Arquivo REDME

**Integrantes:** Kaick Eugenio Reis; Víctor Bruno Costa Evangelista.

### Instruções de Execução

Para executar o jogo, uma pasta de nome "public" deve estar no mesmo diretório onde se localiza o arquivo do servidor, chamado de server.js. Depois disso, o terminal deve ser aberto e o diretório do projeto acessado, em seguida, o comando "npm init -y" deve ser executado para criar um arquivo package.json. A instalação das dependências necessárias é recomendada à ser feita junto da instalação do node, ao permitir o download de quaisquer dependências que estiverem em falta. Após as dependências instaladas, o servidor será iniciado ao executar o comando "node server.js", assim, o endereço "http://localhost:3000" ficará disponível. Nesse momento, será necessário abrir o navegador em duas abas ou dois navegadores diferentes, sendo cada um os jogadores 1 e 2 respectivamente. A partir daí, o jogo estará pronto para ser jogado, tendo uma comunicação em tempo real e funcionando entre os dois jogadores por meio do WebSocket, que transmite os movimentos a medida que cada um realiza sua jogada.

#### Ferramentas Utilizadas

Uma ferramenta muito essencial para o desenvolvimento desse projeto foi o Node.js, sendo o responsável por rodar o servidor backend, enquanto as bibliotecas express e ws permitem servir os arquivos estáticos da aplicação e gerenciar a comunicação em tempo real via WebSocket entre os jogadores. O código foi promadado pelo Visual Studio Code. No lado do cliente, o jogo funciona diretamente em um navegador web, que é responsável por renderizar a interface gráfica construida à partir de HTML e CSS, e que teve também a utilização de JavaScript. O terminal é utilizado para executar o servidor e instalar algumas dependências do projeto. Por último, o padrão WebSocket permitiu a comunicação bidirecional entre o cliente e o servidor, garantindo a troca de mensagens e a sincronização das jogadas em tempo real.

### Funcionalidades Implementadas

O código se trata de um jogo de damas para dois jogadores com comunicação por meio do "WebSocket". O tabuleiro é criado de forma dinâmica, posicionando as peças conforme as regras do jogo. Os players podem selecionar e mover suas peças clicando nas casas e realizando capturas. O jogo controla os turnos de cada jogador e atualiza o placar sempre que é feito uma nova captura de peça. A comunicação entre os dois navegadores é feita por meio de um servidor "Node,js" com "WebSocket", responsável por receber e repassar as jogadas feitas. Tem também a presença de um botão que permite reiniciar a partida a qualquer momento.