Projeto [registro de partidas competitivas]

Documento de Visão do Produto

UPF

Prof: Alexandre Lazaretti Zanatta

Alunos: Luis Fernando & Gabriel Buffon

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 03/05/2017 | 1.0.0 | Versão diagrama de classe | Luis e Gabriel |
| 10/05/2017 | 1.0.1 | Versão mockups | Luis e Gabriel |
| 12/06/2017 | 1.0.2 | Versão final | Luis e Gabriel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Requisitos Não-Funcionais**

**Requisitos de operacionais:**

* O usuário deve conter o java instalado.
* Sistema operacional Windows 10
* O sistema será feito em camadas

**Requisitos de desenvolvimento:**

* O desenvolvimento deve ser feito no padrão net-beans.

**Requisitos de confiabilidade:**

* O programa deve ser capaz de ler os dados salvos por ele mesmo

**Requisitos Funcionais**

**Requisito para inserção e excusão**

**Tipo:** [ X ] funcional [ ] não funcional  
 **Importância:** [ ] crítico [ X ] importante [ ] útil  
 **Priorização:** [ 1 ]

**Problemas/Necessidades Identificadas:**

Partidas deverão ser adicionadas para o programa fazer sua função

A interface gráfica deve conter campos para introdução dos dados representados abaixo.

**Solução**

Inserção por registro

* + K/D/A (ama – abates, mortes e assistências)
  + W/L/D (resultado da partida – vitória, derrota e empate)
  + Pontuação
  + Rank atual (baseado nas divisões do jogo)

**Exclusão**

Caso o usuario insira um dado errado ele pode excluir o dado, selecionando ele pela tabela e pressionando o botão de exclusão

**Requisitos para análise e exibição de dados :**

**Tipo:** [ X ] funcional [ ] não funcional  
**Importância:** [ ] crítico [ X ] importante [ ] útil  
**Priorização:** [ 1 ]

**Problemas/Necessidades Identificadas:**

Deve manter uma tabela de exibição dos dados.

Uma função que devera gerar um gráfico para cada um dos dados apontados abaixo. Comparando entre partidas.

**Solução**

K/D/A (ama – abates, mortes e assistências)

W/L/D (resultado da partida – vitória, derrota e empate)

Pontuação

Rank atual (baseado nas divisões do jogo)

**Requisitos para armazenamento**

**Tipo:** [ X ] funcional [ ] não funcional  
**Importância:** [ ] crítico [ X ] importante [ ] útil  
**Priorização:** [ 1 ]

**Problemas/Necessidades Identificadas:**

O programa será capaz de manter um arquivo , que armazena as informações adicionadas

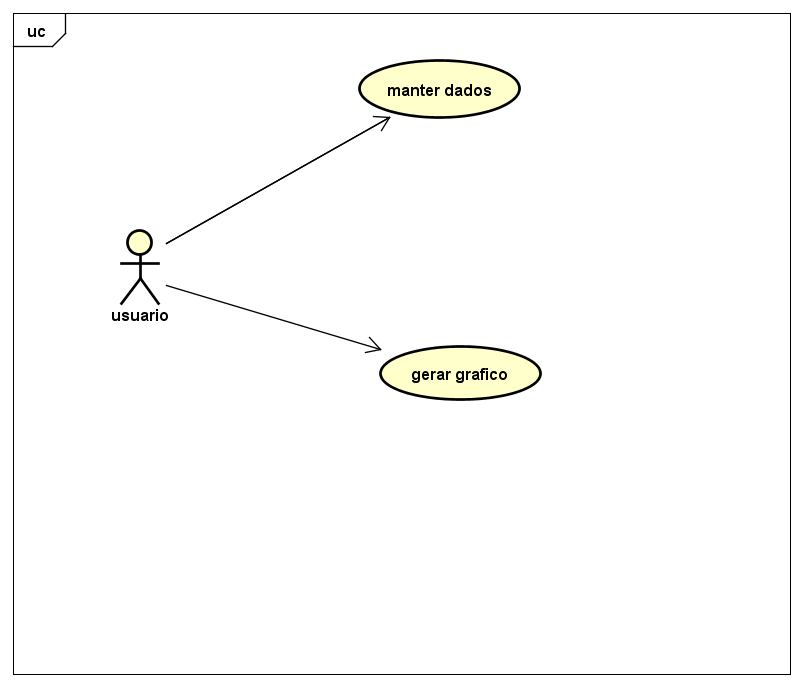
**Solução**

Será armazenado cada dado com separado por partida, sendo que

(K) (D) (A) (W L D) (P) (R)

**Diagrama de casos de uso(uc)**

O diagrama de casos de uso, expresso em UML (*Unified Modeling Language*), expressa os requisitos funcionais do sistema na forma de casos de uso

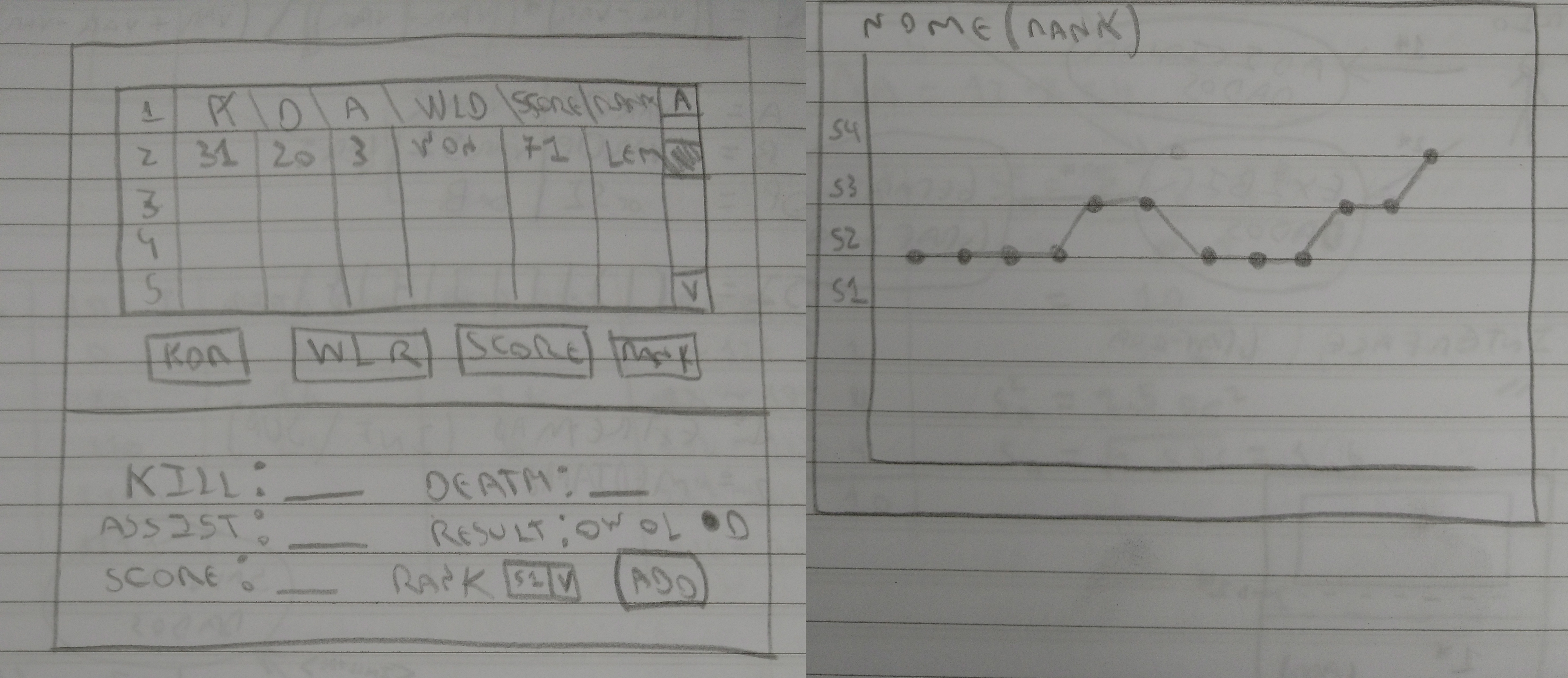
****

**Caso de uso estendido**

Adicionar Dados: o programa apresentara uma área específica para a inserção de dados na janela principal, se escreverá cada dado em seu respectivo campo e apertara o botão de confirmação.

Exibir Dados: todos os dados já inseridos serão exibidos na janela principal do programa.

Gerar Gráfico: os gráficos serão gerados a partir do comando do usuário ao apertar um botão na janela principal, gerando um gráfico diferente dependendo do botão apertado.

 Excluir Dados: deverá ser selecionado o dado a ser excluído em uma tabela presente na janela principal e será realizada sua exclusão ao apertar o botão excluir com o dado selecionado.

**Regra de negócios**

Rn01 – kills/assists/deaths/score : deve só aceitar somente numero inteiros.

Rn02 – score: deve ser maior ou igual a 2 vezes kills mais assists .

Rn03 – win/loss/draw: não pode ser vazio.

Rn04 – O sistrema não deverá gerar gráfico sem dados.

## **Detalhamento de caso de uso**

Manter dados

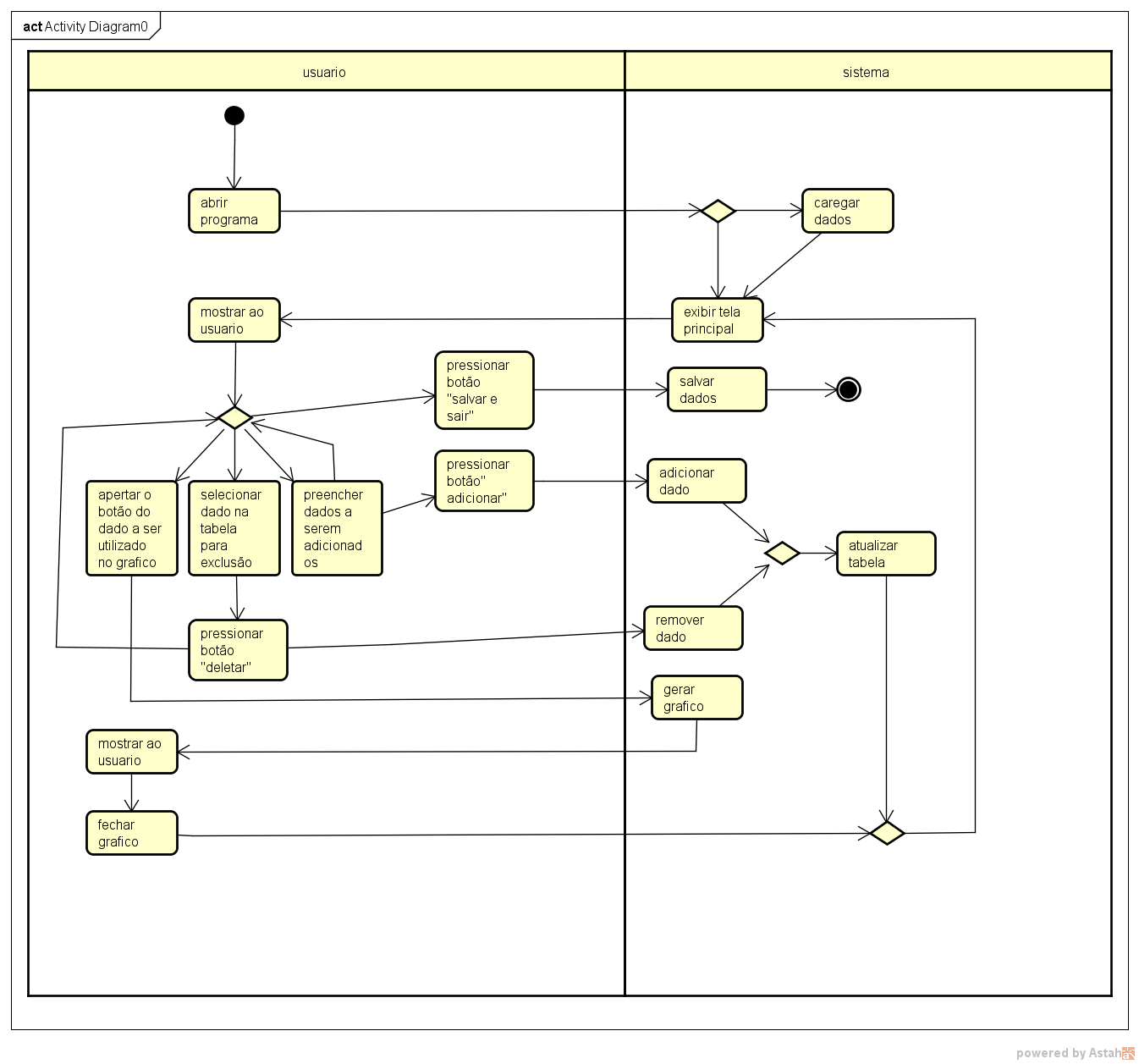
|  |  |
| --- | --- |
| **[UC01]** | |
| Nome: | Manter dados |
| Descrição: | O usuário poderá adicionar e excluir. |
| Atores: | Usuário |
| Prioridade: | [ 1 ] |
| Fluxos de eventos | |
| Fluxo principal: | 1. Escreve o dado no formulário ou seleciona o dado na tabela 2. aperta o botão correspondente. |
| Fluxo de exceção: | **FE01 – Campo obrigatório não informado ou invalido**  O sistema notificará o usuário. |
| Esboço de tela: |  |
| Cenários: | **CEN01 – Usuário com dados corretos para cadastro**  O usuário cadastra os dados no sistema  **CEN02 – Usuário com dados incorretos para cadastro**  o sistema gera uma mensagem de erro  **CEN03 – Usuário seleciona dado para exclusão**  O usuário exclui o dado selecionado  **CEN04 – Usuário exclui sem selecionar dados**  O sistema não gera nada |
| Regras de Interação: |  |

Gerar gráfico

|  |  |  |
| --- | --- | --- |
| **[UC02]** | | |
| Nome: | Gerar gráfico | |
| Descrição: | Este caso de uso descreve a ação do usuário para o sistema gerar o grafico | |
| Atores: | Usuário | |
| Prioridade: | [ 1 ] | |
| Entradas e pré-condições: | | * Deve ter dados para ser gerado |
| Fluxos de eventos | | |
| Fluxo principal: | **1.** O usuário clica no botão correspondente ao dado que o mesmo deseja a gerar | |
| Fluxo de exceção: | **FE01 – Tabela vazia**  O sistema não faz nada | |
| Esboço de tela: |  | |
| Cenários: | **CEN01 – Usuário** **aperta o botão para gerar gráficos com dados na tabela**  O sistema gera gráficos normalmente  **CEN02 – Usuário aperta o botão para gerar gráficos sem dados na tabela**  O sistema não faz nada | |

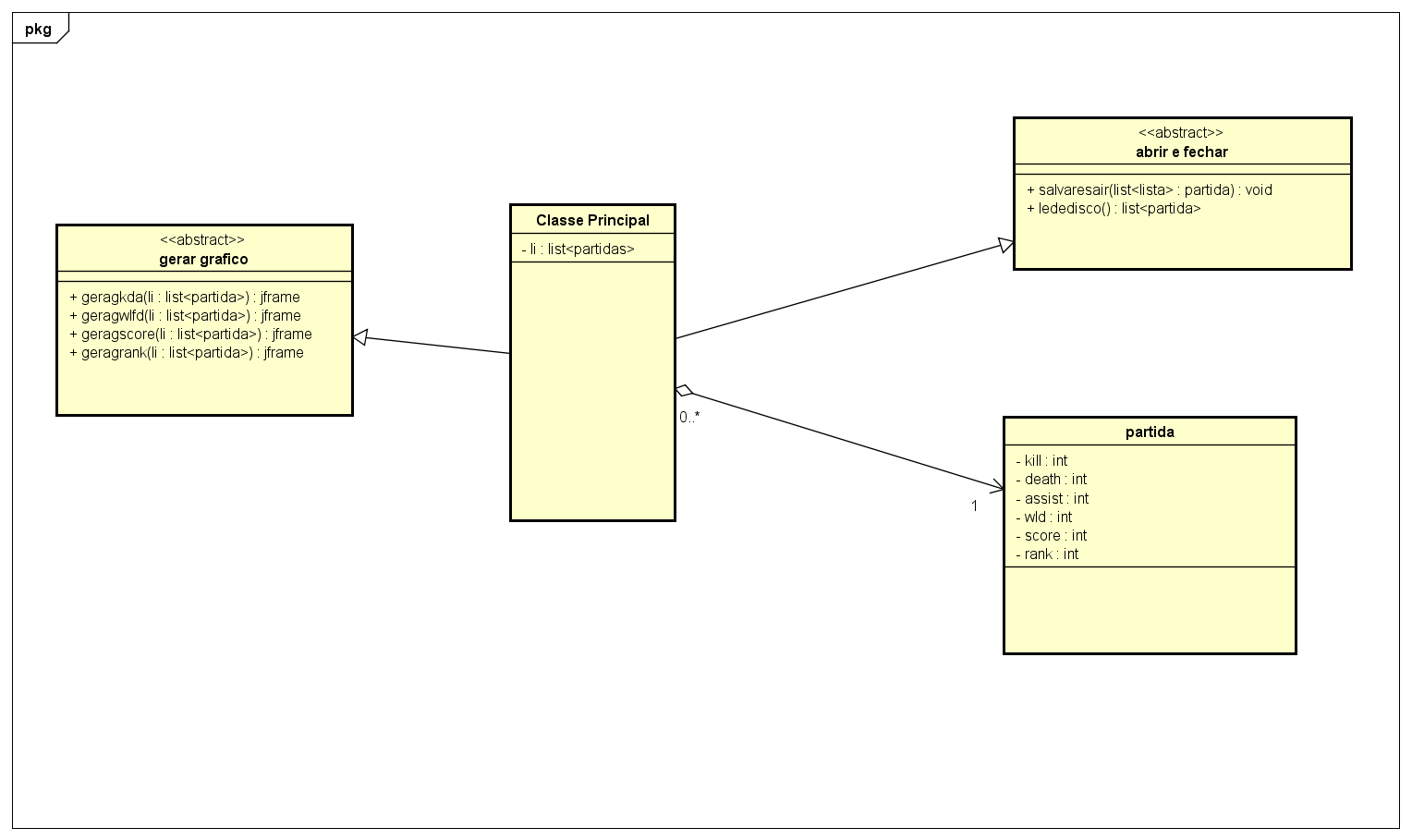
**Diagrama de Atividade(ad)**

O diagrama abaixo descreve atividades executadas nos casos de usos



**Diagrama de classe(cd)**

o diagrama representa a estrutura estática de classes do sistema de gerencia de partidas

****

**Evolução(Ev)**

tratar erro inserção evolução