

## Problema: Gerenciamento de Tarefas em Equipe

Objetivo: Desenvolver uma pequena aplicação em Python que ajude a gerenciar tarefas de um projeto de software. O sistema deve permitir que os usuários adicionem, visualizem e filtrem tarefas de acordo com o status (pendente, em andamento, concluída) e a prioridade (alta, média, baixa).

### Requisitos:

#### 1. Representação das Tarefas:

- Cada tarefa deve ser representada como um dicionário com as seguintes chaves:
  - `"id"`: Identificador único da tarefa (inteiro).
  - `"descricao"`: Descrição da tarefa (string).
  - `"status"`: Status da tarefa (string: "pendente", "em andamento", "concluída").
  - `"prioridade"`: Prioridade da tarefa (string: "alta", "média", "baixa").

#### 2. Lista de Tarefas:

- As tarefas devem ser armazenadas em uma lista.

#### 3. Funções:

- Adicionar Tarefa:

Implemente uma função `adicionar_tarefa(lista_tarefas, descricao, status, prioridade)` que adicione uma nova tarefa à lista de tarefas. O identificador (`id`) deve ser gerado automaticamente, incrementando a partir do maior `id` já presente na lista.

- Visualizar Tarefas:

Implemente uma função `visualizar_tarefas(lista_tarefas)` que exiba todas as tarefas presentes na lista, mostrando `id`, `descricao`, `status` e `prioridade` de cada uma.

- Filtrar Tarefas:

Implemente uma função `filtrar_tarefas(lista_tarefas, status=None, prioridade=None)` que retorne uma nova lista contendo apenas as tarefas que correspondem ao status e/ou prioridade fornecidos. Se nenhum filtro for fornecido, todas as tarefas devem ser retornadas.

#### 4. Exemplo de Fluxo do Programa:

- O programa deve iniciar com uma lista vazia de tarefas.
- Adicione algumas tarefas utilizando a função `adicionar_tarefa`.
- Visualize todas as tarefas utilizando a função `visualizar_tarefas`.
- Filtre e visualize as tarefas de acordo com diferentes critérios (por exemplo, todas as tarefas com status "pendente" e prioridade "alta").

Desafio:

Em programação em pares, um dos programadores deve focar na implementação das funções, enquanto o outro revisa e sugere melhorias ou correções. Troque de papéis após cada função implementada para garantir uma colaboração equilibrada e aprendizado mútuo.

Leitura Recomendada: <https://www.codementor.io/pair-programming>