

Documentação Técnica — Jogo da Memória NFL com WebSockets

Ana Laura, Heitor Ceolin e Gabriel Ferreira

4 de agosto de 2025

1. Instruções de Execução

1.1 Requisitos

- Python 3.11 ou superior
- Biblioteca `websockets`:

```
pip install websockets
```

1.2 Execução

Abra dois terminais:

- **Terminal 1 — Servidor WebSocket**

```
python servidor.py
```

Saída esperada:

```
Servidor WebSocket rodando em ws://127.0.0.1:8765
```

- **Terminal 2 — Cliente (Jogo)**

```
python jogo.py
```

Observações:

- Pressione `ESC` para sair do modo tela cheia.
- Um segundo jogador pode iniciar o mesmo `jogo.py` em outro terminal ou computador na mesma rede.

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 no formato JSON.

3.2 Mensagem Enviada

```
{  
  "action": "reveal",  
  "index": 15  
}
```

3.3 Mensagem Recebida

```
{  
  "action": "reveal",  
  "index": 15  
}
```

3.4 Explicação

- A ação "reveal" indica que uma carta foi virada.
- O campo "index" representa o índice da carta (0 a 47).
- O servidor retransmite a mensagem para os outros jogadores conectados.

4. Logs de Sessões de Teste

4.1 Saída do Servidor

```
Servidor WebSocket rodando em ws://127.0.0.1:8765
```

4.2 Evento de Jogo

- Jogador 1 clica em uma carta de índice 12.
- Enviado ao servidor:

```
{"action": "reveal", "index": 12}
```

- Servidor retransmite para o jogador 2.
- Jogador 2 recebe:

```
{"action": "reveal", "index": 12}
```

5. Atualizações no Protocolo (se aplicável)

Nenhuma alteração foi necessária no protocolo após o TP2. O modelo atual se mostrou funcional para o propósito do jogo.

6. Arquivos Enviados

- `jogo.py` — cliente com interface gráfica e lógica de jogo.
- `servidor.py` — servidor WebSocket simples.
- `pontuacoes.txt` — registro dos vencedores e pontuações.