

# Projeto Truco Gaúcho

## Especificação de Requisitos de Software

Versão 2.0

Versão	Autores	Data	Ação
1.0	Pedro Taglialenha, Vitor Praxedes, Matheus Bigolin	10/04/2024	Estabelecimento dos requisitos
2.0	Pedro Taglialenha, Vitor Praxedes, Matheus Bigolin	30/06/2024	Correção de requisitos

### Conteúdo

<b>1 INTRODUÇÃO.....</b>	<b>2</b>
1.1 OBJETIVOS DO DESENVOLVIMENTO.....	2
1.2 REFERÊNCIAS.....	2
1.3 DESCRIÇÃO E DEFINIÇÕES.....	2
<b>2 VISÃO GERAL.....</b>	<b>3</b>
2.1 ARQUITETURA DO SOFTWARE.....	3
2.2 PREMISSAS DE DESENVOLVIMENTO.....	3
<b>3 REQUISITOS DE SOFTWARE.....</b>	<b>4</b>
3.1 REQUISITOS FUNCIONAIS.....	4
3.2 REQUISITOS NÃO FUNCIONAIS.....	8

# 1 INTRODUÇÃO

Este documento de Especificação de Requisitos tem como objetivo fornecer uma estrutura dos elementos essenciais para o desenvolvimento do jogo Truco Gaúcho. São definidos os requisitos funcionais, que descrevem as funcionalidades específicas que o sistema deve realizar, também é especificado os requisitos não funcionais. Além disso, são apresentadas as premissas fundamentais que orientam o processo de desenvolvimento.

## 1.1 OBJETIVOS DO DESENVOLVIMENTO

Desenvolvimento de um programa que suporte a disputa de partidas de Truco Gaúcho de forma usuário versus usuário, online.

## 1.2 REFERÊNCIAS

As regras escritas do jogo podem ser encontrados nesse link: [Regras do Truco Gaúcho](#) ;

Um vídeo tutorial pode ser encontrado nesse link:

[Truco Gaúcho / Truco Gaudério - Linha Campeira #18](#) .

## 1.3 DESCRIÇÃO E DEFINIÇÕES

Descrição:

- Jogadores - 2.
- Baralhos - um baralho espanhol de 40 cartas, sendo que as cartas 8 e 9 não são utilizadas.
- Distribuição - três cartas para cada participante.
- Objetivo – Fazer 30 pontos.

Definições:

- Envio e Flor - Modos de jogo onde se disputam pontos extras ou uma partida inteira.
- Rodada - uma sequência de 2 jogadas, onde cada jogador joga uma carta.
- Mão - composta de duas a três rodadas, e vale inicialmente 1 ponto.
- Partida - Uma partida vale 30 pontos.
- Truco - proposta inicial para subir o valor da mão (Truco vale 2 pontos, Retruco vale 3 e Vale 4 se disputa 4 pontos ).
- Ordem das cartas (da maior para menor): 1 de espadas, 1 de paus, 7 de espadas, 7 de ouros, todos os 3, todos os 2, 1 de copas e 1 de ouros, todos os 12 (Reis), todos os 11 (Cavalos), todos os 10 (Valetes), 7 de paus e 7 de copas, todos os 6, todos os 5, todos os 4.
- Confronto - disputa em jogo que resulta no futuro ganho de pontos(truco, envio, flor e rodada).

## **2 VISÃO GERAL**

Nessa seção será mostrado a arquitetura do software e as premissas obrigatórias de desenvolvimento do projeto.

### **2.1 ARQUITETURA DO SOFTWARE**

Programa orientado a objetos, cliente-servidor distribuído.

### **2.2 PREMISSAS DE DESENVOLVIMENTO**

O programa deve:

- O programa deve ser implementado em Python;
- O programa deve usar DOG como suporte para execução distribuída;
- Além do código, deve ser produzida especificação de projeto baseada em UML, segunda versão.

### 3 REQUISITOS DE SOFTWARE

#### 3.1 REQUISITOS FUNCIONAIS

- Iniciar programa

Ao ser executado, o programa deve apresentar na interface o jogo em seu estado inicial (todas as cartas no baralho e mão dos jogadores vazias) e solicitar o nome do jogador. Após isso, deve solicitar conexão com DOG Server (utilizando os recursos de DOG). O resultado da tentativa de conexão deve ser informado ao usuário. Apenas em caso de conexão bem sucedida as demais funcionalidades estarão habilitadas. No caso de conexão mal sucedida, a única alternativa deve ser encerrar o programa;

- Começar partida

O programa deve apresentar a opção de menu “iniciar partida” para o início de uma nova partida. O procedimento de início de partida consiste em enviar uma solicitação de início a Dog Server, que retornará o resultado, que será a identificação e a ordem dos jogadores, em caso de êxito, ou a razão da impossibilidade de início de partida, caso contrário. A interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de jogada. Esta funcionalidade só deve estar habilitada se o programa estiver em seu estado inicial, isto é, sem partida em andamento e com o jogo em seu estado inicial

- Iniciar nova mão

O programa deve poder retirar as cartas restantes nas mãos dos jogadores e na mesa de forma a retornarem ao baralho. Em sequência o baralho deve ser embaralhado e as cartas entregues aos jogadores.

- Restaurar estado inicial

O programa deve apresentar a opção de menu “Restaurar estado inicial” para levar o programa ao seu estado inicial, isto é, sem partida em andamento e o jogo em seu estado inicial. Esta funcionalidade só deve estar habilitada se o programa estiver com uma partida finalizada;

- Receber determinação de início

O programa deve poder receber uma notificação de início de partida, originada em Dog Server, em função de solicitação de início de partida por parte de outro jogador conectado ao servidor. O procedimento a partir do recebimento da notificação de início é o mesmo descrito no requisito funcional ‘Iniciar jogo’, isto é, a interface do programa deve ser atualizada com as informações recebidas e caso o jogador local seja quem inicia a partida, a interface deve estar habilitada para seu procedimento de jogada.

- Receber jogada

O programa deve poder receber uma jogada do adversário, enviada por Dog Server, quando for a vez do adversário do jogador local. A jogada recebida deve

corresponder a uma seguintes ações: “Jogar carta”, “Truco”, “Envio”, “Flor”, “Aumentar Valor”, “Correr” ou “Aceitar pedido”. Cada jogada citada anteriormente deve respeitar os requisitos funcionais de seu respectivo nome. Após isso, deve avaliar o encerramento de partida. No caso de encerramento de partida, deve ser notificado o jogador vencedor conforme requisito funcional “Indicar vencedor”; no caso de não encerramento, deve ser habilitado o jogador local, para que possa proceder a sua jogada;

- Receber notificação de abandono

O programa deve poder receber uma notificação de abandono de partida por parte do adversário remoto, enviada pelo DOG Server. Neste caso, a partida deve ser considerada encerrada e o abandono notificado na interface.

- Indicar vencedor

Ao final do jogo deve ser apresentado tanto ao jogador local, quanto ao remoto uma tela indicando se o jogador azul ou vermelho ganhou a partida junto com a pontuação final de cada um dos jogadores.

- Jogar carta

O jogador em seu turno pode escolher uma de suas três cartas e jogá-la na mesa ao clicar na carta escolhida. Caso o jogador opte por “Jogar carta” e já tenha uma carta na mesa na rodada atual, o programa deve também avaliar o vencedor da rodada conforme descrito no requisito funcional “Avaliar vencedor”. Ao jogar uma carta o programa deve enviar a jogada ao adversário utilizando os recurso de DOG, e o turno deve ser passado para o oponente. A jogada enviada ao oponente deve conter a carta selecionada pelo jogador.

- Avaliar vencedor

O programa deve verificar as cartas e atribuir o vencedor entre os jogadores dependendo do confronto e atribuir os pontos da rodada ou confronto ao jogador vencedor.

- Pedir Truco

O jogador, em seu turno antes de jogar uma carta, deve poder pedir truco. Quando o truco é pedido, será enviado para o oponente um popup que irá conter três botões: Aceitar, Correr, Retruco. Se aceite, a mão passa a valer 2 pontos para o vencedor, o turno deve continuar normalmente. Se correr, o jogador que pediu o truco deve ganhar 1 ponto e será iniciada uma nova mão.

Se pedir Retruco, deve ser enviado ao jogador que pediu o truco um popup com 3 botões: Aceitar, Correr, Vale 4. Caso aceite, a mão começa a valer 3 pontos e continua normalmente. Se correr, o jogador que pediu retruco deve ganhar 2 pontos e será iniciada uma nova mão.

Se pedir Vale 4, deve ser enviado um popup para o oponente um popup com duas opções: Aceitar, correr. Caso aceite, a partida começa a valer 4 pontos e continua normalmente. Se correr, o jogador que pediu o vale 4 ganha 3 pontos e será iniciado uma nova mão.

- Pedir Envio

O jogador deve poder pedir Envio uma vez por mão. Ao ser pedido Envio, o oponente deve receber um popup com 3 botões: Aceitar, Correr, Real Envio. Caso aceite, o sistema deve contar os pontos das cartas na mão de cada jogador e da mesa caso o adversário já tenha jogado uma carta, e atribuir 2 pontos para o que tiver maior pontuação. Caso corra, quem pediu o Envio ganha 1 ponto e a partida continua normalmente. Se pedir Real Envio, deve ser enviado a quem pediu o Envio um popup com 3 opções: Aceitar, Correr, Falta Envio. Se aceitar, será feita a avaliação de pontos do envio e entregue 5 pontos ao vencedor, o jogo continua normalmente. Caso corra, o jogador que pediu o Real Envio ganha 2 pontos e o jogo continua normalmente. Se pedir Falta Envio, o jogador que pediu Real Envio deve receber um popup com duas opções: Aceitar, correr. Caso aceite, será feita a avaliação do envio, mas o vencedor do envio receberá os pontos restantes para finalizar a partida (30 - maior pontuação do placar). Caso corra, o jogador que pediu o Falta Envio recebe 5 pontos e o jogo continua normalmente.

O jogador só pode aceitar se cumprir o requisito de ter 2 cartas do mesmo naipe, contando com a carta na mesa. Mas pode blefar um aumento.

- Pedir Flor

O jogador pode pedir Flor uma vez por mão. Não pode mais ser pedido envio na rodada, e é enviado ao oponente um popup com 3 botões: Aceitar, Correr, Contra-Flor. Caso aceitado, é realizada a disputa da flor, onde o sistema irá contar os pontos das cartas da mão dos jogadores e na mesa caso já tenha jogado uma carta, e atribuir ao vencedor 4 pontos. Se correr, o jogador que pediu flor ganha 3 pontos. Se pedir Contra-Flor, será enviado ao jogador que pediu flor um popup com 3 opções: Aceitar, Correr, Contra-Flor-e-o-Resto. Se aceitar é feita a avaliação do vencedor da flor e entregue 6 pontos ao vencedor. Se correr é entregue 4 pontos a quem pediu o Contra-Flor. Se pedir Contra-Flor-e-o-Resto, é apresentado ao jogador que pediu Contra-Flor um popup com 2 opções: Aceitar, Correr. Se aceitar, é feita a avaliação do vencedor da flor e entregue os pontos restantes para finalizar a partida (30 - maior pontuação do placar). Se correr é entregue a quem pediu Contra-Flor-e-o-Resto 6 pontos.

- Enviar popup

Quando um jogador pedir Truco, Flor ou Envio, deve ser enviado para o seu oponente um popup específico para cada confronto (ver requisitos funcionais “Truco”, “Flor” e “Envio” para mais detalhes). Generalizando, o botão aceitar deve iniciar o confronto e fechar o popup. O botão Correr deve negar o confronto e fechar o popup. O botão Aumentar valor, que o texto depende do confronto e do nível do aumento, deve fechar o popup e mandar para o oponente o popup do confronto aumentado.

- Aceitar Pedido

O jogador terá a opção de aceitar um pedido quando receber o popup de um confronto pedido pelo jogador adversário, sendo eles: Truco, Flor e Envido. Aceitar terá um comportamento específico para cada confronto (ver requisitos funcionais “Truco”, “Flor” e “Envido” para mais detalhes). Caso o confronto em questão corresponda a “Flor” ou “Envido” o programa deve avaliar vencedor conforme descrito no requisito funcional “Avaliar Vencedor”.

- Correr

O jogador deve poder pedir Correr em qualquer momento em seu turno. Se o jogador correr, o oponente deve receber os pontos do confronto. Se o confronto for truco ou rodada, é iniciada uma nova mão.

- Aumentar Valor

O jogador deve ser capaz de optar por Aumentar Valor ao receber um popup de confronto (ver requisito funcional “Enviar popup”). Cada jogador pode aumentar o valor do confronto uma vez (ver requisitos funcionais “Truco”, “Flor” e “Envido” para mais detalhes).

## 3.2 REQUISITOS NÃO FUNCIONAIS

- Versão do Python

O programa será desenvolvido na linguagem Python versão 3.10.

- IDE (ambiente de desenvolvimento integrado)

A interface de desenvolvimento escolhida para programar o jogo será o Pycharm e o VSCode.

- Suporte para especificação do projeto

A especificação de projeto deve ser produzida com a ferramenta Visual Paradigm.

- Bibliotecas Python

Será usado a biblioteca gráfica Tkinter, por ser a ferramenta padrão do Python para interface de usuário. Além disso, a biblioteca PIL (Python Imaging Library) é usada para proporcionar um melhor suporte para a manipulação de imagens, especialmente devido a seu suporte à antisserrilhamento (*anti-aliasing*) ao redimensionar as imagens das cartas.

- Interface gráfica

Os principais elementos da interface gráfica do jogo estão presentes nas figuras 1 e 2.

Figura 1 - Interface do jogo.



Fonte: Elaborado pelos autores



Figura 2 - Interface do jogo com popup e menu aberto.



Fonte: Elaborado pelos autores