

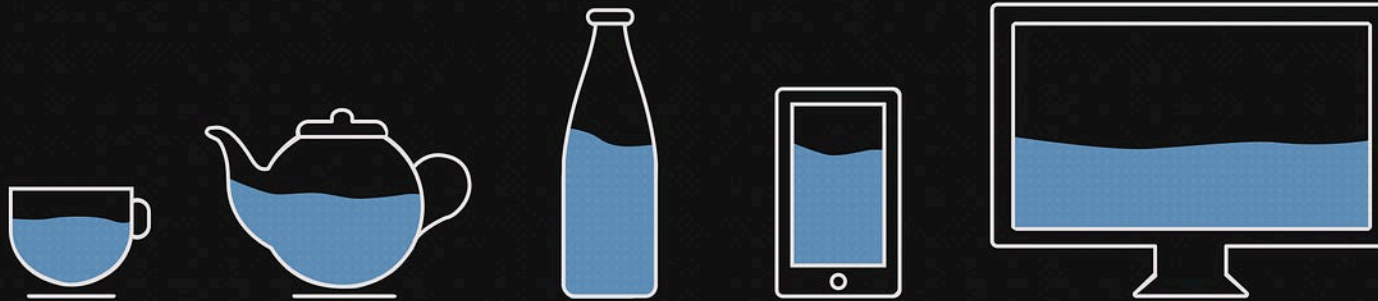
TÓPICO 01: FUNDAMENTOS DE ESTRUTURA DE DADOS

Estrutura de Dados - Professor Ramon Venson - SATC 2026.2

Processamento de dados de um computador



CONTENT IS LIKE WATER



“ You put water into a cup it becomes **the cup**.
You put water into a bottle it becomes **the bottle**.
You put it in a teapot, it becomes **the teapot**. ”

Josh Clark (originally Bruce Lee) - Seven deadly mobile myths

Illustration by Stéphanie Walter

Estruturas de Dados

Estrutura de dados é a forma que organizamos e armazenamos dados em um computador para que possamos **manipulá-los de maneira eficiente**.

Tópico 01 - Fundamentos de Estrutura de Dados



Estrutura de Dados - Professor Ramon Venson - SATC 2026.2

1E : Necromancy

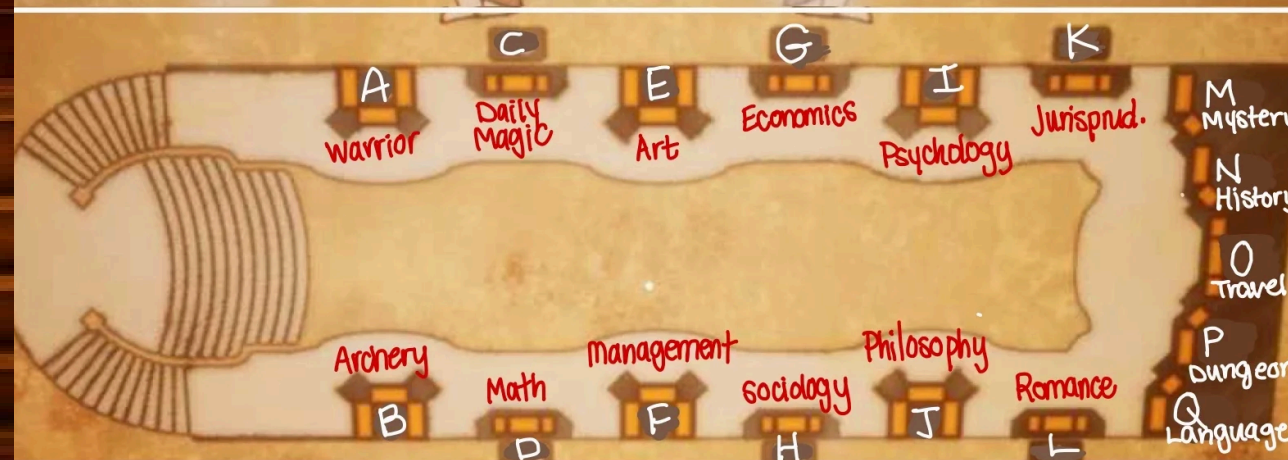
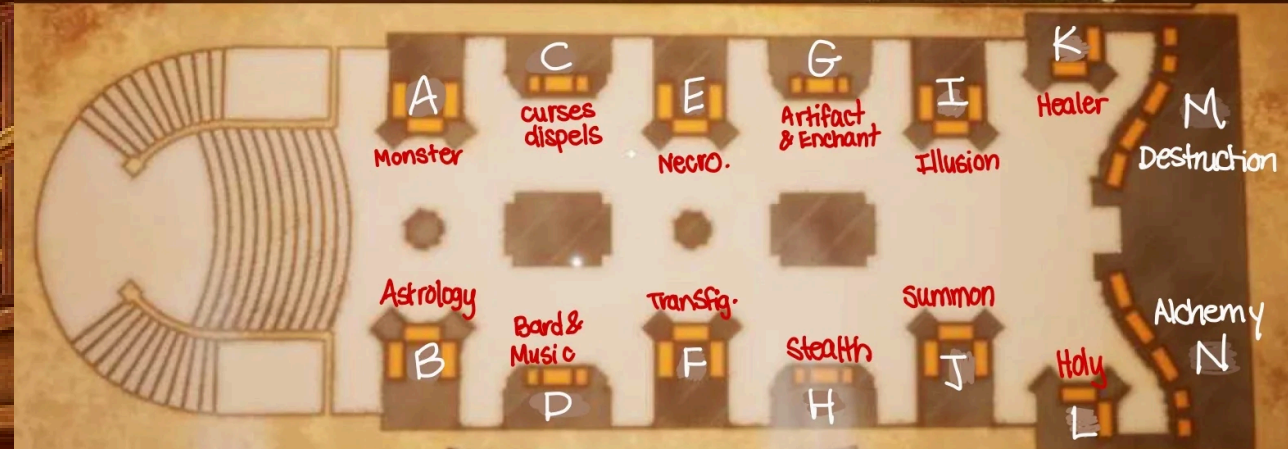
1F : Transfiguration

1G : Magical Artifacts and Enchanting

1L : Holy Magic

1M : Destruction Magic

1N : Alchemy and Potion-Making



2A : Warrior

2B : Archery

2C : Daily Magic

2D : Mathematics

2E : Art

2F : Management

2G : Economics

2H : Sociology

2I : Psychology

2J : Philosophy

2K : Jurisprudence

2L : Romance Novels

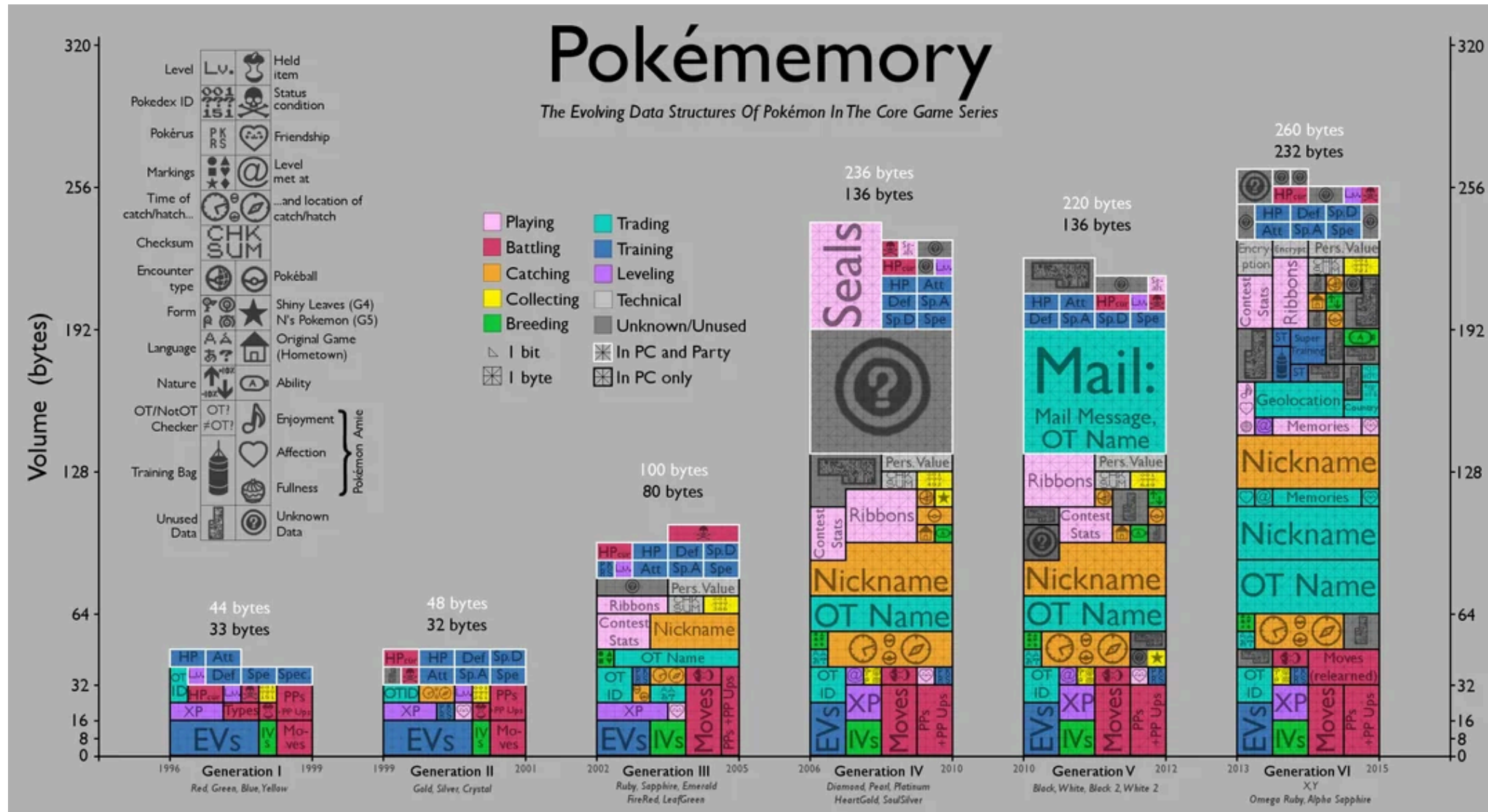
2M : Mystery Novels

2N : History

2O : The Travels of Otherworld

2P : Dungeons

2Q : Language



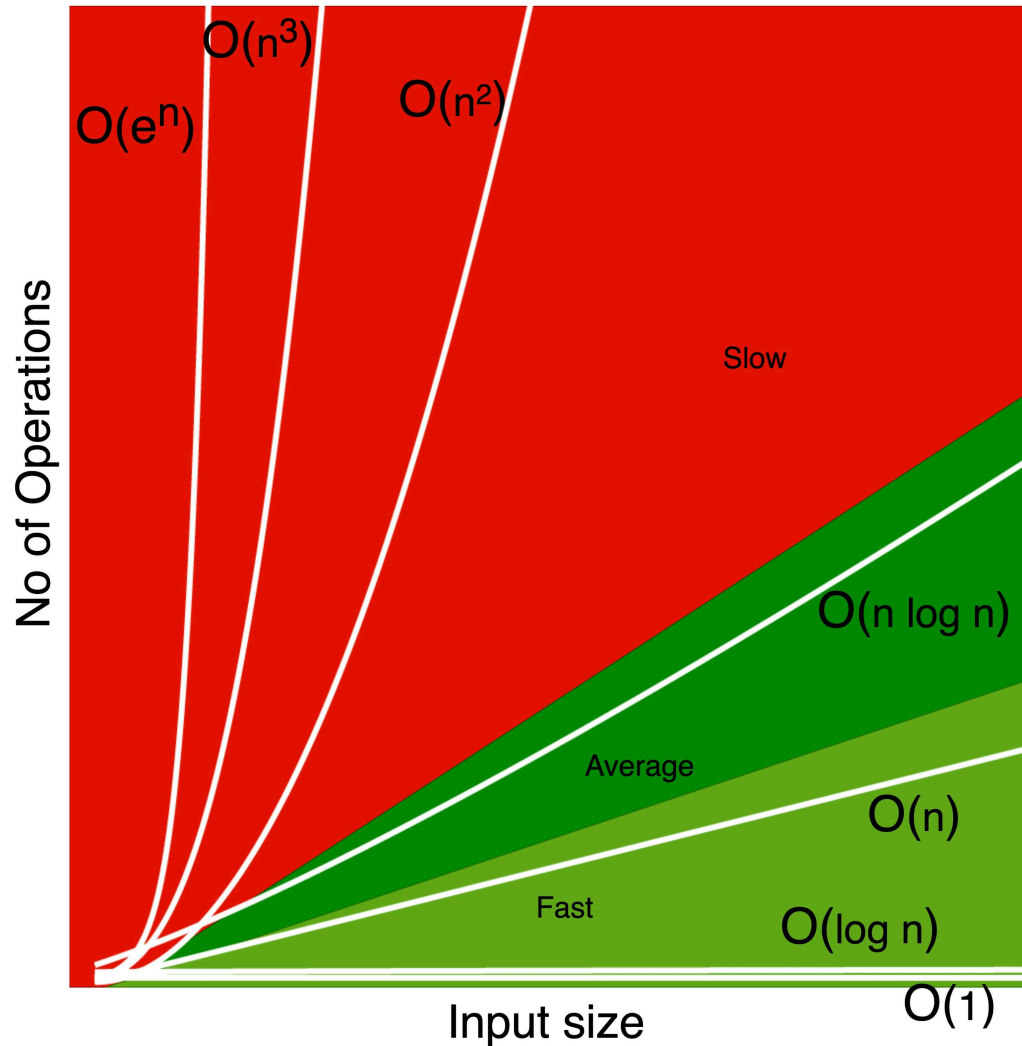
Considere o seguinte exemplo de uma estrutura de vetor:

```
alunos = [  
    ["Ana", 8.5],  
    ["Bruno", 7.2],  
    ["Carlos", 4.1],  
    ["Daniel", 6.8],  
    ["Eduardo", 5.2]  
]  
  
print(alunos[0][0]) # Acessa o nome do primeiro aluno  
print(alunos[1][1]) # Acessa a nota do segundo aluno
```

Vamos agora analisar o mesmo exemplo utilizando um dicionário:

```
alunos = {  
    "Ana": 8.5,  
    "Bruno": 7.2,  
    "Carlos": 4.1,  
    "Daniel": 6.8,  
    "Eduardo": 5.2  
}  
  
print(alunos["Ana"]) # Acessa a nota da Ana  
print(alunos["Bruno"]) # Acessa a nota do Bruno
```

Imagine que de ambos os exemplos anteriores, a única coisa que precisamos verificar é a quantidade de alunos reprovados. **Qual das duas estruturas de dados seria mais eficiente para essa tarefa?**



Complexidade de Algoritmos

Toda operação que realizamos em uma estrutura de dados possui um **custo computacional** associado. Esse custo é medido em termos de **complexidade de tempo** e **complexidade de espaço**.

Complexidade de Tempo

Observe o algoritmo abaixo que mantém uma lista com as notas (N1, N2 e N3) de cada aluno:

```
notas = [  
    [8.5, 7.0, 9.0],  
    [7.2, 6.5, 8.0],  
    [4.1, 5.0, 3.5],  
    [6.8, 7.5, 6.0],  
    [5.2, 4.5, 6.0]  
]
```

A complexidade espacial desse algoritmo é **$O(n * m)$** , onde n é o número de alunos e m é o número de notas por aluno.

```
maior_nota_final = 0
for aluno in notas:
    soma = 0
    for nota in aluno:
        soma += nota
    media = soma / len(aluno)
    if media > maior_nota_final:
        maior_nota_final = media

print(f"A maior nota final é: {maior_nota_final}")
```

Para encontrar a maior nota final, precisamos percorrer toda a lista, somando as três notas e comparando com a maior nota final encontrada.

A complexidade de tempo desse algoritmo é $O(n * m)$, onde n é o número de alunos e m é o número de notas por aluno.

Complexidade de Espaço

Agora vamos analisar o algoritmo anterior em termos de complexidade de espaço. Adicionamos o cálculo prévio da média das notas de cada aluno na própria lista:

```
alunos = [  
    [8.5, 7.0, 9.0, 8.17],  
    [7.2, 6.5, 8.0, 7.23],  
    [4.1, 5.0, 3.5, 4.2],  
    [6.8, 7.5, 6.0, 6.77],  
    [5.2, 4.5, 6.0, 5.23]  
]
```

A complexidade espacial desse algoritmo ainda é **$O(n * m)$** , onde n é o número de alunos e m é o número de notas por aluno.

```
maior_nota_final = 0
for aluno in alunos:
    media = aluno[3] # A média já está armazenada na lista
    if media > maior_nota_final:
        maior_nota_final = media
print(f"A maior nota final é: {maior_nota_final}")
```

Para cada aluno, teremos um campo extra na lista, aumentando o consumo de memória. A complexidade de espaço desse algoritmo é **$O(n)$** , onde n é o número de alunos, pois estamos armazenando uma média para cada aluno.

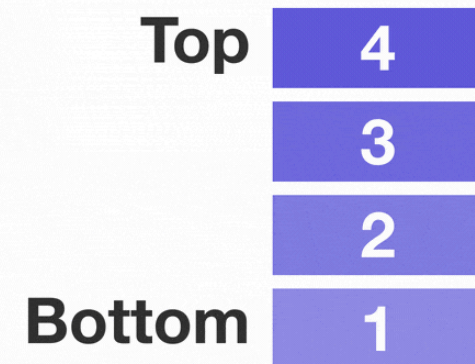
Resumo da Complexidade do Exemplo

Tipo de Estrutura	Complexidade de Tempo	Complexidade de Espaço
Sem armazenar as médias	Mais lento	Menos espaço ocupado
Armazenando as médias	Menos rápido	Mais espaço ocupado

Operações em Estruturas

Estruturas de dados permitem realizar operações como:

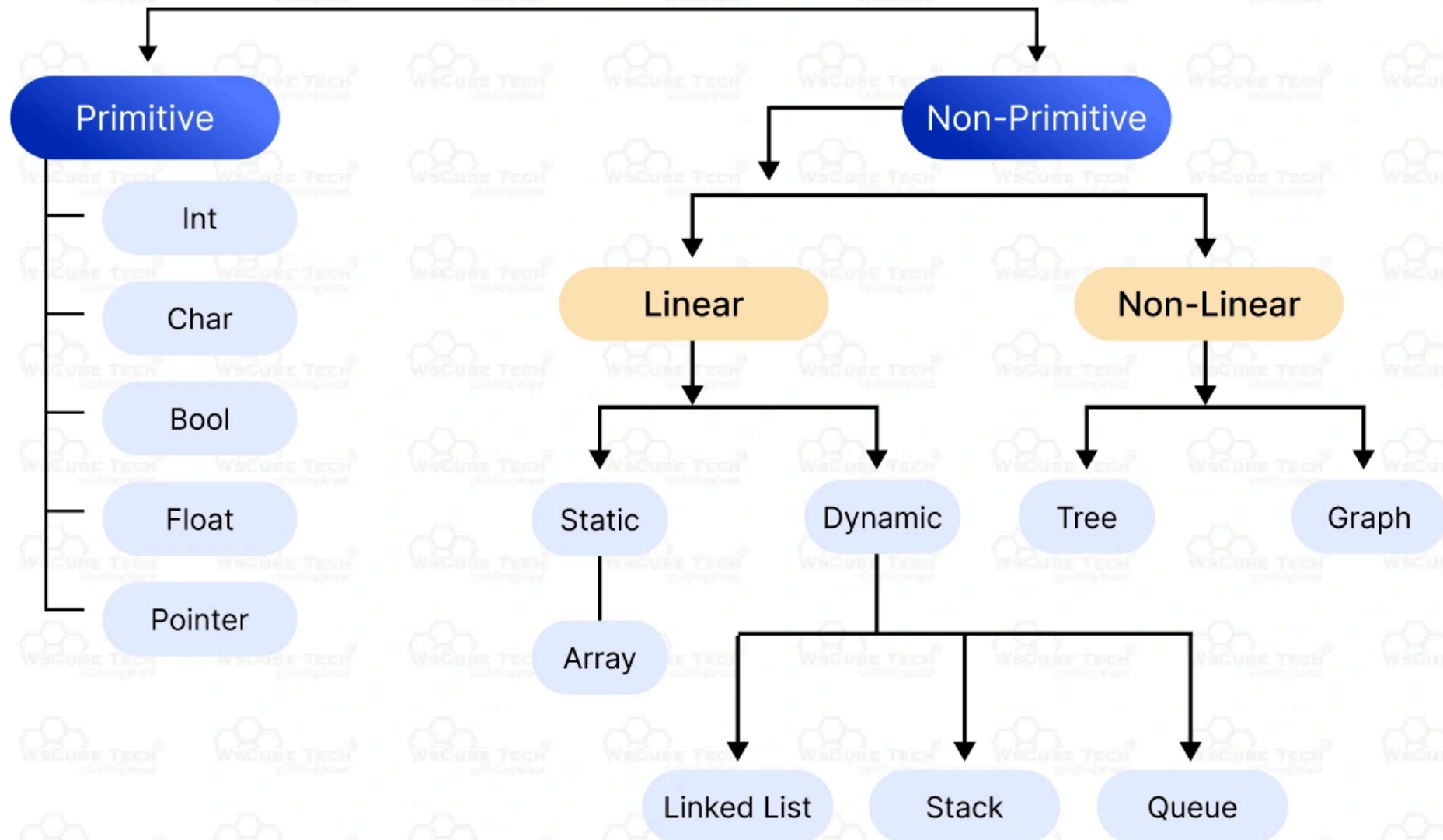
- Inserção de dados
- Modificação de dados
- Remoção de dados
- Busca de dados
- Ordenação de dados
- Filtragem de dados
- Agrupamento de dados



Classificação de Estruturas

As estruturas de dados podem ser classificadas de acordo com a forma como são organizadas e acessadas:

- **Primitivas:** São as estruturas de dados mais simples, como números, caracteres e booleanos.
- **Estruturas Lineares:** Os elementos são organizados em uma sequência linear.
- **Estruturas Não-Lineares:** Os elementos não são organizados em uma sequência linear.



Exemplos de Estruturas de Dados

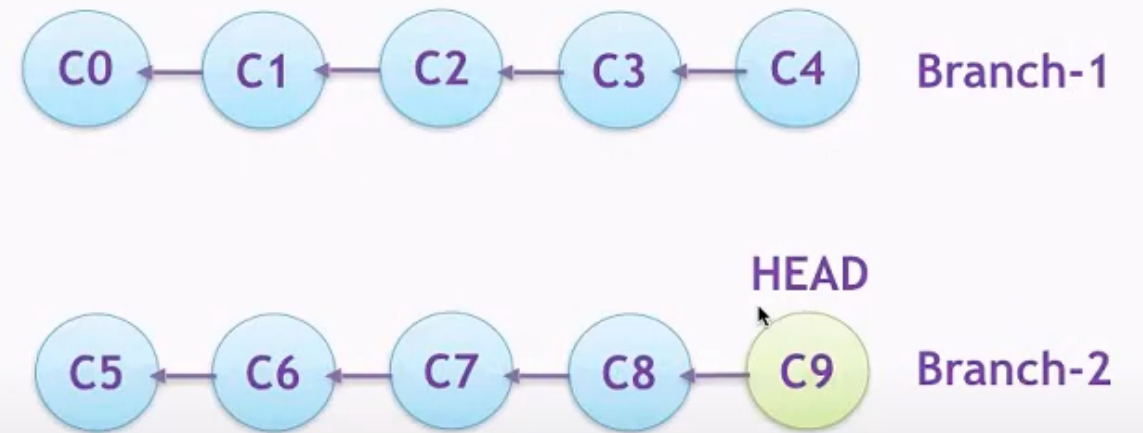


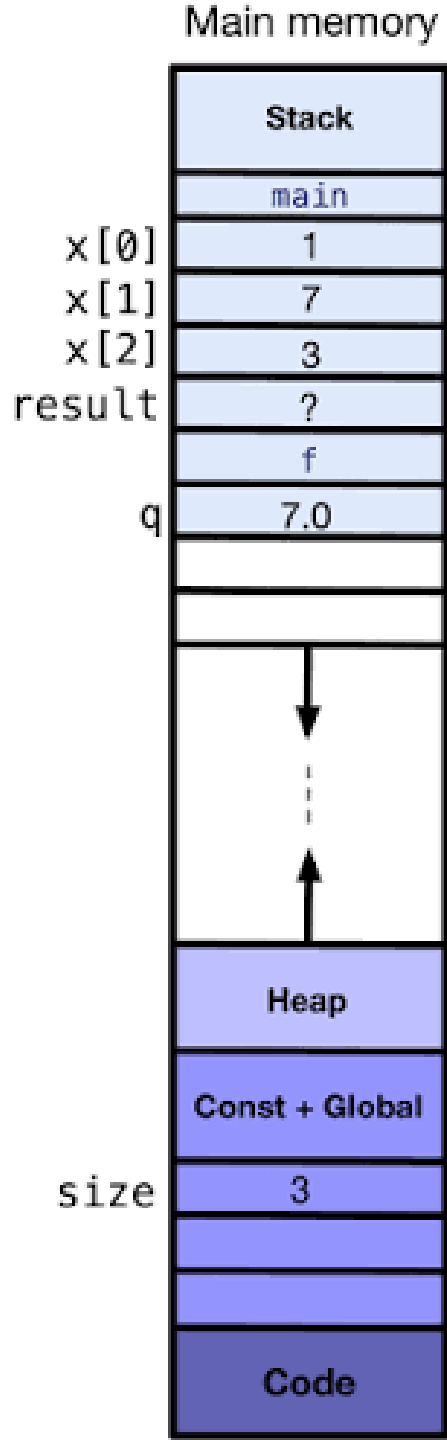
Vetores Estáticos

Uso em tilesets de jogos 2D, onde cada tile é representado por um índice fixo em um vetor.

Listas Encadeadas

Uso em *commits* de sistemas de controle de versão, onde cada commit aponta para o commit anterior, formando uma lista encadeada.





```

1 #include <stdio.h>
2 double f(int[]);
3
4 int main(void) {
5     int x[3] = {1, 7, 3};
6     double result = f(x);
7     return 0;
8 }
9 double f(int list[]) {
10     double q = 7.0;
11     return q;
12 }

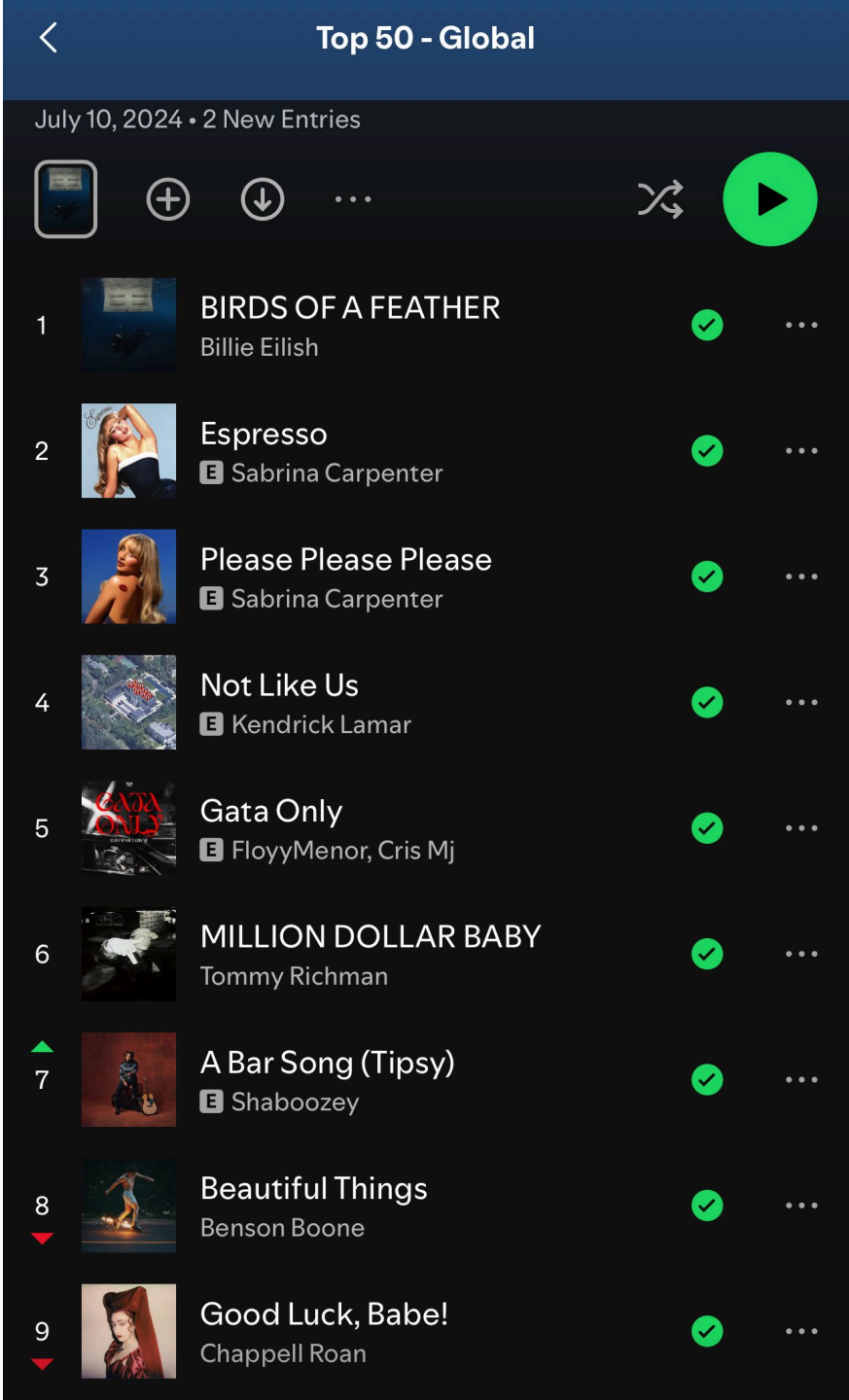
```

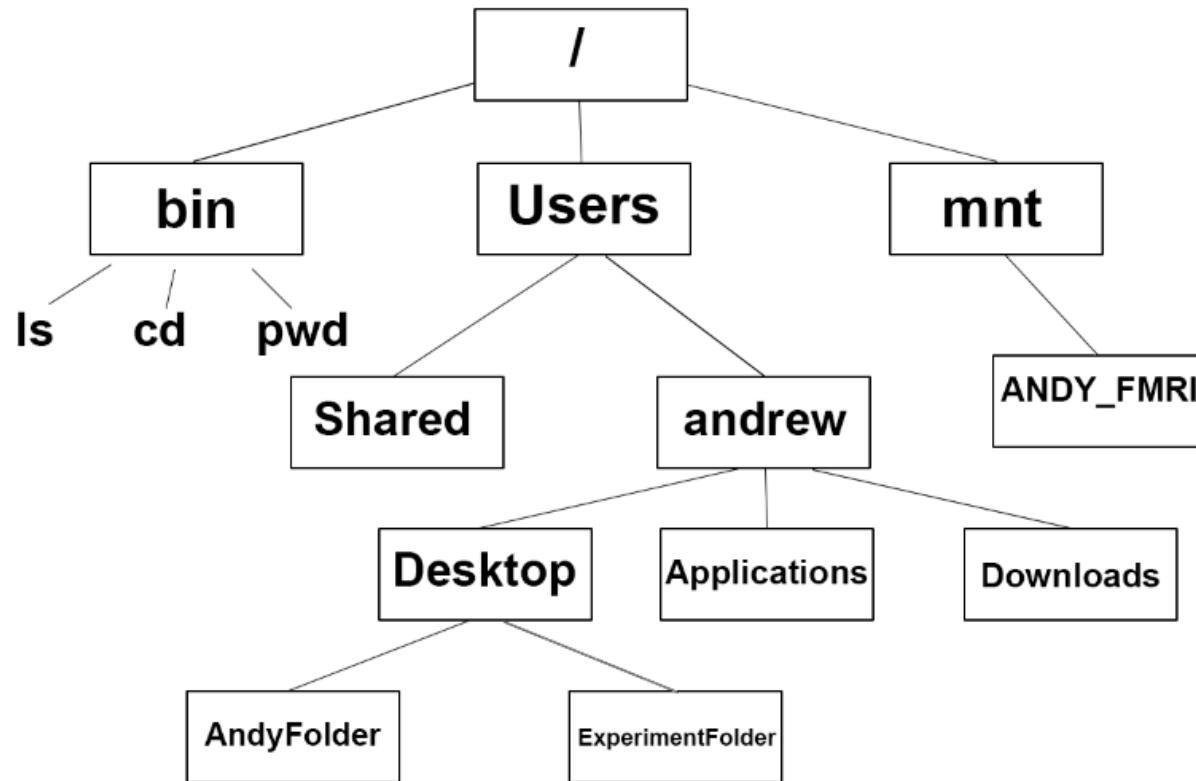
Pilhas

Usadas na execução de chamadas de funções em linguagens de programação, onde cada chamada é empilhada e desempilhada conforme a execução progride.

Filas

Usadas em playlists de música, onde as músicas são reproduzidas na ordem em que foram adicionadas à fila.





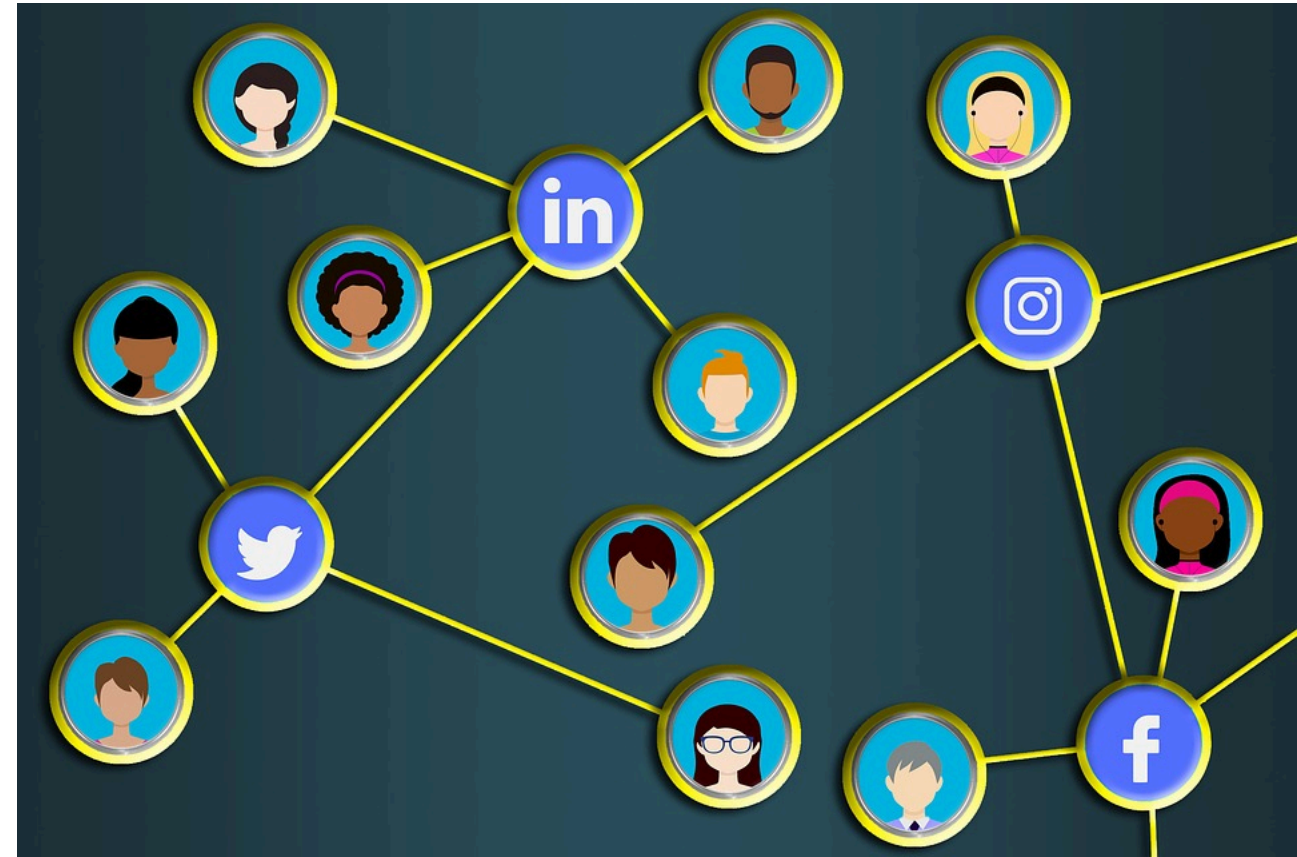
Árvores

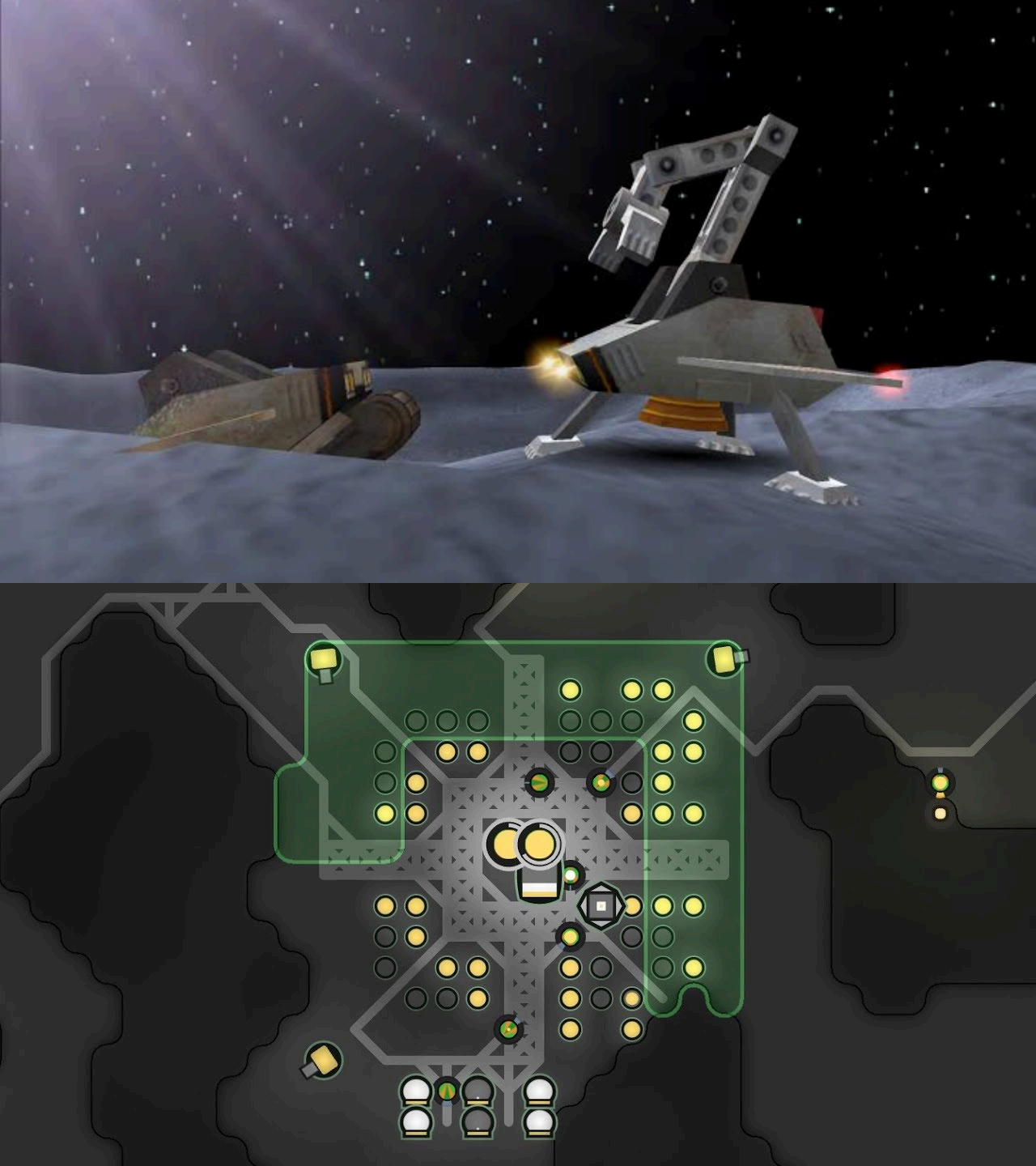
Usadas em sistemas de arquivos, onde cada diretório pode conter subdiretórios e arquivos, formando uma estrutura hierárquica.

Grafos

Usadas em redes sociais, onde cada usuário é um nó e as conexões entre eles são arestas, formando um grafo que representa as relações entre os usuários.

Você já ouviu falar da teoria dos seis graus de separação?





Recursos Úteis

- [Colobot](#)
- [Screeps](#)
- [CodeCombat](#)
- [GitHub](#)