

**Documentação de um**

**Produto de Software**

**SN Game**

**Nome dos Alunos:**

|  |  |
| --- | --- |
| **RA** | **Nome** |
| 24.00204-6 | ARTHUR TRINDADE DE SOUZA |
| 24.01732-9 | BRENO HENRIQUE DE CASTRO ROCHA |
| 23.00941-0 | GUSTAVO NORONHA BOMFIM |
| 24.00884-2 | MURILO HIDEKI SATO |
| 24.01229-7 | RAFAEL YUNO CHOI |
| 24.00920-2 | VITOR BRITTO PASSOS DE FIGUEIREDO |

**2024**

**ÍNDICE DETALHADO**

[1. Descrição/Resumo do Projeto 3](#_Toc162427357)

[2. Levantamento dos Requisitos do Sistema de Software 3](#_Toc162427358)

[2.1. Extração de Requisitos 3](#_Toc162427359)

[2.2. Análise da Coleta de Requisitos 3](#_Toc162427360)

[3. Especificação dos Requisitos do Sistema de Software 3](#_Toc162427361)

[3.1. Requisitos Funcionais 3](#_Toc162427362)

[3.2. Requisitos Não-Funcionais 4](#_Toc162427363)

[4. Análise/Projeto 5](#_Toc162427364)

[4.1. Diagrama de Classes 5](#_Toc162427365)

[4.2. Diagrama de Sequência 5](#_Toc162427366)

[4.3. Modelo de Banco de Dados 5](#_Toc162427367)

[4.4. Diagrama de Atividades (opcional) 5](#_Toc162427368)

[4.5. Diagrama de estados (opcional) 5](#_Toc162427369)

[5. Implementação 5](#_Toc162427370)

[6. Testes 6](#_Toc162427371)

[7. Resultados e Considerações 6](#_Toc162427372)

[8. Registro da Apresentação ao Parceiro 6](#_Toc162427373)

[Referências 7](#_Toc162427374)

[Apêndice I 8](#_Toc162427375)

# Descrição/Resumo do Projeto

|  |
| --- |
| Jogo de exploração, com quiz Integrado. O usuário deverá explorar o mapa do jogo e responder as perguntas relacionadas a teoria da evolução de Charles Darwin. |

# Levantamento dos Requisitos do Sistema de Software

Este capítulo tem como objetivo apresentar o levantamento dos requisitos do Sistema de Software e a forma de extração dos Requisitos.

## Extração de Requisitos

Os métodos utilizados para extrair requisitos do cliente foram a Entrevista, que foi feita com o professor onde tivemos a oportunidade de entender o que o cliente espera do produto e o que ele tem em mente em relação ao game, o brainstorm que foi utilizado entre os integrantes do grupo para definirmos como realizaremos o game e também utilizamos o questionário, para extrairmos o que o público gostaria que o jogo funcionasse. As perguntas do questionaram foram as seguintes:

## Análise da Coleta de Requisitos

O que o cliente quer do jogo:

* O cliente quer um jogo educativo que explore o tema da Seleção Natural, se utilizando de perguntas em formato de quiz para testar e desenvolver o nível de conhecimento dos jogadores em relação a este tema. Porém, trazendo inovações divertidas e cativantes para que o jogo prenda a atenção dos jogadores.

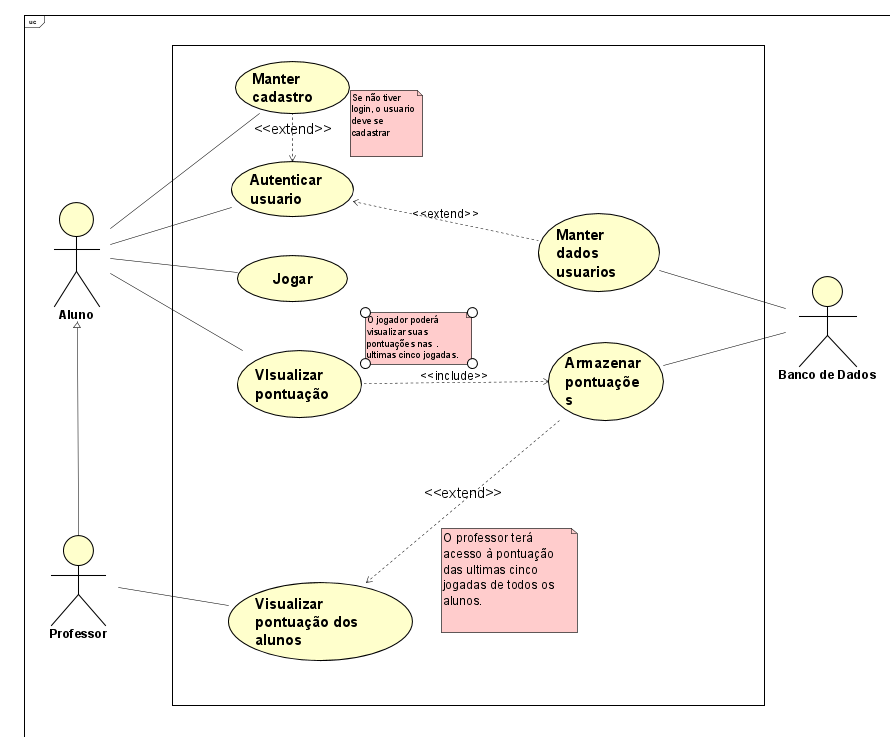
Qual é o público alvo do game:

* O público alvo são alunos do Fundamental 2 e Ensino Médio, da Escola Piaget. As perguntas têm que ser mais focadas em temas específicos ou mais gerais: As perguntas podem ser diversificadas em dificuldade. Não sendo necessário a utilização de perguntas de vestibulares, porém as perguntas têm que ser coesas ao tema. Elas podem explorar todos os temas da seleção natural, desde o processo de criação da teoria até as mutações e evoluções das espécies de animais.

# Especificação dos Requisitos do Sistema de Software

Este capítulo tem como objetivo especificar os requisitos funcionais e não funcionais.

Os requisitos especificados devem ter como base a análise da coleta de requisitos, de acordo, com o resultado da técnica de extração de requisitos.



## Especificações de Casos de usos

**- Manter cadastro:**

**Breve descrição:**

O sistema pedirá ao usuário que insira seu e-mail e senha, para cadastrá-lo no banco de dados. Sendo também possível para o usuário alterar sua senha quando desejar.

**Fluxo básico:**

Ao iniciar o jogo, o sistema irá pedir para que o usuário insira seu e-mail e senha, após ter completado os campos pedidos, o sistema irá guardar os dados do usuário no banco de dados.

**Fluxo Alternativo:**

Caso o usuário insira um e-mail que não seja aquele de sua instituição, o sistema notificará o erro ao usuário.

**Pré-condição:**

- O usuário deverá possuir o e-mail institucional.

- O banco de dados deve possuir um meio de distinguir o e-mail institucional dos demais.

**Pós-condição:**

* Os dados de acesso do usuário serão salvos, para que ele acesse sua conta novamente.

**- Autenticar usuário:**

**Breve Descrição:**

O sistema pedirá para que o usuário insira seus dados de acesso, em seguida determinará se o usuário é professor ou aluno.

**Fluxo Básico:**

Ao acessar o jogo, o sistema pedirá para o usuário informar seus dados de acesso, em seguida o sistema liberará as outras funções aos usuários.

**Fluxo Alternativo:**

Se o sistema identificar que os dados de acesso informados pelo usuário não estão cadastrados no banco de dados, ele pedirá para que o usuário se cadastre.

**Pré-condição:**

* O usuário deverá ter se cadastrado no banco de dados

**Pós-condição:**

* O sistema deverá permitir que o usuário acesse as outras funções do sistema.

**- Jogar:**

**Breve descrição:**

Função que permite ao usuário a acessar aos recursos do jogo.

**Fluxo Básico:**

Após o jogador apertar no botão de play, o sistema abrirá a tela do jogo onde o usuário poderá explorar o mapa e responder as perguntas e exibir uma mensagem de acerto, quando o usuário acertar as respostas certas, após o jogador responder todas as perguntas do quiz, o sistema informará a pontuação do jogador naquela jogada.

**Fluxo Alternativo:**

Caso o jogador erre as respostas das perguntas, o jogo deve exibir uma tela de falha ao acertar a resposta, além de exibir o motivo da resposta estar errada.

**Pré-condição:**

- O usuário deve ter se cadastrado no sistema.

**Pós-condição:**

- Após o jogador responder corretamente a questão o sistema deverá exibir uma tela de acerto.

- Se o jogador responder a questão de maneira incorreta, o sistema deverá exibir uma tela de erro.

**- Visualizar pontuação:**

**Breve descrição:**

Função que permite ao usuário a visualizar suas pontuações feitas.

**Fluxo básico:**

Após o jogador apertar no botão de visualizar pontuação, o sistema deverá mostrar as últimas cinco pontuações do usuário.

**Fluxo Alternativo:**

Se o usuário não tiver jogado nenhuma vez e não possui uma pontuação, o sistema deverá informar que o usuário não possui nenhuma pontuação salva.

**Pré-condição:**

**-** O usuário deve ter completado o jogo ao menos uma vez, para ter seus pontos armazenados.

**Pós-condição:**

- O sistema deve exibir as pontuações do usuário.

**- Visualizar pontuação dos alunos:**

**Breve descrição:**

Função que permite ao professor visualizar as pontuações de todos os alunos.

**Fluxo básico:**

Após o professor apertar no botão de visualizar pontuação dos alunos, o sistema deve exibir as cinco últimas pontuações de todos os alunos.

**Fluxo Alternativo: \*\*\***

Se o professor quiser ver apenas a sua própria pontuação, o sistema deverá exibir apenas a pontuação do professor.

**Pré-condição:**

**-** Os alunos precisaram ter terminado o jogo para visualizar suas pontuações.

**Pós-condição:**

**-** O sistema deverá exibir as últimas cinco pontuações de todos os alunos.

**- Manter dados usuários:**

**Breve descrição:**

O sistema deverá enviar os dados de acesso dos usuários, como e-mail e senha, para o banco de dados.

**Fluxo básico:**

Após o usuario informar seus dados de acesso o sistema deverá armazená-los no banco de dados.

**Fluxo Alternativo:**

Se o usuário informar um e-mail que não seja o institucional, o sistema deverá exibir uma mensagem de erro, pedindo para que o usuário coloque o e-mail institucional.

**Pré-condição:**

- Os usuários não podem ter seus dados salvos.

**Pós-condição:**

- O usuário poderá acessar novamente, com seus dados salvos anteriormente.

**- Armazenar Pontuação:**

**Breve descrição:**

O sistema deverá enviar as pontuações feitas pelos jogadores em cada jogada, para o banco de dados.

**Fluxo básico:**

O banco de dados armazenará a pontuação das últimas cinco jogadas do usuario e poderá exibi-las quando o jogador quiser, através da função visualizar pontuação.

**Fluxo alternativo:**

Se o jogador já tiver cinco pontuações salvas e o jogador jogar mais uma vez, o sistema deve substituir a pontuação mais antiga salva pela nova.

**Pré-condição:**

**-** O usuario deve ter pelo menos finalizado o jogo uma vez.

**Pós-condição:**

- O sistema deve poder acessar o banco de dados, para exibir as pontuações do usuario.

## Requisitos Funcionais

* RF1: O sistema deve solicitar e-mail e senha ao usuário para se autentificar no jogo.
* RF2: O sistema deve aceitar somente o e-mail da instituição ao usuário.
* RF3: O sistema deve possuir um sistema de perguntas e repostas similares a um quiz.
* RF4: O sistema deve apresentar perguntas variadas em relação ao tema da seleção natural.
* RF5: O sistema deve possuir quatro escolhas possíveis para cada pergunta, sendo apenas uma delas certa.
* RF6: O sistema deve possuir um sistema de pontuação, como forma de avaliar o aluno.
* RF7: O sistema deve somar a pontuação dos jogadores com base na quantidade de questões respondidas corretamente.
* RF8: O sistema deve exibir uma tela ao usuário informando que o usuário acertou a questão, quando responder corretamente.
* RF9: O sistema deve exibir uma tela ao usuário informando que o usuário errou a questão, quando responder incorretamente.
* RF10: O sistema deve enviar os dados de e-mail e senha dos usuários ao banco de dados.
* RF11: O sistema deve enviar a pontuação feita pelos usuários ao banco de dados.

## Requisitos Não-Funcionais

* RNF1: O sistema deve ter tempos de carregamento rápidos, não podendo passar de 1 segundo.
* RNF2: O sistema deve respeitar a lei LGPD de proteção aos dados dos usuários.
* RNF3: A interface do usuário deve ser compreensível, com explicações detalhadas fornecidas para cada função do sistema.

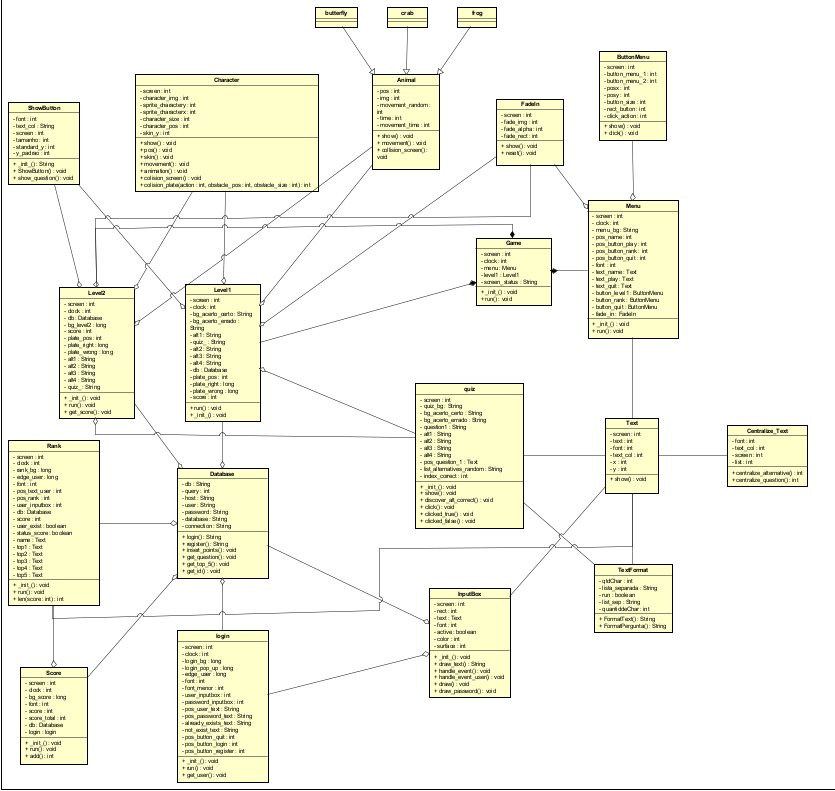
# Análise/Projeto

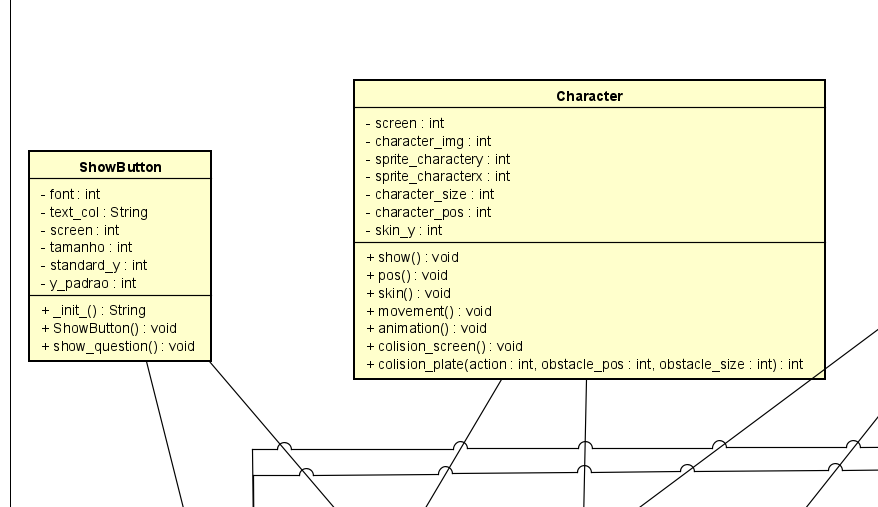
## Diagrama de Classes

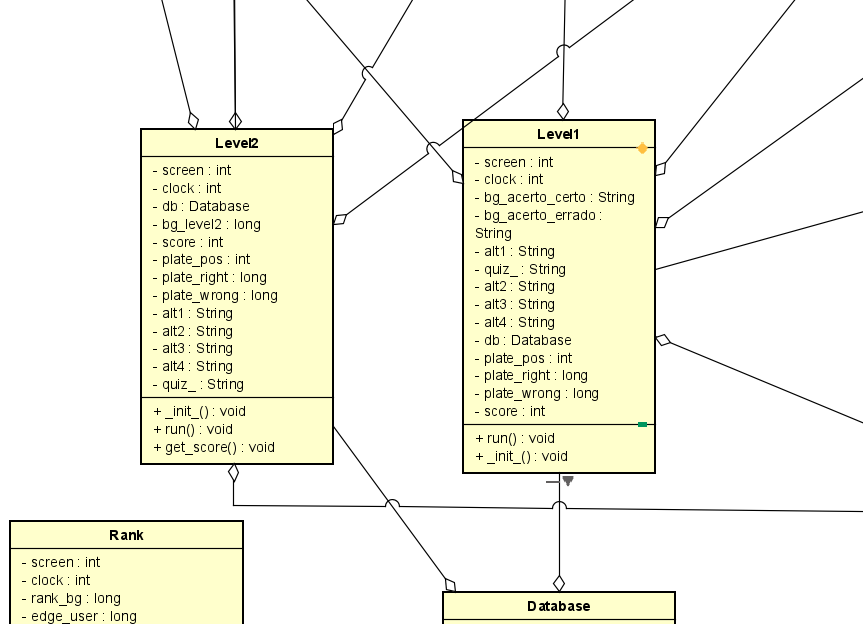
Neste item deve ser apresentado o modelo do domínio, visão de negócio, que representa um primeiro modelo conceitual do diagrama de classes.

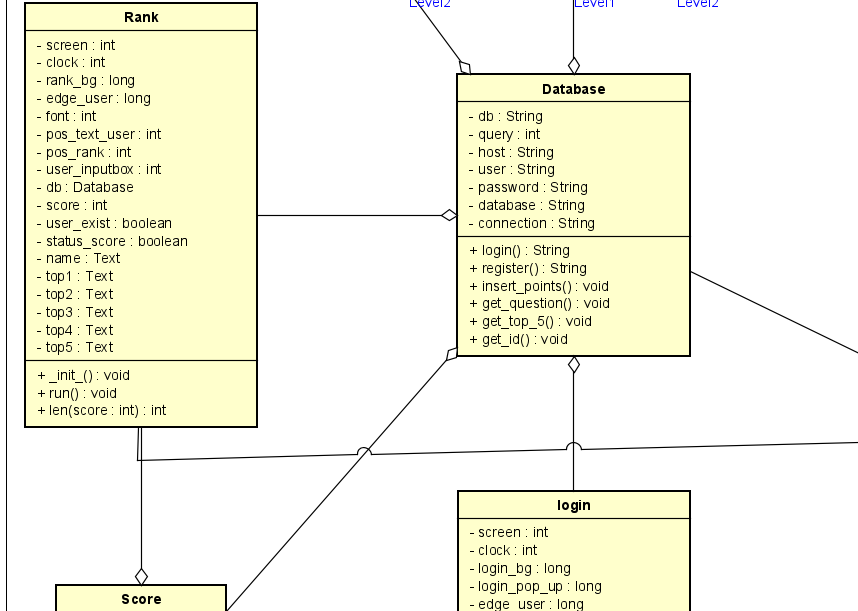
O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamentos entre elas.

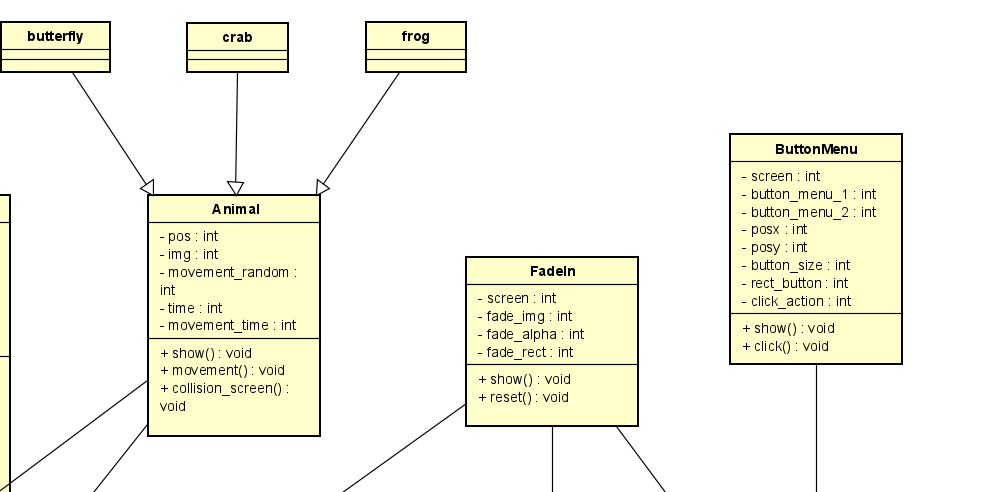
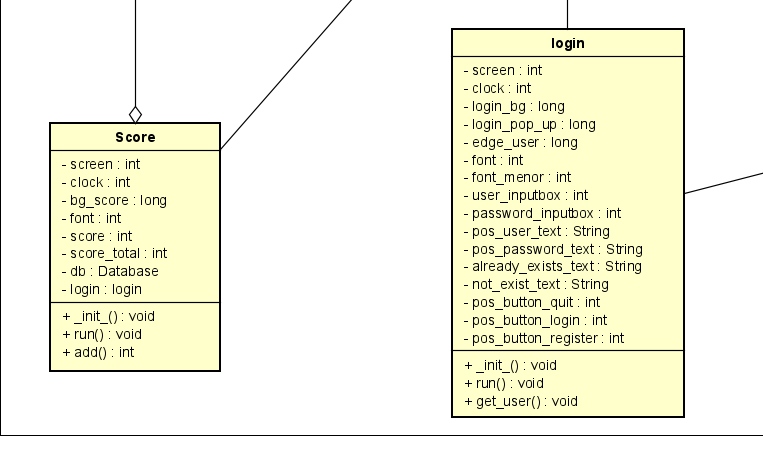
*Disciplina de Apoio: Modelagem Orientada a Objetos.*

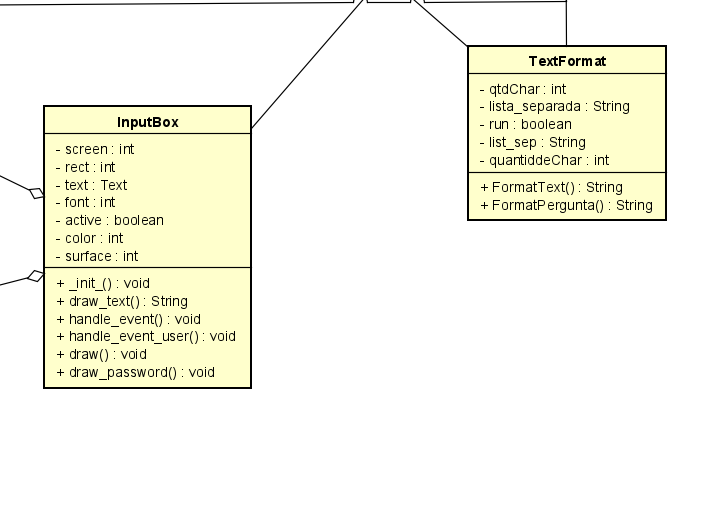
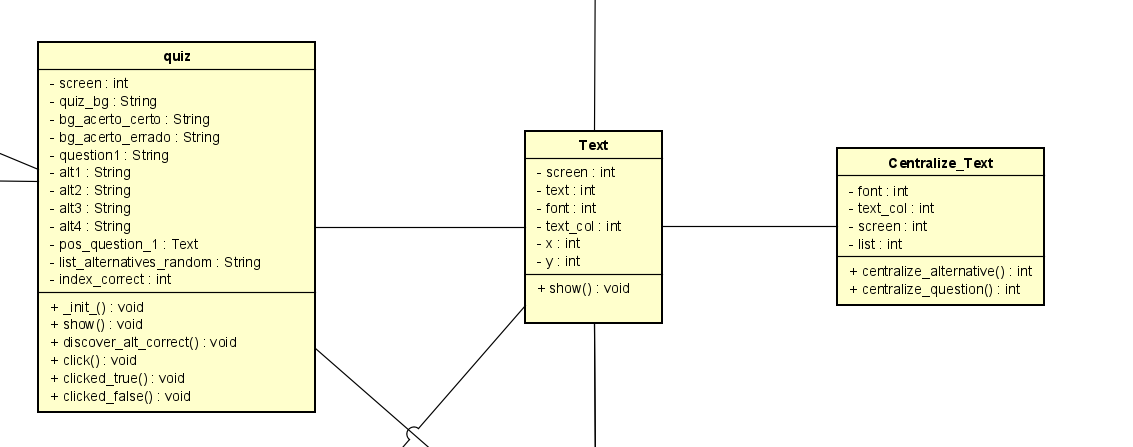
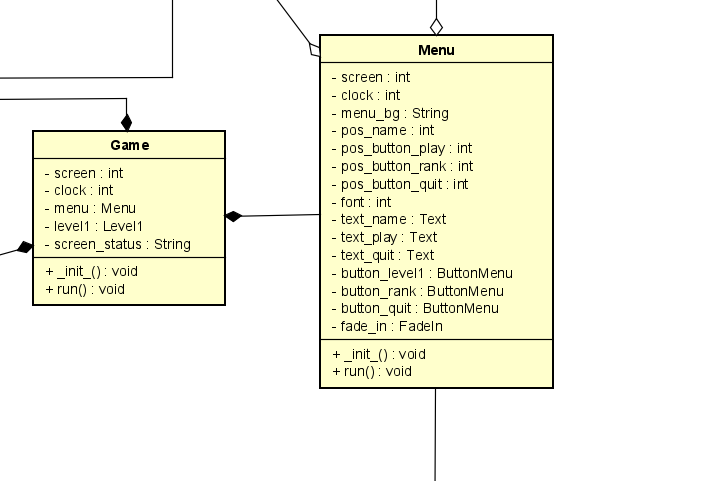










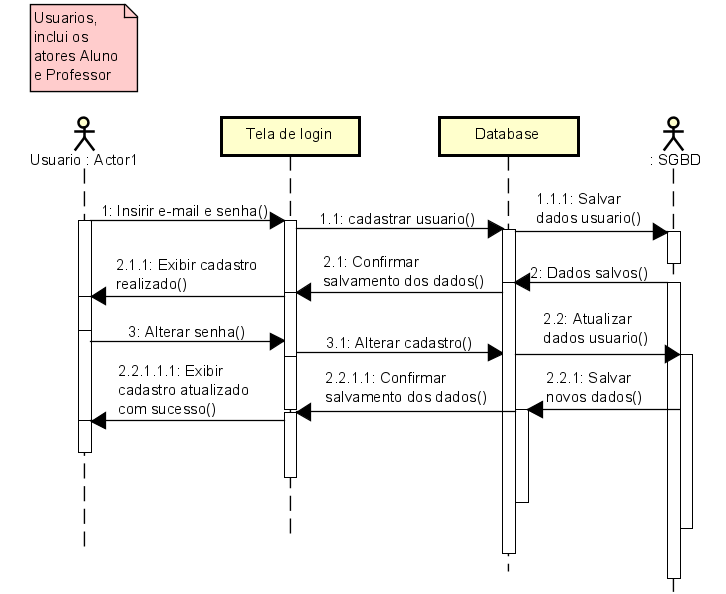


## Diagrama de Sequência

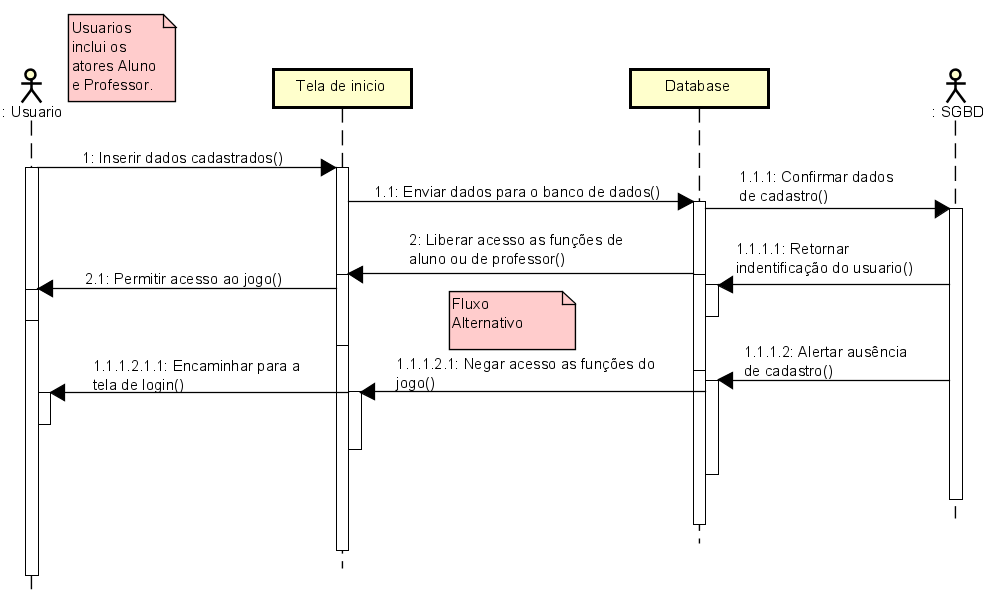
Neste item deve ser apresentado os diagramas de sequência com maior valor de negócio ao sistema. A escolha é realizada por caso de uso. Não há necessidade de realizar o diagrama de sequência para <<crud>>.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

**- Manter cadastro:**

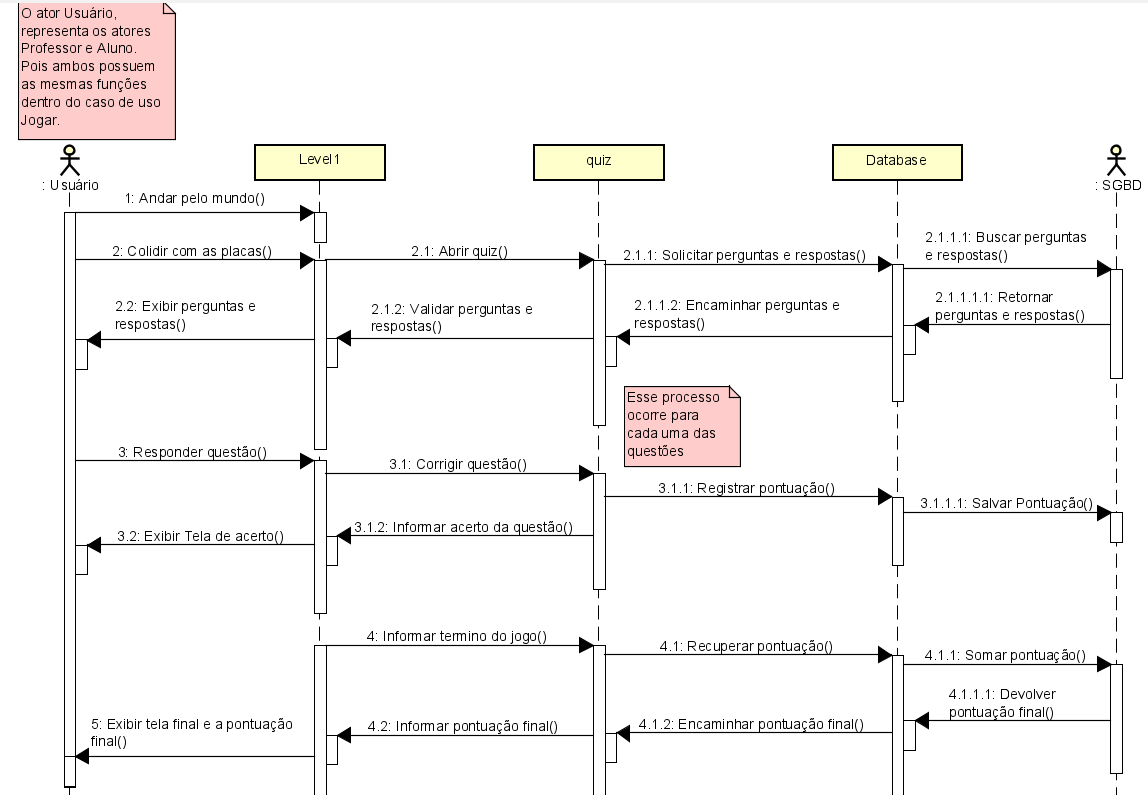


**- Autenticar usuario:**

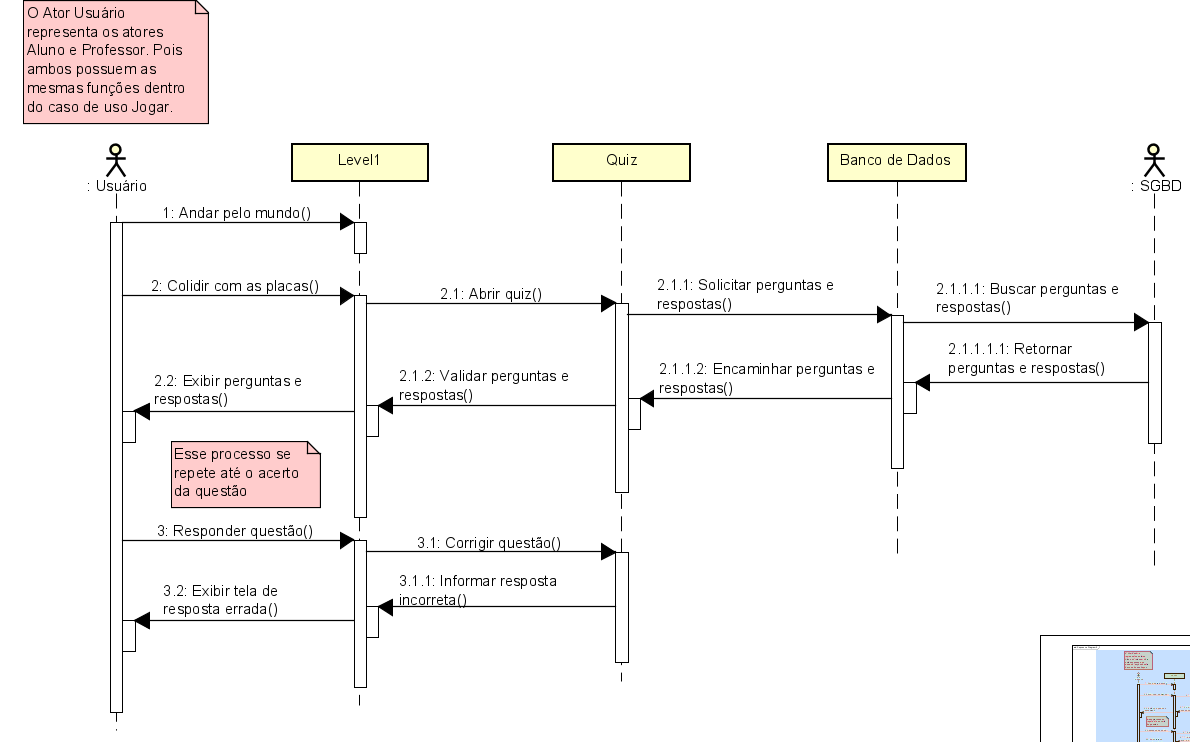


**- Jogar:**

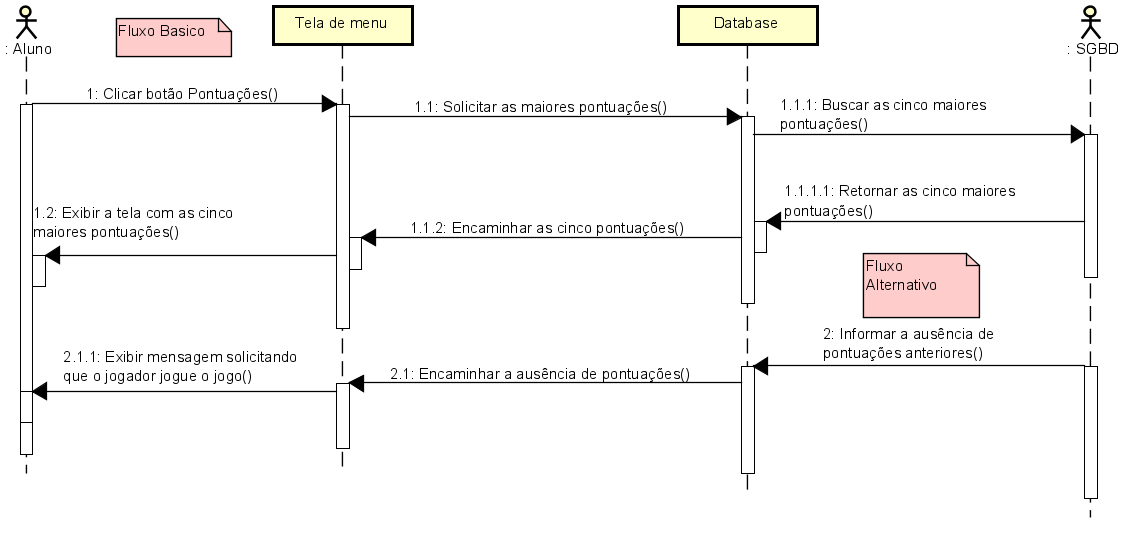
**1 – Fluxo Básico:**

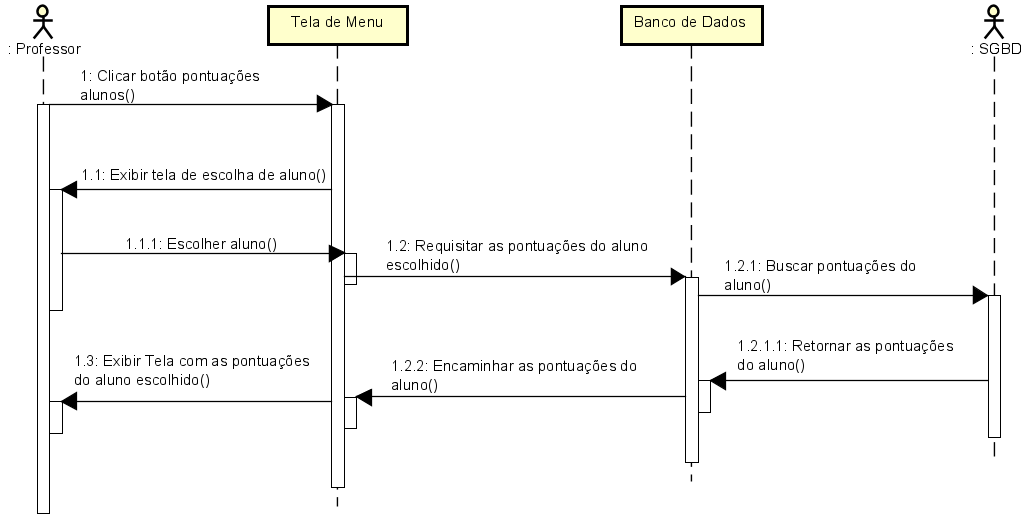


**2 – Fluxo Alternativo:**



**- Visualizar Pontuação:**

**- Visualizar Pontuação dos Alunos:**

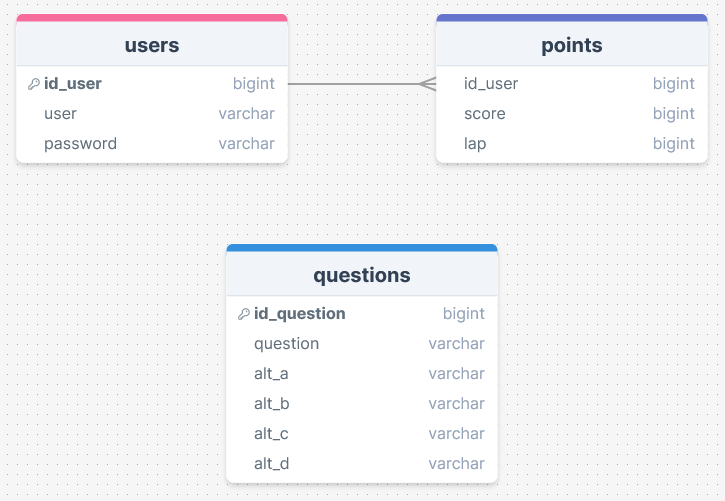


## Modelo de Banco de Dados

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.*

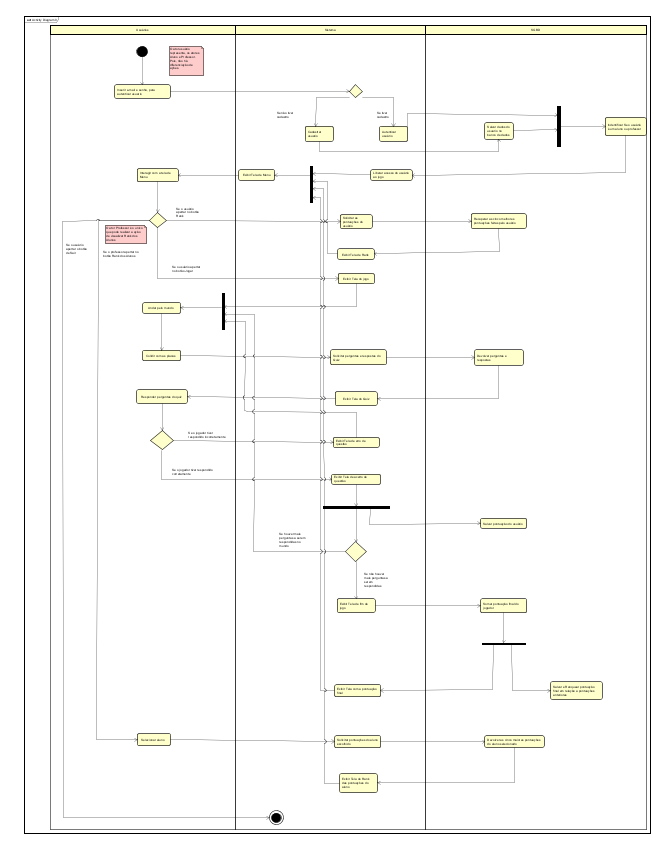


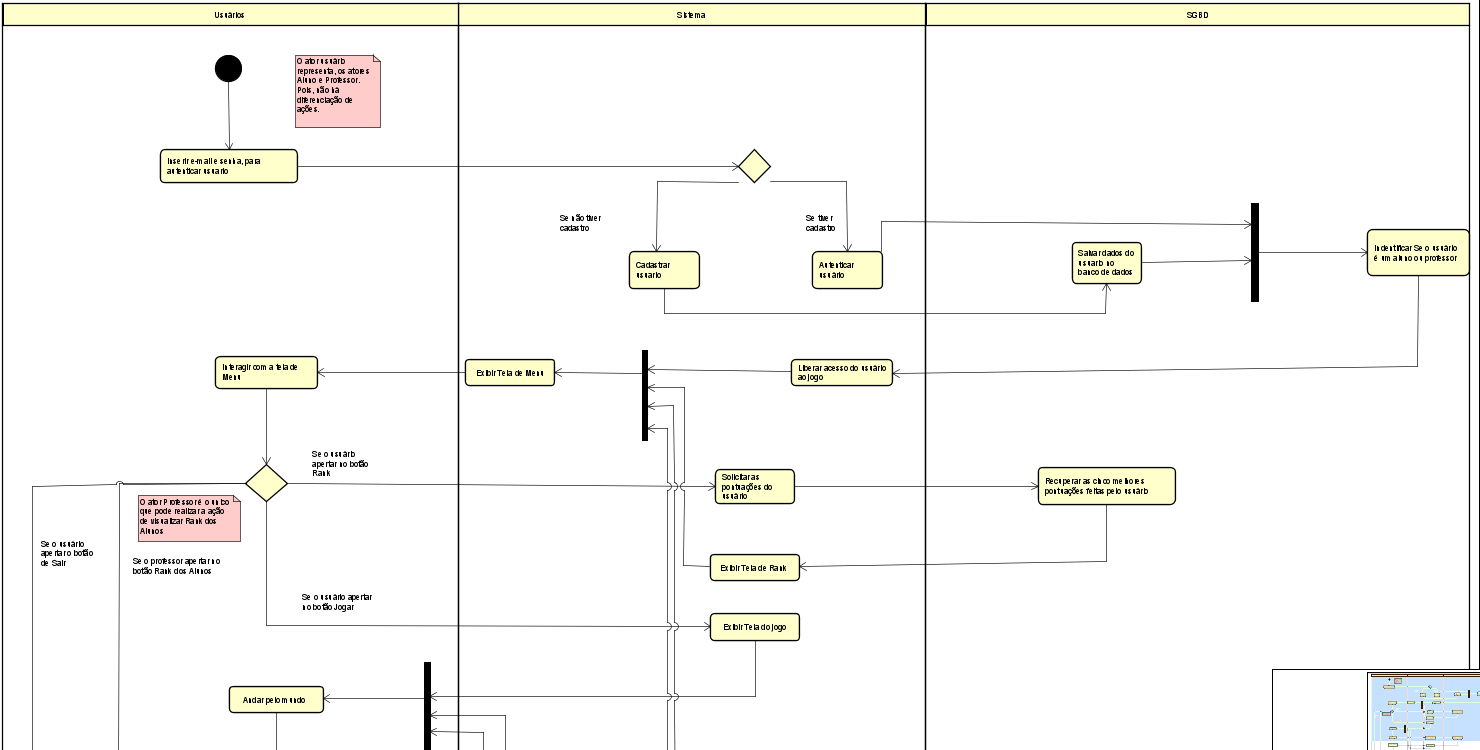
## Diagrama de Atividades (opcional)

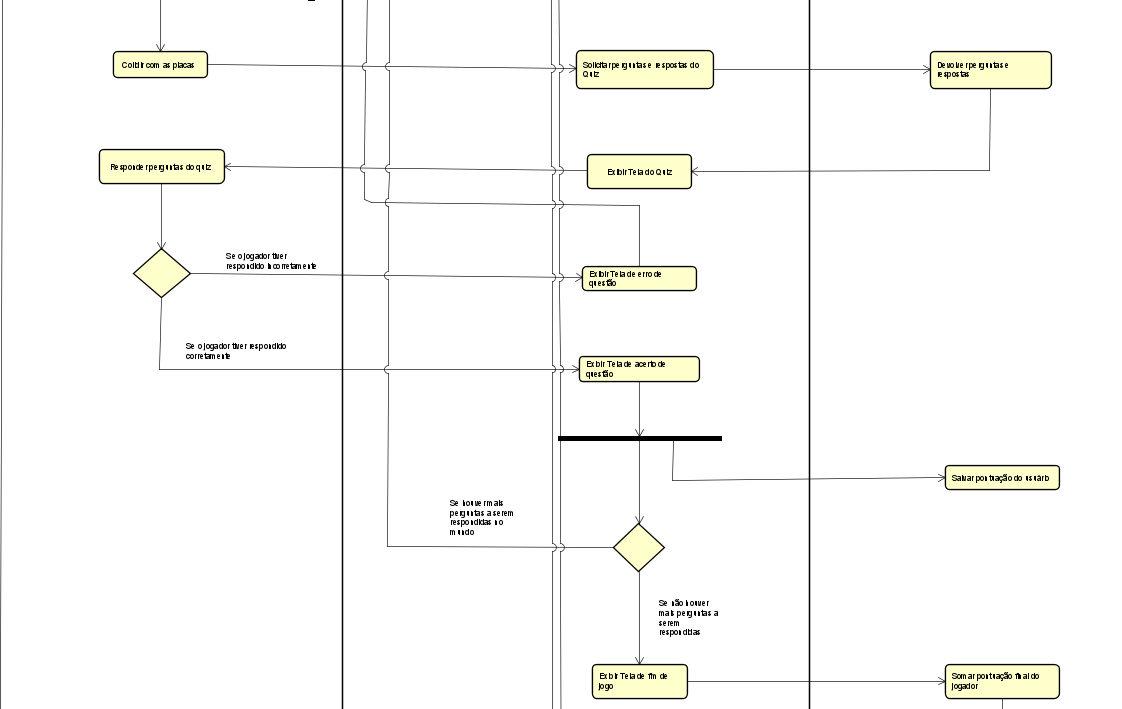
O diagrama de atividades representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema, geralmente utilizado para os métodos que contém regras de negócio.

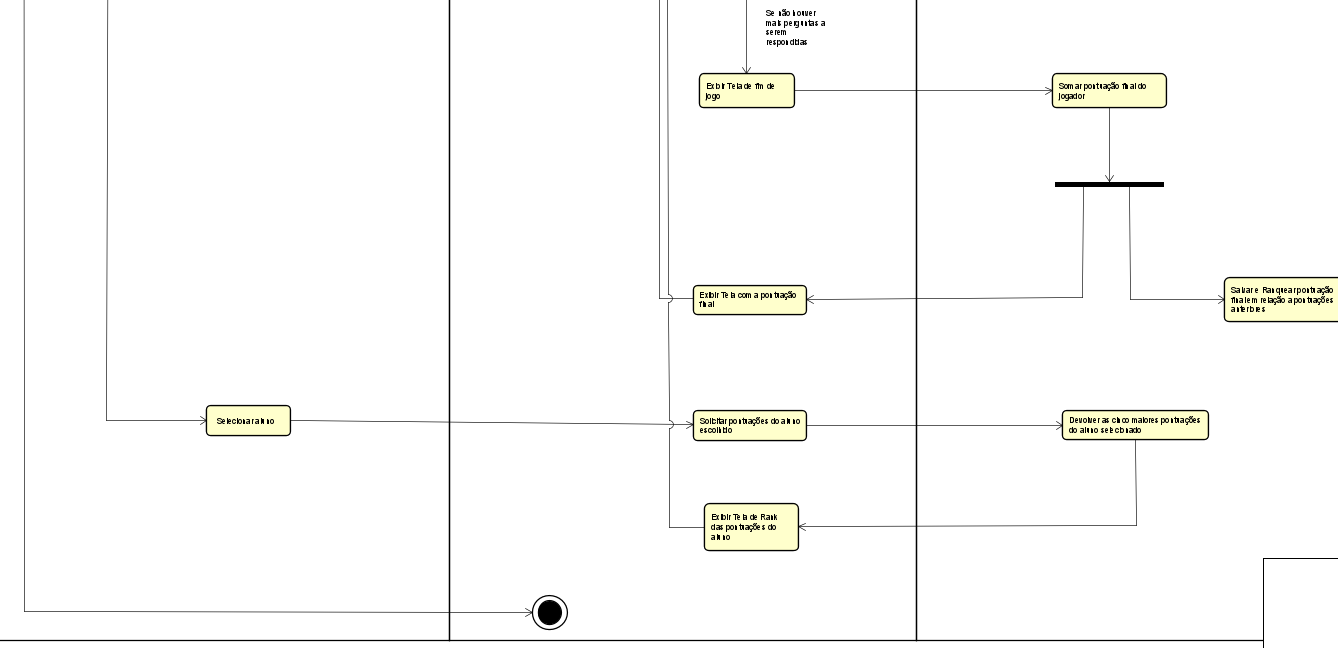
*Esse diagrama deverá ser elaborado se houver necessidade e agregar valor ao projeto.*

*Disciplina de Apoio: Modelagem Orientada a Objetos.*







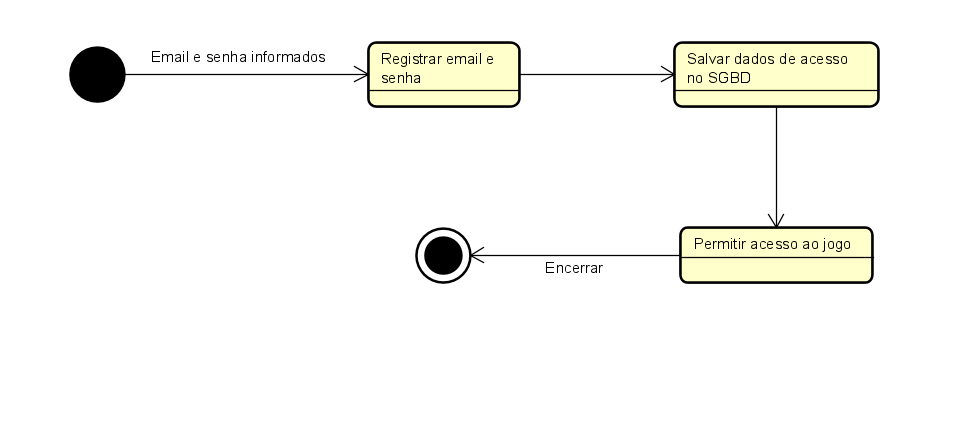


## Diagrama de estados (opcional)

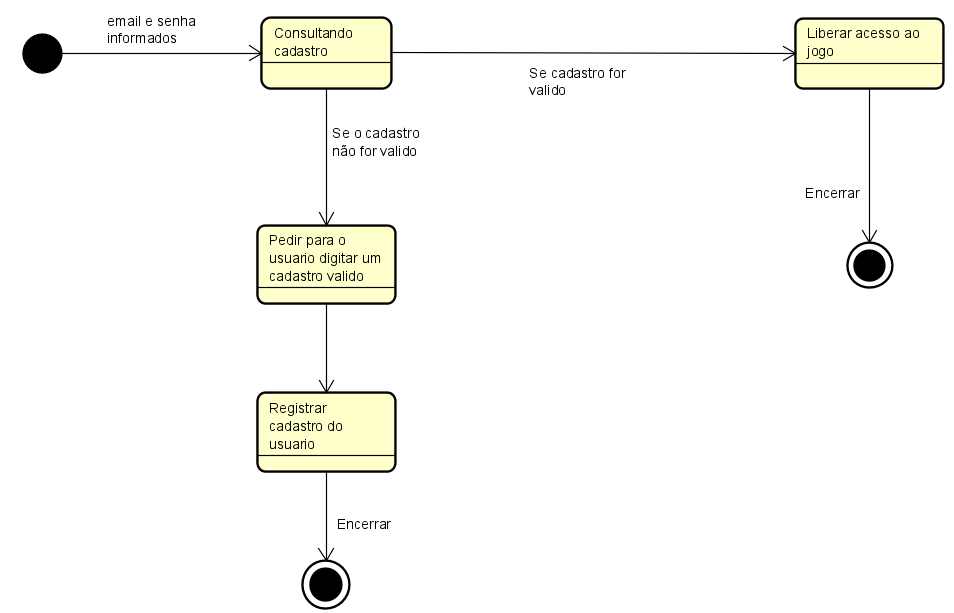
O diagrama de estados especifica as sequências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

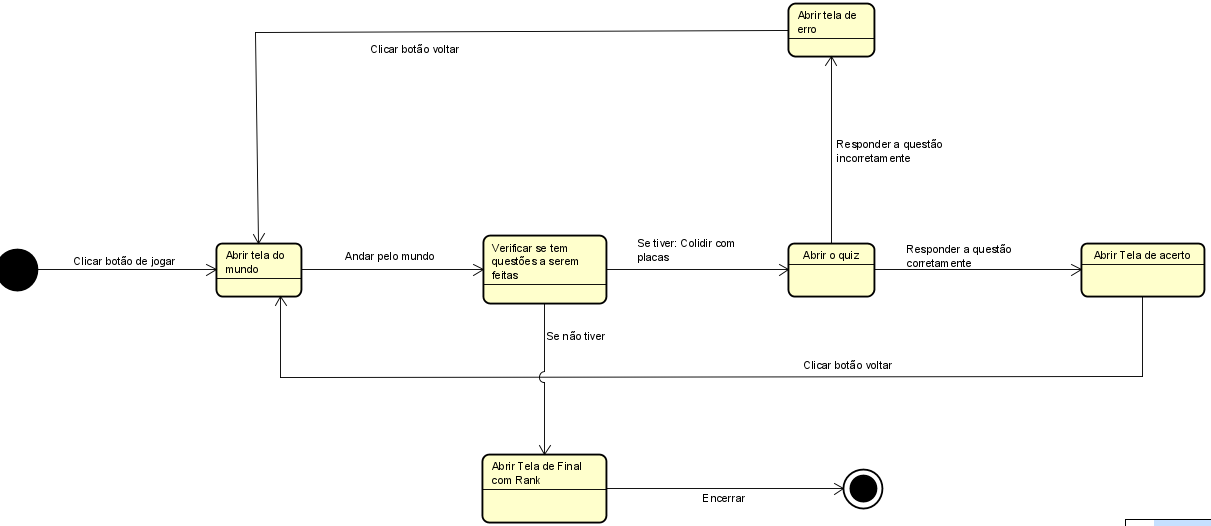
**- Manter Cadastro:**



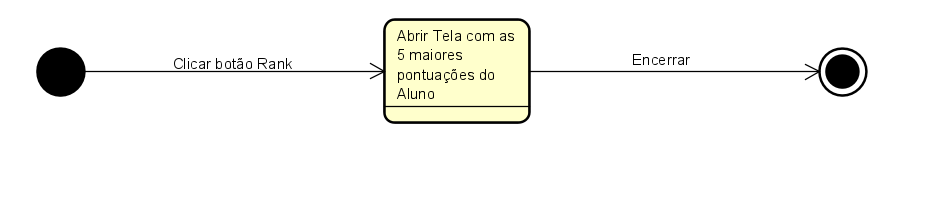
**- Autenticar usuário:**



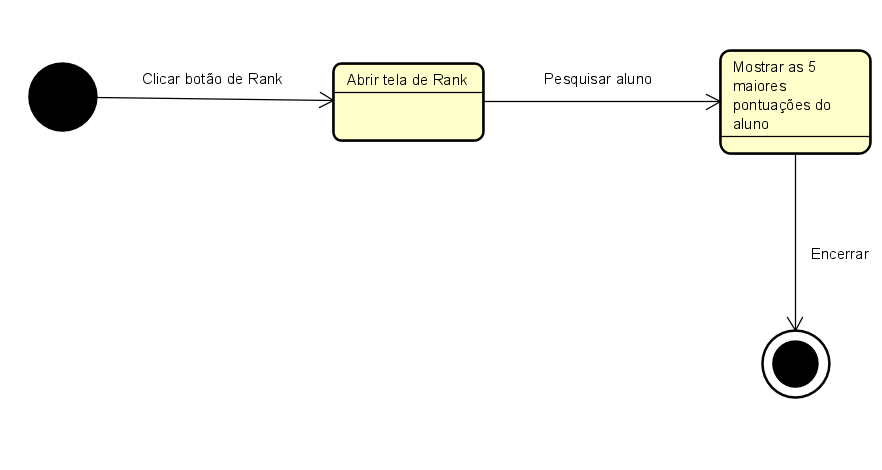
**- Jogar:**



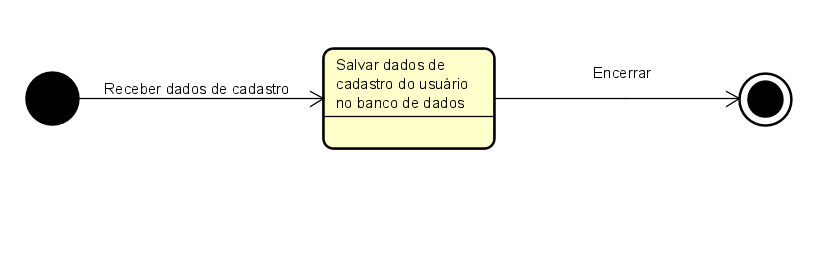
**- Visualizar pontuação:**



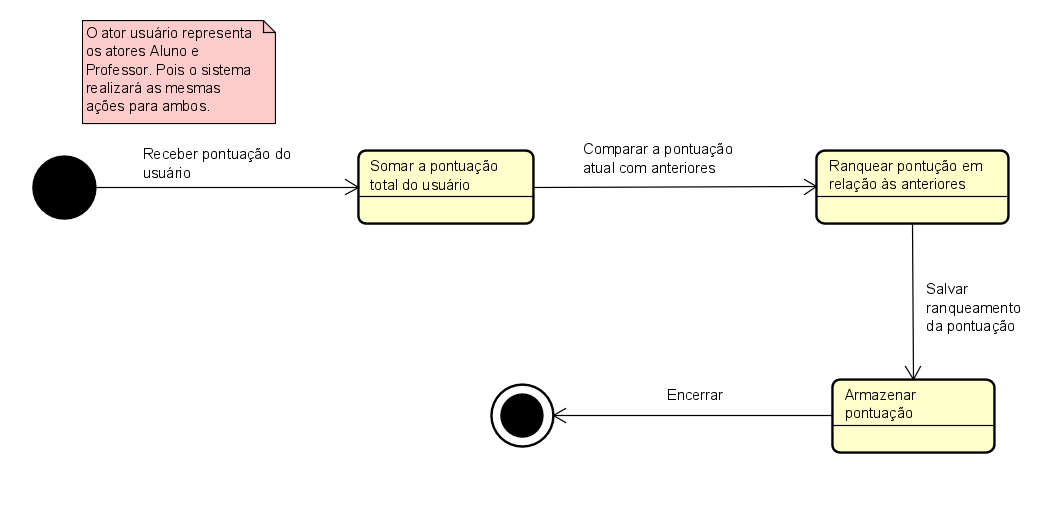
**- Visualizar pontuação dos alunos:**



**- Manter dados usuários:**



**- Armazenar Pontuações:**



# 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.*

**Link de acesso ao repositório no GitHub:**

**https://github.com/gugsgod/ProjetoSelecaoNatural**

# 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.*

# 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.

# Registro da Apresentação ao Parceiro

Neste item devem ser apresentados os registros firmados com os parceiros do projeto.

1. Referências

Neste item incluir as referências bibliográficas que forem utilizadas na documentação, seguindo o padrão ABNT.

Apêndice I

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

*Disciplina de Apoio: Modelagem Orientada a Objetos.*