

**CENTRO PAULA SOUZA
FATEC INDAIATUBA - FACULDADE DE TECNOLOGIA DE
INDAIATUBA DR. ARCHIMEDES LAMMOGLIA**

**GIULIANA CAMPREGHER MISSIO
JULIA ROMANI BARRETA
MATEUS RODRIGUES VIEIRA
ROBSON GARCIA DOS ANJOS
YOLANDA FERREIRA DE SOUZA**

**Season Gems
Jogo desktop**

**PI
PROJETO INTERDISCIPLINAR**

**INDAIATUBA
2021**

GIULIANA CAMPREGHER MISSIO

JULIA ROMANI BARRETA

MATEUS RODRIGUES VIEIRA

ROBSON GARCIA DOS ANJOS

YOLANDA FERREIRA DE SOUZA

Season Gems
Jogo desktop

Este trabalho visa como seu principal objetivo contribuir para o a compilação de todos os recursos e ensinamentos vinculados pelo Projeto Integrador de disciplinas da FATEC Indaiatuba Dr. Archimedes Lammoglia.

INDAIATUBA
2021

RESUMO

Season Gems é um jogo de *RPG* para desktop, que possibilita jogabilidade intuitiva de qualidade. Através da tecnologia desenvolvida, o jogador pode cadastrar quantos usuários quiser, armazenando as informações de cada login, *score* e a fase atual. O software conta também com design e enredo elaborados para melhor entretenimento do cliente. Apesar de ser desenvolvido para o público jovem *SG* é capaz de envolver pessoas de todas as idades.

Palavras-Chave: Jogos, *RPG*, entretenimento.

ABSTRACT

Season Gems is a desktop role-playing game that enables quality intuitive gameplay. Through the developed technology, the player can register as many users as he wants, storing the information of each login, score, and the current phase. The software also has a design and plot designed for better customer entertainment. Despite being developed for a young audience, SG can involve people of all ages.

Keywords: Games, RPG, entertainment.

SUMÁRIO DE TABELAS

Tabela 1: TCO - Cliente	8
Tabela 2: TCO - Fornecedor	9
Tabela 3: ROI - Cliente.....	10
Tabela 4: ROI - Fornecedor.....	11

SUMÁRIO DE FIGURAS

Figura 1: Diagrama de Casos de Uso	15
Figura 2: Diagrama de Classes	16
Figura 3: Diagrama de Sequência.....	18
Figura 4: Diagrama de Atividades	19
Figura 5: Tela inicial – Season Gems.....	20
Figura 6: Tela de Login – Season Gems.....	20
Figura 7: Tela de Cadastro – Season Gems	21
Figura 8: Tela história – Season Gems	21
Figura 9: Tela Jogo 1 – Season Gems.....	21
Figura 10: Tela Jogo 2 – Season Gems.....	22
Figura 11: Tela Jogo 3 – Season Gems.....	22
Figura 12: Tela Jogo 4 – Season Gems.....	22
Figura 13: Tela Vitória – Season Gems	23
Figura 14: Tela Ranking – Season Gems	23
Figura 15: Folder Inglês – Season Gems.....	24
Figura 16: MER	25
Figura 17: Script das Tabelas.....	26
Figura 18: Script - Inserção das Fases.....	26
Figura 19: Manual do Usuário	28

SUMÁRIO

1. INTRODUÇÃO	7
2. CONTEÚDO	8
2.1. PAPEL DAS DISCIPLINAS	8
2.1.1. GESTÃO E GOVERNANÇA DA TECNOLOGIA DA INFORMAÇÃO	8
2.1.1.1. Custo Total do Investimento	8
2.1.1.1.1. Prisma do cliente	8
2.1.1.1.2. Prisma do fornecedor	9
2.1.1.2. Retorno do Investimento	10
2.1.1.2.1. Prisma do cliente	10
2.1.1.2.2. Prisma do do fornecedor	11
2.1.2. ENGENHARIA DE SOFTWARE II	12
2.1.2.1. Escopo	12
2.1.2.1.1. Contextualização	12
2.1.2.1.2. Objetivos e justificativa	12
2.1.2.1.3. Evolução do sistema	12
2.1.2.1.4. Macro funções	13
2.1.2.1.5. Restrições	13
2.1.2.1.6. Benefícios esperados	13
2.1.2.1.7. Requisitos	13
2.1.2.1.7.1. Requisitos Funcionais	13
2.1.2.1.7.1. Requisitos Não-Funcionais	14
2.1.2.1.8. Regras de negócio	14
2.1.2.2. Diagramas da UML	15
2.1.2.2.1. Diagrama de Casos de Uso	15
2.1.2.2.2. Diagrama de Classes	16
2.1.2.2.3. Diagrama de Sequência	18
2.1.2.2.4. Diagrama de Atividades	19
2.1.3. LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS	20
2.1.3.1. Interface do Usuário	20
2.1.4. INGLÊS	24
2.1.4.1. Resumo em Português	24
2.1.4.2. Resumo em Inglês	24
2.1.4.3. Folder	24
2.1.5. BANCO DE DADOS	25
2.1.5.1. Modelo Entidade Relacionamento	25
2.1.5.2. Script das Tabelas	25
2.1.5.3. Inserção das Fases	26
2.2. DESCRIÇÃO DO PROJETO PRÁTICO	27

2.2.1. Manuais	28
2.3. LIÇÕES APRENDIDAS E DIFICULDADES	29
2.4. OPORTUNIDADE DE MELHORIA	30
2.5. ANEXOS	31
2.6. BIBLIOGRAFIA	32
3. CONCLUSÕES	33

1. INTRODUÇÃO

Season Gems define-se como um jogo de *RPG* para *desktop*, que tem como objetivo principal entreter o usuário de forma intuitiva. O intuito é contribuir com o atual mercado de games *RPG* brasileiro, de modo que se mantenha acessível e abrangente para todos os públicos.

Para tornar isto possível foi preciso projetar um software qual o usuário pudesse se cadastrar com facilidade em quantas contas quiser, desta forma podendo repetir o percurso do jogo o quanto necessário para aperfeiçoar suas habilidades e bater seus próprios *records*, que serão apresentados ao final do jogo. Possuir diversos usuários também possibilita que mais de uma pessoa possa jogar em um mesmo computador.

Toda estrutura do software foi desenvolvida por cinco discentes da Faculdade de Tecnologia de Indaiatuba, Dr. Archimedes Lammoglia, Robson Garcia dos Anjos, Julia Romani Barreta, Mateus Rodrigues Vieira, Giuliana Campregher Missio e Yolanda Ferreira de Souza. Deste modo, cada um efetuando suas atividades a fim de alcançar o melhor desempenho.

O propósito dos desenvolvedores é trazer um modelo de jogo *RPG* simples e inovador. Com isto em vista, a ideia surgiu ao longo do semestre, em meio a discussões dos alunos que traziam como bagagem experiências anteriores, o conhecimento adquirido em aula e um olhar visionário.

2. CONTEÚDO

2.1. PAPEL DAS DISCIPLINAS

2.1.1. GESTÃO E GOVERNANÇA DA TECNOLOGIA DA INFORMAÇÃO

Esta disciplina foi um fator importante para o desenvolvimento do projeto, pois através dos conhecimentos adquiridos com ela foi possível fazer um planejamento e gerenciar o sistema de acordo com as nossas necessidades.

2.1.1.1. Custo Total do Investimento

2.1.1.1.1. Prisma do cliente

No exemplo escolhido o usuário é competidor de campeonatos de jogos, para assim podermos aplicar as análises de investimento. Tendo isso em vista o único custo é o valor da licença do software, que é pago uma vez e equivale a uma instalação.

Tabela 1: TCO - Cliente

Custo					
Item	Quantidade	Valor		Total	
Licença do Jogo	1	R\$	50,00	R\$	50,00
custo	R\$			50,00	

Fonte: Criação Própria

2.1.1.1.2. Prisma do fornecedor

A licença de uso do jogo é um serviço, portanto desprovida de material físico, e por se tratar de um Jogo Indie, ou seja, criado por uma equipe sem apoio financeiro, o custo fixo com hardware é nulo (Giorgi, 2020). Há, entretanto, três custos associados à licença de uso vendida: o suporte técnico que utiliza mão de obra, com a particularidade de que um atendente suporta “n” licenciados, e o uso da infraestrutura de Cloud Computing, cujo custo é função das características de uso de cada licenciado e o gerenciamento de atualização, que visa aprimorar o site e melhorar a experiência do usuário. Além disto, existe a despesa única de mão de obra direta, que só é cobrada uma vez.

Tabela 2: TCO - Fornecedor

Mão de obra direta				
Item	Quantidade	Valor/h	Horas	Total
Programador	2	R\$ 27,69	48	R\$ 2.658,24
Banco de dados	1	R\$ 34,75	8	R\$ 278,00
Engenharia de Software	1	R\$ 43,26	40	R\$ 1.730,40
Designer	1	R\$ 15,00	36	R\$ 540,00
Finanças e vendas	1	R\$ 8,67	8	R\$ 69,36
R\$				5.276,00

Custo fixo				
Item	Valor	Ruim	Médio	Bom
Cloud Computing por licença	R\$ 1,50	R\$ 75,00	R\$ 150,00	R\$ 750,00
Profissional para atualizações	R\$ 1.772,16	R\$ 1.772,16	R\$ 1.772,16	R\$ 1.772,16
Profissional do suporte por licenciado	R\$ 1,00	R\$ 50,00	R\$ 100,00	R\$ 500,00
custo		R\$ 1.897,16	R\$ 2.022,16	R\$ 3.022,16

Fonte: Criação Própria

2.1.1.2. Retorno do Investimento

2.1.1.2.1. Prisma do cliente

Supondo que dentro de um ano um competidor se prepare e participe de um campeonato, o retorno do seu investimento ocorrerá ao fim do campeonato. O valor do investimento é o mesmo em todo os casos, o retorno varia segundo a sua colocação. A tabela abaixo exhibe o Retorno do investimento em três possíveis cenários.

Tabela 3: ROI - Cliente

	caso 1	caso 2	caso 3
custo	50	50	50
receita	500	2000	5000

Cenário ruim - 50 licenças por mês												
SLO	mês 1	mês 2	mês 3	mês 4	mês 5	mês 6	mês 7	mês 8	mês 9	mês 10	mês 11	mês 12
receita	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	500,00
custo	50,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
lucratividade	-50,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	500,00
lucratividade%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	100%
acumulado	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	R\$ 450,00
												PAYBACK

Cenário médio - 100 licenças por mês												
SLO	mês 1	mês 2	mês 3	mês 4	mês 5	mês 6	mês 7	mês 8	mês 9	mês 10	mês 11	mês 12
receita	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 2.000,00
custo	R\$ 50,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00
lucratividade	-R\$ 50,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 2.000,00
lucratividade%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	100%
acumulado	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	R\$ 1.950,00
												PAYBACK

Cenário bom - 500 licenças por mês												
SLO	mês 1	mês 2	mês 3	mês 4	mês 5	mês 6	mês 7	mês 8	mês 9	mês 10	mês 11	mês 12
receita	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 5.000,00
custo	R\$ 50,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00
lucratividade	-R\$ 50,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 0,00	R\$ 5.000,00
lucratividade%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	-100%	100%
acumulado	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	-R\$ 50,00	R\$ 4.950,00
												PAYBACK

Fonte: Criação Própria

2.1.1.2.2. Prisma do fornecedor

Usando recursos de *cloud computing*, a expansão do volume de produção (quantidade de licenciamentos em produção) é ilimitada. Dado que o custo fixo é diluído pela extensão dos licenciamentos e que o preço e o custo dependem dos mesmos indicadores, então a receita marginal é sempre crescente. Com isso, o ganho de escala é contínuo. A tabela abaixo exhibe o Retorno do investimento em três possíveis cenários.

Tabela 4: ROI - Fornecedor

	caso 1	caso 2	caso 3
mão de obra d	5276	5276	5276
custo fixo	R\$ 1.897,16	R\$ 2.022,16	R\$ 3.022,16
receita	2500	5000	25000

Cenário ruim - 50 licenças por mês												
	TRANSIÇÃO	SLO	SLO	SLO	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA
	mês 1	mês 2	mês 3	mês 4	mês 5	mês 6	mês 7	mês 8	mês 9	mês 10	mês 11	mês 12
receita	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00	2500,00
custo	7173,16	1897,16	1897,16	1897,16	1897,16	1897,16	1897,16	1897,16	1897,16	1897,16	1897,16	1897,16
lucratividade	-4673,16	602,84	602,84	602,84	602,84	602,84	602,84	602,84	602,84	602,84	602,84	602,84
lucratividade%	-48%	14%	14%	14%	14%	14%	14%	14%	14%	14%	14%	14%
acumulado	-R\$ 4.673,16	-R\$ 4.070,32	-R\$ 3.467,48	-R\$ 2.864,64	-R\$ 2.261,80	-R\$ 1.658,96	-R\$ 1.056,12	-R\$ 453,28	R\$ 149,56	R\$ 752,40	R\$ 1.355,24	R\$ 1.958,08
									PAYBACK			

Cenário médio - 100 licenças por mês												
	TRANSIÇÃO	SLO	SLO	SLO	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA
	mês 1	mês 2	mês 3	mês 4	mês 5	mês 6	mês 7	mês 8	mês 9	mês 10	mês 11	mês 12
receita	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00	R\$ 5.000,00
custo	R\$ 7.298,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16	R\$ 2.022,16
lucratividade	-R\$ 2.298,16	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84	R\$ 2.977,84
lucratividade%	-19%	42%	42%	42%	42%	42%	42%	42%	42%	42%	42%	42%
acumulado	-R\$ 2.298,16	R\$ 679,68	R\$ 3.657,52	R\$ 6.635,36	R\$ 9.613,20	R\$ 12.591,04	R\$ 15.568,88	R\$ 18.546,72	R\$ 21.524,56	R\$ 24.502,40	R\$ 27.480,24	R\$ 30.458,08
		PAYBACK										

Cenário bom - 500 licenças por mês												
	SLO	SLO	SLO	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA	SLA
	mês 1	mês 2	mês 3	mês 4	mês 5	mês 6	mês 7	mês 8	mês 9	mês 10	mês 11	mês 12
receita	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00	R\$ 25.000,00
custo	R\$ 8.298,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16	R\$ 3.022,16
lucratividade	R\$ 16.701,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84	R\$ 21.977,84
lucratividade%	50%	78%	78%	78%	78%	78%	78%	78%	78%	78%	78%	78%
acumulado	R\$ 16.701,84	R\$ 38.679,68	R\$ 60.657,52	R\$ 82.635,36	R\$ 104.613,20	R\$ 126.591,04	R\$ 148.568,88	R\$ 170.546,72	R\$ 192.524,56	R\$ 214.502,40	R\$ 236.480,24	R\$ 258.458,08
	PAYBACK											

Fonte: Criação Própria

2.1.2. ENGENHARIA DE SOFTWARE II

O principal fator ligado a estrutura do jogo, e a escolha dos processos do software. A partir dos conhecimentos adquiridos nesta matéria, elaborou-se as os diagramas para obter os principais requisitos do sistema para assim oferecer uma usabilidade de qualidade.

2.1.2.1. Escopo

2.1.2.1.1. Contextualização

O *software* em desenvolvimento é um jogo para desktop, não há necessidade de incluir modelos de negócios ou quaisquer processos empresariais.

O objetivo do programa está ligado ao contexto mundial de pandemia, quando se faz necessário praticar o distanciamento social os jogos se revelam como entretenimento divertido, seguro e acessível.

2.1.2.1.2. Objetivos e justificativa

O software de entretenimento tem como objetivo, divertir com uma mecânica de movimentação e interação com o mapa de forma simples e intuitiva, sem a necessidade de que o jogador esteja familiarizado no mundo dos jogos, tornando-se assim, mais acessível.

Ideia da criação de um jogo está diretamente relacionada ao momento histórico, no qual o mesmo está sendo desenvolvido (pandemia mundial do COVID-19). Na realidade que estamos inseridos, todos estão se preservando em suas casas a maior parte do tempo, sendo constantemente bombardeados de informações negativas através de inúmeras plataformas. O programa se apresenta como uma escapatória desse ambiente, para entreter e divertir de uma maneira segura qualquer cidadão.

2.1.2.1.3. Evolução do sistema

Será desenvolvido um novo projeto, não existe nenhuma ligação com sistemas apresentados nos semestres anteriores.

2.1.2.1.4. Macro funções

- Funções de movimento do personagem.
- Função de mudança de cenário/fases.
- Função de contagem de pontos.
- Função de armazenamento de desempenho.
- Função de cadastro.
- Função de login.
- Função de cronometro.
- Listagem de ranking.

2.1.2.1.5. Restrições

- O jogo deverá ser desenvolvido em JAVA na plataforma “Eclipse”.
- Em consequência ao modelo remoto, os integrantes contam com recursos tecnológicos individuais, que quando em falha podem atrapalhar na produtividade e processo de desenvolvimento.
- O projeto necessita da participação de todos os integrantes durante o desenvolvimento do projeto. Portanto, caso haja algum problema com saúde dos mesmos o processo poderá ser afetado.

2.1.2.1.6. Benefícios esperados

- Contribuir com a sociedade através de entretenimento, durante o distanciamento social e combate ao COVID-19.
- O jogo irá contar com uma arte em pixel, levando assim muitos amantes dos indies como por exemplo, fãs de “Zelda” e “Undertale” virem a se interessar ao nosso produto pela experiência que será oferecida.

2.1.2.1.7. Requisitos

2.1.2.1.7.1. Requisitos Funcionais

RF01 – O sistema deve cadastrar nome de usuário e senha (no banco de dados).

RF02 – O sistema deve movimentar o personagem.

RF03 – O sistema deve salvar (através do banco de dados), o *checkpoint* da personagem (a última fase desbloqueada) para que o usuário não perca seu progresso.

RF04 – Caso o usuário não consiga concluir os requisitos específicos da fase, o sistema deverá retirar uma vida do personagem.

RF05 – O sistema deve registrar o mais alto nível de *checkpoint* do usuário.

RF06 – O jogador pode consultar o *ranking* dos melhores jogadores.

2.1.2.1.7.1. Requisitos Não-Funcionais

RQ01 – O jogo deve ter facilidade de uso para o jogador. - **Usabilidade**

RQ02 – O *Design* deve ser agradável e atrativo ao usuário. - **Design**

RQ03 – O sistema deve ser de fácil manutenção, incluindo código limpo e comentários.
- **Manutenibilidade**

RQ04 – O jogo deve possuir um bom desempenho a fim de evitar lentidão ou problemas que possam prejudicar o jogador de alguma maneira. - **Desempenho**

RQ05 – O jogo deve ter seu sistema de desenvolvimento totalmente padronizado de acordo com a metodologia de gestão e planejamento de projetos de software. – **Padronização.**

2.1.2.1.8. Regras de negócio

RN01 – O jogador deverá começar na primeira fase.

RN02 – O jogador não poderá avançar de fase sem antes completar a atual.

RN03 – O jogador não pode acessar as fases anteriores.

RN04 – O jogador tem apenas três chances de para errar durante toda jornada, estas que são chamadas de “vidas”.

RN05– Quando o jogador estiver com o número de vidas zerado, ele retorna à primeira fase, e todas as outras fases são novamente bloqueadas.

RN06 – O jogador possui movimentos limitados, só podendo se movimentar na vertical e na horizontal, (não possui movimentos diagonais).

2.1.2.2. Diagramas da UML

UML (*Unified Modeling Language*) é uma linguagem que define uma série de artefatos que nos ajuda na tarefa de modelar e documentar os sistemas orientados a objetos que desenvolvemos. (Leandro, 2012)

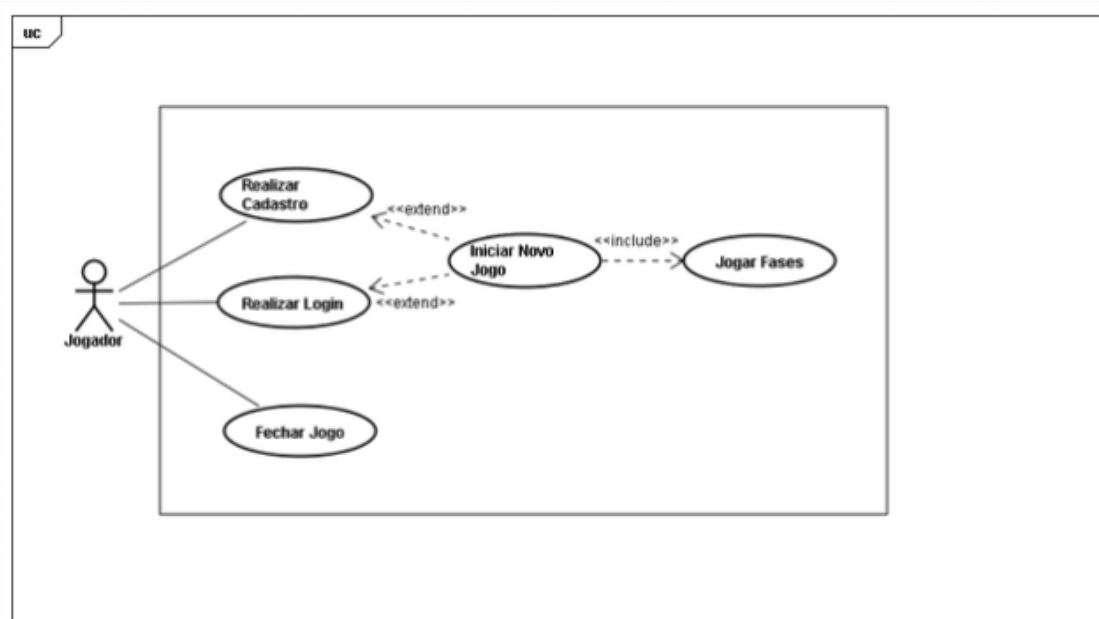
2.1.2.2.1. Diagrama de Casos de Uso

O planejamento do nosso software conta com cinco casos de uso, entre eles estão:

- O caso de uso “Realizar login” que é responsável por identificar o jogador.
- O segundo caso de uso se chama “Realizar cadastro”, é ele o que cadastra o novo jogador.
- O caso de uso “Iniciar novo Jogo” possui um nome autoexplicativo, começa um jogo novo da fase um.
- “Jogar fases” é o caso de uso de jogabilidade.
- Por último temos o caso de uso “Fechar jogo”, que fecha a aplicação.

Este diagrama documenta o que o sistema faz do ponto de vista do usuário. Em outras palavras, ele descreve as principais funcionalidades do sistema e a interação dessas funcionalidades com os usuários do mesmo sistema. Nesse diagrama não nos aprofundamos em detalhes técnicos que dizem como o sistema faz.

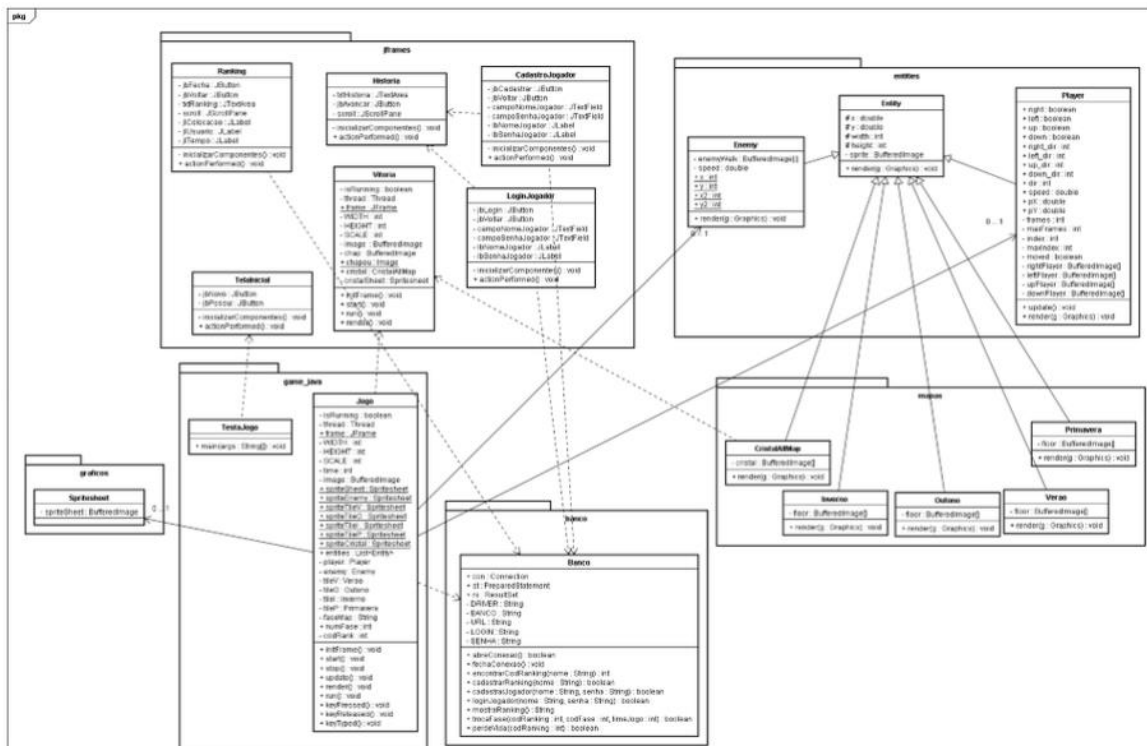
Figura 1: Diagrama de Casos de Uso



2.1.2.2.2. Diagrama de Classes

O diagrama de classes descreve os vários tipos de classes existentes num sistema e os relacionamentos que existem entre elas, também mostra os atributos (propriedades) e as operações (serviços) de uma classe. Diagramas de classe permitem, planejar como as classes/objetos funcionarão e interagirão. O Diagrama estará anexo a esta documentação para melhor visualização.

Figura 2: Diagrama de Classes



Fonte: Criação Própria

As classes representadas no diagrama são, respectivamente:

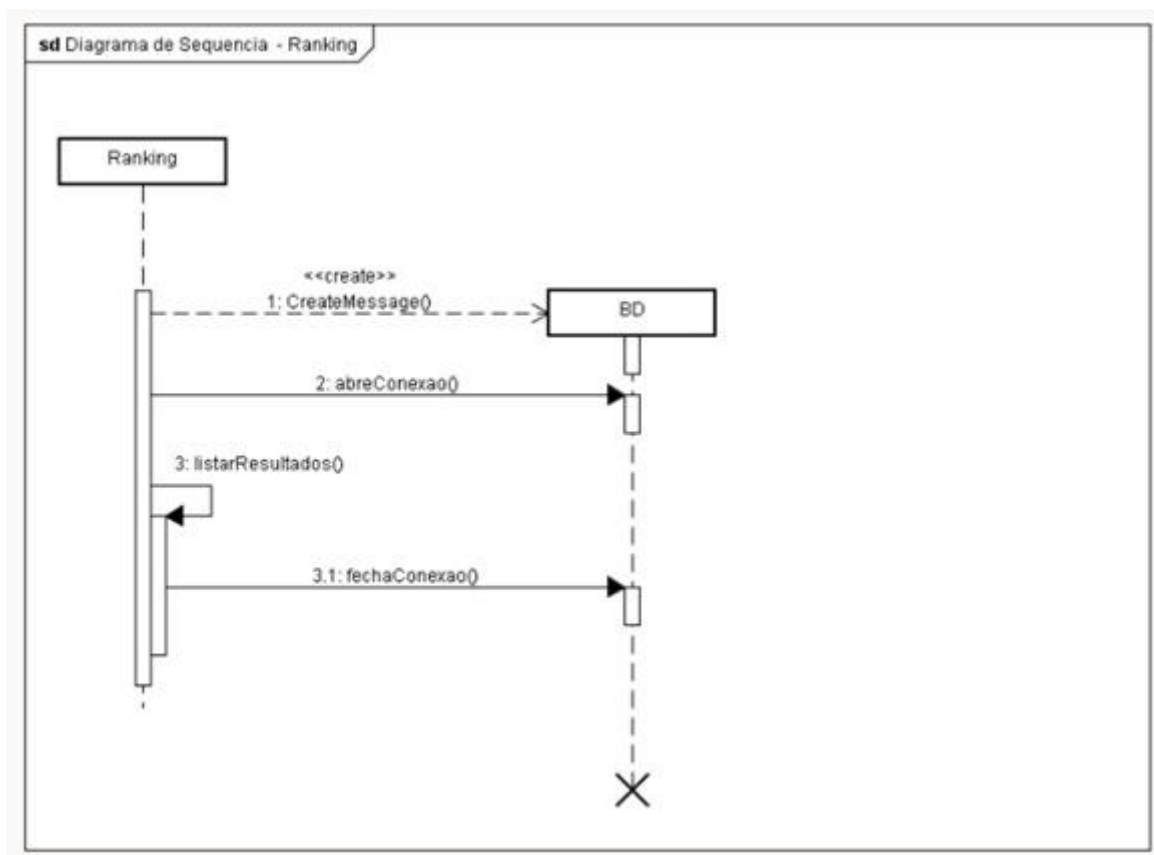
- TestaJogo: é a classe que possui o *main*, ou seja, que inicia o sistema e encaminha o usuário a TelaInicial.
- Jogo: possui todos os métodos e atributos para a renderização do mapa e dos personagens e processa as fases do jogo.
- Banco: possui todos os atributos e métodos para conexão o SQL Server e os métodos consulta, inserção e alteração de dados do jogador e seu ranking durante todo o jogo.
- TelaInicial: é a primeira classe estendida a JFrame que possibilita o usuário a escolher se irá se cadastrar ou realizar login.
- CadastroJogador: é a classe estendida a JFrame que realiza o cadastro do jogador utilizando a classe Banco.

- LoginJogador: é a classe estendida a JFrame que realiza o login do jogador utilizando a classe Banco.
- Historia: é a classe estendida a JFrame que apresenta o contexto do jogo e encaminha o usuário ao início do jogo.
- Vitoria: é uma classe estendida do Canvas que apresenta a mensagem de vitória o usuário e ela instancia e abre o JFrame do ranking.
- Ranking: é a classe estendida a JFrame que apresenta todos os jogadores que conquistaram as 5 fases em ordem de menor tempo utilizando o Banco.
- Entity: é a classe modelo para todas aquelas que precisam renderizar algum elemento.
- Enemy: é utilizada para renderizar o inimigo.
- Player: é utilizada para renderizar o jogador.
- Inverno, Primavera, Outono, Verão: cada classe renderiza as imagens respectivas, das suas estações, dos mapas de cada fase.
- cristalAllMap: é a classe que utiliza entity como modelo para renderizar o cristal mostrado no final do jogo.
- Spritesheet: recebe a imagem e retorna os parâmetros de recorte da imagem.

2.1.2.2.3. Diagrama de Sequência

Os diagramas de sequência, comumente usados pelos desenvolvedores, modelam as interações entre objetos em um único caso de uso. Eles ilustram como as diferentes partes de um sistema interagem entre si para realizar uma função, e a ordem em que as interações ocorrem quando um determinado caso de uso é executado. Ele é estruturado de tal forma que representa uma linha de tempo que começa no topo e desce gradualmente para marcar a sequência de interações. (Glossário UFMG, s.d.)

Figura 3: Diagrama de Sequência



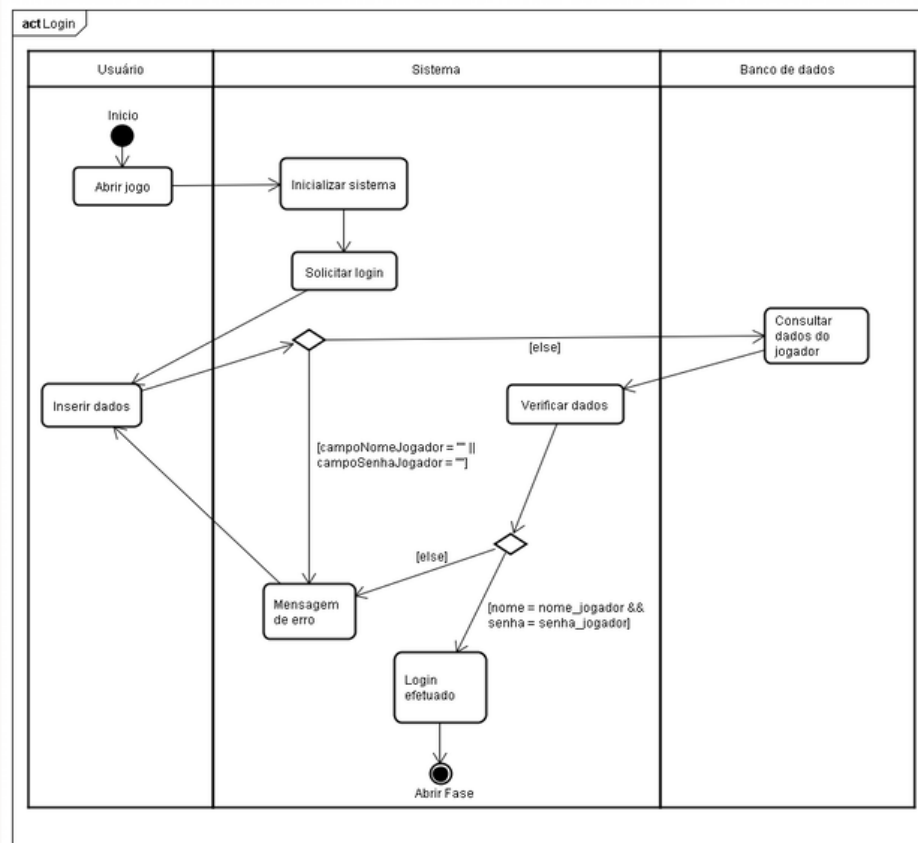
Fonte: Criação Própria

Quando o jogador passa todas as fases é exibido um ranking. Essa classe possui ligação com a classe BD que cria uma conexão com o banco de dados e através disso é mostrada uma lista, em ordem, dos jogadores que completaram o jogo em menos tempo. Após essa lista ser exibida a conexão com o banco de dados é fechada.

2.1.2.2.4. Diagrama de Atividades

Um diagrama de atividade ilustra a natureza dinâmica de um sistema pela modelagem do fluxo de controle de atividade a atividade. Uma atividade representa uma operação em alguma classe no sistema que resulta em uma mudança no estado do sistema. O diagrama de atividades é representado por um gráfico de atividades que mostram o fluxo de uma atividade para outra. (Glossário UFMG, s.d.)

Figura 4: Diagrama de Atividades



Fonte: Criação Própria

Esse diagrama de atividades representa o fluxo de atividades para realização do login. A ordem seria abrir o jogo, inicializar o sistema, preencher os dados, se ao apertar o botão “Fazer Login” um dos campos Nome (campoNomeJogador) ou Senha (campoSenhaJogador) estiver vazio, será exibida uma mensagem de erro e voltará para a tela de preenchimento de dados. Se os dois campos estiverem preenchidos, o sistema irá se comunicar com o banco de dados que fará uma consulta com os dados preenchidos, logo após o sistema irá verificar os mesmos caso as duas informações sejam existentes e válidas, o login será efetuado e abrirá a fase, caso não, abrirá uma mensagem de erro que voltará para tela de inserção de dados.

2.1.3. LINGUAGEM DE PROGRAMAÇÃO ORIENTADA A OBJETOS

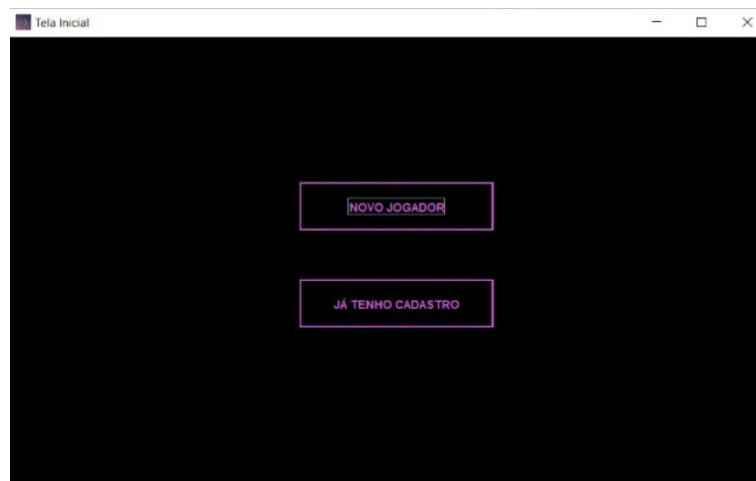
Esta disciplina foi essencial para o desenvolvimento do projeto, à partir dela encontramos o necessário quanto a codificação e desenvolvimento visual do sistema. Contribuindo também para identificar o melhor método para criação de jogos em Java e estabelecer conexão com o banco de dados.

2.1.3.1. Interface do Usuário

A aplicação conta com dez telas, todo design foi desenvolvido pelos discentes e todas as imagens são de autoria própria.

- Tela inicial

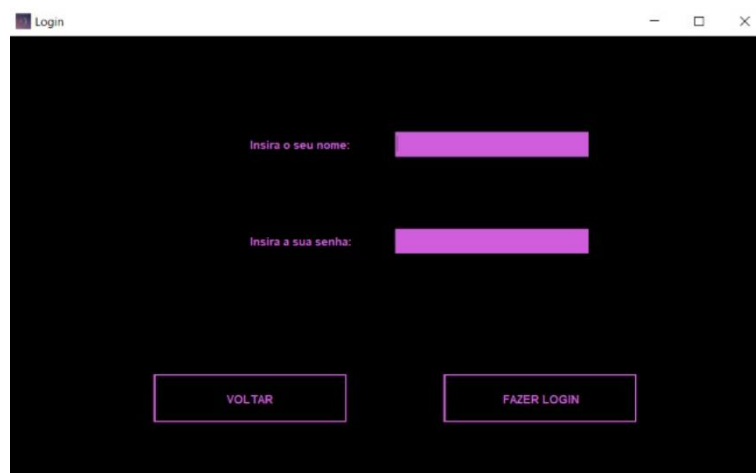
Figura 5: Tela inicial – Season Gems



Fonte: Criação Própria

- Tela de Login

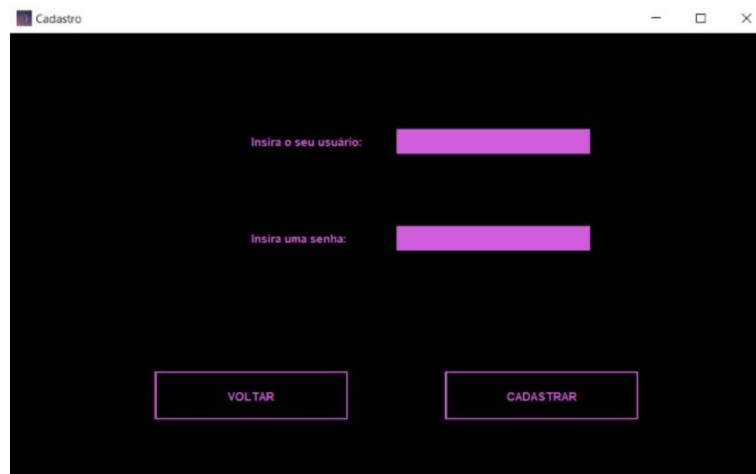
Figura 6: Tela de Login – Season Gems



Fonte: Criação Própria

- Tela de Cadastro

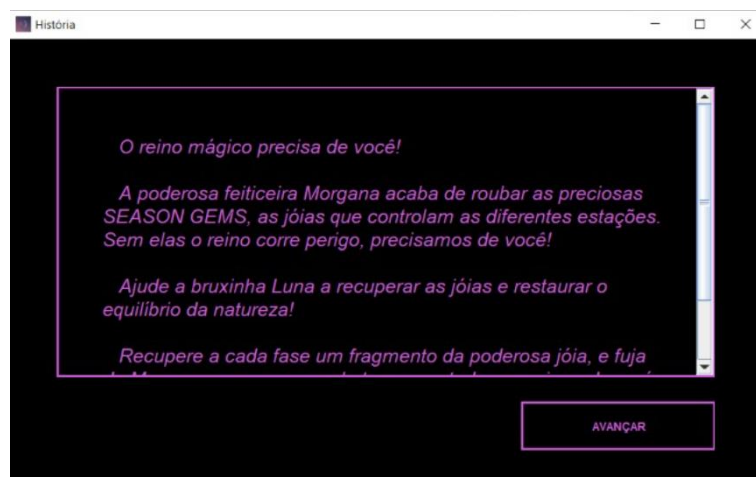
Figura 7: Tela de Cadastro – Season Gems



Fonte: Criação Própria

- Tela história do jogo

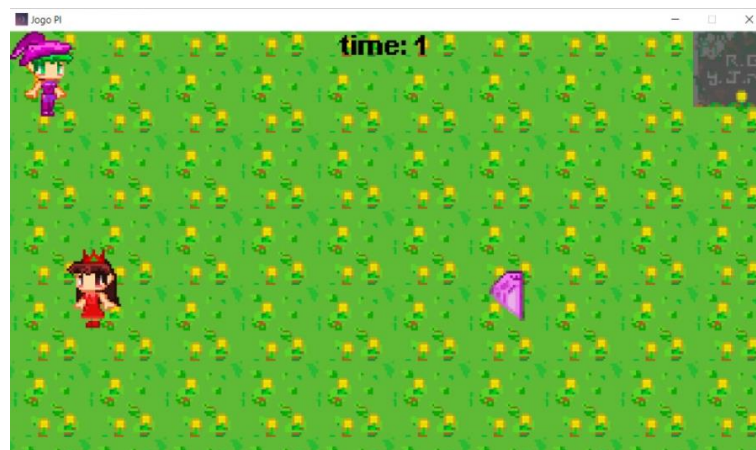
Figura 8: Tela história – Season Gems



Fonte: Criação Própria

- Tela de jogo 1 - Verão

Figura 9: Tela Jogo 1 – Season Gems



Fonte: Criação Própria

- Tela de jogo 2 - Outono

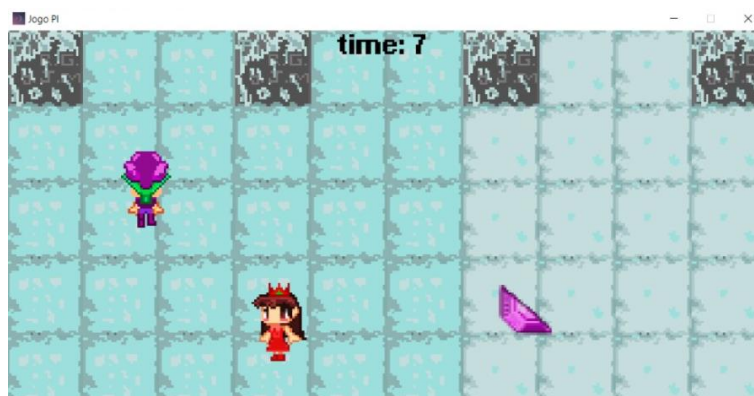
Figura 10: Tela Jogo 2 – Season Gems



Fonte: Criação Própria

- Tela de jogo 3 - Inverno

Figura 11: Tela Jogo 3 – Season Gems



Fonte: Criação Própria

- Tela de jogo 4 - Primavera

Figura 12: Tela Jogo 4 – Season Gems



Fonte: Criação Própria

- Tela vitória

Figura 13: Tela Vitória – Season Gems



Fonte: Criação Própria

- Tela ranking

Figura 14: Tela Ranking – Season Gems



Fonte: Criação Própria

2.1.3.2. Codificação

O projeto foi desenvolvido na linguagem de programação Java, através da ferramenta Eclipse. O projeto com todos os seus pacotes e bibliotecas estarão anexos a este documento.

2.1.4. INGLÊS

2.1.4.1. Resumo em português

A matéria nos deu suporte para conseguirmos concluir o vídeo de apresentação do sistema em inglês, visando assim alcançar o maior público possível.

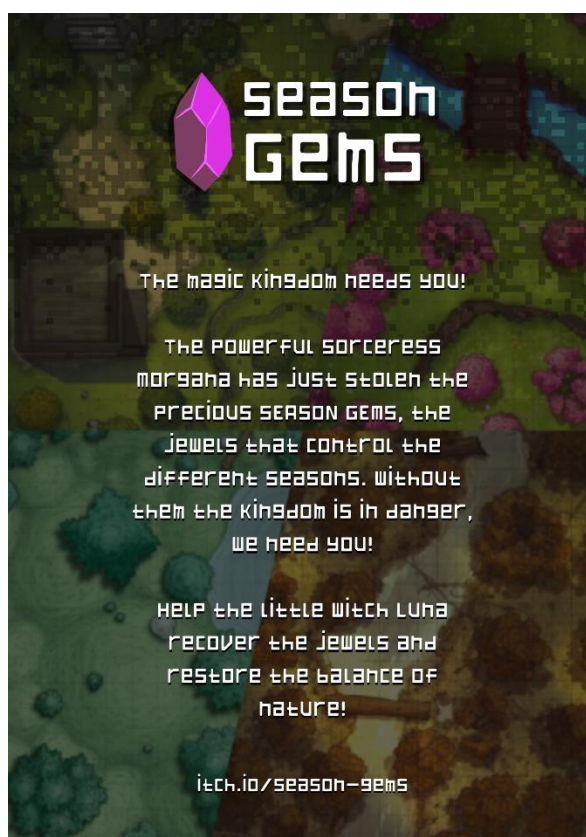
2.1.4.2. Resumo em inglês

The discipline gave us support to be able to complete the presentation video of the system in English, this aiming to reach the largest possible audience.

2.1.4.3. Folder

Elaboramos um folder em inglês, para divulgação mais eficiente do software.

Figura 15: Folder Inglês – Season Gems



Fonte: Criação Própria

2.1.5. BANCO DE DADOS

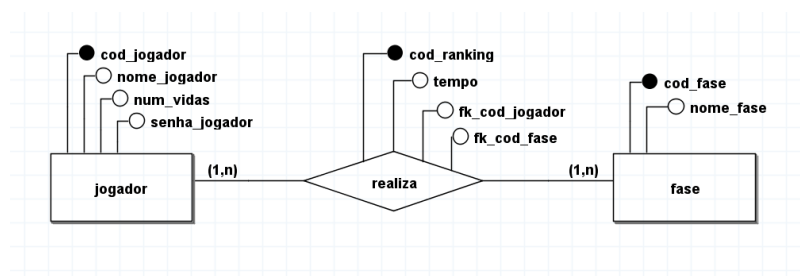
Esta disciplina foi essencial para o desenvolvimento do projeto, a partir dela encontramos o necessário quanto a representação visual do banco de dados e desenvolvimento do banco. Contribuindo também para identificar as tabelas necessárias e fazer o script das mesmas.

A ferramenta de gerenciamento de dados que escolhemos é o SQL Server, durante o semestre fomos instruídos sobre como utilizar o SQL Server Management Studio.

2.1.5.1. Modelo Entidade Relacionamento

O diagrama abaixo é o Modelo Entidade Relacionamento, um modelo de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requisitos de processo, de uma maneira abstrata. O modelo deste trabalho representa três tabelas, esta foi a base para a codificação.

Figura 16: MER



Fonte: Criação Própria

2.1.5.2. Script das Tabelas

O script abaixo efetua a criação do banco de dados e das tabelas do projeto, que são respectivamente:

- Tabela “jogador” que possui como colunas o código do jogador, o seu nome, senha e o número de vidas.
- A tabela “fase” conta com somente o número e nome das fases.
- E a tabela “ranking” é responsável por relacionar o jogador com a fase e armazenado o tempo utilizado pelo jogador para concluir cada fase. Esta é a tabela atualizada para fazer o ranking dos melhores jogadores.

Figura 17: Script das Tabelas

```
create DATABASE projetoPI

use projetoPI

create table jogador(
    cod_jogador int PRIMARY KEY IDENTITY,
    nome_jogador varchar(50) not null,
    num_vidas int not null,
    senha_jogador varchar(50) not null
)

create table ranking(
    cod_ranking int PRIMARY KEY IDENTITY,
    tempo time,
    fk_cod_jogador int not null,
    fk_cod_fase int not null,
)

create table fase(
    cod_fase int PRIMARY KEY IDENTITY,
    nome_fase varchar(10) not null
)

ALTER TABLE ranking ADD FOREIGN KEY (fk_cod_jogador) REFERENCES jogador(cod_jogador);
ALTER TABLE ranking ADD FOREIGN KEY (fk_cod_fase) REFERENCES fase(cod_fase);
```

Fonte: Criação Própria

2.1.5.3. Inserção das Fases

O script abaixo trata-se da inserção das fases no banco, estas não poderão ser alteradas da interface do usuário, por este motivo inclui-se diretamente em uma consulta no SQL Server Management Studio.

Figura 18: Script - Inserção das Fases

```
use projetoPI

INSERT INTO fase(nome_fase)
VALUES ('Primavera');

INSERT INTO fase(nome_fase)
VALUES ('Verão');

INSERT INTO fase(nome_fase)
VALUES ('Outono');

INSERT INTO fase(nome_fase)
VALUES ('Inverno');

INSERT INTO fase(nome_fase)
VALUES ('Boss');

select * from fase
```

Fonte: Criação Própria

2.2. DESCRIÇÃO DO PROJETO PRÁTICO

O projeto levou em consideração a diversão do usuário/jogador com gráficos e artes suaves e coloridas. Não deixando de lado a eficiência em processos como cadastro e login.

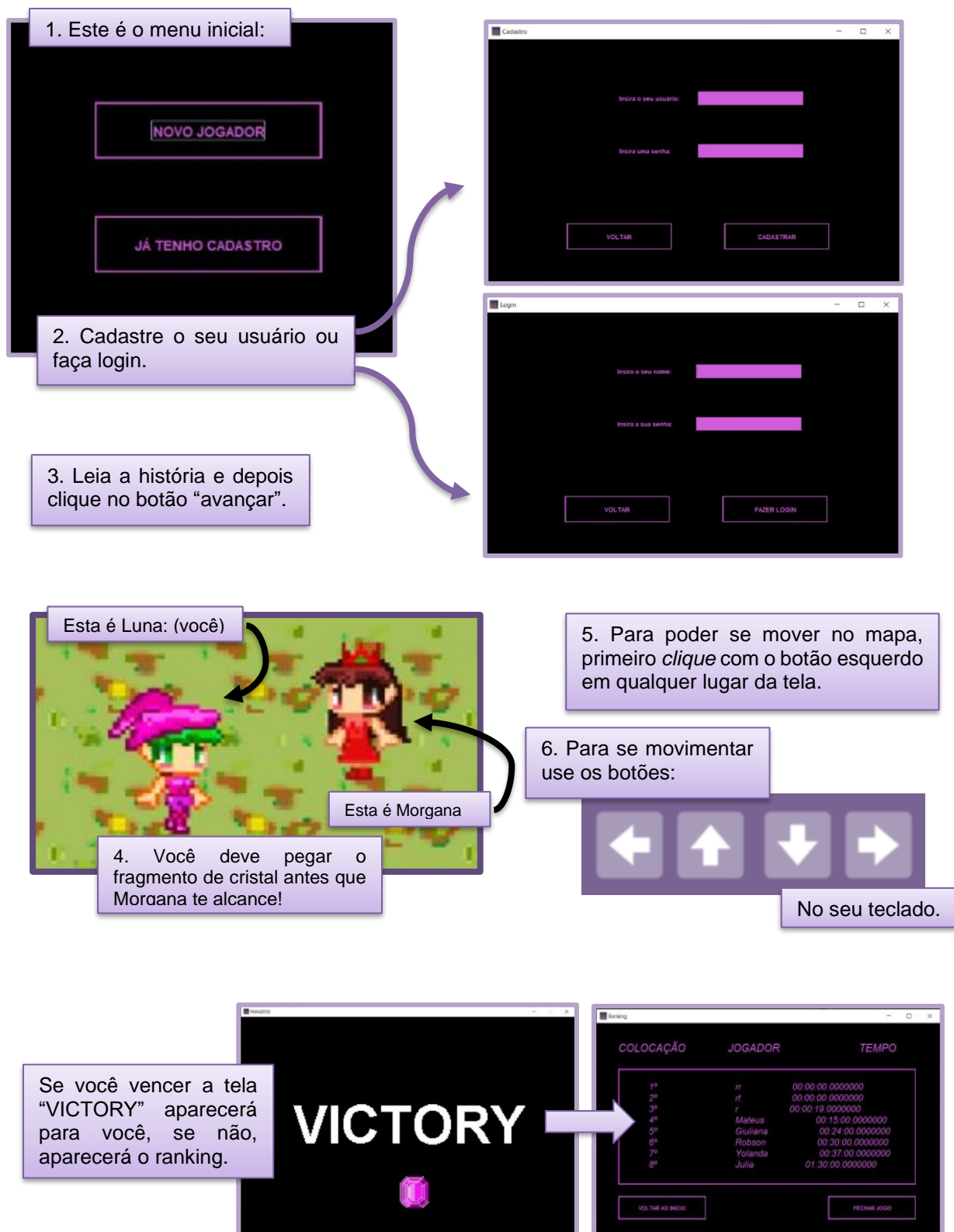
Para o desenvolvimento do projeto, foi usado a combinação de linguagens e bibliotecas em ambientes de desenvolvimento, e artes desenvolvidas pelos integrantes da equipe.

O sistema oportunizou o cadastro e login de usuário; as quatro fases propostas (estações); além a geração de um ranking com os melhores jogadores que concluíram a jornada. A língua utilizada na interface do jogo é o português, mas o jogo será divulgado em língua nativa e estrangeira (inglês), utilizando-se do folder apresentado anteriormente.

Todo o sistema segue anexo á este documento, em versão executável e todo o seu projeto de codificação em Java.

2.2.1. Manuais

Figura 19: Manual do Usuário



Fonte: Criação Própria

2.3. LIÇÕES APRENDIDAS E DIFICULDADES

Aprendemos a elaborar um sistema de modo prático e objetivo, mantendo tudo organizado e dinâmico para fácil utilização. Levantamos os requisitos de forma cooperativa, reunimos bibliografia e definimos regras de negócio e como elaborar a comunicação de diferentes classes e representá-las em diagramas.

Obtivemos um grande aprendizado nas questões referentes à estrutura de dados, como gerenciar bancos de dados, gerar consultas e fazer conexão em Java. Fomos instruídos sobre desenvolvimento de programas orientado a objetos, conhecemos novas bibliotecas e funções relacionadas a renderização e criação de interface de usuário.

Durante o percurso enfrentamos alguns desafios, entre eles estão, o distanciamento social e conexão com o banco de dados. Destarte a todo o aprendizado, é válido ressaltar que este processo é somente o começo de uma jornada de aprendizado para o desenvolvimento de jogos e outros tipos de softwares.

2.4. OPORTUNIDADE DE MELHORIA

Uma das possibilidades é ampliar o jogo para plataformas web, que possibilitam mais acessibilidade e versatilidade. Acreditamos que Season Games também chegará ao universo dos jogos mobile.

Além disto pretendemos ampliar o jogo com novas fases e níveis de dificuldade, acrescentar uma função para que o jogador visualize o número de vidas que restam e um contador para quando suas vidas acabarem ele volte ao início.

2.5. ANEXOS

Anexo a documentação temos:

- Diagrama de Classes
- Projeto Java
- Tabelas – Gestão e Governança de TI
- Consultas SQL Server – Banco de Dados
- Repositório ao programa através do GITHUB:
- <https://github.com/yoleihu>

2.6. BIBLIOGRAFIA

Giorgi, F. D. (03 de Janeiro de 2020). *Ecommerce Brasil*. Fonte:

<https://www.ecommercebrasil.com.br/>:

<https://www.ecommercebrasil.com.br/artigos/precificacao-custo-producao-software/>

Glossário UFMG. (s.d.). *DCC UFMG*. Fonte: homepages.dcc.ufmg.br:

https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/classse/diagrama_de_classe.htm

Glossário UFMG. (s.d.). *DCC UFMG*. Fonte: homepages.dcc.ufmg.br:

https://homepages.dcc.ufmg.br/~amendes/GlossarioUML/glossario/conteudo/atividades/diagrama_de_atividades.htm

Leandro. (2012). *DevMedia*. Fonte: www.devmedia.com.br:

<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>

3. CONCLUSÕES

Logo no início avaliamos muitas possibilidades, e ao analisar alguns dos caminhos que poderíamos seguir na programação orientada a objetos, optamos por desenvolver um jogo, a fim de nos desafiar e adquirir o maior conhecimento possível.

Conhecemos as múltiplas variáveis do gerenciamento de um novo software e fomos instruídos para tomar as decisões certas para obtermos o resultado esperado em diversos cenários de vendas.

Através da disciplina de Engenharia de Software, aprendemos como elaborar os principais modelos de diagramas da UML, levantamos os requisitos e as regras de negócios do nosso software, elementos decisivos no desenvolvimento.

Ao planejar a estrutura e organização do programa, colocamos em prática todo o conhecimento adquirido na sala de aula, descobrindo as principais tabelas que deveriam ser incluídas no banco de dados, como representá-las em um modelo Entidade Relacionamento, e principalmente, como conectá-las ao programa.

Em geral a experiência de criar o Season Gems e colocar em prova as competências conquistadas durante todo o semestre, foi enriquecedor para todos os membros da equipe. Construímos uma base para que planos futuros se concretizem com mais naturalidade e eficiência.

Deixamos aqui o nosso agradecimento a todos os profissionais envolvidos no auxílio do projeto e aos nossos colegas da turma de Análise e Desenvolvimento de Sistemas do 3º Semestre da Faculdade de Tecnologia Dr. Archimedes Lammoglia de Indaiatuba.