

## Documentação – Jogo Batalha Naval

Este projeto é uma versão simples do jogo Batalha Naval desenvolvida em Python. A partida é disputada entre o jogador e a máquina, onde o objetivo é afundar todos os navios do adversário antes que os próprios navios sejam destruídos.

O tabuleiro é representado por uma matriz de 10 por 10, preenchida inicialmente com o símbolo “~” representando a água. Em cada tabuleiro são posicionados automaticamente cinco tipos de navios, de tamanhos diferentes, que ocupam posições aleatórias. Os navios são: Porta-aviões (4 espaços), Submarino (1 espaço), Cruzador (2 espaços), Destroyer (5 espaços) e Guardião dos Mares (3 espaços). O código garante que eles não fiquem sobrepostos.

Durante a partida, o jogador escolhe uma posição do tabuleiro inimigo informando linha e coluna. Se o ataque acertar parte de um navio, o local é marcado com “X”. Se for um erro, é marcado com “O”. A máquina, por sua vez, realiza ataques de forma aleatória no tabuleiro do jogador, também respeitando as mesmas regras.

O jogo acompanha em tempo real os acertos e informa quando um navio é totalmente afundado. A partida termina quando todos os navios de um dos jogadores forem destruídos. No final, o sistema mostra quem foi o vencedor e pergunta se o jogador deseja iniciar uma nova partida.

De forma geral, o código está dividido em funções para criar o tabuleiro, mostrar as jogadas, posicionar os navios, realizar ataques e verificar a vitória. A função principal, chamada `jogar_batalha_naval`, organiza todo o fluxo da partida, desde a apresentação das regras até a verificação do resultado final.

## Funcionalidades

**Criação do tabuleiro:** O jogo gera um tabuleiro quadrado (10x10) preenchido inicialmente com “~”.

**Posicionamento aleatório de navios:** Os navios são colocados aleatoriamente de forma horizontal ou vertical no tabuleiro. O código garante que não haja sobreposição de navios.

### Tipos de Navios:

- (P) Porta-aviões → 4 espaços;
- (S) Submarino → 1 espaço;
- (C) Cruzador → 2 espaços;
- (D) Destroyer → 5 espaços;
- (G) Guardião dos Mares → 3 espaços;

### Ataques

- O jogador escolhe uma coordenada (linha e coluna) para atacar.
- A máquina realiza ataques aleatórios no tabuleiro do jogador.
- Se houver acerto, a posição é marcada com “X”.
- Se houver erro, a posição é marcada com “O”.

### Verificação de navios afundados

- O jogo verifica automaticamente quando um navio foi totalmente destruído.
- Mensagens informam o jogador sobre quais navios foram afundados.

### Condição de Vitória

O jogo termina quando todos os navios de um dos lados forem afundados.

### Reinício

Ao final da partida, o jogador pode escolher jogar novamente ou encerrar o jogo.

## Estrutura do Código

- **criar\_tabuleiro(tamanho)** → Cria a matriz do tabuleiro.
- **mostrar\_tabuleiro(tabuleiro, ocultar=False)** → Exibe o tabuleiro (com a opção de ocultar as informações do tabuleiro ou não).
- **colocar\_navios(tabuleiro)** → Posiciona navios aleatoriamente sem sobreposição.
- **ataque\_jogador(tabuleiro)** → Controla o ataque do jogador, com validações.
- **ataque\_maquina(tabuleiro)** → Gera ataques aleatórios da máquina.
- **verificar\_navios\_afundados(tabuleiro, navios\_afundados, navios\_restantes, quem)** → Checa se navios foram destruídos, essa função também é utilizada para verificar vitória.
- **jogar\_batalhanaval()** → Função principal que organiza o fluxo do jogo.

### Alunos:

Isabela dos Santos Pinto – **RM:** 563422

Manuela de Lacerda Soares – **RM:** 564887