

# **Projeto Integrador**

## **Estruturas de Dados**

### **Documento de Especificações do Projeto**

**Nome do projeto:** Jogo de cartas

**Nome da equipe:** Naruto

**Integrantes:** [Felipe Fontoura](#), [Amanda Viana](#) e [Leonardo Paiva](#)

# 1. Introdução

Fizemos a escolha do jogo Black Jack com o intuito de criar um jogo que fosse tão fácil e lúdico quanto o real para jogadores de computador. Assim sendo mais fácil de ser jogado e de modo leve, pois não é necessário downloads ou a compra de um baralho. Você pode jogar com múltiplos amigos e se divertir.

## 1.1. Proposta do projeto

O projeto tem como objetivo a formalização do jogo Black Jack em linguagem C.

## 1.2. Objetivos

A equipe teve como objetivo a criação do jogo Black Jack com estratégias aprendidas dentro de sala e com outras técnicas que encontramos por meio de estudos. Entre esses objetivos, além de rodar o jogo perfeitamente, salvando jogadores, rankings, pontuações, tivemos o objetivo de colocar o desenho da carta na tela e rodar o jogo com lista encadeada circular, indicada pelo professor.

## 2. Diagrama de estados do projeto

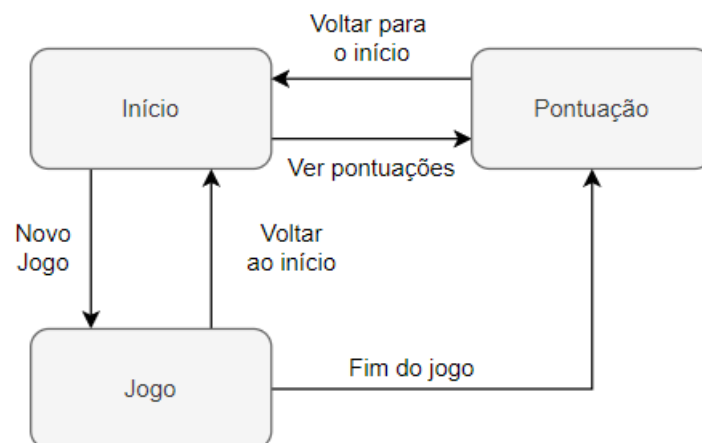


Figura 1: Diagrama de estados do projeto.

### 2.1. Detalhamento dos estados

**ES1 - Início. Descrição:** Apresenta o menu da tela inicial, contendo as opções de Novo Jogo, Acessar Ranking, Zerar Ranking e Sair. **Transições:** É capaz de ir para o estado de pontuação (**ES2**) e para o estado **ES3** ao selecionar Novo Jogo.

**ES2 - Pontuação. Descrição:** Exibe a pontuação do ranking de jogadores, além do ranking dos jogadores de cada rodada. **Transições:** É capaz de voltar apenas para a tela inicial (ES1).

**ES3 Jogo. Descrição:** Realiza a execução da dinâmica do jogo, composto pelas funções de impressão de cartas, IA do *dealer*, e a lógica para os próprios jogadores. **Transições:** É capaz de voltar apenas para a tela inicial (ES1).

### 3. Especificação das estruturas de dados

---

#### ED1 - Lista de Jogadores

Descrição:

```
typedef struct {
    t_player *primeiro;
    t_player *ultimo;
} t_lista;
```

- A estrutura `t_lista` contém dados do primeiro e último jogador na lista, sendo assim, efetivamente uma lista encadeada circular.
- É utilizada na função `criar_jogador`, inserindo o primeiro jogador e o direcionando para o último.

#### ED2 - Jogadores

Descrição:

```
typedef struct Player {
    char nomePlayer[50];
    int ganhou, perdeu, dinheiro, indice;
    struct Player *proximo;
    t_manilha manilha; // lista encadeada de cartas
} t_player;
```

- A estrutura `t_player` contém dados dos jogadores, sendo estes o nome, endereço do próximo jogador, seu índice, quantidade de dinheiro, quantidade de vezes que ganhou/perdeu, e a pontuação da manilha atual do jogador.
- É utilizada na função `dinamica_jogo` para guardar as informações dos jogadores ao longo do jogo.

### ED3 - Cartas (Pilha)

Descrição:

```
typedef struct {
    int value;
    char naipe[10];
    char color[10];
    int is_hidden;
} t_card;
```

- A estrutura `t_card` contém dados possíveis para uma carta, como seu valor, naipe, cor, e se está virada ou não.
- É utilizada na função `print_carta`, para padronizar as cartas a serem impressas na tela.

### ED4 - Manilha (Mão de cartas)

Descrição:

```
typedef struct {
    int pontuacao;
    t_card *inicio;
} t_manilha;
```

- A estrutura `t_manilha` contém dados da estrutura `t_card`, e a pontuação total da manilha.
- É utilizada nas funções `print_carta` e `dinamica_jogo`, para receber a pontuação da carta tirada da pilha, e para a lógica do jogo.

## 4. Conclusões

Ao longo do projeto encontramos diversas dificuldades, como falta de horário juntos, por cada um ter problemas para resolver, saber necessariamente exatamente o que íamos fazer, muitas vezes tínhamos uma ideia mas não conseguíamos colocá-la em prática, pedimos diversas ajudas para o professor pedindo sua opinião sobre partes do nosso projeto, etc. E com o passar do tempo e de outras aulas que ensinavam a própria programação em C, tivemos recursos e ideias suficientes para criar este programa. Não foi uma jornada fácil, foi cheia de desafios, mas em equipe conseguimos realizar.

## 5. Trabalhos futuros

Futuramente, teríamos planos de consertar os bugs presentes atualmente no projeto, implementar mais gráficos para o terminal; além disso, poderíamos implementar partidas através da internet para facilitar que vários jogadores tenham uma boa experiência.

Código: <https://replit.com/@FelpsFon/Grupo-Naurto>