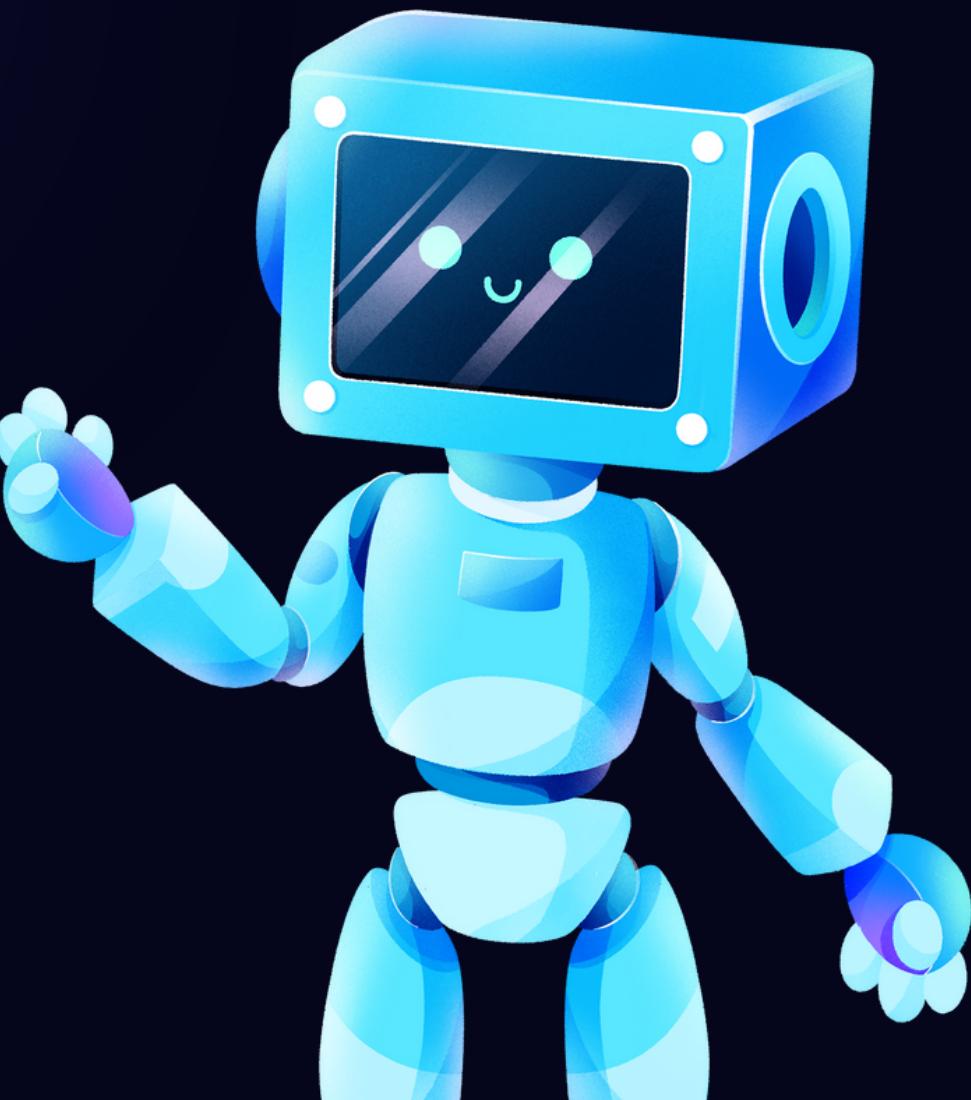


DESING PATTERNS

ITERATOR

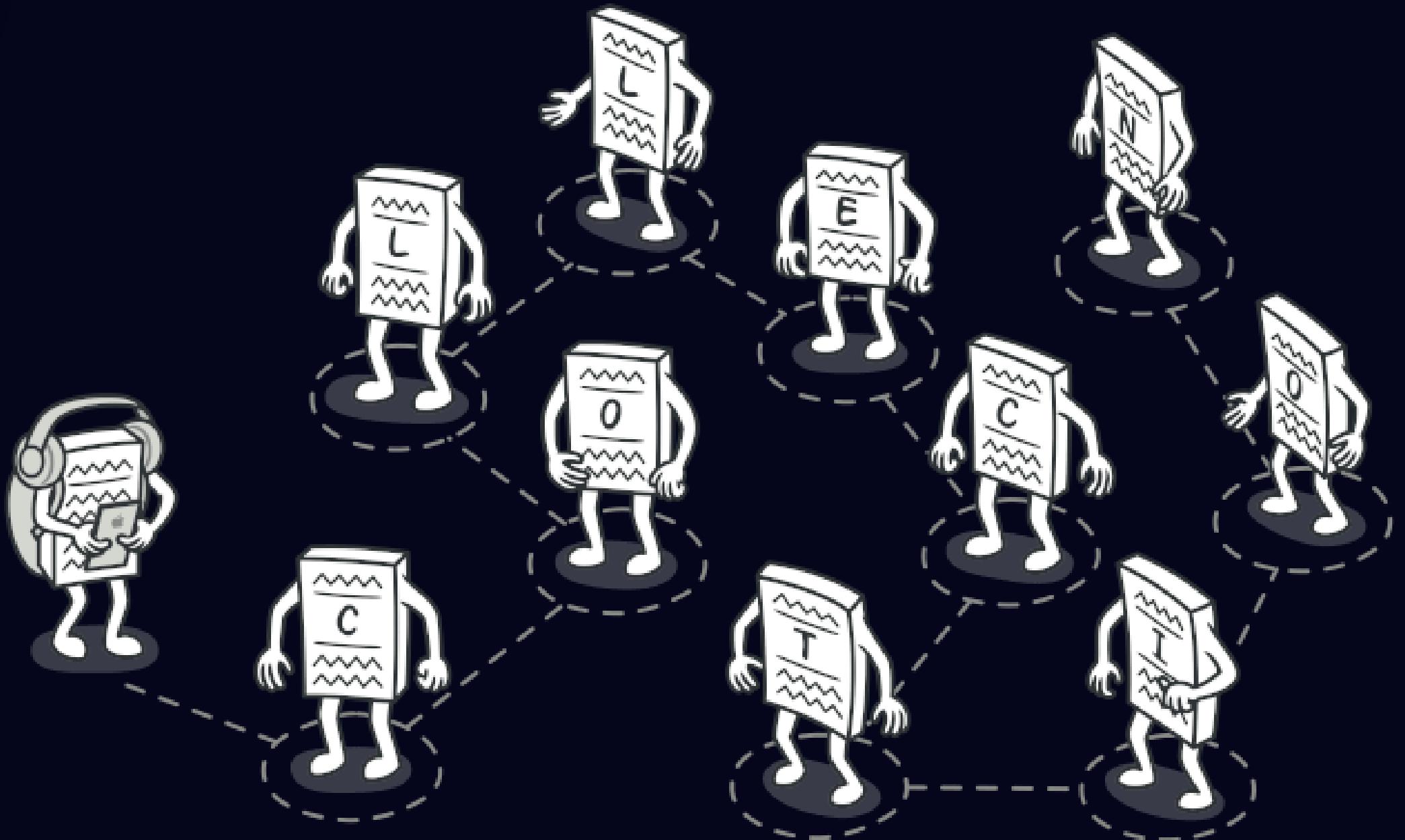
Luis Marcelo
Vitória Oliveira

ADS - 3º Período



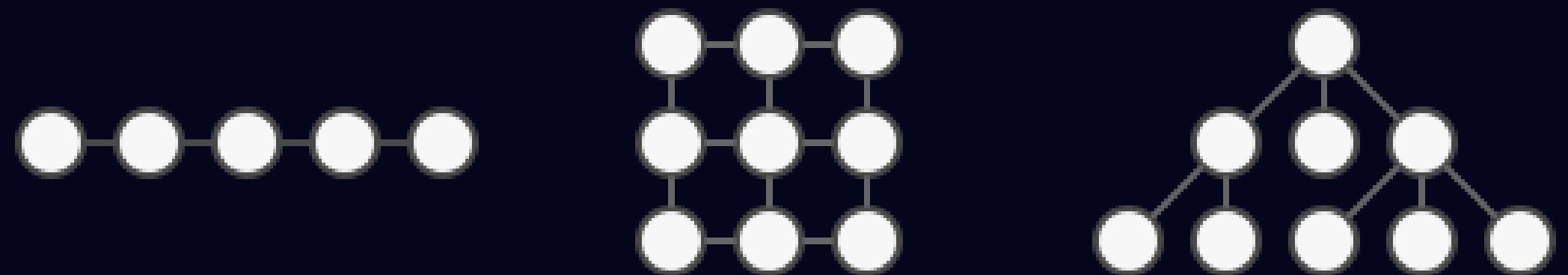
PROpósito

O Iterator é um padrão de projeto comportamental que permite percorrer elementos de uma coleção sem precisar saber como isso está acontecendo debaixo dos panos, independente de ser lista, pilha, árvore, etc.

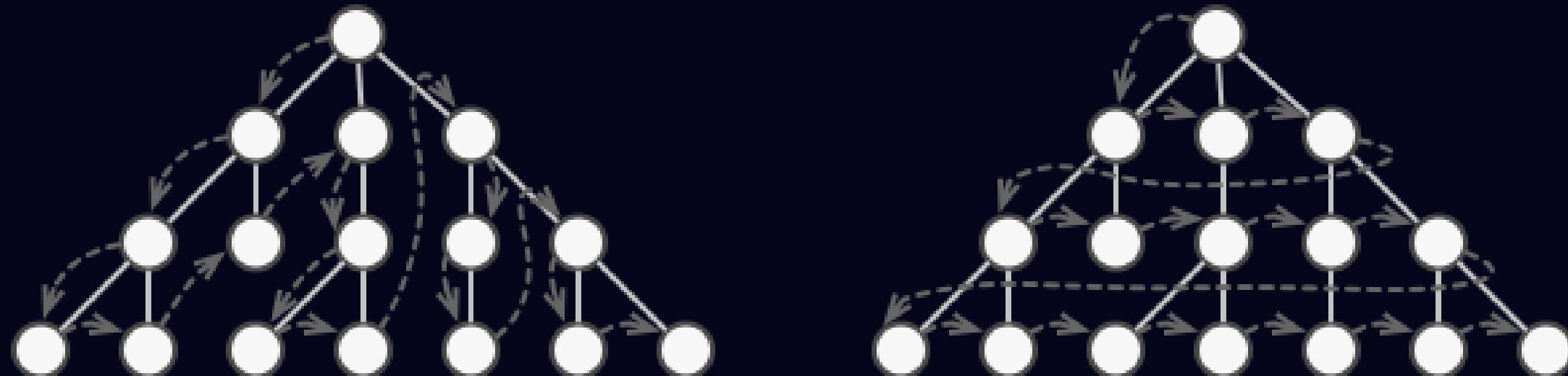


PROBLEMA

Vários tipos de coleções.



A mesma coleção pode ser atravessada de diferentes formas.



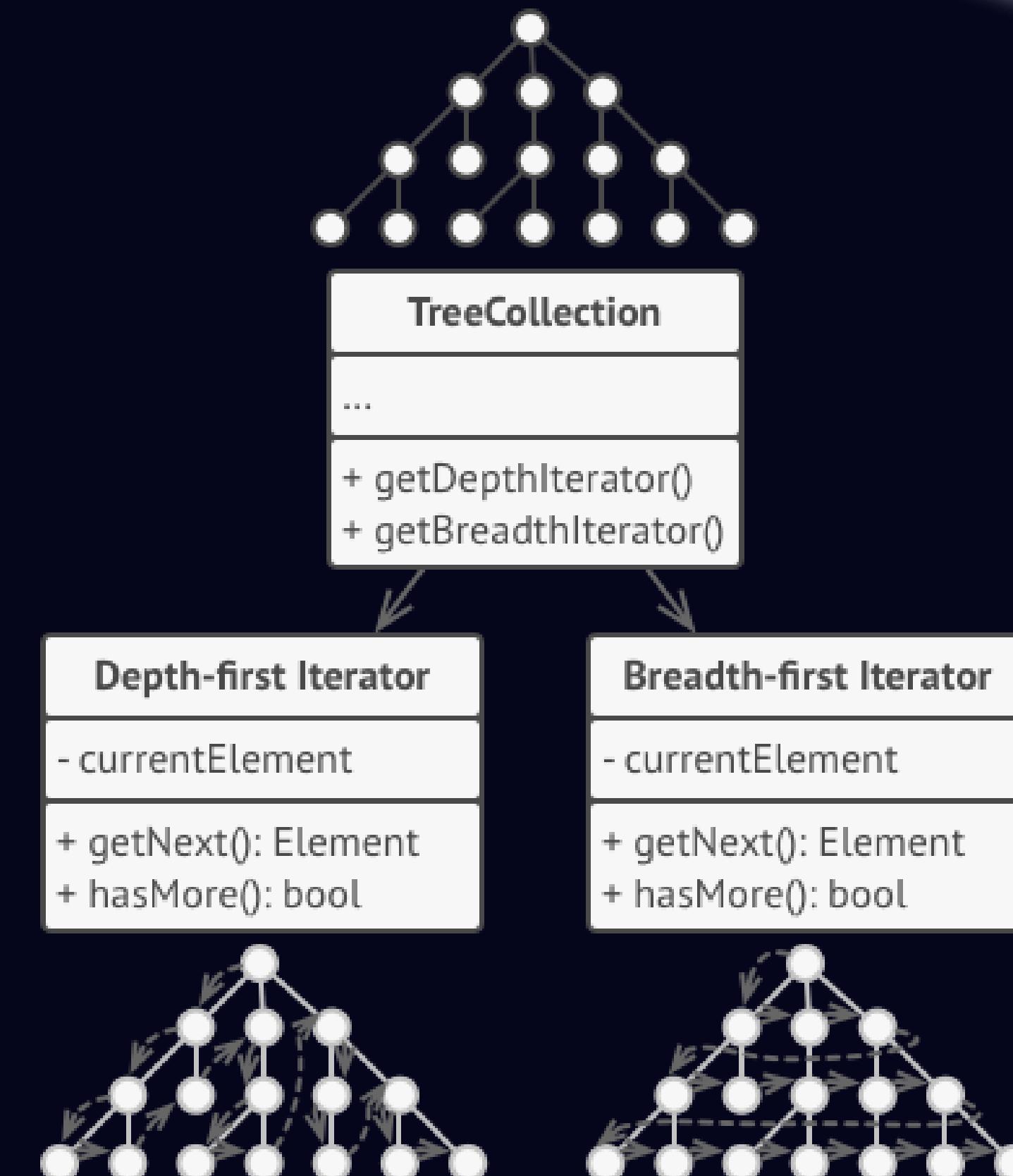
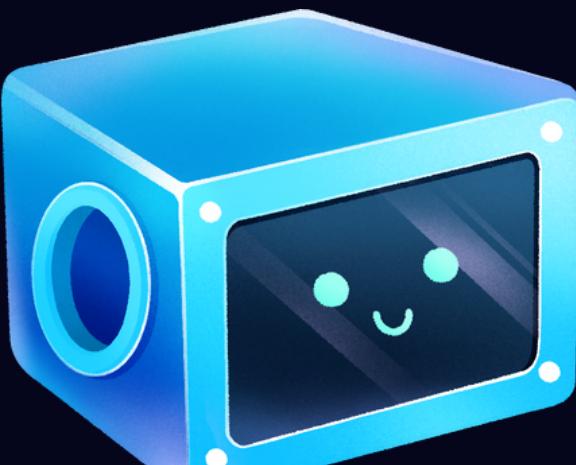
PROBLEMA

Como podemos garantir uma travessia eficiente de coleções, considerando a diversidade de estruturas de dados e algoritmos de travessia disponíveis?

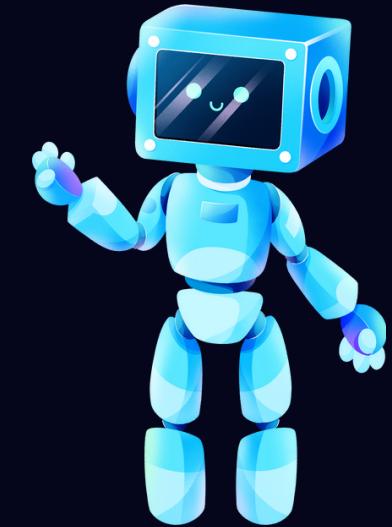
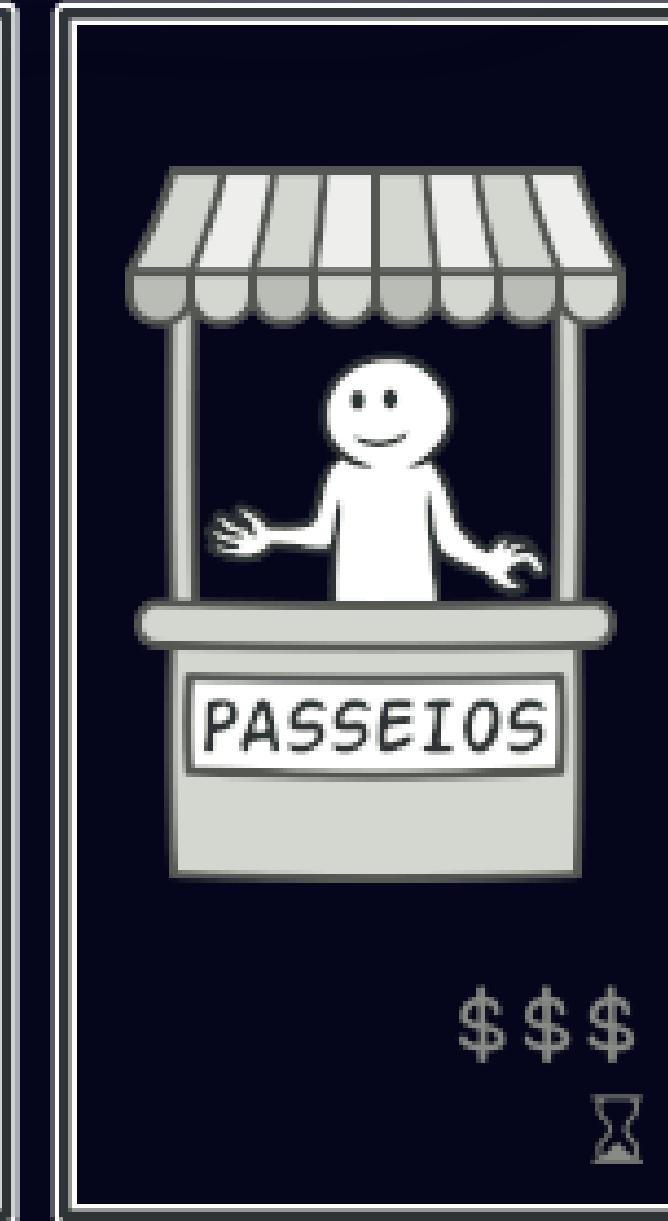
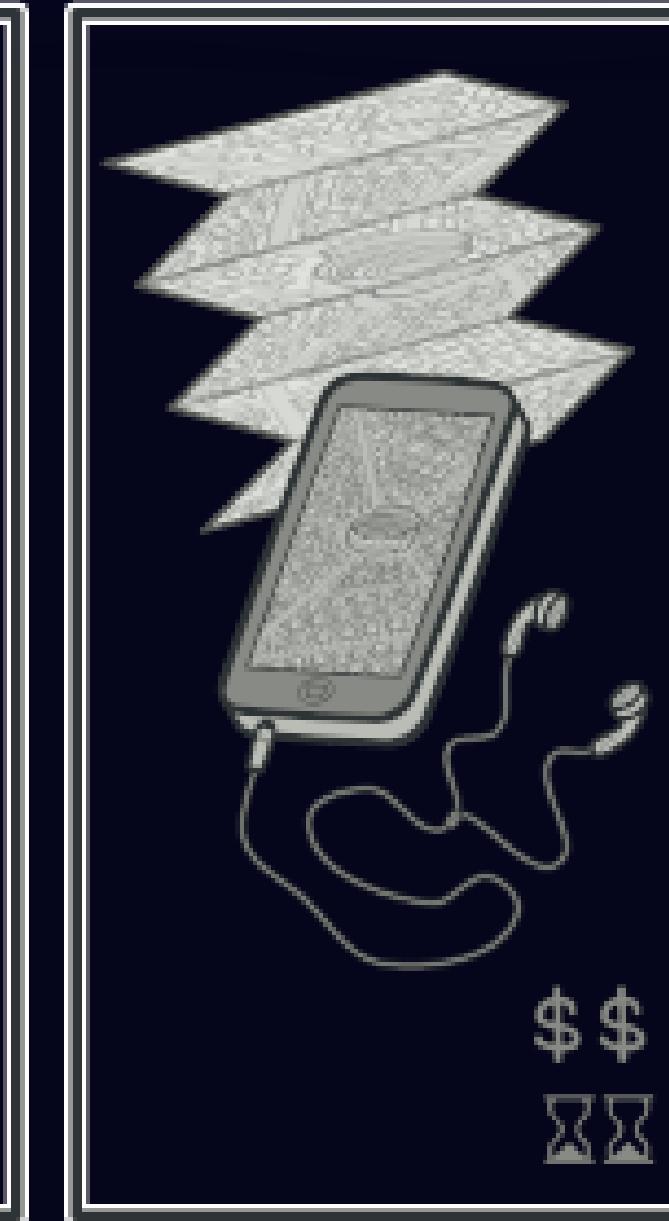
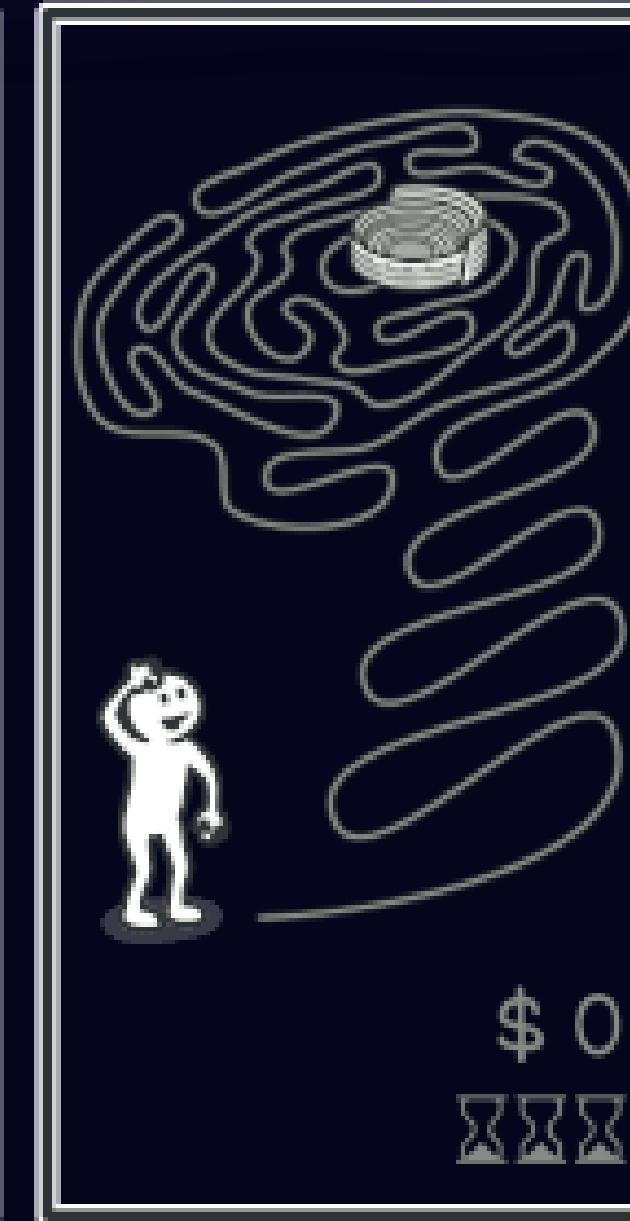
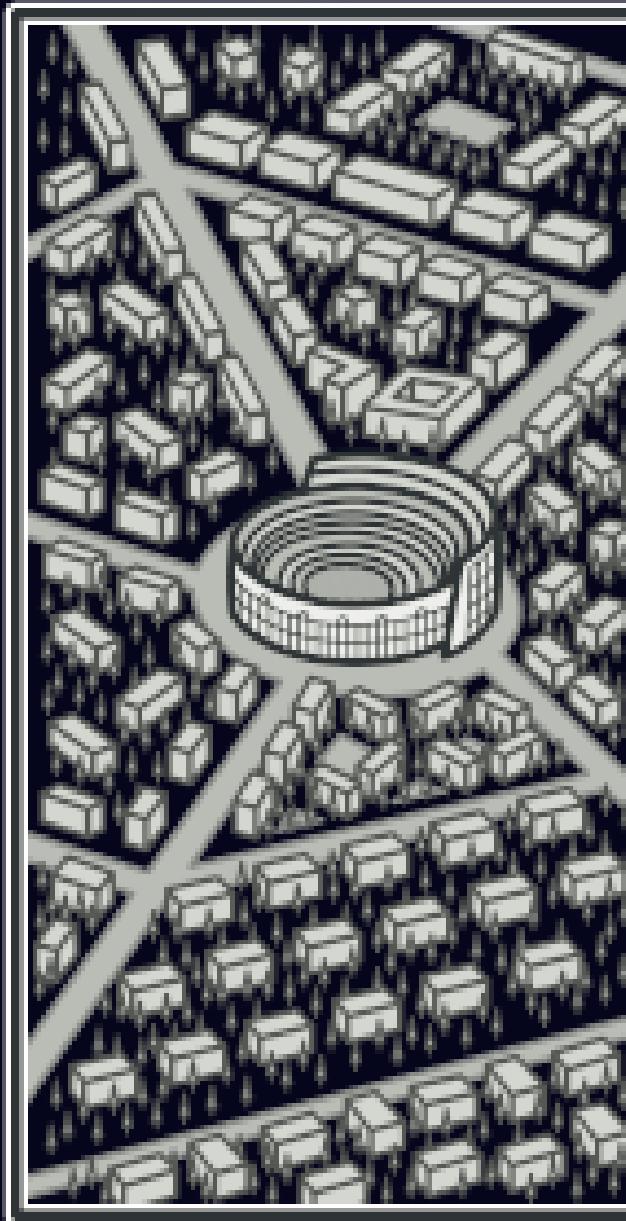


SOLUÇÃO

A ideia principal do padrão Iterator é extrair o comportamento de travessia de uma coleção para um objeto separado chamado um iterador.

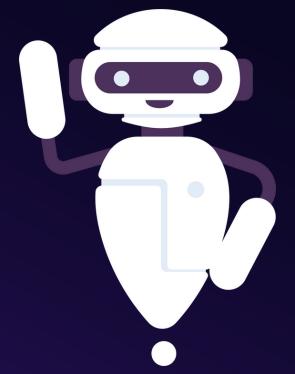


ANALOGIA COM O MUNDO REAL

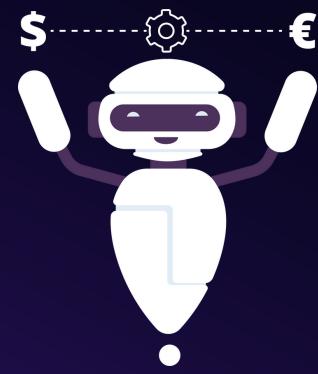


Várias maneiras de se caminhar por Roma.

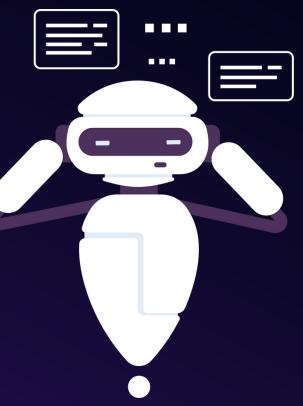
APLICABILIDADE



Utilize o Iterator quando sua coleção tiver uma estrutura de dados complexa, mas você quer esconder a complexidade dela, por motivos de conveniência ou segurança.



Utilize o iterator para reduzir a duplicação de código de travessia em sua aplicação.



Utilize o Iterator quando você quer que seu código seja capaz de percorrer diferentes estruturas de dados ou quando os tipos dessas estruturas são desconhecidos de antemão.

PRÓS



É possível pausar a travessia e continuar posteriormente.



É possível atravessar várias vezes a mesma coleção em paralelo usando outro objeto iterador.



É fácil adicionar novos objetos iteradores com algoritmos de travessia completamente diferentes.

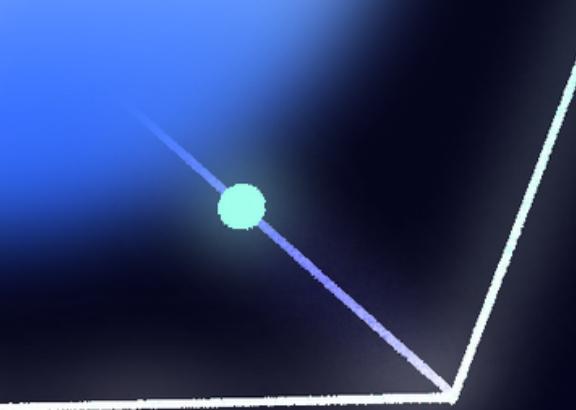


Não polui o código do objeto principal com vários métodos e algoritmos de travessia diferentes.

CONTRAS



Aplicar o iterator pode adicionar complexidade ao código, especialmente para coleções simples.

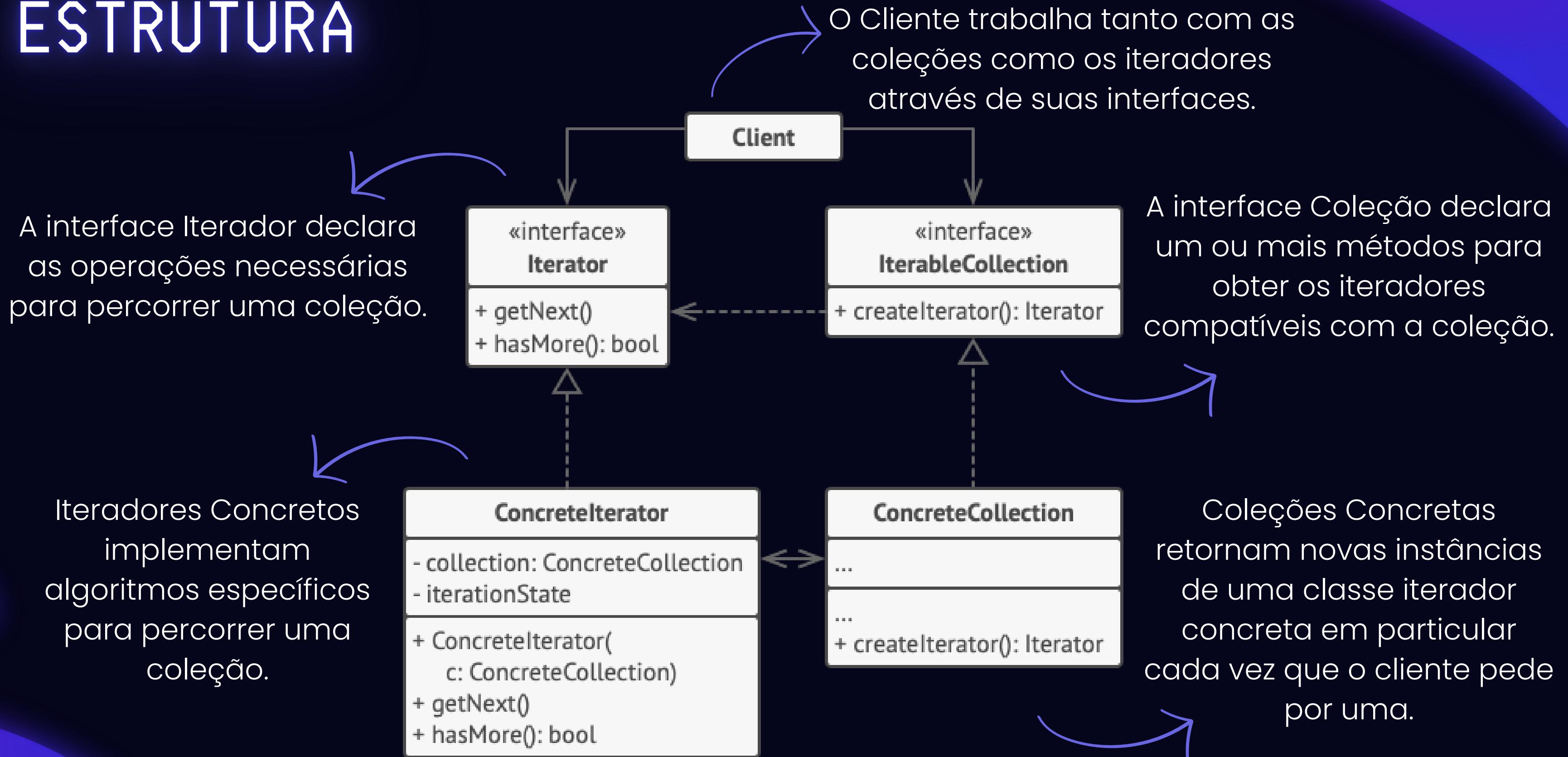


RELAÇÕES COM OUTROS PADRÕES

- **Composite:** Iterador pode ser usado para percorrer árvores composite.
- **Factory Method:** Fornecer diferentes tipos de iteradores para diferentes tipos de coleções.
- **Decorator:** Capturar e restaurar o estado de iteração.
- **Visitor:** Percorrer estruturas de dados complexas e executar operações sobre seus elementos.



ESTRUTURA



TIME TO CODE !

