# Projeto TPG

# Nome: Felipe Augusto Pedroso

## Introdução

Para o projeto proposto na aula de TPG foi escolhido o desenvolvimento de um jogo genérico de cartas, aonde os algoritmos desenvolvidos serão:

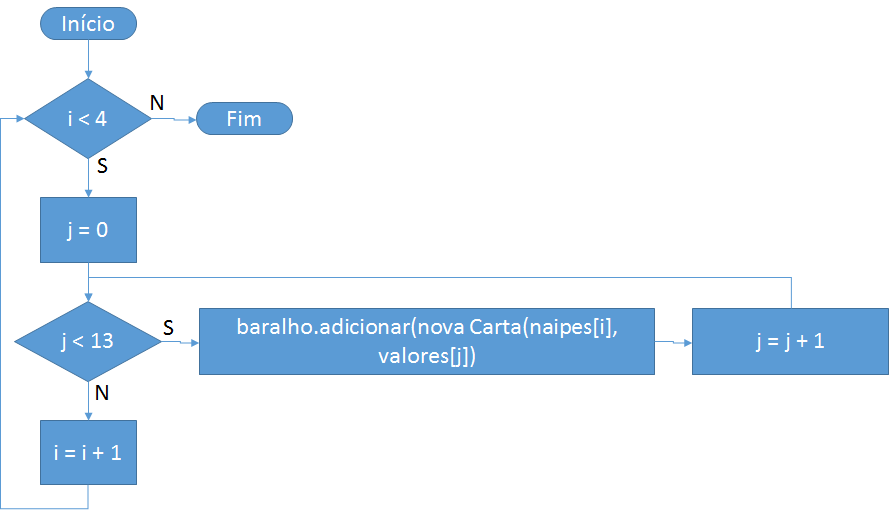
* *CriarBaralho*: que tem como propósito criar um baralho dada a combinação de naipes (Copas, Espadas, Ouros e Paus) e valores (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K) num total de 52 cartas
* *Embaralhar*: aonde um baralho criado pelo algoritmo “CriarBaralho” possa ser embaralhado, fazendo com que suas cartas estejam ordenadas randomicamente.
* *DistribuirCartas*: após a criação e embaralhamento das cartas, este algoritmo irá distribuir uma quantidade n de cartas para uma dada quantidade de jogadores

As implementações dos algoritmos em Java estão disponíveis em anexo ao trabalho e também no repositório de código: <https://github.com/felipepedroso/Projeto-TPG>

## Parte 1 – Algoritmos

### Algoritmo 1 - “CriarBaralho”

* **Fluxograma**



* **Pseudocódigo**

inicio

enumeracao Naipes

enumeracao Valores

var cartas : Lista<Carta>

var carta : Carta

var naipe : Naipes

var valor : Valores

para cada naipe de Naipes

para cada valor de Valores

carta = nova Carta (naipe, valor)

cartas.adicionarCarta(carta)

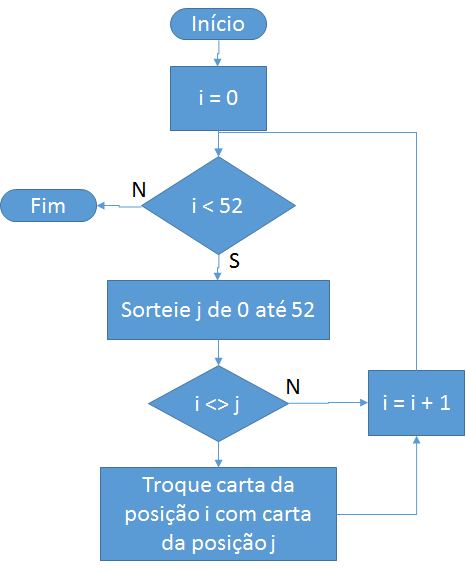
fim para

fim para

fim

### Algoritmo 2 – “Embaralhar”

* **Fluxograma**



* **Pseudocódigo**

inicio

var cartas : Lista<Cartas>

var aux : Carta

var tamanhoBaralho, i, j : inteiro

tamanhoBaralho <- cartas.tamanho()

para i de 0 ate (tamanhoBaralho - 1) passo 1

j <- sorteia (0, tamanhoBaralho - 1)

se i <> j

aux = carta[i]

carta[i] = carta[j]

carta[j] = aux

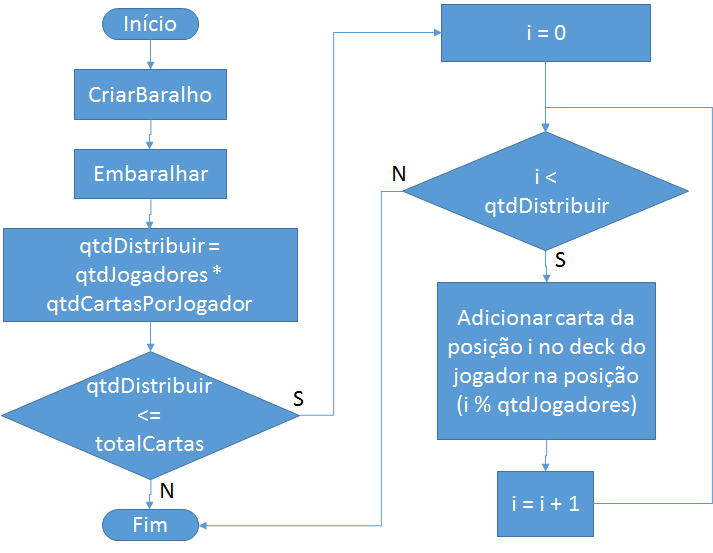
fim se

fim para

fim

### Algoritmo 3 – “DistribuirCartas”

* **Fluxograma**



* **Pseudocódigo**

inicio

constante inteiro qtdJogadores <- 2

constante inteiro qtdCartasPorJogador <- 3

var cartas : Lista<Carta>

var jogadores : Lista<Deck>

var carta : Carta

cartas <- CriarBaralho()

Embaralhar(cartas)

para i de 0 ate qtdJogadores \* qtdCartasPorJogador passo 1

carta <- cartas[i]

deckJogadores[i % qtdJogadores].adicionarCarta(carta)

fim para

fim

## Parte 2 – Extreme Programming

### Definição do Time

* **Project Owner:**
  + Mr. Church
* **Scrum Master**
  + Barney Ross
  + Lee Christmas (backup)
* **Time de Desenvolvimento**
  + Gunner Jensen
  + Yin Yang
  + Hale Caesar
  + Lee Christmas

### Planejamento

O desenvolvimento do produto se dará em 3 Sprints com 2 semanas em cada um. Cada Sprint irá contemplar o desenvolvimento da funcionalidade, documentação e desenvolvimento dos testes. Da a importância dos algoritmos selecionados, foi determinado que o desenvolvimento do produto será na seguinte ordem:

* Sprint 1: Algoritmo de criação do baralho (incluindo a modelagem dos objetos envolvidos)
* Sprint 2: Algoritmo de embaralhamento
* Sprint 3 Algoritmo de distribuição de cartas

### Testes dos três algoritmos

Os testes dos algoritmos estão inclusos no código em anexo.