

SENACK

ANALISE DE ALGORITMOS E COMPLEXIDADE EM UMA APLICAÇÃO WEB

AUTOR: Antonio Carlos Sena | DATA: 2025



OBJETIVO DO PROJETO

- > Desenvolver uma aplicação web completa Jogo da Cobrinha para aplicar conceitos teóricos.
- > Implementar e analisar **Estruturas de Dados** (Filas e Arrays).
- > Aplicar **Algoritmos de Ordenação** (QuickSort) e Busca.
- > Analisar a **Complexidade Assintótica** (Big O) de cada funcionalidade.

SENACK - Jogo da Cobrinha



TECNOLOGIAS UTILIZADAS



FRONTEND

HTML5, CSS3 (Pixel Art), JavaScript Puro (Vanilla JS) para lógica do jogo.



BACKEND

Node.js com Express.
API RESTful para gerenciamento de pontuação e ranking.



DEPLOY

Vercel. Integração contínua via GitHub para atualizações automáticas.

ESTRUTURAS DE DADOS

FILA (QUEUE)

Utilizada para representar a **Cobra**.

- > **Movimento:** ``unshift()`` (Adiciona Cabeça).
- > **Remoção:** ``pop()`` (Remove Cauda).
- > **Complexidade:** Simula perfeitamente o movimento deslizando mantendo o tamanho.

ARRAY (LISTA)

Utilizado para armazenar o **Ranking**.

- > Armazena objetos ``{ nome, score }``.
- > Facilita a iteração para exibição no frontend.
- > Estrutura base para a aplicação do algoritmo de ordenação.

ALGORITMOS: QUICKSORT

DIVISÃO E CONQUISTA

Implementado manualmente no backend para ordenar o ranking.

```
const pivot = arr[Math.floor(arr.length / 2)];  
// Particiona em Left, Right, Equal
```

- > **Escolha:** Eficiente para casos médios.
- > **Recursividade:** Função chama a si mesma para sub-arrays menores.
- > **Equação:** $T(n) = 2T(n/2) + O(n)$

ARRAY ORIGINAL



↓ Pivô: 5

MENORES



MAIORES



ANALISE DE COMPLEXIDADE

FUNCIONALIDADE	COMPLEXIDADE (Time)	JUSTIFICATIVA
POST /scores (Inserir)	$O(1)$	Inserção direta no final do array (push).
GET /ranking (Ordenar)	$O(n \log n)$	Uso do QuickSort (caso médio).
DELETE /ranking (Reset)	$O(1)$	Reatribuição de array vazio.
Colisão (Frontend)	$O(k)$	Busca linear onde k é o tamanho da cobra.

DEMONSTRAÇÃO

ACESSE O JOGO AO VIVO:

[SENACK.VERCEL.APP](https://senack.vercel.app)

Fim!

OBRIGADO!

