

UNIVERSIDADE ESTADUAL DA PARAÍBA

DEPARTAMENTO DE COMPUTAÇÃO

**ALUNOS:**AURELIANA LELIS DA SILVA, 132084287

BIANCA LAYSE FURTADO, 142080519

SERGIO DAVI MARINHO, 142080462

LARRISSA DANTAS XAVIER, 142083526

**PROFESSOR:** JANDERSON JASON BARBOSA AGUIAR

**DISCIPLINA:** ENGENHARIA DE SOFTWARE

**DOCUMENTO DE REQUISITOS - JOGO UNO**

Campina Grande, Novembro de 2017

# **Histórico de Revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor(a)** |
| 07/11/2017 | 0.1 | Criação do documento de requisitos. | Larrissa Dantas |
| 08/11/2017 | 0.2 | Definição dos requisitos funcionais e não-funcionais. | Larrissa Dantas |
| 09/11/2017 | 0.3 | Diagrama de Caso de Uso | Bianca Layse |
| 09/11/2017 | 0.4 | Projeto Arquitetural | Sérgio Davi |
| 09/11/2017 | 0.5 | Definição dos Diagramas de Atividades. | Aureliana Lelis |
| 09/11/2017 | 0.6 | Projeto Simplificado do BD | Sérgio Davi |
| 09/11/2017 | 0.7 | Diagrama de Classes | Sérgio Davi |

**Sumário**

[**Histórico de Alterações**](#_vpovehxl1fp5) **2**

[**Introdução**](#_mg8btefx3t2y) **4**

[Visão geral do documento](#_lo5ij5550agh) 4

[Requisitos Funcionais](#_lphik1s61j04) 5

[Requisitos Não-Funcionais](#_3podhpdl66q1) 5

[**Descrição Geral do Sistema**](#_tlap0wai3db7) **6**

[Diagrama de Caso de Uso](#_hts6j2rtiwqg) 6

[Diagrama de Classes](#_39sdb7pj9tua) 7

[Diagramas de Atividades](#_b6y63k9kbhhr) 8

[Projeto Simplificado do BD (Esquema Relacional Diagramático)](#_sj4diqus2auj) 9

[Projeto Arquitetural](#_9rbdp2bdzpfp) 9

[**Referências**](#_sl90llcmit8g) **10**

# 

# **Introdução**

Este documento apresenta a especificação de requisitos do jogo UNO. É fornecido de forma objetiva, as informações do projeto. Como os requisitos funcionais e não-funcionais, o projeto arquitetural, projeto simplificado do Banco de Dados, diagrama de Classes, diagrama de Casos de Uso, sendo especificados os fluxos de eventos por meio de Diagramas de Atividades. Sistematizando assim, o projeto, de acordo com o que foi solicitado.

## **Visão geral do documento**

Foi apresentado inicialmente a seção introdutória(seção 1), onde é mostrado o histórico de alterações, a introdução e a visão geral do documento. As seguintes seções são:

**Seção 2:** Será mostrado os **Requisitos funcionais -** Funcionalidades que se espera que o sistema disponibilize, de uma forma completa e consistente. E os **Requisitos Não-Funcionais -**  Aspectos não-funcionais do sistema, como restrições nas quais o sistema deve operar.

**Seção 3:** Esta seção é destinada para mostrar a visão geral do sistema, será mostrado o diagrama de caso de uso, de classe, de atividade, o projeto arquitetural e o projeto simplificado do Banco de Dados.

**Seção 4:** Referência Bibliográfica - Todo material utilizado como base teórica e também, base prática, para a conclusão do projeto.

**Identificação dos Requisitos**

Os requisitos definem os serviços que o sistema deve oferecer, o conjunto deles determina a operação do sistema. Será detalhado neste tópico todos os requisitos funcionais e todos os requisitos não-funcionais do sistema que iremos desenvolver.

Para que seja possível a compreensão do nível de prioridade de cada requisito, tanto o requisito funcional, quanto o requisito não-funcional, será dividido em três níveis de prioridade: Essencial, Importante e desejável.

## **Requisitos Funcionais**

Os requisitos funcionais referem-se aos requisitos que estão relacionados com a maneira com que o sistema deve operar, onde se especificam as entradas e saídas do sistema e o relacionamento comportamental entre elas, assim como a interação com o usuário.

Desta forma, os requisitos encontrados para o desenvolvimento do Jogo UNO, são:

**[RF001] Cadastrar usuário:** Para utilizar o sistema, um novo usuário terá que se cadastrar. Fornecerá o seu nome, data de nascimento, vai disponibilizar uma imagem(que será seu avatar durante todas as partidas jogadas), vai disponibilizar um nome para o login(que poderá conter letras ou números), que terá no mínimo 4 caracteres e senha de no mínimo 8 caracteres. Todos os dados terão que ser salvos no Banco de Dados.

Como a realização do cadastro, indica que o usuário está tendo seu primeiro contato com o sistema, logo após o usuário clicar no botão de “salvar cadastro”, será disponibilizado um tutorial de todas as funcionalidades do sistema.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF002] Login:** O usuário após realizar seu cadastro terá seu acesso para fazer o login. Como todo o cadastro foi salvo no banco de dados, o usuário na tela de login, irá disponibilizar o “login” que terá no mínimo 4 caracteres e sua respectiva “senha”, que terá no mínimo 8 caracteres. Quando for o primeiro login do usuário, será mostrado um tutorial com todas as funcionalidades do sistema. Caso não seja o primeiro login do usuário, o mesmo será encaminhado para tela inicial, que será o menu principal.

Os dados fornecidos no login, terão que ser os mesmos fornecidos pelo usuário no momento do cadastro, caso não seja o mesmo, mostrará um erro na tela, informando que os dados informados estão incorretos.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF003] Tela inicial ou Menu Principal:** Este requisito apresenta um menu inicial do sistema, onde vai conter uma tela interativa, em que o usuário vai poder navegar e escolher opções, como: Iniciar Jogo, Continuar Jogo, Retornar Jogo Salvo, Visualizar Ranking, Sair do Jogo, “Help” e vai poder ver o seu avatar como sua identificação no sistema. É uma das telas principais do sistema.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF004] Tutorial no primeiro login:** O usuário no primeiro contato com o sistema, estará um pouco perdido com suas funcionalidades. Com essa problemática, o sistema irá dispor de um tutorial para aqueles usuários que estão realizando o primeiro login. O tutorial será explicativo, contendo de forma fácil, todas as funcionalidades do sistema.

Prioridade: () Essencial (**x**) Importante ( ) Desejável

**[RF005] Iniciar o jogo:** Este requisito permite que o jogador possa iniciar uma partida.Terá os avatares de cada jogador durante a partida. Será disponibilizado um botão “Iniciar jogo”, o usuário ao clicar terá acesso ao ambiente do jogo.

Inicialmente, estará disponível a versão simples, no caso, sem inteligência artificial.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF006] Salvar e/ou continuar jogo:** O usuário poderá salvar o jogo. Será disponibilizado o botão “salvar jogo”, onde a partida que o usuário está jogando será salva. Terá também a opção para o usuário continuar o jogo, caso ele tenha que sair da partida antes que a partida tenha acabado. O sistema irá disponibilizar o botão “continuar jogo”. Caso o usuário não queira salvar o jogo ou continuar, vai contar como uma derrota do usuário.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF007] Sair do jogo:** Este requisito permite o usuário sair do jogo, através do botão que terá, intitulado “Sair do jogo”.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF008] Pontuação:** O primeiro jogador que ficar sem cartas vai ganhar o jogo. O resto dos jogadores mostram suas cartas e somam o valor delas. Essa soma será a pontuação do vencedor. Para salvar a pontuação, vai pro ranking a cada término de partida, o vencedor da mesma acumula pontos e vai atualizando no ranking.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF009] Jogador que inicia:** Para determinar o jogador que vai iniciar a partida, no início, cada jogador vai receber uma carta, o jogador que tiver com a carta maior irá começar a partida. As cartas que foram usadas para esse início, serão embaralhadas com as outras cartas, para a partir daí distribui-las e iniciar o jogo com o jogador que obteve anteriormente a maior carta. O sentido do jogo é sempre horário.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF010] Ranking:** O ranking vai mostrar a quantidade de vitórias de cada jogador, por colocação. De melhores para piores. Se caso o usuário não possua nenhuma vitória não irá para o ranking. A tela do ranking, vai mostrar além da colocação, o nome do jogador, a quantidade de vitórias, a pontuação e o avatar de cada jogador, por fim, a média de todas as vitórias. O Rank é composto da soma das pontuações e da média do tempo das vitórias , em caso de empate o tempo será critério de desempate no rank. Será possível também, a opção de zerar o ranking.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF011] Tempo máximo para jogada:** O usuário não vai poder passar muito tempo para jogar, ele terá um tempo de no máximo 5 segundos para jogar. Caso isso não aconteça é forçada a jogada. Colocando de toda forma uma carta na jogada.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF012] A troca da carta 0:** Este requisito é para, quando o jogador lança a cara “0” na mesa, ele escolhe para qual sentido o jogo irá rodar. No caso, horário ou anti-horário.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF013] Método para embaralhar:** No jogo UNO, é necessário em alguns momentos que as cartas sejam embaralhadas. Para isso, será implementado de modo aleatório, apenas um range trocando posições de um vetor, por um determinado tempo.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF014] Carta Pênalti:** Essas cartas não se acumulam na jogada do próximo participante. O primeiro que lança, o próximo já “paga” o pênalti, mesmo que ele possa ter uma carta da mesma para dar continuidade. Isso não vai acontecer.

Prioridade: () Essencial (**x**) Importante ( ) Desejável

**[RF015] Regra do corte:** Quando um jogador possui uma carta idêntica(número ou símbolo + cor) a da última descartada, ele poderá jogá-la antes que o próximo jogador descarte. Assim ele corta a vez dos jogadores e o jogo continua a partir dele.

Prioridade: ( ) Essencial ( ) Importante (**x**) Desejável

**[RF016] Jogadas seguidas ganhar presente:** O sistema usará método de bonificação. Pois, quando o usuário jogar 5 partidas ele irá ganhar um presente(bônus), exemplo: um coringa para partida. Podem ter até 5 tipos de presente mas será disponibilizado 3 para o usuário, de forma surpresa.

Prioridade: (**x**) Essencial ( ) Importante () Desejável

**[RF017] Botão UNO:** Quando algum jogador estiver com apenas 1 carta, será disponibilizado um botão, para o mesmo apertar. Terá um temporizador, que contará até 5 segundos para o jogador apertar, caso não aperte duas cartas serão adicionadas para ele. O botão ao ser apertado emitirá um som.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF018] Temporizador do jogo:** Haverá um cronômetro que irá iniciar do início da partida até o final da partida. Esse cronômetro vai ajudar a realizar a média de tempo gasto de cada vitória, o tempo para o jogador iniciar a partida e para os jogadores terem uma noção de tempo gasto da partida.

Prioridade: (**x**) Essencial ( ) Importante () Desejável

**[RF019] Gerar Relatório em PDF:** Para se ter um controle geral das partidas, vai ser possível gerar um relatório em formato PDF, com as estatísticas do jogo, o ranking, histórico do jogo, nome dos desenvolvedores, o nome do usuário e seu respectivo avatar, a data de nascimento, e por fim, a logo do jogo.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF020] Jogada:** Este requisito é da regra da jogada. Só pode jogar as cartas da mesma cor ou do mesmo número. Apenas uma carta jogada por vez para cada jogador.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF021] Valor das cartas:** A carta +2 equivale a 20 pontos. Inversão/bloqueio/coringa equivale a 20 pontos. A carta +4 equivale a 50 pontos. A carta numérica(normal) equivale ao valor dela mesma.

Prioridade: (**x**) Essencial ( ) Importante ( ) Desejável

**[RF022] Atribuir as IA’s:** Será construído mais a frente, a inteligência artificial. Será disponível, antes de iniciar o jogo duas modalidades:

* Jogo individual + 3 IA’s.
* O usuário formando dupla com uma IA, contra uma dupla de IA’s.

Prioridade: () Essencial (**x**) Importante ( ) Desejável

## 

## **Requisitos Não-Funcionais**

Os requisitos não-funcionais são aqueles que não estão especificamente relacionados com a funcionalidade do sistema. Eles impõem restrições no produto a ser desenvolvido e/ou no processo de desenvolvimento do sistema como também especificam restrições externas as quais o produto precisa atender.

Desta forma, os requisitos não funcionais encontrados para o desenvolvimento do Jogo UNO, são:

**[RNF001] Help:** Para ajudar os usuários que estão sentindo dificuldade ao utilizar o sistema, terá disponível o botão “Help”. Esse botão irá conter informações que serão necessárias para os usuários utilizarem o sistema sem nenhum problema.

Prioridade: ( ) Essencial ( ) Importante (**x**) Desejável

**[RNF002] Sobre:** Esse requisito é relacionado ao botão “Sobre”, este botão vai informar o que é o jogo e quem desenvolveu o jogo.

Prioridade: ( ) Essencial ( ) Importante (**x**) Desejável

**[RNF003] Efeitos sonoros:** O sistema deve possuir música no menu, durante a partida e quando um jogador ganhar uma partida. Será possível também desabilitar o efeito sonoro.

Prioridade: ( ) Essencial ( ) Importante (**x**) Desejável

**[RNF004] Especificação da linguagem de programação:** Será utilizado em todo o desenvolvimento do projeto a linguagem JAVA.

Prioridade: (**x**) Essencial ( ) Importante () Desejável

**[RNF005] O uso de mouse e teclado:** Para utilizar o sistema é necessário o uso do teclado e mouse.

Prioridade: (**x**) Essencial ( ) Importante () Desejável

**[RNF006] Segurança:** Será utilizado um sistema de segurança, para criptografar a senha do usuário.

Prioridade: (**x**) Essencial () Importante () Desejável

**[RNF007] Mensagens de Alerta:** Será informado mensagens de erro para o usuário, caso o mesmo informe login e senha incorreto.

Prioridade: (**x**) Essencial ( ) Importante () Desejável

**[RNF008] UIxUX:** Terá uma interface intuitiva para o usuário. A usabilidade também será implementada de forma correta.

Prioridade: () Essencial (**x**) Importante () Desejável

# **Descrição Geral do Sistema**

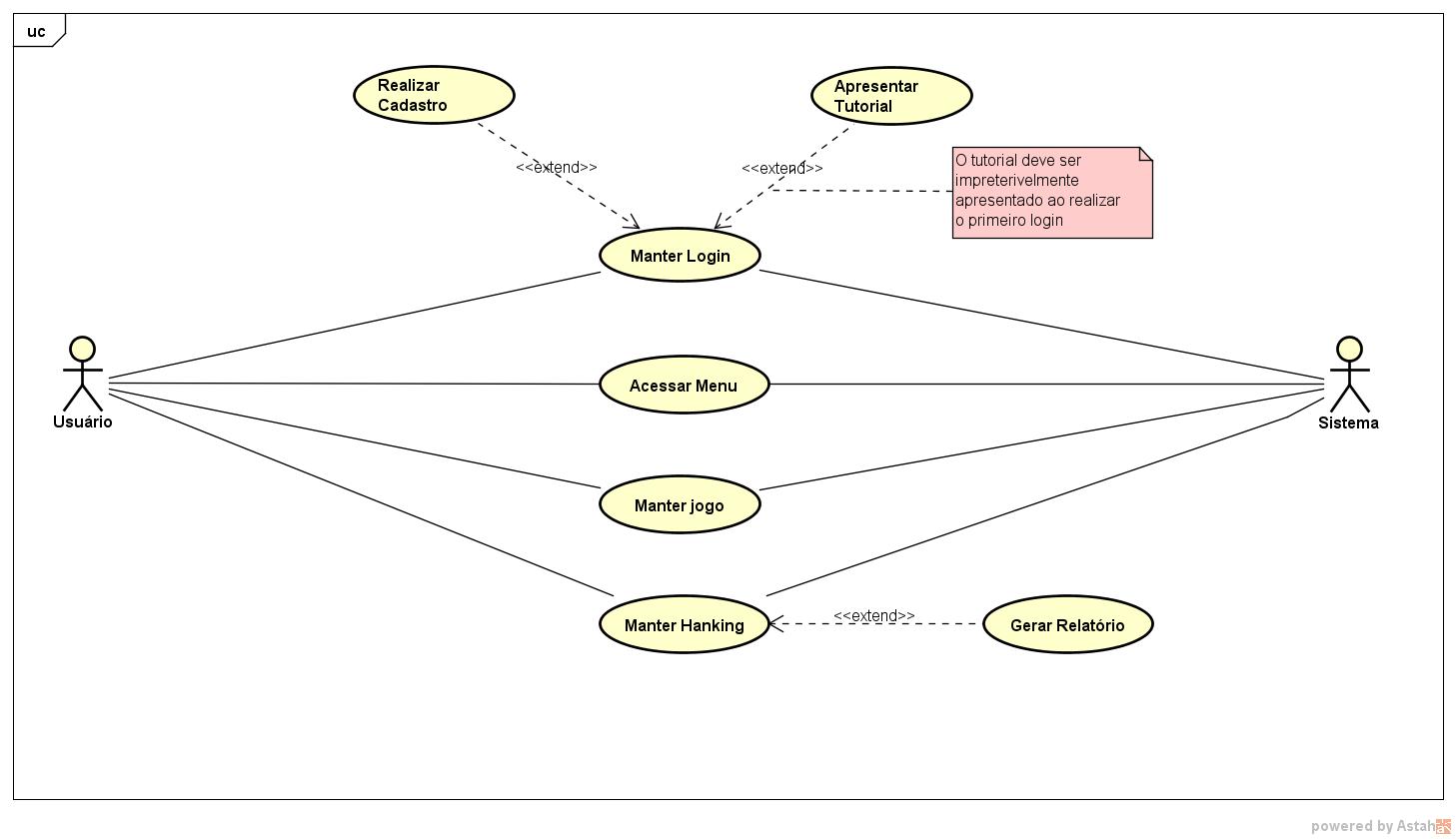
O UNO, é um jogo de cartas. O baralho é composto por cartas de quatro cores: verde, amarelo, azul e vermelho. As fileiras de cada cor variam entre 0 e 9. Existem três ações especiais para cada tipo de cor de carta, identificadas como "pular", "comprar duas" e "inverter". Há também cartas de ações especiais com fundo preto, "coringa" e "coringa comprar quatro". Para cada carta regular ou de ação, existem duas das mesmas no baralho, com exceção do 0, que só possui uma unidade. Há quatro "coringas que mudam de cor" e quatro "coringas comprar quatro", o que resulta num total de 108 cartas.

O nosso sistema vai priorizar todas as regras do UNO. Adicionaremos características extras, como efeitos sonoros, ranking de pontuação e relatório de desempenho do jogador. Terá também, a interação homem-máquina. O usuário poderá jogar com um computador com inteligência artificial.

## 

## 

## **Diagrama de Caso de Uso**



## 

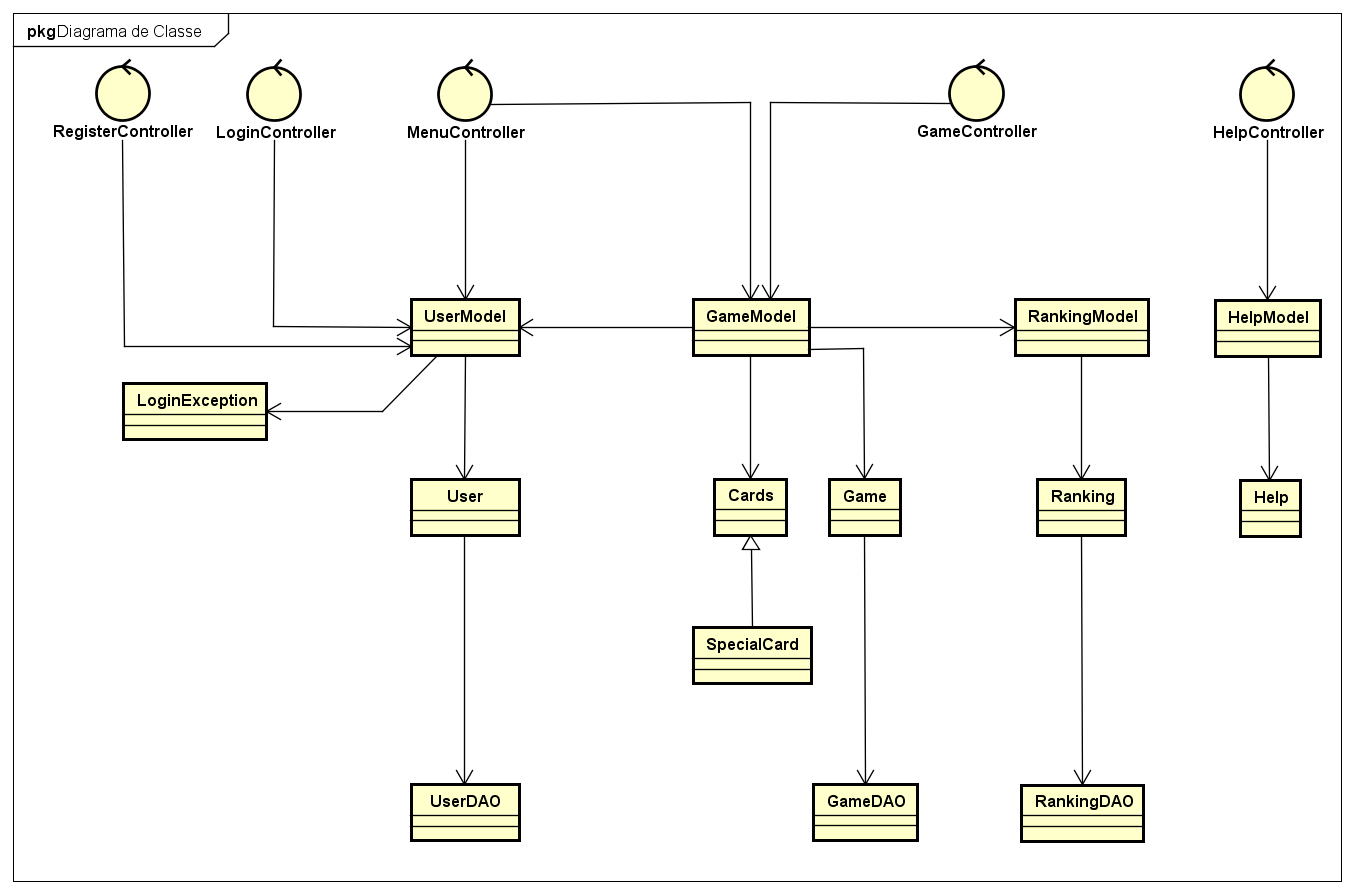
## 

## 

## 

## 

## **Diagrama de Classes**

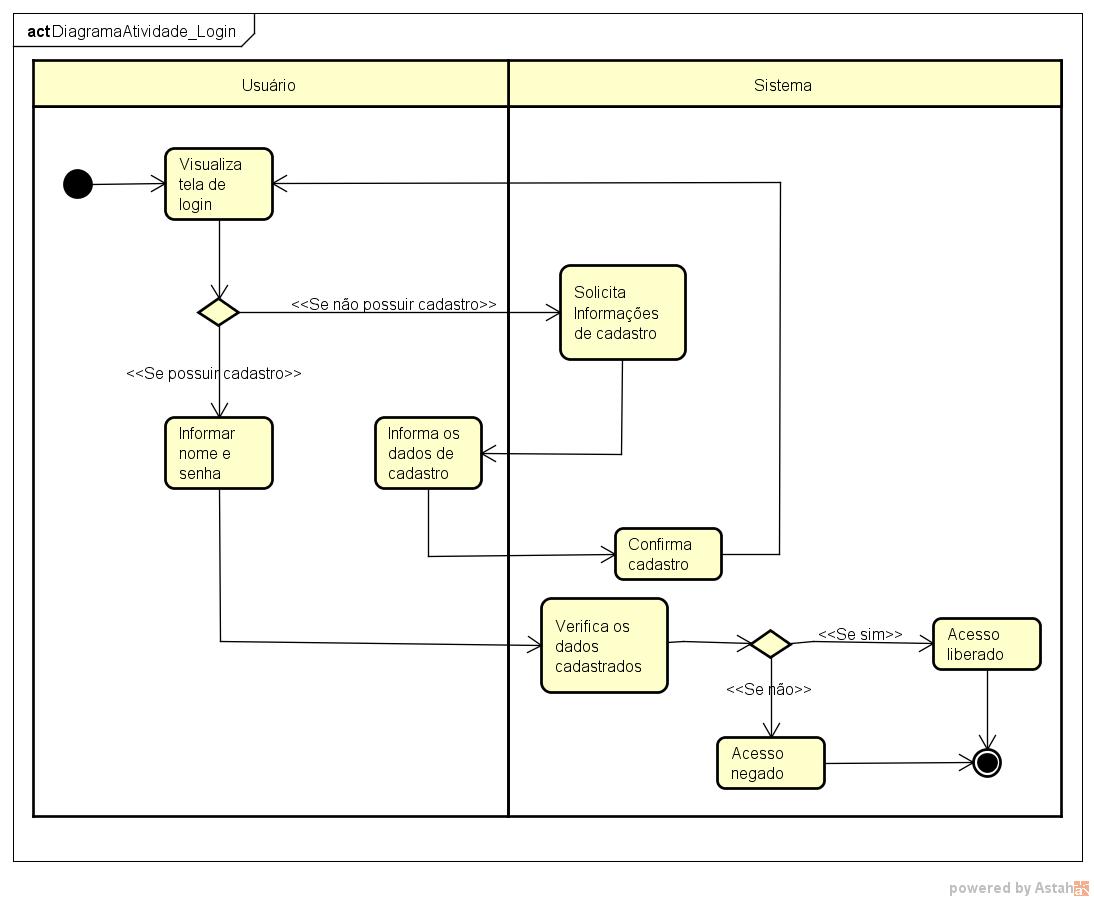


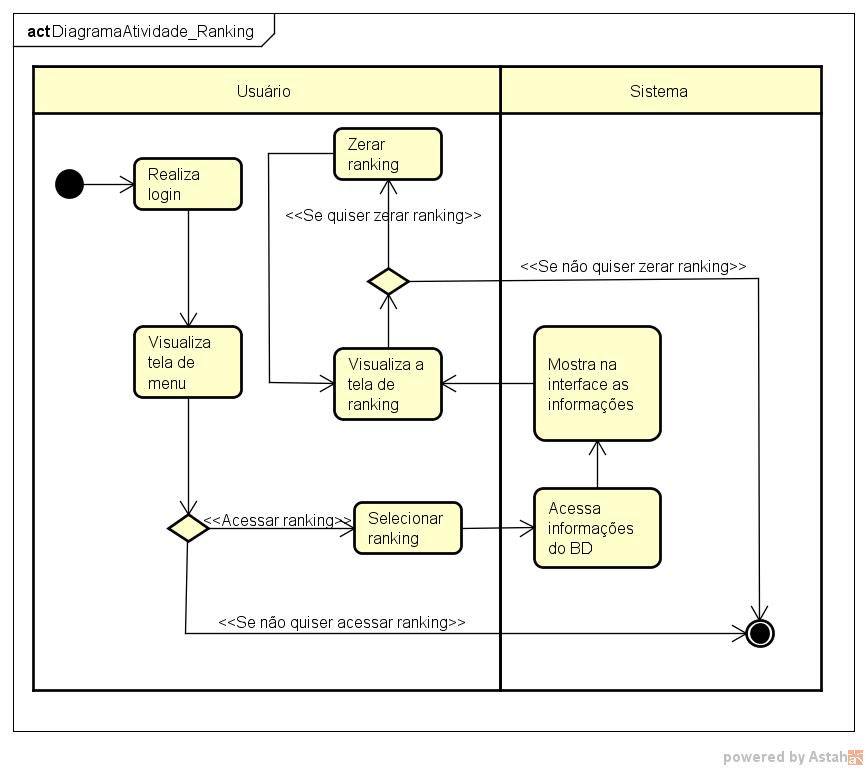
Obs: Torna-se inviável definir os atributos e métodos das classes durante a fase inicial do diagrama, visto que o mesmo sofrerá alterações durante o desenvolvimento.

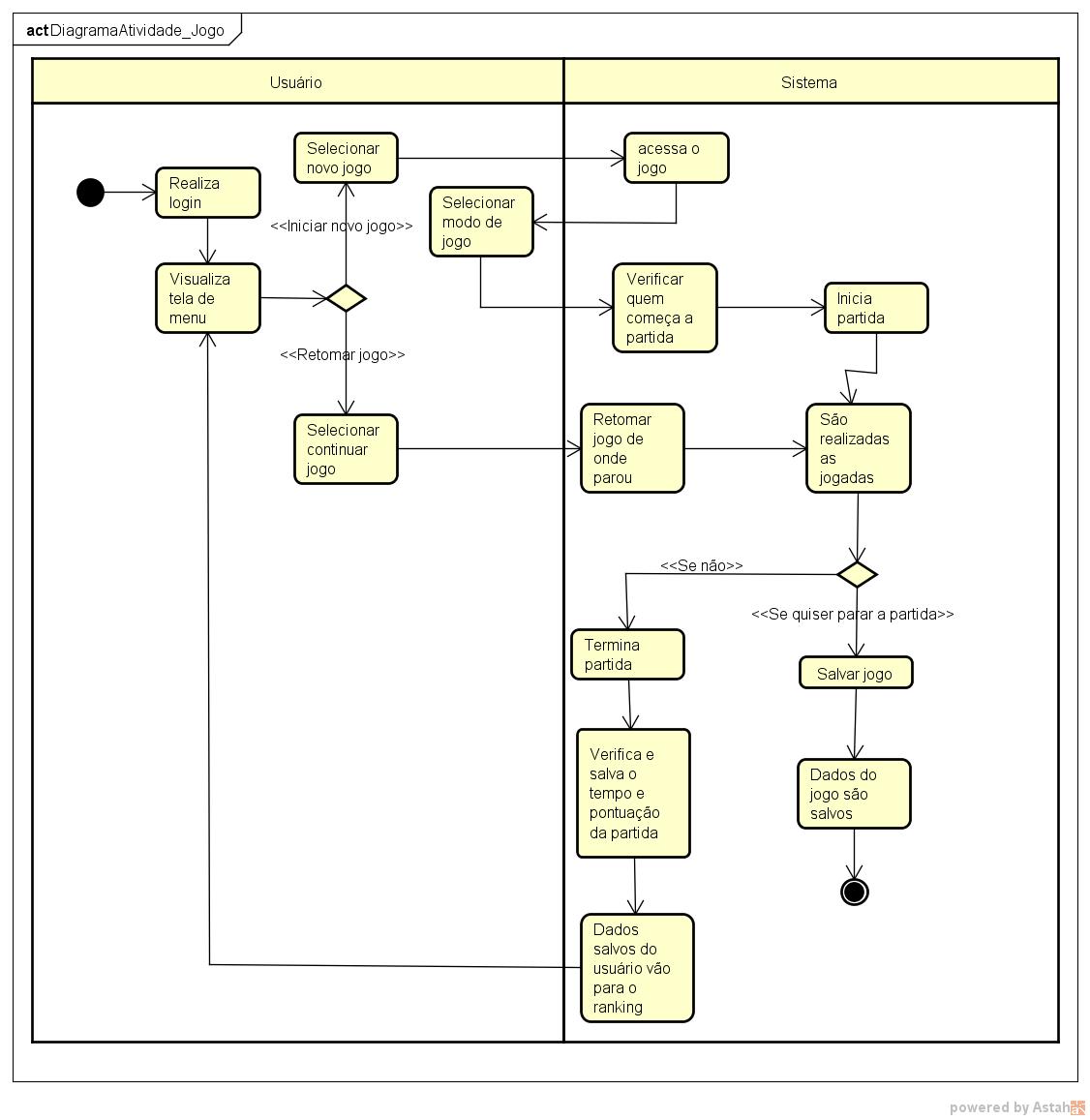
## 

## 

## **Diagramas de Atividades**







## 

## 

## **Projeto Simplificado do BD (Esquema Relacional Diagramático)**

## 

## 

## 

## 

## 

## 

## 

## 

## **Projeto Arquitetural**

# 

O Projeto arquitetural escolhido será o MVC (Model, View, Controller), pois, o mesmo possui uma melhor distribuição de responsabilidades dentre os componentes do sistema, atendendo às necessidades do projeto.

Camadas do MVC:

* A camada **Controller** envia comandos para o modelo para atualizar o seu estado, recebe comandos da **View** evice-versa**.**
* A camada **View** recebe os “inputs” do usuário (Botões, Cliques e etc) e aciona a **Controller** responsável por ela.
* A camada **Model** é responsável pela lógica do sistema, acesso aos dados do banco e modelagem das entidades.

# 

# 

# 

# **Referências**

Engenharia de Software / Ian Sommerville ; tradução Ivan Bosnic e Kalinka G. de O. Gonçalves ; revisão técnica Kechi Hirama. — 9. ed. — São Paulo : Pearson Prentice Hall, 2011. Acesso em 07/11/2017.

AGUIAR, Janderson. Material de Apoio da Disciplina Engenharia de Software. Disponível em:<<https://drive.google.com/drive/folders/0Byo3wMu7LEjOMC03aVlsV0hza3c>>. Acesso em 08/11/2017.