{abstract}
Task

id: int

titulo: string

descricao: string

prioridade: int

dataVencimento: string

+ Task()

+ Task(id: int, titulo: string, descricao: string, prioridade: int, dataVencimento: string)

+ Task(id: int, titulo: string, descricao: string, prioridade: int)

+ Task(id: int, titulo: string, descricao: string)

+ Task(id: int, titulo: string)

+ getId(): int

+ getTitulo(): string

+ getDescricao(): string

+ getPrioridade(): int

+ getDataVencimento(): string

+ setId(id: int): void

+ setTitulo(titulo: string): void

+ setDescricao(descricao: string): void

+ setPrioridade(prioridade: int): void

+ setDataVencimento(dataVencimento: string): void

+ printTasks(): void (virtual puro)

+ ~Tasks()

<<Template>>
No (lista encadeada)

- valor: T

- prev: No<T>*

- next: No<T>*

+ No(valor: T)

0..*

1

<<Template>>
ListaDuplamenteEncadeada

- head: No<T>*

- tail: No<T>*

+ ListaDuplamenteEncadeada()
+ pushFront(value: T): void
+ pushBack(value: T): void
+ removeBack(): void
+ removeFront(): void
+ isEmpty(): bool
+ print(): void
+ getSize(): int
+ removeValue(value: T): bool
+ clear(): void

1..1

Extends

KanbanBoard

- tarefas: ListaDuplamenteEncadeada<KanbanTask*>

+ addTask(task: KanbanTask*): void

+ removeTask(task: KanbanTask*): void

+ sortTasks(prioridade: int): void

+ moveTask(taskId: int, statusAtual: string, statusDestino:): void

+ printBoard(): void

+ getTasks(): ListaDuplamenteEncadeada<KanbanTask*>

+ KanbanBoard()

1

0..*

KanbanTask

- status : string

+ KanbanTask(id: int, titulo: string)

+ KanbanTask(id: int, titulo: string, descricao: string)

+ KanbanTask(id: int, titulo: string, descricao: string, prioridade: int)

+ KanbanTask(id: int, titulo: string, descricao: string, prioridade: int, dataVencimento: string)

+ setStatus(status: string): void

+ getStatus(): string

+ printTask(): void

+ ~KanbanTask()

<<Template>>
Fila

- head: No*

- tail: No*

+ Fila()
+ ~Fila()
+ isEmpty(): bool
+ enqueue(value: T): void
+ dequeue(): void
+ peek(): T&

<<Template>>
Pilha

- top: No*

+ Pilha()
+ ~Pilha()
+ isEmpty(): bool
+ push(value: T): void
+ pop(): void
+ peek(): T&

<<Template>>
AlgoritmosDeOrdenacao

+ bubbleSort(arr: T[], size: int): void
+ selectionSort(arr: T[], size: int): void
+ mergeSort(arr: T[], left: int, right: int): void
+ merge(arr: T[], left: int, mid: int, right: int): void

<<Template>>
AlgoritmosDeBuscaBinaria

+ BuscaBinariaIterativa(arr: T[], tamanho: int, chave: T): int

+ BuscaBinariaRecursiva(arr: T[], inicio: int, fim: int, chave: T): int