Arcade* antigas, o jogo deve ser capaz de registrar localmente as maiores pontuações, além de permitir o registro das iniciais do jogador que as obteve.

RF020 - Tabela de Resultados

Deve ser exibida ao final de cada fase, em forma de tabela ou similar, os resultados dos testes com quantas repostas corretas, tempo total, e pontuação da respectiva fase.

RF021 – Opção *Replay*

Após completa uma fase deve ser fornecida uma opção de *replay* da fase para que possa ser jogada novamente caso o jogador acredite que não teve um bom desempenho e precise estudar mais um determinado tópico.

RF022 - Seleção de Fases

Após completo, o jogo deve fornecer uma opção de seleção de fases, que poderá ser acessada a partir do menu principal ou liberada por um código digitado, para que qualquer fase possa ser jogada novamente caso o jogador deseje.

RF023 - Dificuldade Aumentada

Após completo, o jogo deve fornecer uma opção, que poderá ser acessada a partir do menu principal ou liberada por um código digitado, de dificuldade aumentada, com diversos fatores alterados:

- Menos vida disponível ao jogador;
- Inimigos mais fortes;
- Mais perguntas por fase.

RF024 - Teste Final

Após completo, o jogo deve fornecer uma opção, que poderá ser acessada a partir do menu principal ou liberada por um código digitado, denominada teste final, para que o jogador desafie em sequência os Inimigos Chefes mais fortes e com todas as perguntas disponíveis no banco fixo de questões. A recompensa por completar com sucesso o teste final liberaria o equivalente ao modo desenvolvedor, o jogador não tomaria dano e poderia jogar como um jogo de plataforma comum após ter concluído seus estudos por completo.

RF025 - Códigos

O menu do jogo deverá fornecer as seguintes opções:

- Iniciar jogo
- *Leaderboards* (Tabela de Resultados)
- Códigos

Os Códigos deverão permitir ao jogador liberar novas opções e modos de jogo caso o mesmo deseje desafios maiores. Esses códigos serão liberados ao final de cada jogatina de acordo com o desempenho final do jogador.

- Desempenho Máximo: Teste Final
- Desempenho Mediano: Dificuldade Aumentada
- Desempenho Mínimo Necessário: Seleção de Fases

4 Descrição dos Requisitos Não Funcionais (RNF)

RNF001 – Hospedar o OOPPG num website

O jogo deverá ser *hospedado* no site Heroku.

RNF002 – Compatibilidade com Browsers

O software deve ser compatível com os *browsers* Google Chrome, Firefox e Brave.

RNF003 – Compatibilidade com Sistemas Operacionais

O software deve ser compatível com os Sistemas Operacionais Windows e Linux em suas distribuições: Arch-Linux, Ubuntu 20.04.4 LTS e Linux Mint 18.3.

RNF004 – Linguagem de Programação

A linguagem de programação a ser utilizada é o *TypeScript*.

RNF005 – Restrições de Hardware

O sistema não deve executar em mais de 8GB de memória RAM.

RNF006 – Banco de questões

O banco de questões é fixo, no formato JSON, e deverá ter pelo menos 15 questões sobre os pilares da Orientação a Objetos.

RNF007 – Fluidez

O jogo deve garantir fluidez, mesmo que em caso de erros, para que não haja uma estagnação na jogatina.

RNF008 – Inputs

O jogo deve possuir tempo de resposta de no máximo 500 milissegundos aos comandos dados pelo jogador, evitando dessa forma frustrações com os controles.

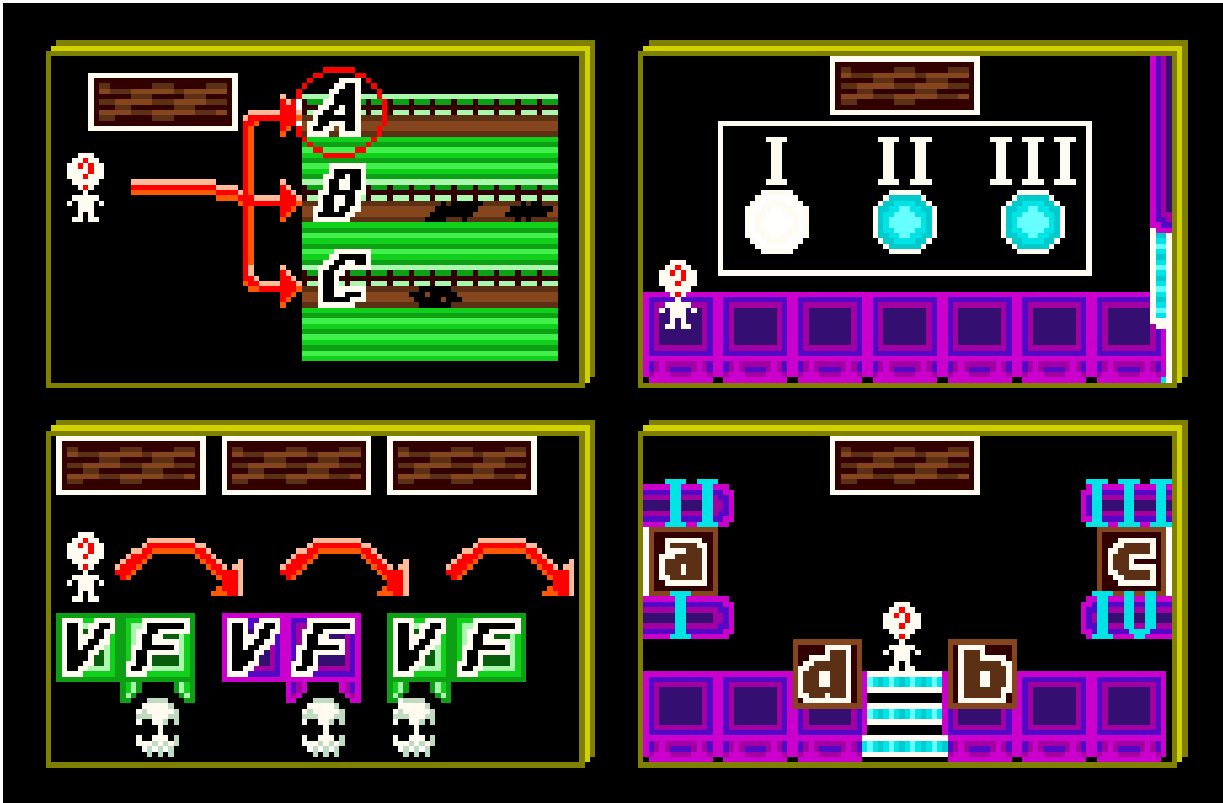
RNF009 – Padrão de codificação

O código seguirá o seguinte padrão de codificação: [JavaScript Coding Guidelines](#).

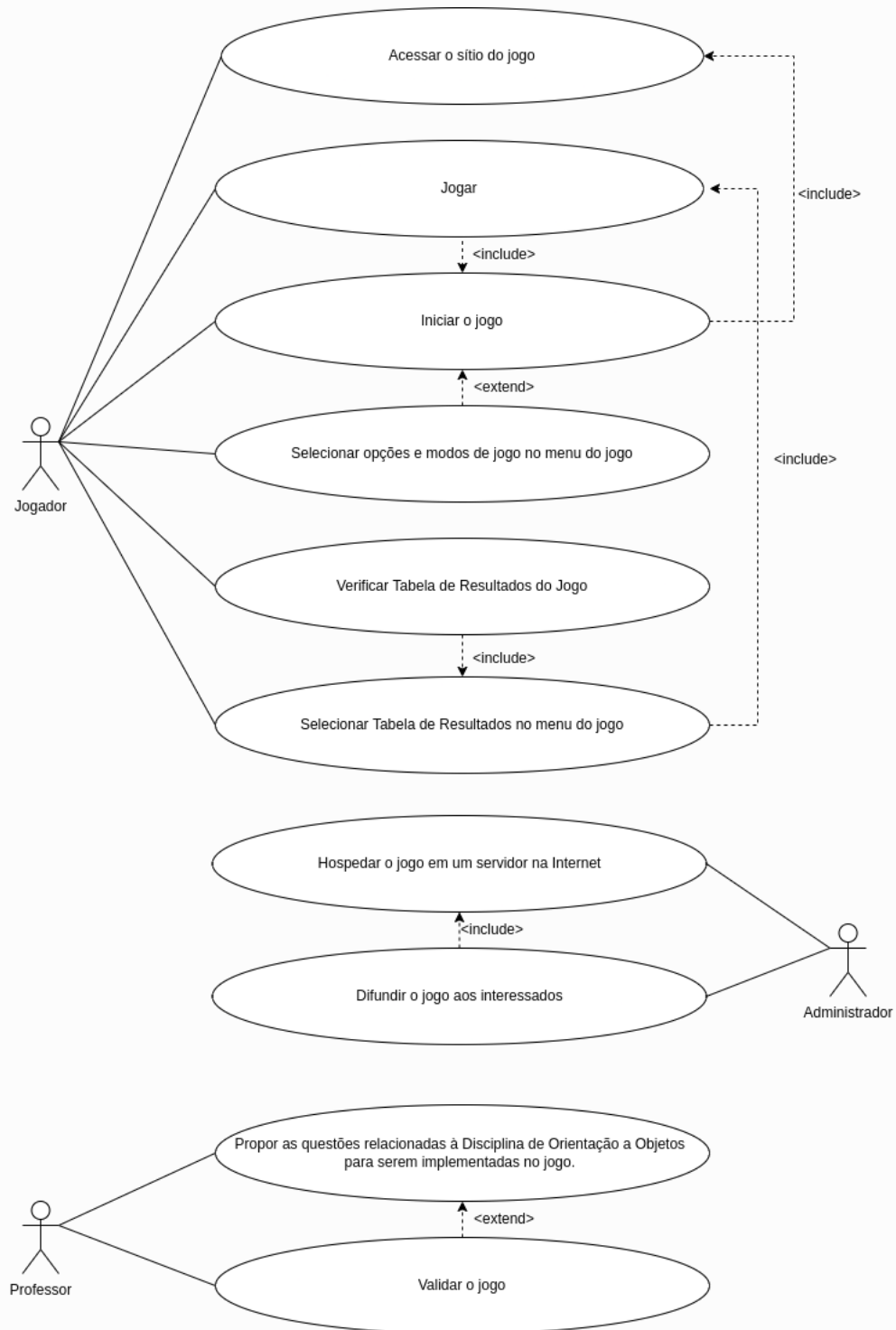
RNF010 – Padrão de commits

Os commits seguirão as regras encontradas em [Conventional Commits](#).

5 Artes Conceituais



6 Diagrama de casos de uso



7 Diagrama de classes

