

# Trabalho Prático 2

## Jogo de Baralho - Burro

Édipo Fernandes Vieira de Oliveira - 2011054324

Diego Henrique de Castro Aniceto - 2011054286

Departamento de Ciência da Computação – Universidade Federal de Minas Gerais

18 de maio de 2015

### Resumo:

*Este relatório descreve a implementação do Jogo de baralho Burro. Para que fosse possível essa implemetação foi utilizado a linguagem de programação Java além de teorias de Orientação a Objeto, dentre elas Modularização, Encapsulamento, Herança, Polimorfismo, entre outras. O resultado obtido foi satisfatório, tanto em relação a solução do problema, quanto aos conceitos envolvidos.*

## 1. Introdução

Este trabalho tem como objetivo, observar na prática os principais conceitos da Programação Orientada ao Objeto. Para que estes fossem exercitados foi proposto a implementação de um jogo de baralho. O jogo escolhido é chamado **Burro**, um jogo simples jogado por no mínimo duas pessoas, sem limite máximo de jogadores.

O jogo foi modelado utilizando como base as entidades principais do jogo, *Jogador* e *Baralho*, e a partir destas foram geradas as classes utilizadas para o desenvolvimento do jogo e que também serão descritas nas próximas sessões.

Para que a implementação deste trabalho fosse possível, foram utilizados conceitos de Orientação a Objeto, como herança, polimorfismo, restrição de acesso, entre outros, pois dessa forma o algoritmo cobre os requisitos básicos solicitados e também facilita a compreensão do trabalho.

O trabalho está composto das seguintes seções:

- A seção 2 discute detalhes de implementação e da modelagem do problema.
- A seção 3 traz os testes realizados para verificar a solução do trabalho, bem como a saída gerada
- A seção 4 apresenta uma breve conclusão sobre o trabalho.
- E por fim a seção 5 traz as referências bibliográficas.

## 2. Implementação

### 2.1. Modelagem do Jogo

O jogo foi modelado em torno de duas classes principais, *Jogador* e *Baralho*.

A classe *Jogador* é abstrata, logo contém métodos abstratos e concretos, ela também é a classe base de outras duas, *JogadorHumano* e *JogadorBot* que estendem da super classe. A classe *JogadorHumano* como o próprio nome diz, modela um jogador humano e de acordo com o conceito de herança, ela "É-UM" *Jogador*, assim ela só pode ser um jogador e a classe *JogadorBot* modela um jogador controlado pelo computador e segue o mesmo conceito da classe anterior, porém essa classe não foi completamente implementada.

A classe *Baralho* é uma classe concreta que é composta por *Cartas*, estas por sua vez são compostas de *Naipes* e *Valores*, essas duas últimas, são enumerações, uma forma encontrada para facilitar o desenvolvimento e deixá-lo mais simplificado.

Essa modelagem pode ser visualizada de forma simples no Diagrama de Classes a seguir:

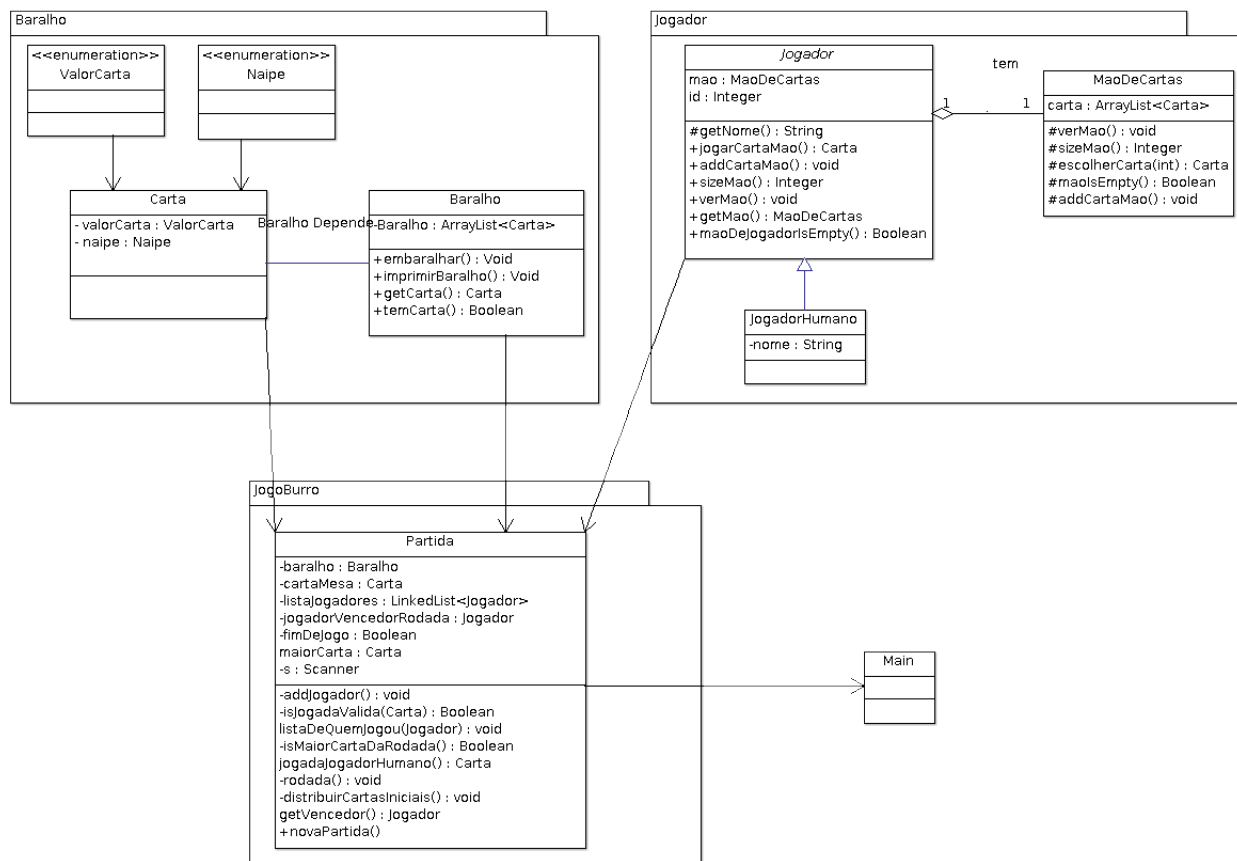


Figura 1: Diagrama de Classes

São necessários no mínimo duas pessoas para jogar, porém não contém um número máximo de jogadores. O jogo se inicia com as pessoas recebendo quatro cartas cada, uma destas deverá jogar uma carta na mesa, as demais deverão jogar uma carta do mesmo naipe daquela que está na mesa, caso não tenham o naipe solicitado em mãos a pessoa deverá comprar cartas no monte de cartas até que encontre o naipe necessário, quem jogar a maior carta será vencedor da jogada, e deverá iniciar a próxima rodada. O vencedor do jogo será aquele que jogar todas as

suas cartas da mão na mesa. O perdedor será aquele que ficar com cartas na mão por ultimo. o numero de cartas que estiver na mão dele serão os 'anos de burrice' da pessoa.

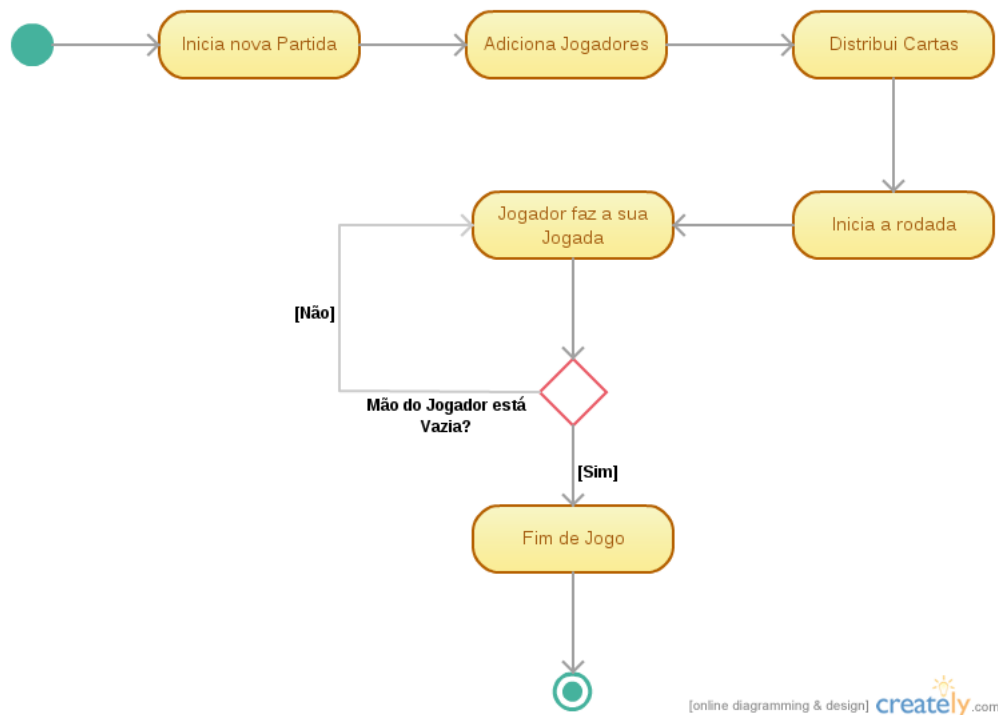


Figura 2: Diagrama de Atividades

## 2.2. Conceitos OO

Foram utilizados para o desenvolvimento do produto final varios conceitos de orientação objeto. Dentre eles, foi utilizado o conceito de **Herança** para especificar a hierarquia de classes envolvendo as classes *Jogador*, *JogadorHumano* e *JogadorBot* onde *Jogador* é a super classe da hierarquia tendo como classes filhas as classes *JogadorHumano* e *JogadorBot*.

Além de herança foi utiizado também foi utilizado o conceito de **Polimorfismo**. Esse conceito pode ser identificado na utilização da sobrescrita do metodo que “pega” o nome do jogodar, sendo ele bot ou humano, ai acontece o Polimorfismo, pois nestas classes eles possuem implementações diferentes.

Ainda passando pelos conceitos de OO, foram utilizados conceitos mais gerais em quase todos os pontos do desenvolvimento, como Encapsulamento de metodos, atributos e classes, composição de classes, manipulação de objetos, entre outros conceitos.

## 2.3. Execução Jogo

## 3. Testes

## 4. Conclusão

## 5. Referências bibliográficas