

Documentação de um Produto de Software

Academia de Dados

Nome dos Alunos:

*Bruno Teruya Ra: 22.01053-0;
Fernando de Oliveira Ra: 19.00468-0;
Gustavo Yudji Hiromoto Ra: 22.00839-0;
Daniel Djinishian de Briquez Ra: 22.00251-0
Diogo Carvalho Bernardini Ra: 22.01179-0;*

ÍNDICE DETALHADO

1. LEVANTAMENTO DOS REQUISITOS DO SISTEMA DE SOFTWARE.....	3
1.1. EXTRAÇÃO DE REQUISITOS.....	3
1.2. ANÁLISE DA COLETA DE REQUISITOS	3
2. ESPECIFICAÇÃO DOS REQUISITOS DO SISTEMA DE SOFTWARE	4
2.1. REQUISITOS FUNCIONAIS.....	4
2.2. REQUISITOS NÃO-FUNCIONAIS	9
3. ANÁLISE/PROJETO	9
3.1. DIAGRAMA DE CLASSES	9
3.2. DIAGRAMA DE SEQUÊNCIA	10
3.3. MODELO DE BANCO DE DADOS	15
3.4. DIAGRAMA DE ATIVIDADES (OPCIONAL)	15
3.5. DIAGRAMA DE ESTADOS (OPCIONAL).....	16
4. IMPLEMENTAÇÃO	19
5. TESTES.....	19
6. RESULTADOS E CONSIDERAÇÕES	19
APÊNDICE I	19

1. Levantamento dos Requisitos do Sistema de Software

1.1. Extração de Requisitos

Após contarmos, em linhas gerais, como seria nosso jogo, entrevistamos o professor Aparecido, passando por diferentes tópicos e começando com perguntas mais abrangentes. Nosso objeto tomar decisões e organizarmos tópicos muito importantes para a fase básica do desenvolvimento do nosso jogo. As áreas são: conteúdo didático a ser cobrado no jogo, Estrutura do Jogo e Programação das Estruturas do Jogo. Toda a entrevista foi, com a autorização de Aparecido, gravada em áudio.

A. Conteúdo didático:

- Quais são as prioridades de conteúdo que deveriam ser abordadas?
- Existe uma matéria que devemos destacar?
- Até qual matéria devemos nos limitar?
- Qual o enfoque do jogo? Teórico ou prático?
- Qual material devemos nos basear para que possamos deixá-lo didático.

B. Conteúdo Estrutura do Jogo:

- Podemos utilizar elementos do cotidiano de um aluno Mauá? (Ex: introduzir personagens como o professor Aparecido e lugares como lojas do ambiente)
- Devemos implementar um jogo rápido ou algo mais elaborado?
- Quem é o público-alvo do jogo?

C. Programação das Estruturas do Jogo:

- Qual linguagem de programação você sugere que seja implementada?

1.2. Análise da Coleta de Requisitos

Tópico: Conteúdo didático

O curso de Banco de Dados só pede, com conteúdo prévio, que o aluno tenha conhecimento acerca da Teoria dos Conjuntos Numéricos, o que, mais à diante, será importante.

Além do conteúdo prévio, nosso curso é dividido em quatro setores, na seguinte sequência:

-
1. **Modelagem conceitual:** entender requisitos do sistema, necessidades do usuário, o que é armazenado no banco de dados, quais as consultas etc. Tudo dentro da modelagem.
 2. **Modelagem Física:** pensar em como as tabelas se estruturam, montar relacionamentos, conseguir distinguir quem assume o papel de Chave Primária, Chave Estrangeira, Super Chave, notações de DER, Atributos de Relacionamentos etc.
 3. **Programação de tabelas (DDL):** definir tipos de atributos, criar tabelas, criar, modificar e inserir chaves etc.
 4. **Consultas:** pôr-se no lugar do usuário e criar queries no próprio banco de dados.

Não foi mencionado sobre dar enfoque à alguma matéria em particular, mas o professor comentou que, nos tópicos [2], [3], [4], -apresentados acima -, deveríamos abordar de forma mais prática. Já, no item [1], mais teórico conceitual.

Poderemos utilizar o material didático de Aparecido no jogo, o que indica que a dificuldade conteudista será semelhante aos questionários e tarefas cobradas em sua disciplina.

Houve uma preocupação bem evidente quanto à sobrecarga que nosso projeto poderia causar, por ser muito trabalhoso. Por isso, Aparecido sugeriu que pudéssemos seguir, por exemplo, até a matéria de definição dos esquemas.

Tópico: Estrutura

Aparecido gostou da ideia de utilizarmos um elemento do dia a dia, como, por exemplo, a cantina da Mauá. Porém, assim como mencionado anteriormente, houve uma preocupação na complexidade do jogo, ou seja, na quantidade de trabalho necessário para colocar a ideia em prática.

Desta forma, um jogo que seja mais rápido e que tenha como público-alvo alunos da IMT que estudam Banco de Dados – que já tem uma noção do conteúdo-, seria importante e nos ajudaria no processo de simplificação da estrutura do jogo

Tópico: Programação das Estruturas do Jogo

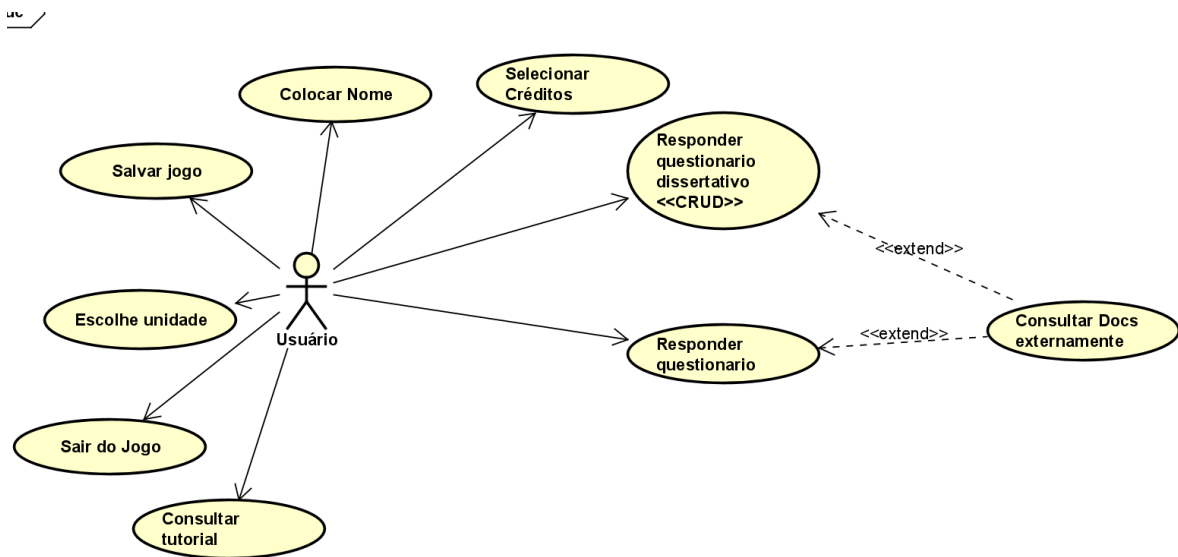
A linguagem SQL tem conectividade com várias outras linguagens de programação, como, por exemplo, C#, Java, Python etc. Todas são boas.

2. Especificação dos Requisitos do Sistema de Software

2.1. Requisitos Funcionais

Histórico da Revisão

Data	Versão	Descrição	Autor
30/03/2022	<beta>	<etapa inicial, especificação caso de uso>	< >



Breve Descrição

Este caso de uso tem como objetivo fazer o usuário escolher qual Unidade será jogada durante o jogo. Dentre elas são:

Fluxo Básico:

1. O ator Usuário, dentro da tela de criação de jogador, clica em jogar [FA1]
2. O ator Usuário automaticamente entrada tela de escolha de Unidade
3. O ator Usuário escolhe a unidade desejada

Fluxos Alternativos

[FA1] Fluxo Alternativo 1: Outra forma de entrar na tela de unidades.

1. O ator usuário clica no ícone de menu
2. O ator usuário acessa a tela de menu
3. O Ator usuário escolhe a opção de Unidade

Pré-condições

Não há pré-condições.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

Nome do Caso de Uso: Salvar Jogo

Breve Descrição: A qualquer momento o Ator usuário pode selecionar a opção de salvar o jogo, ou seja, ao acessar o Sistema novamente será possível retornar aonde parou. Ou, também, caso queira poder retornar à um carregamento previamente feito.

Fluxo Básico

1. O ator Usuário escolhe a opção de verificar a resposta – dissertativa ou múltipla-escolha
2. O Sistema salva a questão e, conseqüentemente, progresso do jogo

Fluxos Alternativos

Não há fluxos alternativos.

Pré-condições:

Não há pré-condições.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

Nome do Caso de Uso: Sair do Jogo

Breve Descrição: Este caso de uso tem como objetivo fazer o usuário escolher sair do jogo. Ao sair do jogo e reentrar, ele retornará ao último Load efetuado.

Fluxo Básico:

1. O ator Usuário escolhe uma opção na tela do jogo de Sair. [FA1]

Fluxos Alternativos**[FA1] Fluxo Alternativo 1: Sair do jogo**

1. Ator usuário seleciona uma opção presente de sair do jogo.
2. ator Usuário recusa a confirmação de saída do jogo.
3. O fluxo retorna ao passo do fluxo básico.

Pré-condições

Não há pré-condições.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

Nome do Caso de Uso: Consultar Material teórico

Breve Descrição: Dentro da fase, o Cliente usuário terá a possibilidade de consultar o material teórico referente à fase a qual o usuário está jogando no momento.

Fluxo de Eventos**Fluxo Básico**

1. O ator Usuário escolhe uma opção de acessar ao menu do jogo
2. O ator Usuário escolhe a opção de acessar a documentação
3. Uma pasta local é aberta automaticamente, contendo todos os PDFs de documentação das unidades
4. O ator Usuário consulta o material
5. O ator usuário seleciona novamente a janela do programa

Fluxos Alternativos

Não há fluxo alternativo

Pré-condições

Não há pré-condições.

Pós condições

Não há pós-condições

Regras de Negócio

Não há regras de negócio.

Nome do Caso de Uso: Consultar Tutorial

Breve Descrição: Dentro do Menu, o Cliente usuário terá a possibilidade de fazer ou refazer o tutorial do jogo, onde encontrará informações básicas de como o jogo se estrutura.

Fluxo de Eventos**Fluxo Básico**

1. O ator Usuário escolhe uma opção na tela do menu do jogo para acessar ao tutorial
2. O ator Usuário consulta o material didático
3. O ator usuário seleciona a opção de próximo na última tela do tutorial para voltar à tela anterior, que é o Menu do jogo [FA1]

Fluxos Alternativos**[FA1]Retornar à uma tela anterior dentro do tutorial**

- 1.O Ator usuário seleciona a funcionalidade de voltar para retornar à tela anterior do tutorial.

Fluxos Alternativos

Não há fluxo alternativo

Pré-condições

Não há pré-condições.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

Nome do Caso de Uso: Responder questionário**Breve Descrição**

O Ator usuário responderá questões de múltipla-escolha. O objetivo é testar os conhecimentos do usuário e qualificá-lo, ou não, para a próxima fase do jogo. Sendo uma fase correspondente a uma unidade.

Fluxo Básico

1. O ator usuário lê as questões expostas na tela.
2. O ator usuário escolhe a alternativa desejada.
3. O usuário confirma a alternativa selecionada.

4. O sistema verifica a quantidade de alternativas certas [FA1], [FA2]

Fluxos Alternativos

[FA1]: Fluxos Alternativo 1: Aprovado de fase.

- 1.O sistema calcula 70% ou mais de acertos dentre todas as questões da fase.
- 2.Uma mensagem confirmando o sucesso do questionário da fase é enviado ao usuário.

Pré-condições

1.O usuário deve ter finalizado a parte de material teórico referente à fase, de modo a não ser possível responder o questionário sem antes ser apresentado ao material.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

[FA2]: Fluxo Alternativo 2 Reprovado de fase

1. O sistema calcula menos de 70% de acertos dentre todas as questões da fase.
2. Uma mensagem confirmando o fracasso do aluno do questionário da fase é enviado ao usuário.
3. O usuário retorna à etapa 1 do fluxo básico.

Pré-condições

O usuário deve ter finalizado a parte de material teórico referente à fase, de modo a não ser possível responder o questionário sem antes ser apresentado ao material.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

Nome do Caso de Uso: CODIFICAR

Breve descrição: O Ator usuário preencherá lacunas e fará opções múltipla-escolha relativas ao conteúdo de programação em si. O objetivo é testar os conhecimentos do usuário e qualificá-lo, ou não, para a próxima etapa do jogo.

Fluxo Básico

1. O ator Usuário digita o código corresponde ao que o enunciado
2. O ator Usuário confirma o código
3. O Sistema verifica o que o usuário confirmou [FA1], [FA2]

Fluxos Alternativos

[FA1] Fluxo Alternativo 1: Fracasso na codificação

1. O sistema calcula menos de 70% de acertos dentre todas as questões da fase.
2. Uma mensagem confirmando o fracasso do aluno do questionário da fase é enviado ao usuário.
- 3.O sistema retorna o Autor usuário à etapa 1 do Fluxo Básico

Pré-condições

Não há pré-condições.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

[FA2] Fluxo Alternativo 2: Sucesso na codificação

1. O sistema calcula mais de 70% de acertos dentre todas as questões da fase.
2. Uma mensagem confirmando o sucesso do aluno do questionário da fase é enviado ao usuário.
3. O Ator usuário continua para a fase seguinte.

Pré-condições

Não há pré-condições.

Pós-condições

Não há pós-condições.

Regras de Negócio

Não há regras de negócio.

2.2. Requisitos Não-Funcionais

O Sistema deverá se inicializar rapidamente;

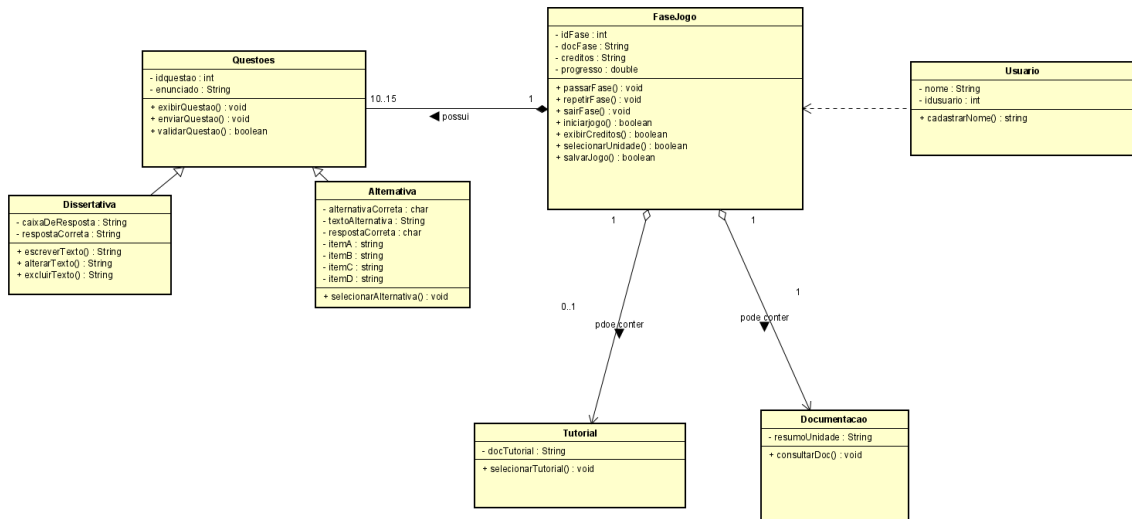
Usabilidade amigável e de fácil uso;

O Sistema deve ter bom desempenho: durante a inicialização e gameplay;

O Sistema deve ser didático para o usuário;

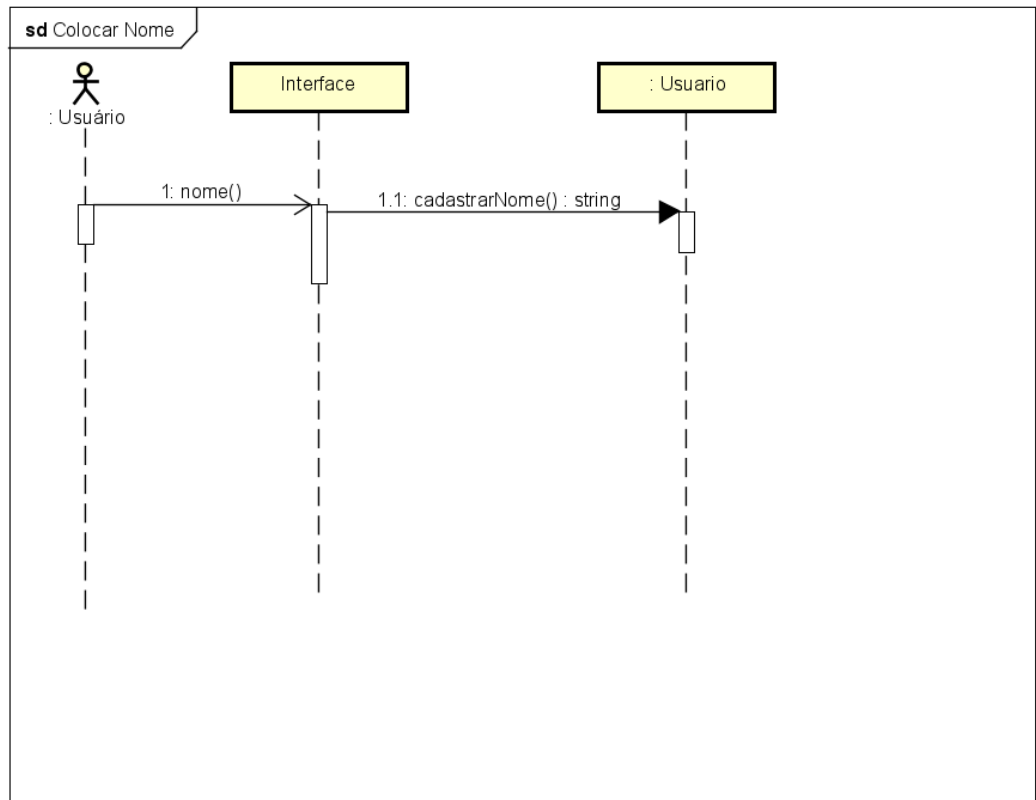
3. Análise/Projeto

3.1. Diagrama de Classes

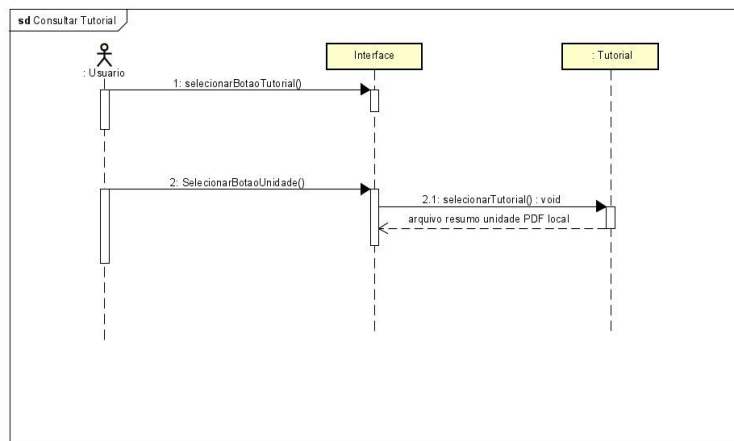


3.2. Diagrama de Sequência

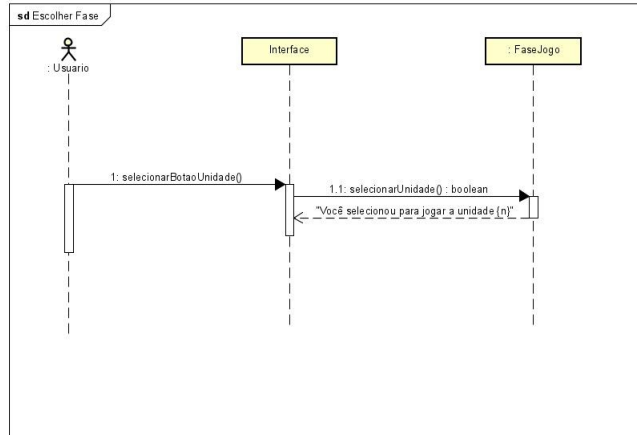
Cadastrar Nome



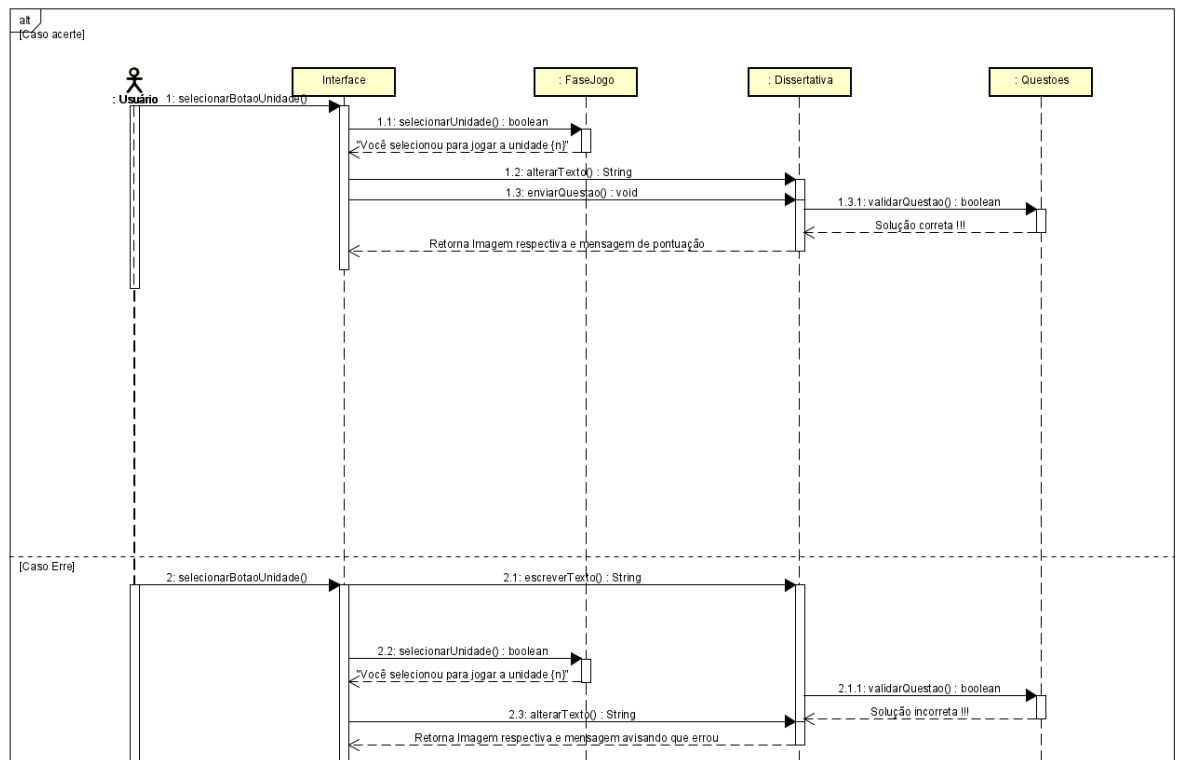
Consultar Tutorial



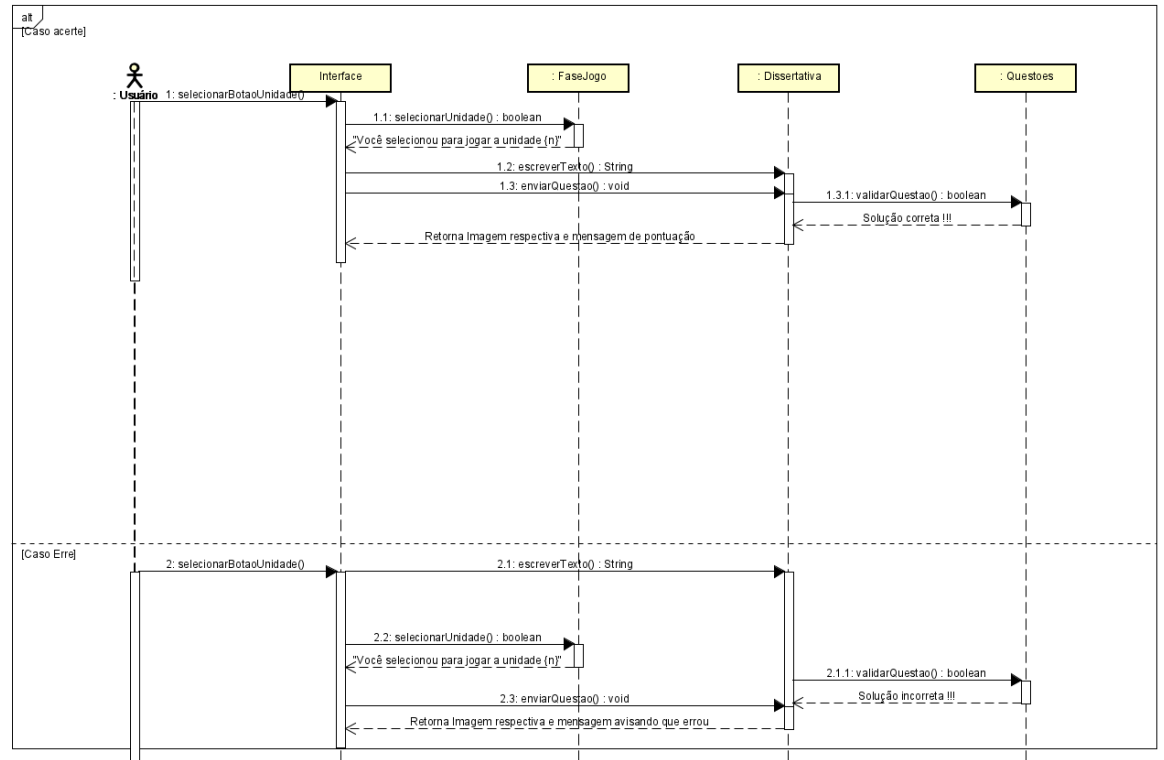
Escolher Fase



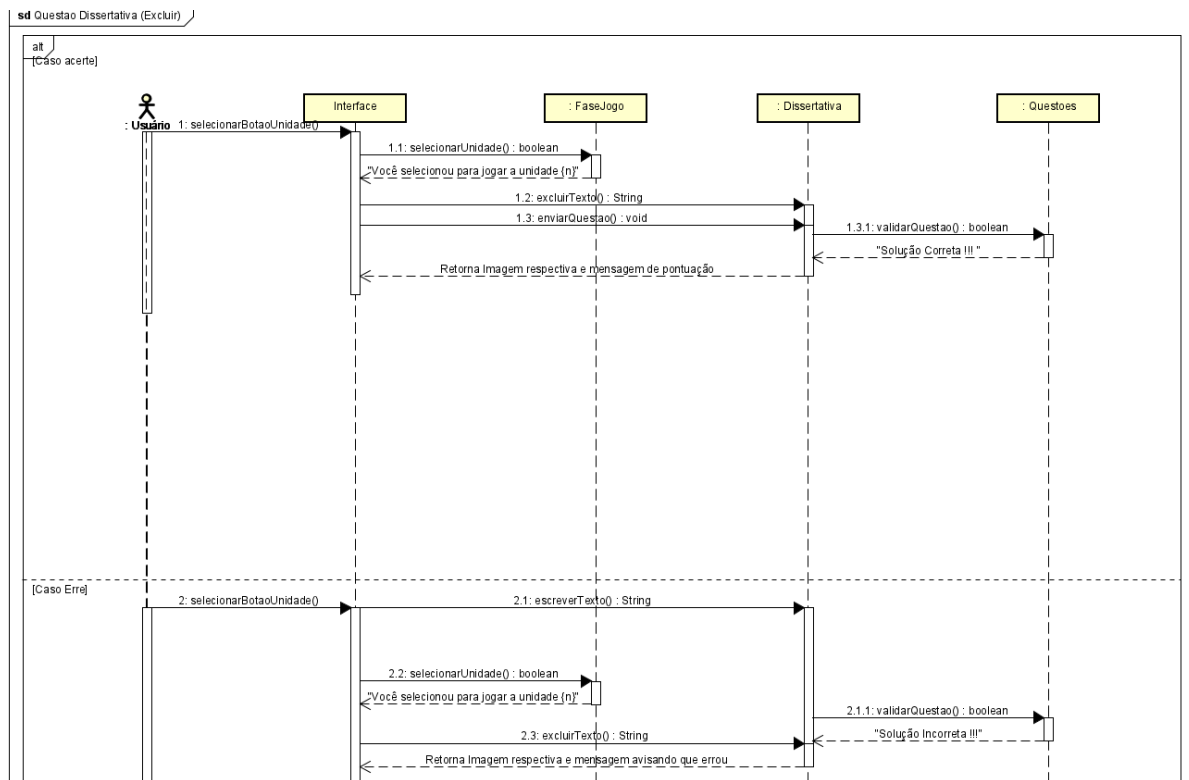
Questão Dissertativa Alterar



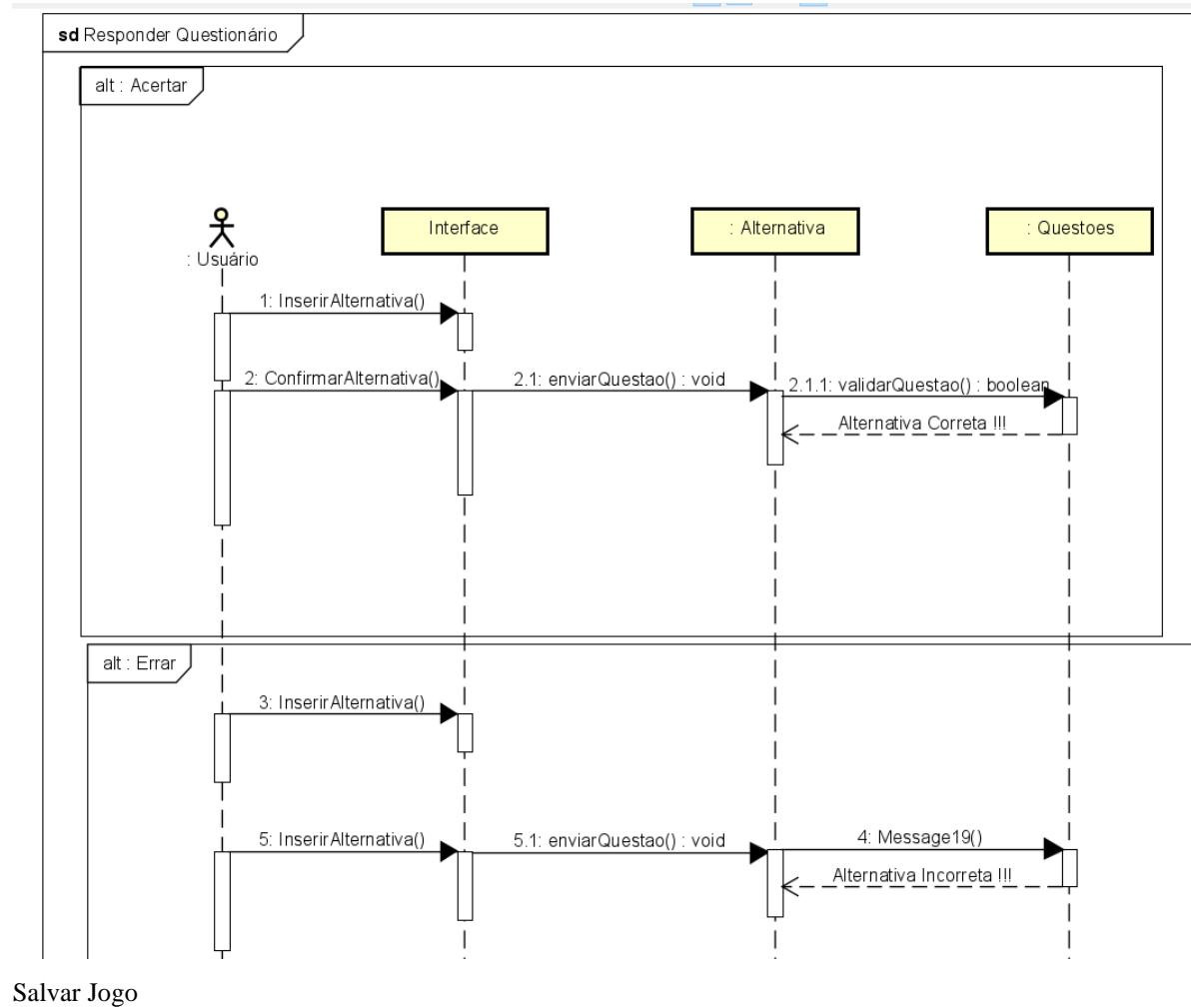
Questão Dissertativa Escrever

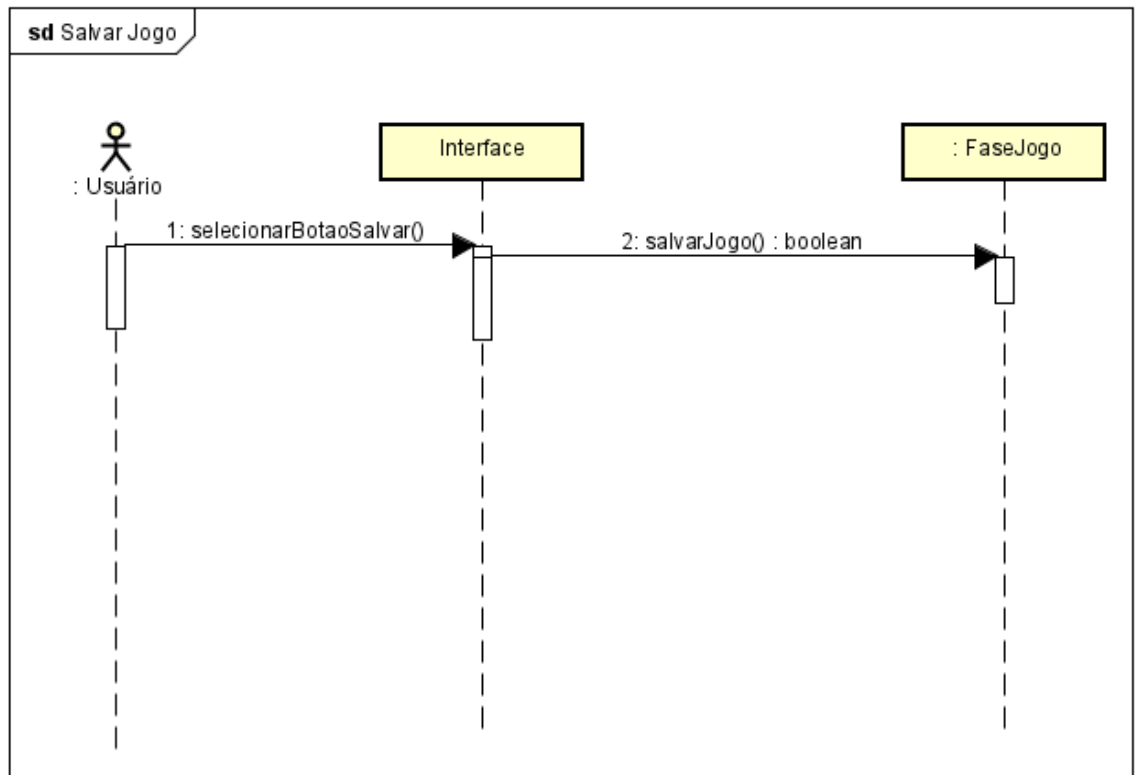


Questão Dissertativa Excluir

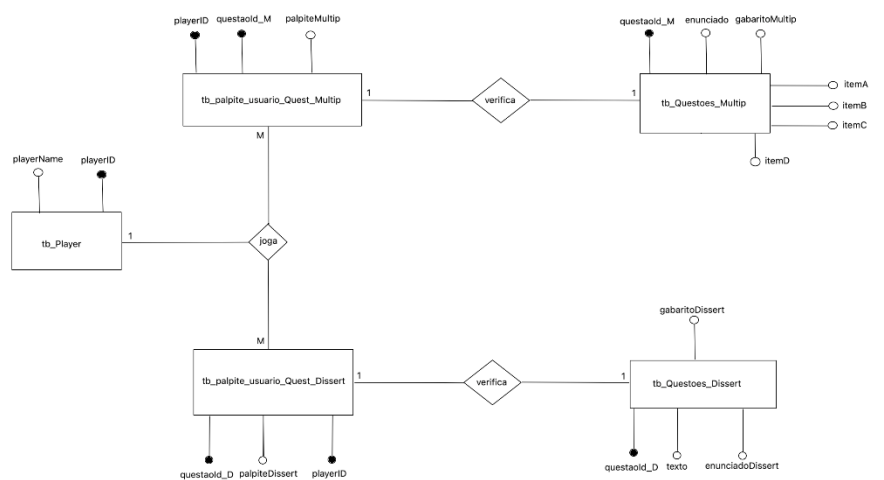


Responder Questionário Alternativa



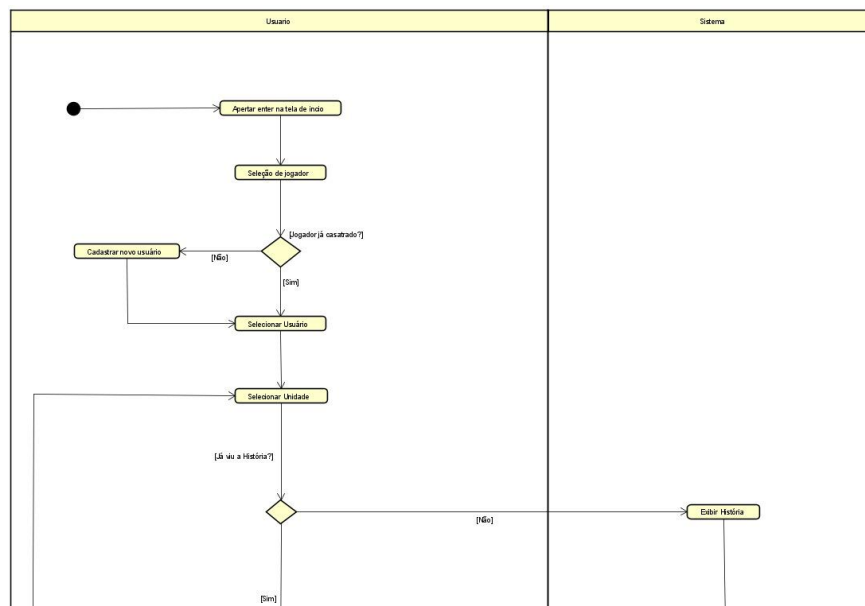


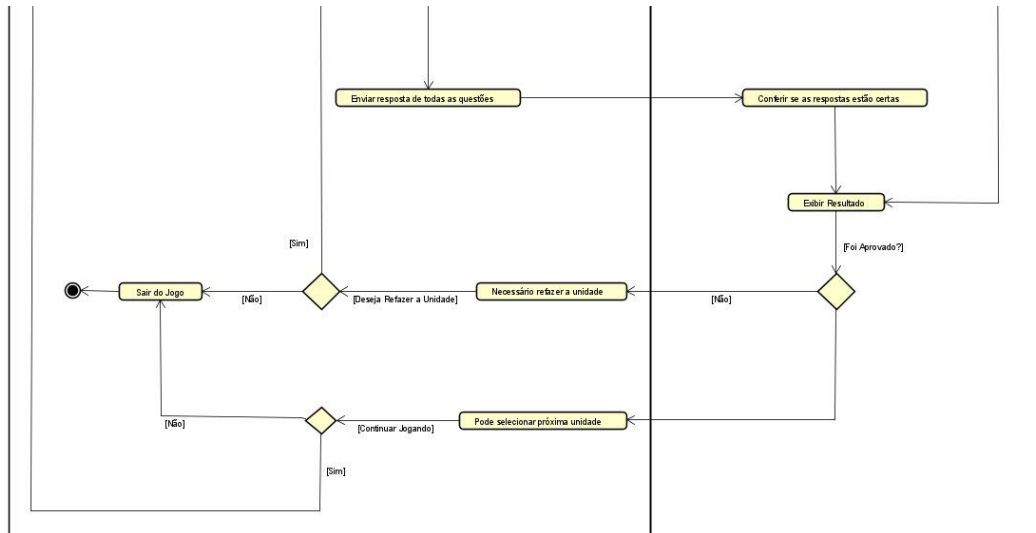
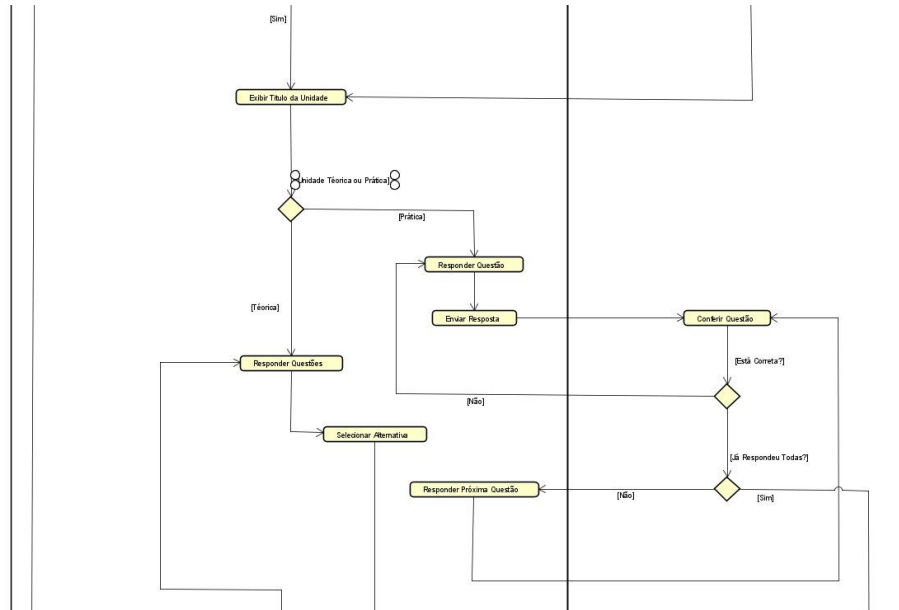
3.3. Modelo de Banco de Dados



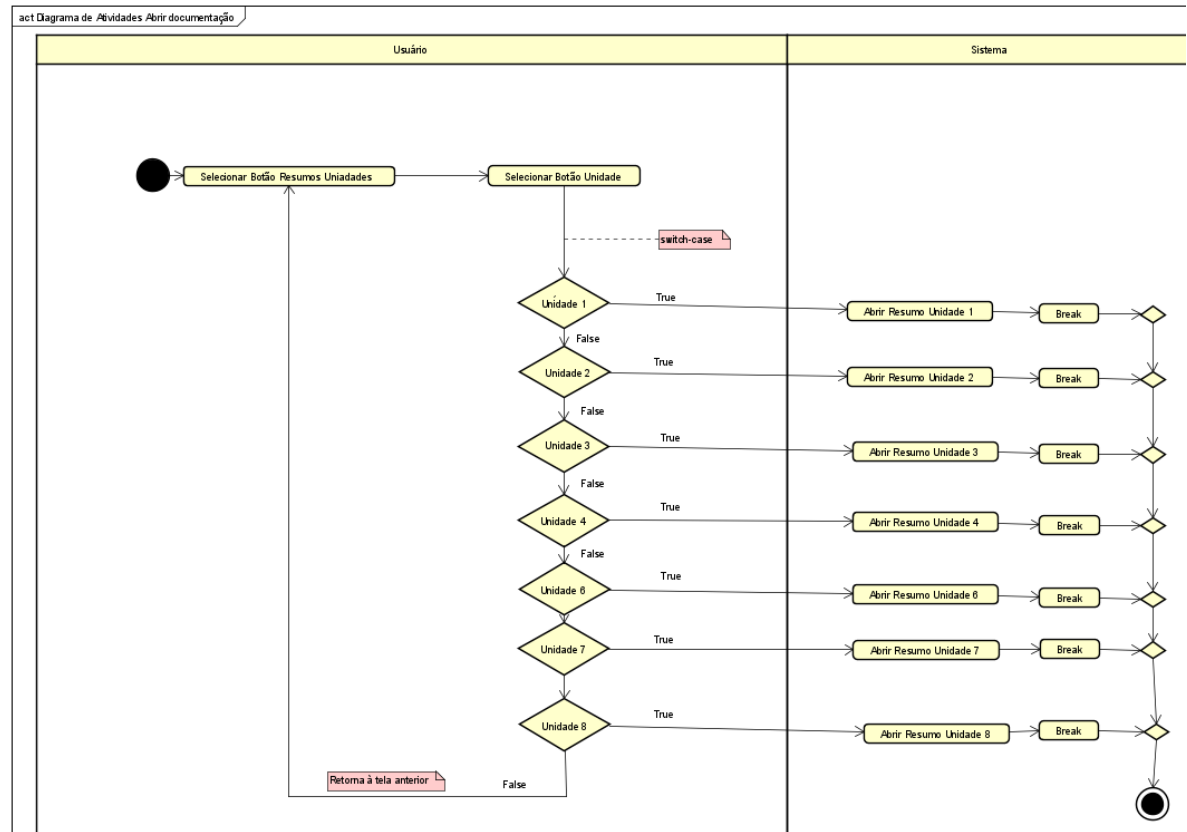
3.4. Diagrama de Atividades (opcional)

Jogando o jogo

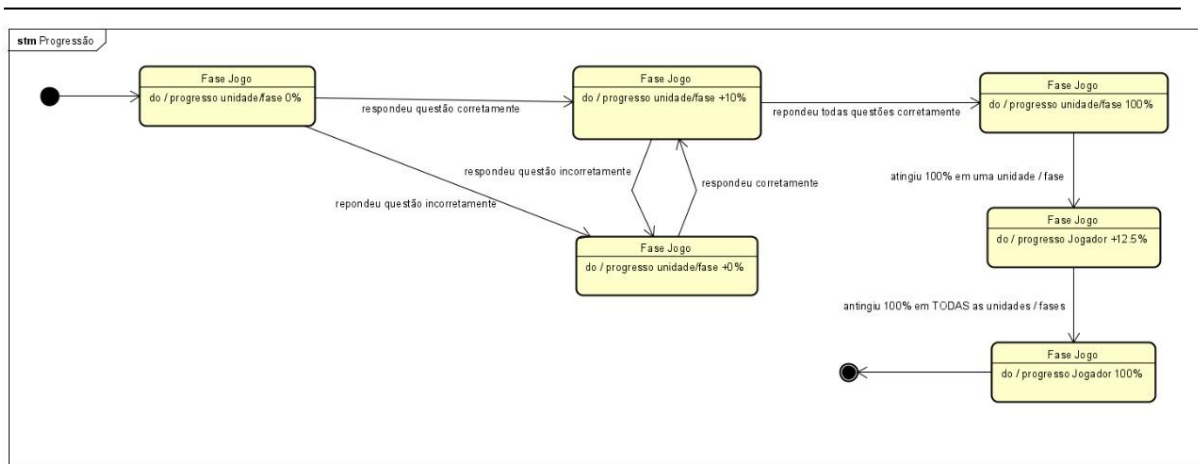




Selecionar Documentação



3.5. Diagrama de estados (opcional)



4. Implementação

https://github.com/Briqz23/Projeto_PI.git

5. Testes

Funcional: Testamos, perto do final do projeto, a conexão entre as telas, se as questões e enunciados estavam sendo importados corretamente e se o programa estava demorando para executar.

Integração: Testamos a integração com Windows 11, 10, Linux (Ubuntu) e macOS.

6. Resultados e Considerações

Gostamos muito de desenvolver o jogo. Pudemos aprender sobre todas as disciplinas envolvidas, principalmente a de banco de dados – exceto Python. Pudemos aprender bastante com os professores, com colegas de turma e pesquisando em lugares diversos na internet.

Tivemos mais problemas do que esperávamos na importação de informações do banco de dados, assim como na modelagem de telas na IDE do NetBeans.

Conseguimos nos organizar dividindo nossas responsabilidades e funções, assim, trabalhando de forma mais eficiente.

Apêndice I

Telas:

<https://www.figma.com/file/uNKltcfoAIMAODY9o2FctX/BD-Academy>

Fonte Mont Serrat:

<https://fonts.google.com/specimen/Montserrat>

IDE:

<https://netbeans.apache.org/download/index.html>

JDK:

<https://www.oracle.com/java/technologies/downloads/>

MySQL Community:

<https://dev.mysql.com/downloads/workbench/>

Aseprite (pixelart)

<https://www.aseprite.org/>