* Classe Tarefa:

Atributos: Essa classe representa uma tarefa a ser realizada. Os atributos definem informações relacionadas à tarefa, como seu ID, título, descrição, datas de início e conclusão, prioridade e status.

Métodos:

definirPrioridade(prioridade: int): void: Permite definir a prioridade da tarefa.

atualizarStatus(status: enum): void: Atualiza o status da tarefa para um novo status.

adicionarSubtarefa(subtarefa: Subtarefa): void: Adiciona uma sub tarefa à tarefa principal.

listarSubtarefas(): List<Subtarefa>: Retorna uma lista de sub tarefas associadas a esta tarefa.

* Classe Subtarefa:

Atributos: Essa classe representa uma sub tarefa associada a uma tarefa principal. Os atributos incluem um ID, uma descrição e um indicador de conclusão.

Métodos:

marcarConcluida(): void: Marca a sub tarefa como concluída.

desmarcarConcluida(): void: Marca a sub tarefa como não concluída.

* Classe ListaDeTarefas:

Atributos: Essa classe representa uma lista de tarefas, que pode conter várias tarefas.

Métodos:

adicionarTarefa(tarefa: Tarefa): void: Adiciona uma tarefa à lista de tarefas.

removerTarefa(tarefa: Tarefa): void: Remove uma tarefa da lista de tarefas.

listarTarefas(): List<Tarefa>: Retorna uma lista de tarefas na lista.

buscarTarefaPorStatus(status: enum): List<Tarefa>: Retorna uma lista de tarefas com base em um status específico.

* Classe Usuário:

Atributos: Essa classe representa um usuário do sistema, que pode criar e gerenciar listas de tarefas.

Métodos:

CriarListaDeTarefas(nome: string): ListaDeTarefas: Cria uma nova lista de tarefas com um nome especificado.

AdicionarListaDeTarefas(lista: ListaDeTarefas): void: Adiciona uma lista de tarefas à lista de listas de tarefas do usuário.

removerListaDeTarefas(lista: ListaDeTarefas): void: Remove uma lista de tarefas da lista de listas de tarefas do usuário.

* Classe Tarefa:

Atributos: Essa classe representa uma tarefa a ser realizada. Os atributos definem informações relacionadas à tarefa, como seu ID, título, descrição, datas de início e conclusão, prioridade e status.

Métodos:

DefinirPrioridade(prioridade: int): void: Permite definir a prioridade da tarefa.

atualizarStatus(status: enum): void: Atualiza o status da tarefa para um novo status.

adicionarSubtarefa(subtarefa: Subtarefa): void: Adiciona uma sub tarefa à tarefa principal.

listarSubtarefas(): List<Subtarefa>: Retorna uma lista de sub tarefas associadas a esta tarefa.

Classe Subtarefa:

Atributos: Essa classe representa uma sub tarefa associada a uma tarefa principal. Os atributos incluem um ID, uma descrição e um indicador de conclusão.

Métodos:

MarcarConcluida(): void: Marca a sub tarefa como concluída.

desmarcarConcluida(): void: Marca a sub tarefa como não concluída.

Classe ListaDeTarefas:

Atributos: Essa classe representa uma lista de tarefas, que pode conter várias tarefas.

Métodos:

AdicionarTarefa(tarefa: Tarefa): void: Adiciona uma tarefa à lista de tarefas.

RemoverTarefa(tarefa: Tarefa): void: Remove uma tarefa da lista de tarefas.

listarTarefas(): List<Tarefa>: Retorna uma lista de tarefas na lista.

buscarTarefaPorStatus(status: enum): List<Tarefa>: Retorna uma lista de tarefas com base em um status específico.

Classe Usuário:

Atributos: Essa classe representa um usuário do sistema, que pode criar e gerenciar listas de tarefas.

Métodos:

CriarListaDeTarefas(nome: string): ListaDeTarefas: Cria uma nova lista de tarefas com um nome especificado.

AdicionarListaDeTarefas(lista: ListaDeTarefas): void: Adiciona uma lista de tarefas à lista de listas de tarefas do usuário.

RemoverListaDeTarefas(lista: ListaDeTarefas): void: Remove uma lista de tarefas da lista de listas de tarefas do usuário.