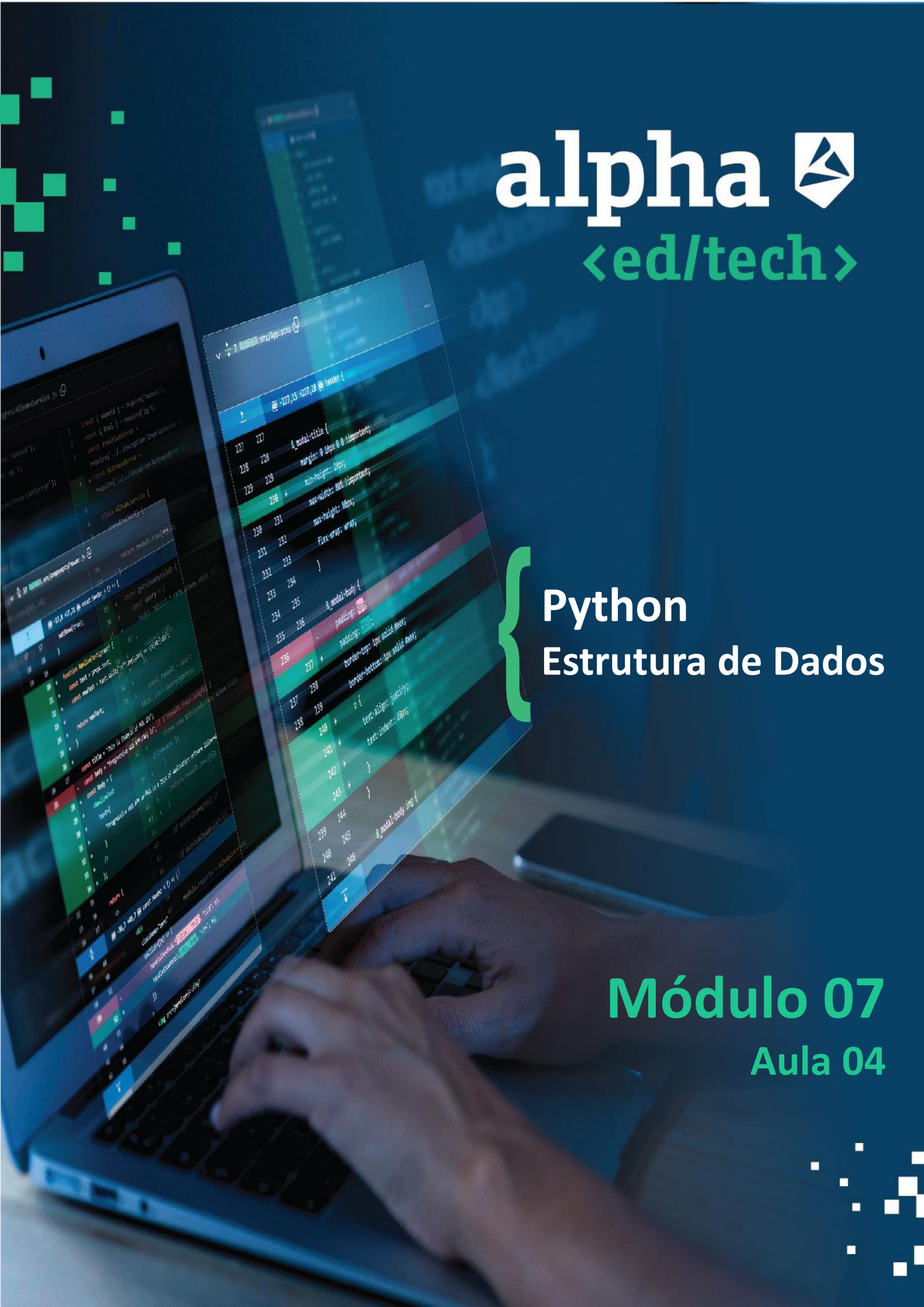
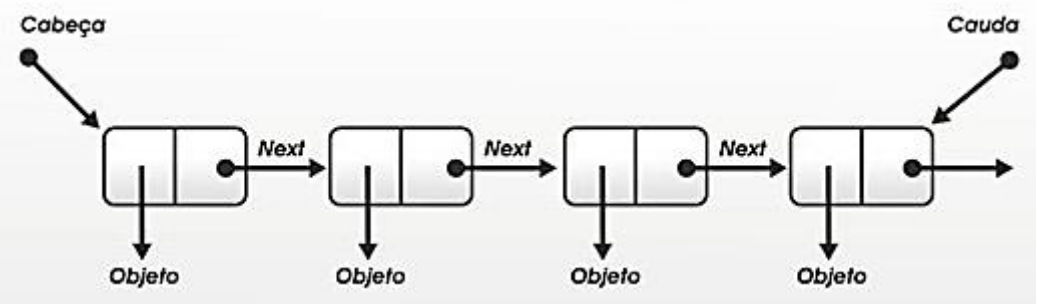

alpha
<ed/tech>
Python
Estrutura de Dados
Módulo 07
Aula 04

Estrutura de Dados

Lista Encadeada Simples

Uma lista encadeada é uma coleção sequencial de elementos, chamados **nós**, onde cada nó armazena **seu próprio valor** e um **ponteiro** (link) para o próximo nó na sequência. Os elementos inicial e final desta lista, possuem referência para ponteiro NULO, pois não tem um elemento anterior (início) e posterior (fim).

São estruturas de dados úteis para representar um conjunto de dados dinâmico onde você não precisa definir um tamanho máximo para sua lista, pois ela permite que os elementos sejam adicionados sob demanda, e isto permite um melhor gerenciamento de memória pois ela só ocupa o espaço dos elementos existentes na lista.



Características

- Flexibilidade no Armazenamento de Dados: Listas encadeadas podem crescer dinamicamente, não havendo necessidade de definir um tamanho fixo no momento da criação.
- Eficiência em Inserções e Deleções: Inserir ou remover um elemento de uma lista encadeada é geralmente mais eficiente que em um array, pois não requer realocação ou reorganização de outros elementos.
- Uso de Memória: Cada nó em uma lista encadeada requer espaço adicional para armazenar o endereço do próximo nó.

Contraste com o Uso de Arrays

Arrays

- **Armazenamento Contíguo na Memória:** Todos os elementos de um array são armazenados em locais de memória contíguos, o que facilita o acesso rápido aos elementos por meio de índices.
- **Tamanho Fixo:** O tamanho de um array é definido na criação e não pode ser alterado dinamicamente (arrays dinâmicos como listas em Python são uma exceção, mas internamente eles são realocados quando seu tamanho muda).
- **Eficiência no Acesso a Elementos:** Arrays permitem acesso direto e rápido aos seus elementos, o que é ideal para operações de leitura frequentes.
- **Ineficiência em Operações de Inserção/Remoção:** Inserir ou remover elementos, especialmente no início ou no meio de um array, pode ser ineficiente, pois pode exigir o deslocamento de muitos elementos.

Listas Encadeadas

- **Armazenamento Não Contíguo:** Os elementos (nós) são armazenados em locais de memória separados, ligados por ponteiros.
- **Tamanho Dinâmico:** Podem crescer ou diminuir de tamanho conforme necessário, sem a necessidade de realocação frequente.
- **Inserções e Deleções Eficientes:** Adicionar ou remover elementos é geralmente mais rápido, pois apenas envolve a alteração de ponteiros.
- **Acesso Indireto aos Elementos:** Acessar elementos específicos é menos eficiente, pois pode exigir percorrer a lista desde o início.

Em resumo, a escolha entre usar um array ou uma lista encadeada depende das necessidades específicas de uma aplicação, particularmente se a prioridade é o acesso rápido aos elementos (arrays) ou a eficiência em operações de inserção e remoção (listas encadeadas).

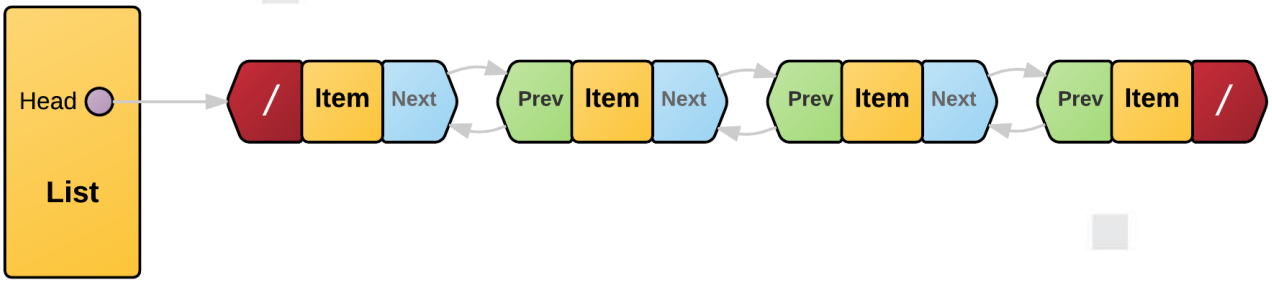
Operações Básicas

- **Inserção:** Adicionar um nó ao início, final ou em uma posição específica.
- **Busca:** Encontrar um elemento na lista.
- **Deleção:** Remover um nó da lista.
- **Travessia:** Percorrer todos os elementos da lista.

Estrutura de Dados

Lista Encadeada Dupla

As listas encadeadas duplas são uma variação avançada das listas encadeadas simples. Enquanto nas listas simples cada nó contém um dado e um ponteiro para o próximo nó, nas listas encadeadas duplas, cada nó possui dois ponteiros: um para o nó seguinte e outro para o nó anterior. Esta característica adicional oferece maior flexibilidade e eficiência em certas operações.



Características das Listas Encadeadas Duplas

- **Nó Duplamente Vinculado:** cada nó em uma lista encadeada dupla contém três partes: uma seção de dados e dois ponteiros (links). O primeiro ponteiro aponta para o próximo nó na lista, enquanto o segundo aponta para o nó anterior.
- **Cabeçalho e Cauda:** a lista tem um nó de cabeçalho (head) que marca o início da lista e um nó de cauda (tail) que marca o fim. Em uma lista vazia, tanto head quanto tail são null.
- **Travessia Bidirecional:** diferentemente das listas encadeadas simples, as listas duplas permitem a travessia tanto na direção frente quanto na direção reversa, proporcionando uma maior flexibilidade na navegação pela lista.
- **Inserção e Remoção Eficientes:** inserções e remoções podem ser mais eficientes em comparação com listas encadeadas simples, pois não é necessário percorrer a lista para encontrar o nó anterior ao ponto de inserção ou remoção.
- **Uso de Memória:** cada nó em uma lista encadeada dupla consome mais memória que um nó em uma lista simples devido ao ponteiro adicional.

Aplicações das Listas Encadeadas Duplas

- **Implementação de Estruturas de Dados Complexas:** como filas de prioridade, deques (double-ended queues) e listas de acesso frequente.
- **Navegação em Ambas as Direções:** útil em aplicações onde a navegação para frente e para trás é frequentemente requerida, como em aplicativos de edição de texto ou navegadores de histórico.
- **Gerenciamento de Recursos:** em sistemas operacionais e gerenciadores de memória, para manter listas de recursos ou processos.

As listas encadeadas duplas são poderosas e versáteis, oferecendo vantagens significativas em termos de flexibilidade operacional, embora ao custo de maior complexidade e uso de memória.