

# Case de Avaliação

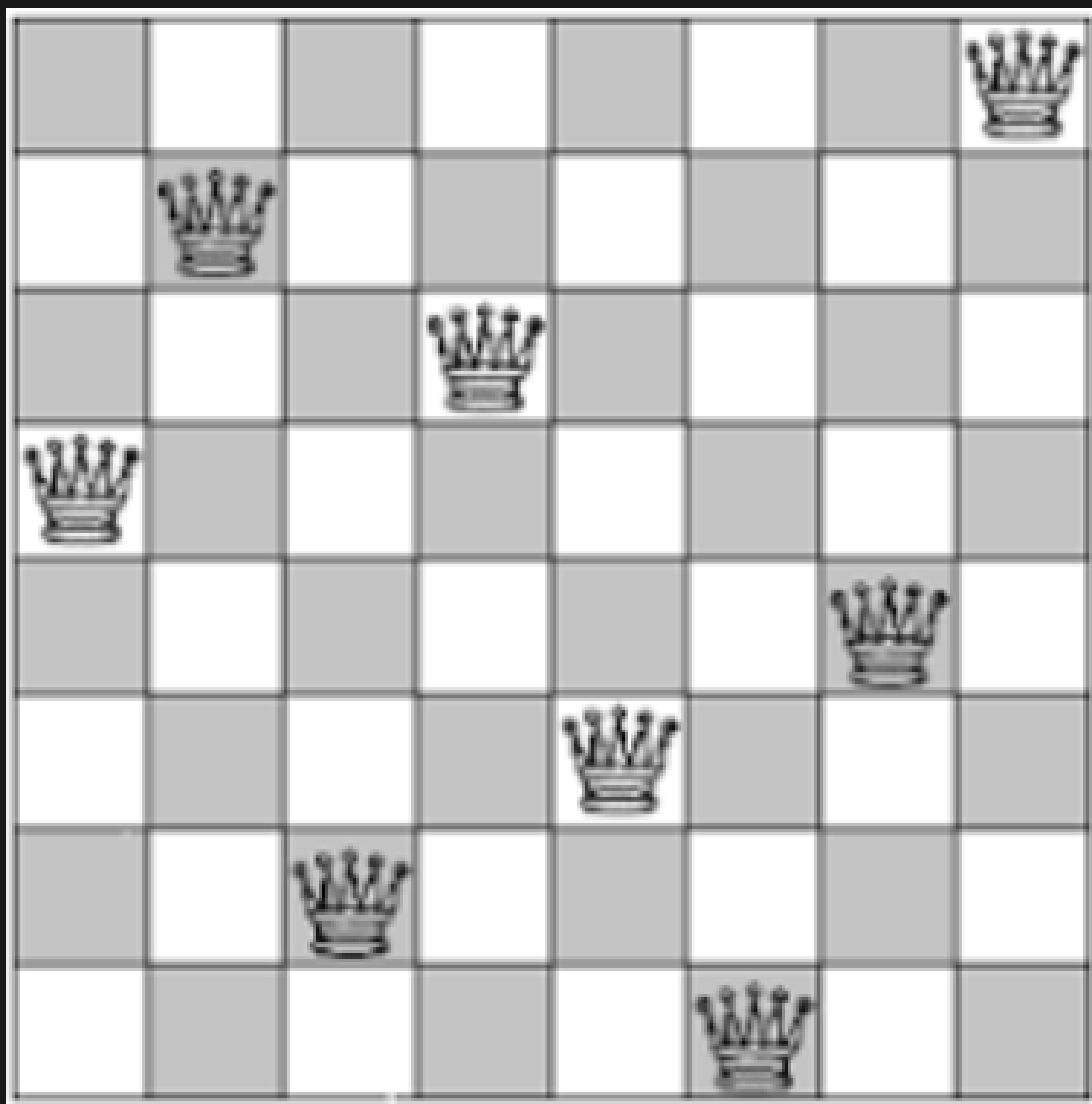
Python



# Cases

- Desafio das Oito Rainhas
- Sudoku

# Case: Desafio das Oito Rainhas



O desafio consiste em quebra-cabeça lógico que envolve posicionar oito rainhas em um tabuleiro de xadrez 8x8 de forma que nenhuma delas possa atacar outra.



# Solução

## Backtracking

- Encontrar todas as soluções válidas para o problema das N Rainhas e exibir visualmente no tabuleiro.

## Exibição na interface gráfica

- Exibição a solução de problemas com o Tkinter.





# Comentários do Código

## **is\_safe(board, row, col)**

- Verifica se é seguro colocar uma rainha em uma determinada posição do tabuleiro.

## **draw\_queens(canvas, n, board)**

- Desenha o tabuleiro e as rainhas no canvas da interface gráfica.

## **solve\_n\_queens(n, board, col, solutions)**

- Utiliza o algoritmo de backtracking para encontrar todas as soluções válidas para o problema.

## **n\_queens(n)**

- Encontra todas as soluções possíveis para o problema das N Rainhas.

# Dificuldades



**Case complexo**

**Visualização para  
usuários**

**A rainha ser uma  
grande peça no jogo**

[Voltar ao índice](#)

# Case: Sudoku

3			8		1			2
2		1		3		6		4
			2		4			
8		9				1		6
	6						5	
7		2				4		9
			5		9			
9		4		8		7		5
6			1		7			3

O Sudoku é um quebra-cabeça numérico amplamente conhecido que demanda o preenchimento de espaços em branco em uma grade 9x9 com dígitos de 1 a 9, garantindo que cada coluna, linha e subgrade de 3x3 contenha todos os números de forma única.



# Solução

## Backtracking

- Encontrar todas as soluções válidas para o problema das N Rainhas e exibir visualmente no tabuleiro.

## Biblioteca NumPy

- Empregada para organizar os números do Sudoku em uma matriz, facilitando a manipulação e resolução do quebra-cabeça.







# Comentários do Código

## **isSafe(grid, row, col, num)**

- Verifica se é seguro inserir um número (num) em uma posição específica (row, col) no grid do Sudoku.

## **solveSudoku(grid)**

- Implementa o algoritmo de resolução do Sudoku usando backtracking para preencher o grid fornecido..

## **printGrid(grid)**

- Imprime o grid do Sudoku na tela de uma maneira formatada.

# Dificuldades



**Quantidade de  
números**

**As regras especiais do  
Sudoku**

**Complexidade em  
espaços vazios**



**Obrigada pela atenção!**

