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 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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 de texto que armazena as informações adicionadas

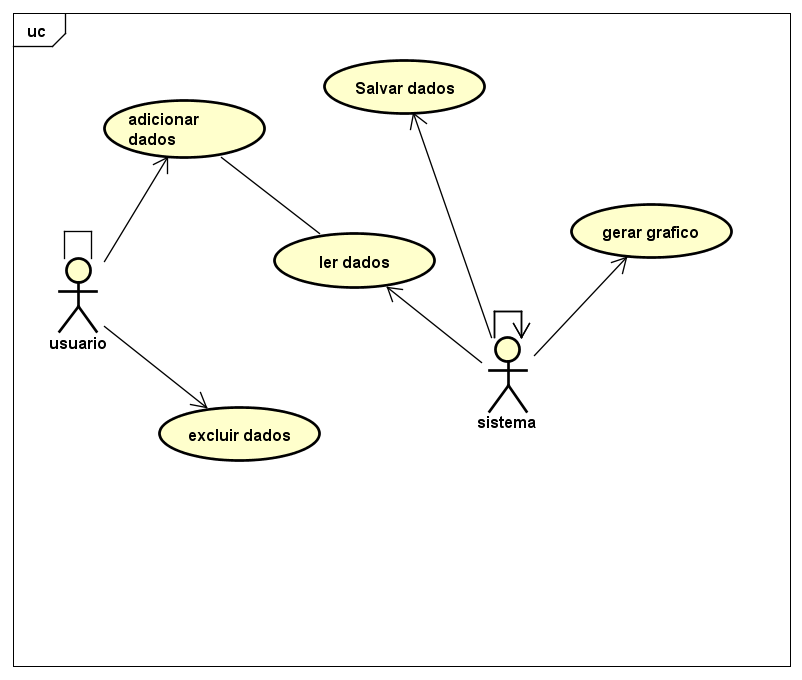
**Solução**

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

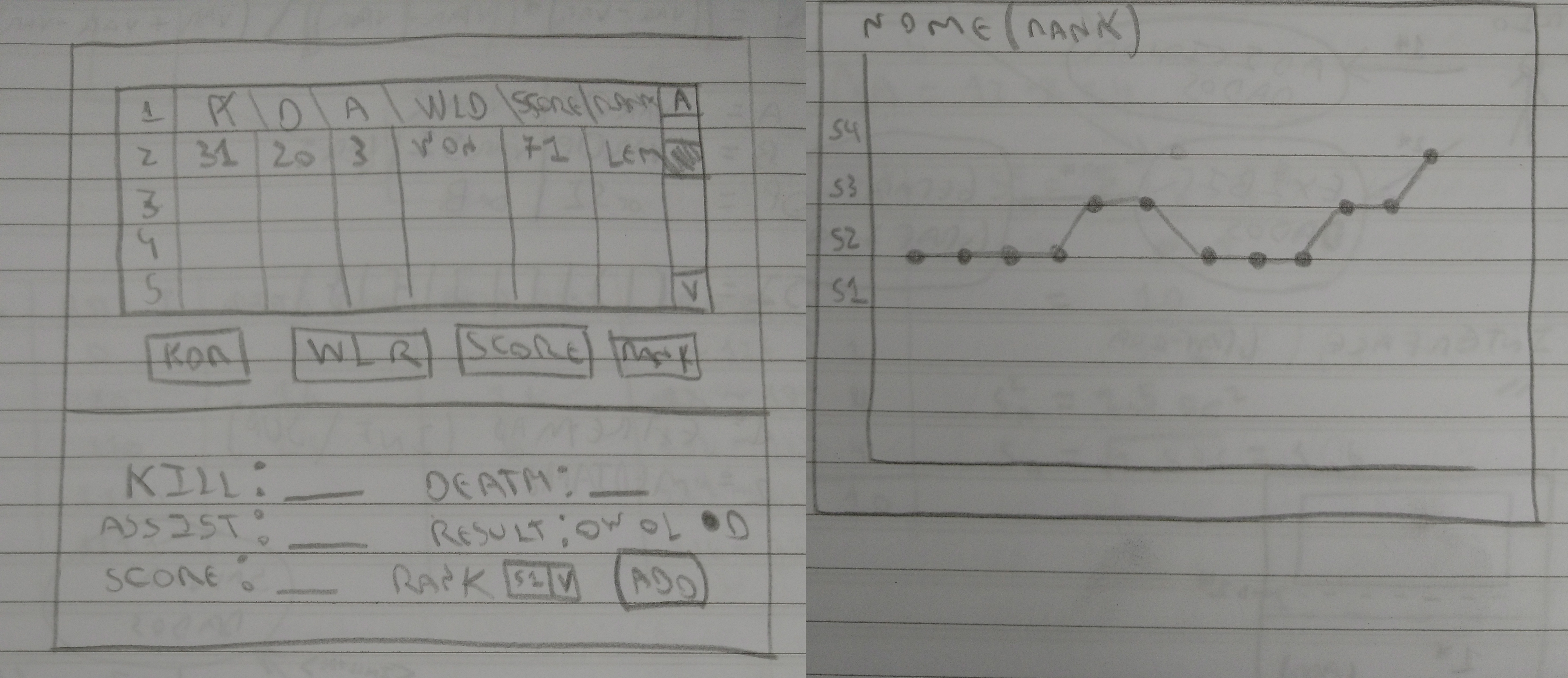


**Caso de uso estendido**

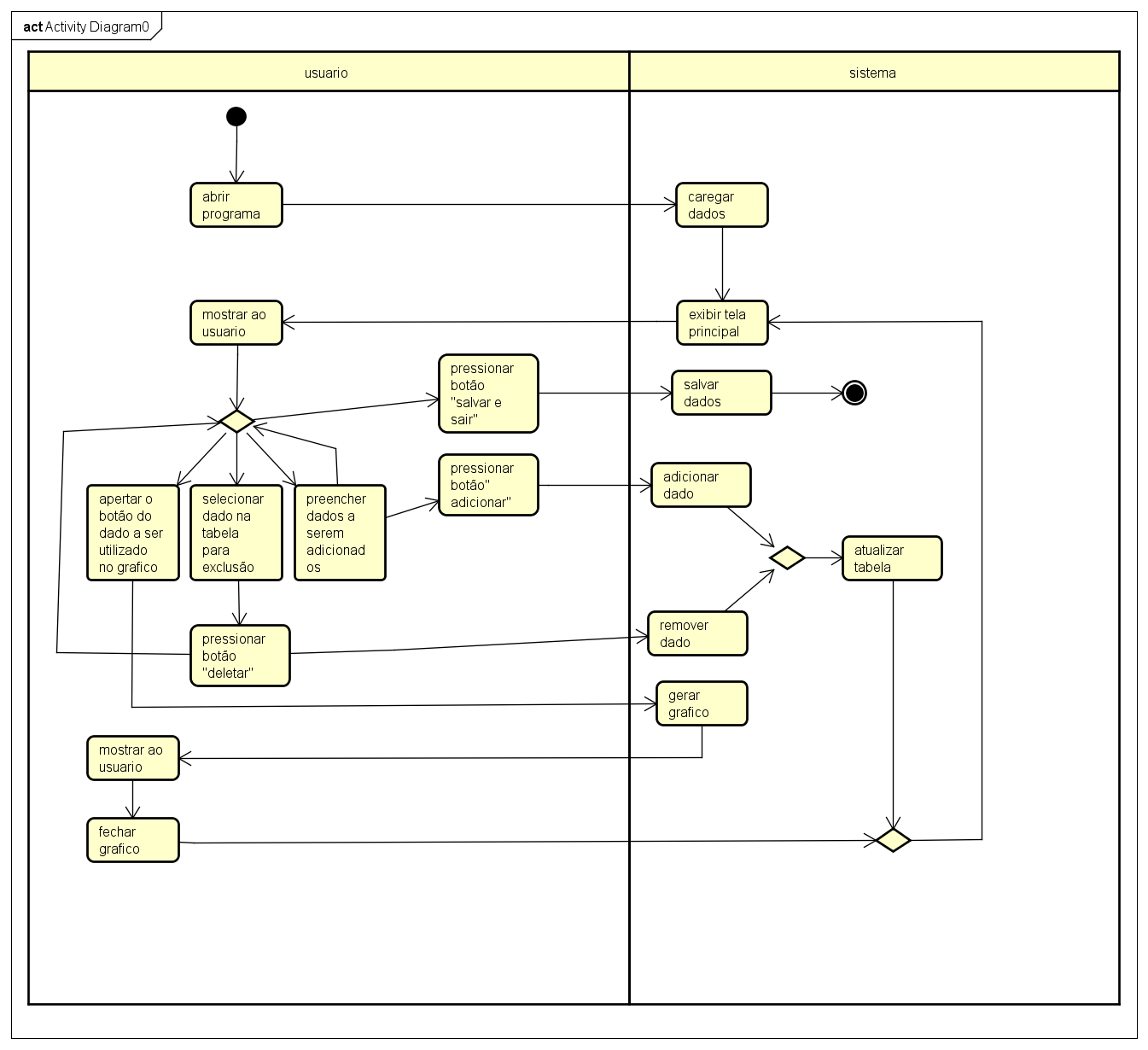
Adicionar Dados: o programa apresentara uma área específica para a inserção de dados na janela principal, se escreverá cada dado em seu respectivo campo e apertara o botão de confirmação.

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

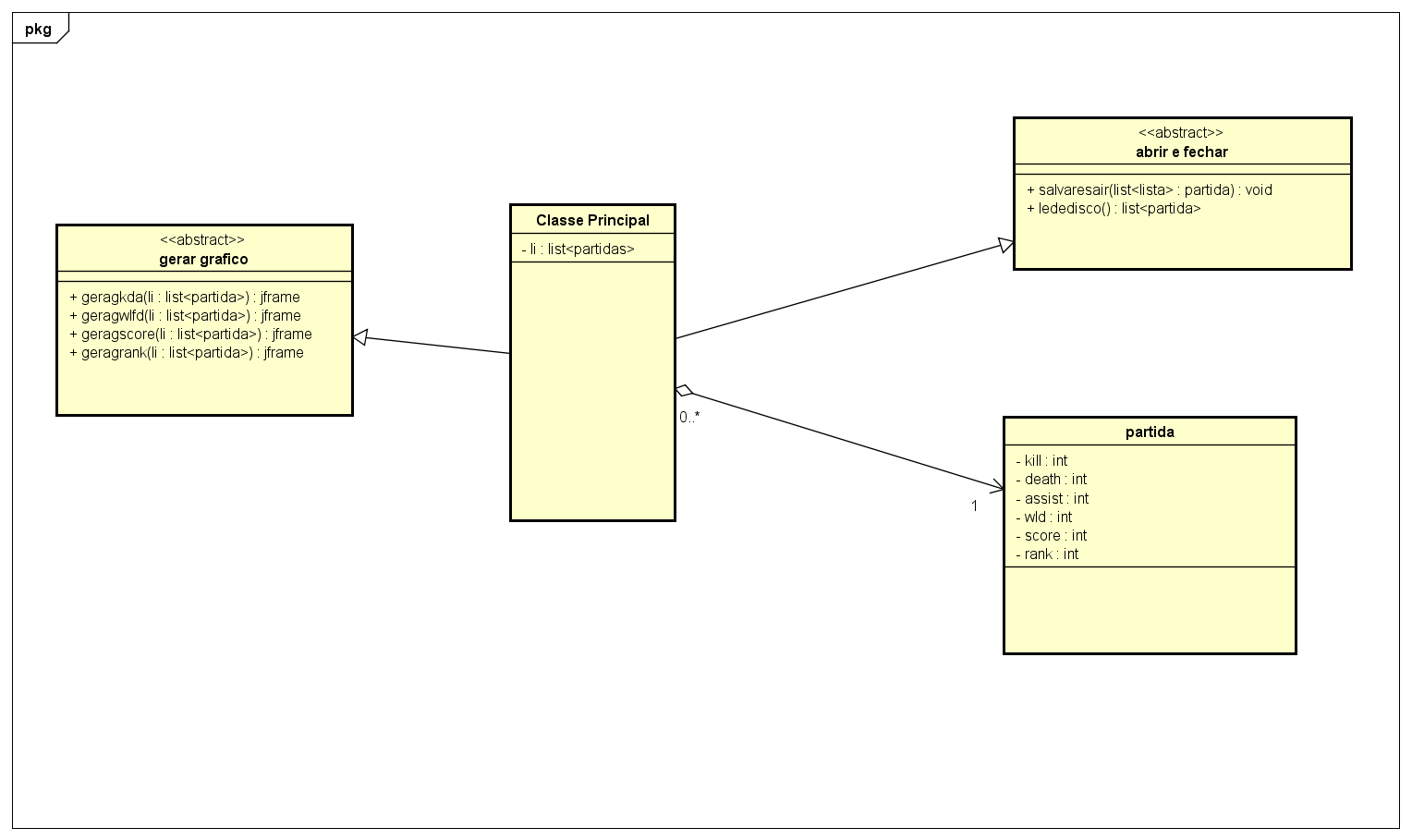
Gerar Gráfico: os gráficos serão gerados a partir do comando do usuário ao apertar um botão na janela principal, gerando um gráfico diferente dependendo do botão apertado.

 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.

**Diagrama de Atividade(ad)**



**Diagrama de classe(cd)**

****