



**INSTITUTO FEDERAL  
DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  
São Paulo

## **Linguagem de Programação 2**

---

Prof. Dr. Domingos Bernardo Gomes Santos

# Ementa

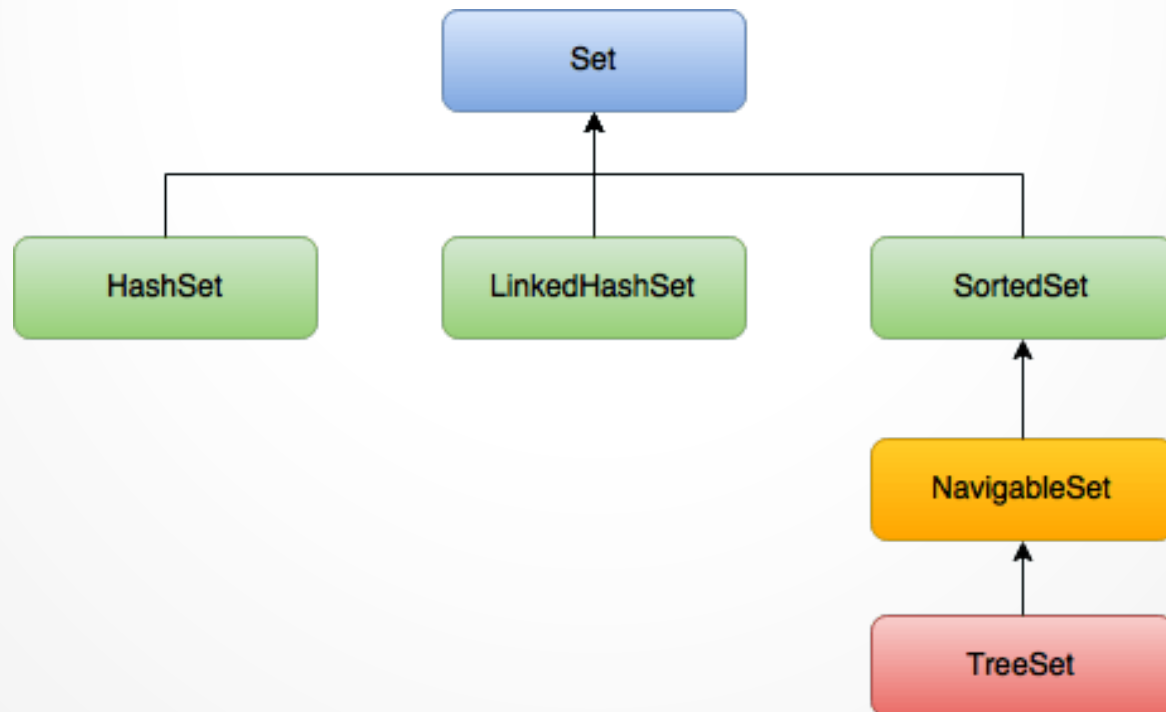
- Apresentação de conceitos avançados sobre **Collections Framework Java**;
- Apresentação de conceitos da Interface **Set**;
- Implementações da interface **Set**: *HashSet*, *TreeSet* e *LinkedHashSet*;
- Apresentação de conceitos da Interface **Queue**;
- Implementações da interface **Queue**: *Deque* e *Stack*.

# Collections

- A classe **Collections** apresenta uma grande quantidade de métodos estáticos úteis para a manipulação de coleções;
- **binarySearch(List, Object)**: busca binária por determinado elemento na lista ordenada e retorna a posição ou número negativo, caso não encontrado;
- **max(Collection)**: maior elemento da coleção;
- **min(Collection)**: menor elemento da coleção;
- **reverse(List)**: inverte a lista.

# Interface Set

- A interface **Set** não permite a inclusão de elementos duplicados;
- Os objetos não permanecem na ordem em que foram inseridos;



# Interface `java.util.Set`

- A comparação entre objetos depende da implementação da sobreescrita do método ***equals*** e ***hashCode***;
- Pode-se empregar o comando ***enhanced-for*** para observar os elementos armazenados no ***Set***;
- A implementação ***TreeSet*** insere os elementos em ordem definida pelo método de comparação. Este método é definido através da implementação da interface ***java.lang.Comparable***;
- O ***LinkedHashSet*** mantém a ordem de inserção dos elementos.

# Java.util.Queue

- A interface **Queue** define métodos de entrada e de saída para filas com critérios de ordenação definidos através das implementações Java;
- Por exemplo implementações de filas do tipo **LIFO** e **FIFO**.

