

ALEXANDRE DIAMANTE

**FUNDAMENTOS DE ESTRUTURAS DE DADOS**  
**2-Listas Encadeadas**

Trabalho apresentado para Atividade Prática na disciplina ESTRUTURA DE DADOS, do curso de CIÊNCIA DE DADOS, turno EAD da Instituição de Ensino Ampli Anhanguera.

Umuarama-PR

2024

```

class Node:

    def __init__(self, data):
        self.data = data # Armazena o dado do nó
        self.next = None # Aponta para o próximo nó da lista, inicialmente None

class LinkedList:

    def __init__(self):
        self.head = None # Inicializa a lista com o primeiro nó (cabeça) como None

    def append(self, data):
        new_node = Node(data) # Cria um novo nó com o dado fornecido
        if self.head is None: # Se a lista estiver vazia, o novo nó será a cabeça
            self.head = new_node
        else:
            current = self.head
            while current.next: # Percorre a lista até o final
                current = current.next
            current.next = new_node # Adiciona o novo nó ao final da lista

    def print_list(self):
        current = self.head
        while current: # Percorre a lista até o final
            print(current.data, end=" -> ") # Imprime o dado do nó
            current = current.next
        print("None") # Indica o fim da lista

# Implementar uma função para contar o número de nós em uma lista encadeada.
def count_nodes(linked_list):
    count = 0
    current = linked_list.head
    while current: # Percorre a lista até o final
        count += 1 # Incrementa o contador para cada nó
        current = current.next
    return count # Retorna o número total de nós

# Exemplo de uso
ll = LinkedList()
ll.append(10)
ll.append(20)
ll.append(30)
ll.append(40)

ll.print_list() # Output: 10 -> 20 -> 30 -> None

print("Número de nós:", count_nodes(ll)) # Output: Número de nós: 3

```

Publicado no Git:

<https://github.com/alexandrediamante/cienciadedados.git>