

# Fundamentos de Algoritmos e Estrutura de Dados - Aula 02 - Listas Encadeadas

Prof. André Gustavo Hochuli

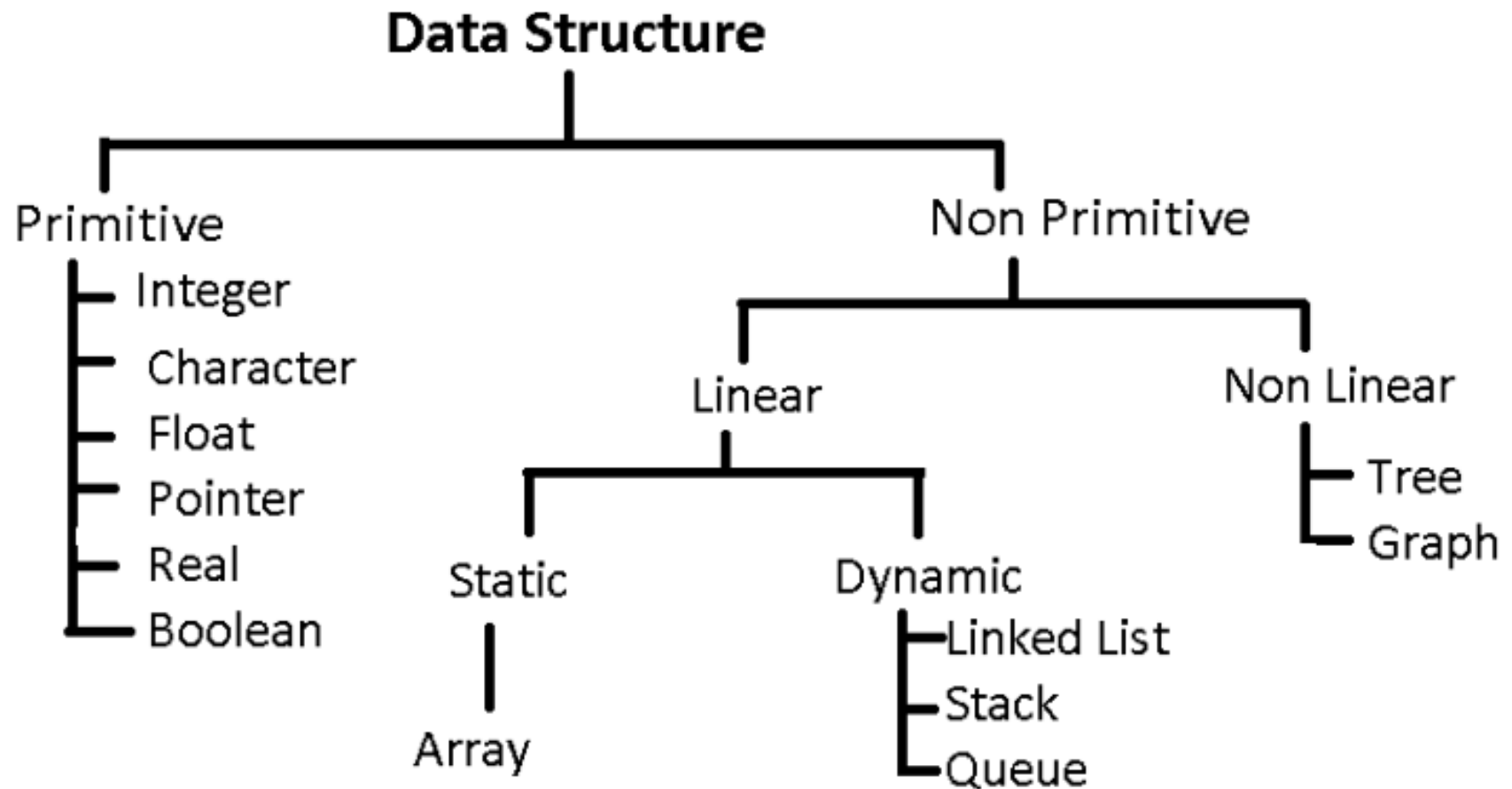
[gustavo.hochuli@pucpr.br](mailto:gustavo.hochuli@pucpr.br)

aghochuli@ppgia.pucpr.br

# Plano de Aula

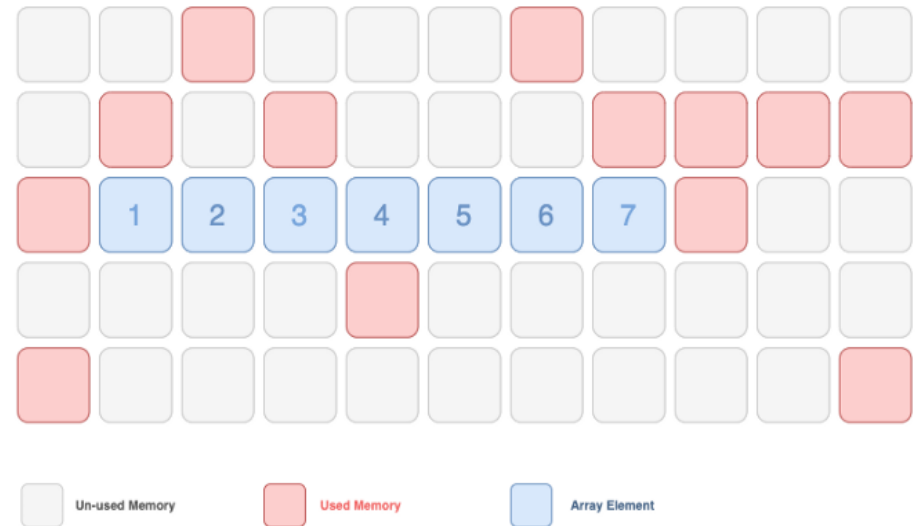
- **Listas Encadeadas**
- **Filas e Pilhas**
- **Exercícios**

# Estruturas de Dados



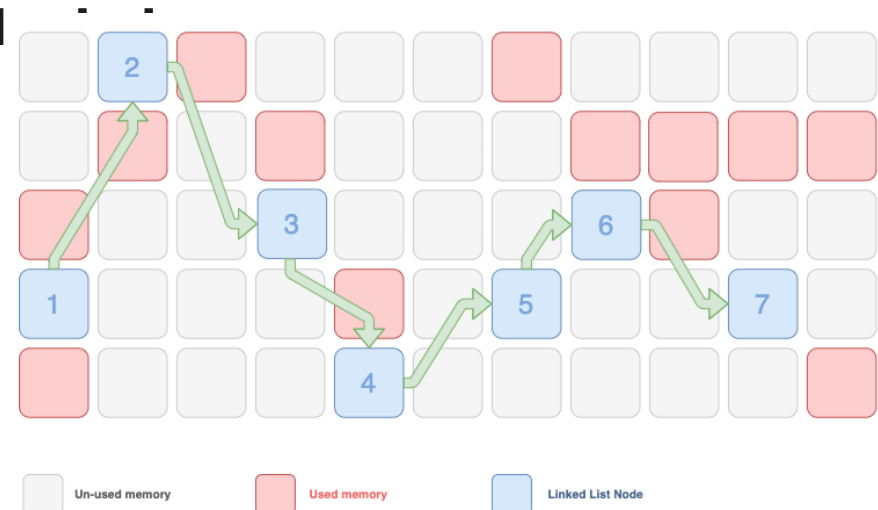
# Estruturas de Dados Estáticas (Ou contíguas)

- **Alocação contígua**
- **Vantagens**
  - **Acesso é rápido e sequencial**
  - **Baixo Overhead**
  - **Requer baixo nível de program**
- **Desvantagem**
  - **Inviável para grandes massas**
  - **Limitado ao número de blocos sequenciais livres**



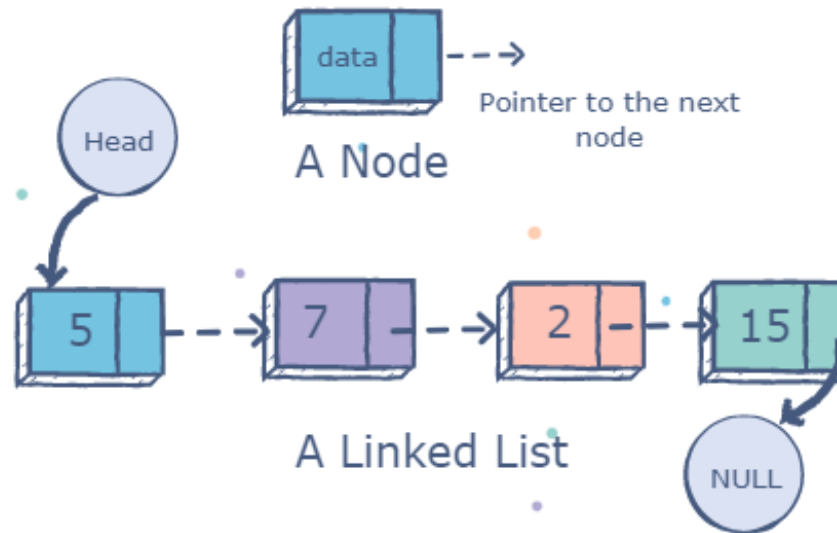
# Estruturas de Dados Dinâmicas (Ou Encadeadas)

- **Alocação não-contígua**
- **Vantagens**
  - **Armazenar grandes massas d**
  - **Memória física é o limite**
- **Desvantagem**
  - **Desempenho**
  - **Alto Overhead**
  - **Elevado nível de abstração**



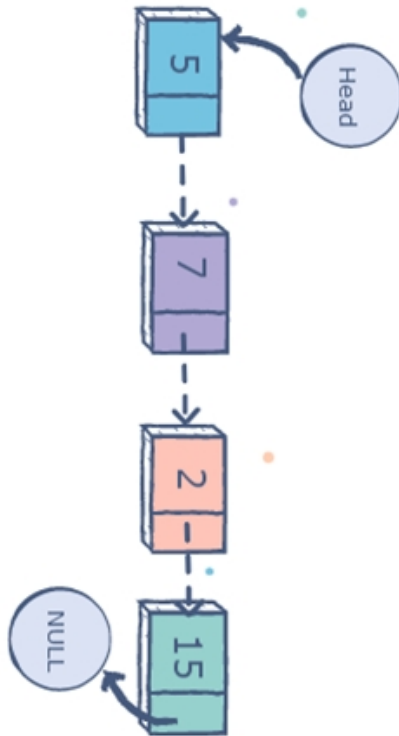
# Estruturas de Dados Dinâmicas (Ou Encadeadas)

- **Topologia**

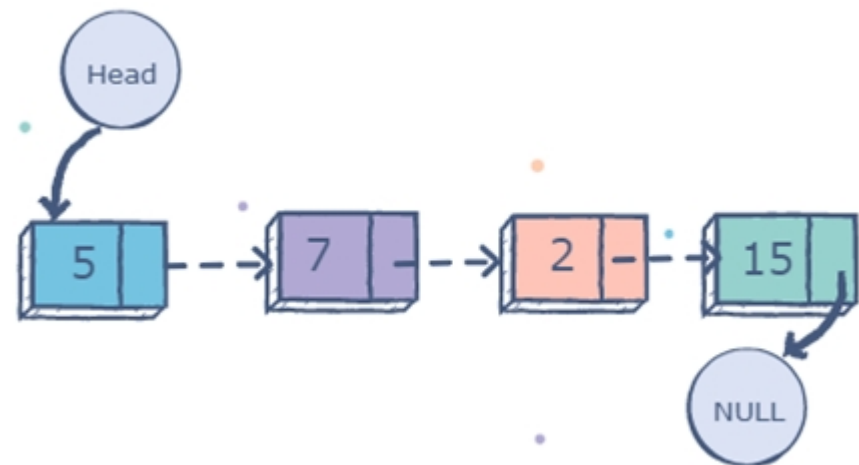


# Estruturas de Dados Dinâmicas (Ou Encadeadas)

**Pilha (First In - Last Out)  
(First In - First Out)**

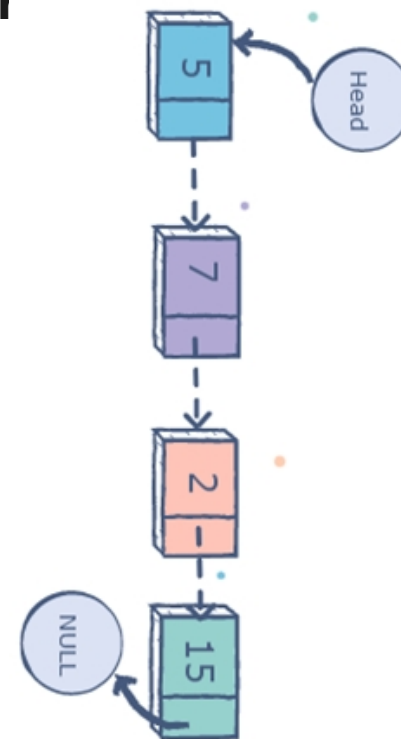


**Fila**



# Pilha (Stack)

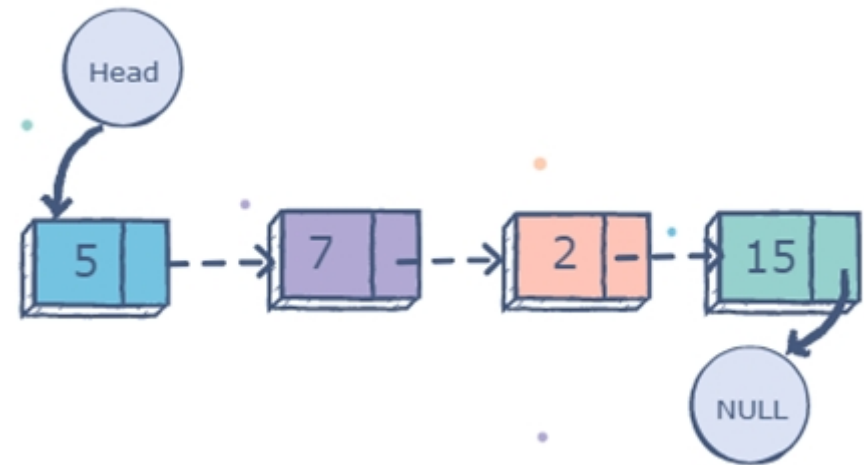
- **Pilha ou (*Stack*)**
  - **Inserção e Remoção da cabeça (First In Last Out)**
- **Aplicações**
  - **Recursão (Programação)**
  - **Reverter Vetores**
  - **Histórico de Navegação**
  - **Etc**





# Fila (Queue)

- **Fila (Queue)**
  - **Inserção da Cauda**
  - **Remoção da Cabeça**
  - **First In - Last Out**
- **Aplicações**
  - **Compartilhamento de Recursos**
    - **CPU, Interrupções, Harwades e Perifericos**
  - **Controle de Acesso**
  - **Transfêrência de Dados**
  - **Playlists**



# Implementação e Discussão

**Vamos codificar!!**

**Estrutura de Dados - DeepNote**

# Exercícios

- **Implemente um lista encadeada, com inserção e remoção em qualquer posição**
- **Implemente um algoritmo de ordenação utilizando listas ligadas**
- **Implemente o jogo da torre de Hanói**
- **Implemente a impressão da de forma recursiva (em ordem e ordem-inversa)**
- **Implemente a busca por elemento recursiva**