

Documentação Técnica — Jogo da Memória NFL com WebSockets

Ana Laura da Silva Nascimento
Gabriel Ferreira de Sousa
Heitor Ceolin Ribeiro

26 de agosto de 2025

1 Instruções de Execução

1.1 Requisitos

- Python 3.8 ou superior
- Bibliotecas:
 - `customtkinter`, `pillow`, `tkinter`
 - `asyncio`, `websockets`

1.2 Execução — Interface Gráfica

1. Certifique-se de ter o Python 3.8+ instalado em sua máquina.
2. Instale as dependências com:

```
pip install customtkinter pillow tk
```
3. Extraia o conteúdo do arquivo `Jogo_NFL.zip`.
4. Navegue até o diretório do projeto.
5. Execute:

```
python jogo.py
```

6. A interface será exibida em tela cheia (pressione `ESC` para sair).

1.3 Execução — Terminal Multiplayer

O jogo também pode ser jogado no terminal em modo multiplayer.

Servidor:

```
python servidor.py
```

Cliente (Linux):

```
sudo apt install python3-venv
python3 -m venv venv
source venv/bin/activate
python jogo.py
```

2 IPs e Portas Utilizados para Testes

- IP: 127.0.0.1 (localhost)
- Porta: 8765
- URI: ws://127.0.0.1:8765

3 Protocolo de Aplicação

3.1 Tipo de Protocolo

WebSocket, com mensagens de texto simples enviadas entre cliente e servidor.

3.2 Exemplo de Mensagens

```
# Cliente solicita virar carta na posição (2,3):
TURN 2 3

# Servidor confirma jogada para todos os clientes:
CARD 2 3 17

# Servidor atualiza placar:
SCORE 1 0

# Servidor anuncia fim de jogo:
WINNER 1
END
```

3.3 Explicação

- O comando `TURN r c` indica uma jogada.
- O servidor responde com `CARD r c valor`, mostrando a carta.
- Se houver par, envia `MATCH`, caso contrário, `NO_MATCH`.
- O placar é atualizado com `SCORE`.
- Ao final, é enviada a mensagem `WINNER` seguida de `END`.

4 Logs de Sessões de Teste

4.1 Servidor

```
Servidor rodando em ws://127.0.0.1:8765  
Jogador 1 conectado.  
Jogador 2 conectado.
```

4.2 Troca de Mensagens

```
Jogador 1 envia: TURN 2 3  
Servidor responde: CARD 2 3 17  
Servidor responde: BOARD [...]  
Servidor responde: SCORE 1 0
```

5 Atualizações no Protocolo

Até o momento, não foram necessárias alterações no protocolo de aplicação. O modelo atual com comandos textuais mostrou-se suficiente para a jogabilidade.

6 Arquivos do Projeto

- `servidor.py` — servidor WebSocket que gerencia regras e turnos.
- `jogo.py` — cliente/interface gráfica com suporte a multiplayer.
- `imagens/` — pasta com cartas e verso.