

**Nomes:** Bruno Duarte Barreto Borges,  
Fábio Oliveira de Abreu,  
João Vitor de Souza Costa.

# **Especificação de Requisitos de Software INTERPOL**

Florianópolis, 10 de setembro de 2019

## Índice

1. Introdução .....	3
1.1 Objetivo do desenvolvimento .....	3
1.2 Definição, Abreviaturas .....	3
1.3 Referências .....	3
1.4 Regras do jogo (adaptadas) .....	4
2. Visão Geral .....	4
2.1 Arquitetura da aplicação .....	4
2.2 Premissas de desenvolvimento .....	4
3. Requisitos da Aplicação .....	5
3.1 Requisitos funcionais .....	5
3.2 Requisitos não-funcionais .....	5

## 1. Introdução

### 1.1 Objetivo do desenvolvimento

O projeto que será descrito a seguir é a especificação de requisitos referente ao trabalho semestral da disciplina de Engenharia de Software I do curso de Ciências da Computação. A atividade se resume ao desenvolvimento de um software (jogo) passando por cada uma de várias etapas desse processo. O objetivo é testar e demonstrar todos os nossos conhecimentos adquiridos durante o curso no desenvolvimento e modelagem de um programa, tanto lógico, visual e programático. O jogo decidido para ser o tema deste trabalho foi o famoso jogo de tabuleiro Interpol, produzido e distribuído no Brasil pela Grow, porém em uma versão mais simplificada para ser um jogo rápido e para menos jogadores. No decorrer da introdução, será explicado melhor sobre como é o jogo e como será o processo de desenvolvimento.

### 1.2 Definições, Abreviaturas

\*Abreviação 1: Mister X (bandido) = X

\*Abreviação 2: Policial (agente) = P

\*Ficha 1: Táxi = ficha com valor de movimento 1 (regra adaptada)

\*Ficha 2: Ônibus = ficha com valor de movimento 2 (regra adaptada)

\*Ficha 3: Metrô = ficha com valor de movimento 3 (regra adaptada)

\*Ficha 4: Coringa = ficha com valor de movimento qualquer (regra adaptada)

\*Conceito 1: Turno de revelação = turno onde a posição inicial de X é visível para P.

### 1.3 Referências

As regras do jogo são baseadas nas do jogo original: (link abaixo)

<http://www.gametrack.com.br/jogos/interpol/instrucoes/help.asp>

## 1.4 Regras do jogo (adaptadas)

Para esse projeto foi decidido fazer uma versão mais simplificada do jogo, ao invés de cada casa ter determinadas estações disponíveis, foi decidido que no lugar, cada ficha vai ter uma quantidade de casas de movimento diferente. Os táxis poderão se mover apenas 1 casa em qualquer direção (curvas não permitidas), os ônibus, 1 ou 2, e os metrô, de 1 a 3, de acordo com a escolha do jogador. A carta coringa, que originalmente serve para escolher um método de movimento secretamente, será adaptada para poder se mover quantas casas quiser. O mapa também será reduzido do original e baseado em um tabuleiro matricial, consequentemente diminuindo o número de jogadas. O número de jogador também foi adequado a possuir apenas 2 jogadores, um Mister X e um policial.

## 2. Visão Geral

### 2.1 Arquitetura da aplicação

Programa orientado a objetos de distribuído com base no modelo cliente-servidor

### 2.2 Premissas de desenvolvimento

- Programa deve possuir interface gráfica;
- Linguagem de programação adotada para desenvolvimento deve ser Java;
- A modelagem da especificação deve ser baseada em UML 2;
- O programa deve obrigatoriamente executar distribuído, com o suporte para aplicações distribuídas NetGamesNRT.

## 3. Requisitos da Aplicação

### 3.1 Requisitos funcionais

- Conectar-se ao servidor: o programa deve ter uma opção de menu para que o jogador possa se conectar ao servidor de uma rede de jogadores e disputar partidas;
- Iniciar nova partida: o programa deve ter uma opção de menu para que o jogador possa iniciar uma nova partida do jogo, se já estiver conectado a um servidor e ter pelo menos mais um jogador conectado;
- Desconectar-se do servidor: o programa deve ter uma opção de menu para que o jogador se desconecte do servidor, se estiver conectado a um;
- Receber solicitação de início: recebe uma solicitação do servidor, que sorteia o jogador que será X;
- Reiniciar partida: o programa deve ter a opção de reiniciar a partida. No caso, o programa deve resortear X e as posições iniciais dos jogadores; Durante uma partida os jogadores devem ter a possibilidade de:
- Executar jogada: o programa deve suportar a opção de movimento dos jogadores
- Histórico de movimento de X: os jogadores devem ser capazes de ver um histórico com os meios de movimento de X e as posições dos turnos de revelação para poder manter informações das jogadas do X;

### 3.2 Requisitos não-funcionais

- Interface gráfica: o programa deve ter interface gráfica para cada um dos usuários;
- Especificação de projeto: além do código Java, deve ser produzida especificação de projeto baseada em UML 2, modelada no Visual Paradigm;
- Execução distribuída: o jogo deve executar distribuído, com o suporte para aplicações distribuídas, NetGamesNRT.

### **Esboço da interface gráfica inicial:**

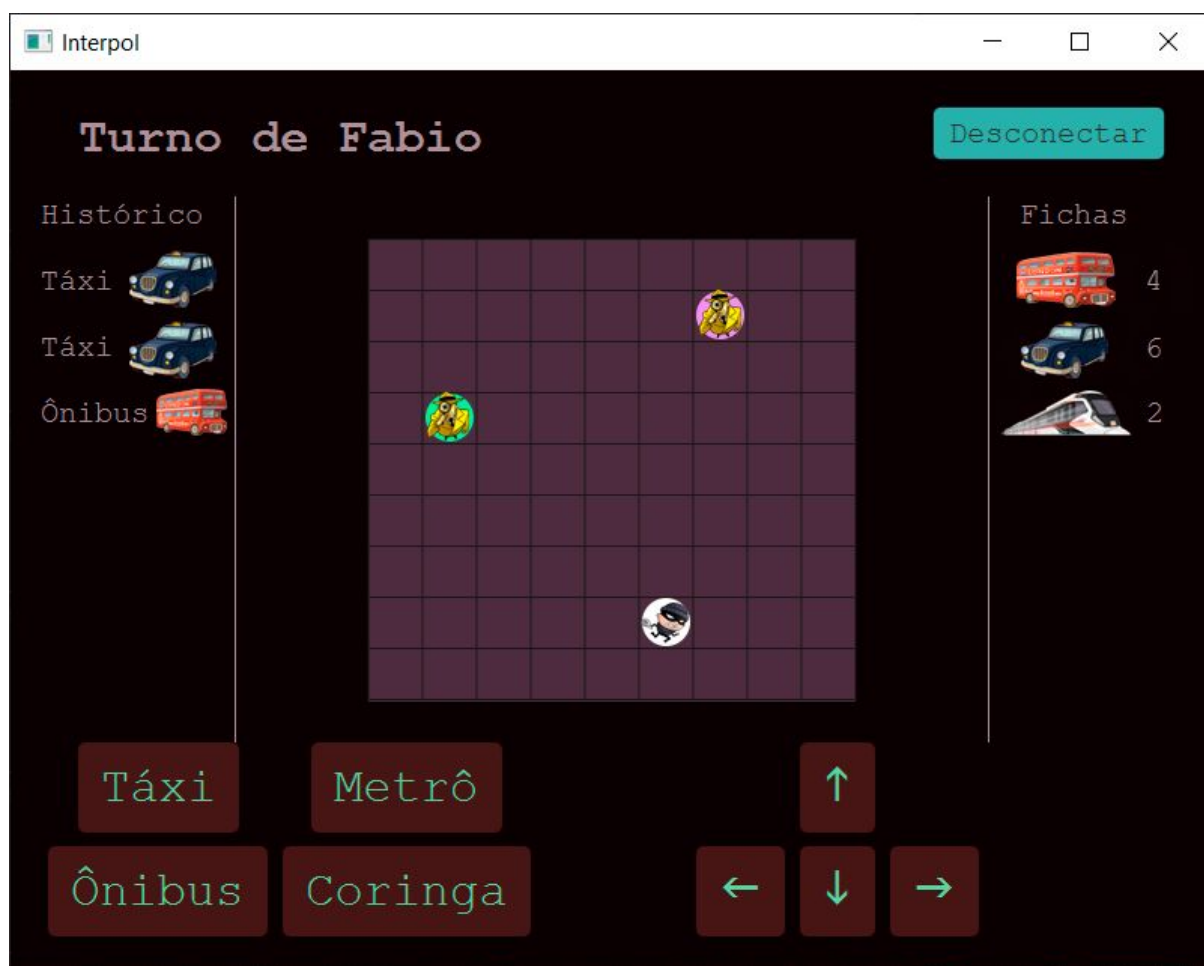


Imagem somente para referência, sujeita a alteração de tabuleiro por mapa.