

TESTES DE SOFTWARE

AVALIAÇÃO (N2)

Professor Joe Jonas Vogel

Avaliação

- Hoje teremos nossa Avaliação pertinente a N2
- De forma resumida, vocês terão que criar uma aplicação simples e então os testes da mesma
- Entrega hoje até às 23:59
- Individual
- A entrega deverá conter um vídeo onde você demonstra os códigos (programa e testes) e a execução dos testes. Não é necessário entregar os códigos
 - **Importante: deverá conter a sua explicação (pode ser por fala ou legenda)**
 - Detalhes nos próximos slides
- Critérios:
 - Completude (programa, testes unitários e integrados, cobertura)
 - Organização (pastas e arquivos)
 - Funcionalidade (tem que estar passando os testes)
 - Testes de fato testam a funcionalidade
 - Vídeo (deve ser possível ler o conteúdo dos códigos, possui explicação e demonstra todos os elementos)

Programa

- Crie um programa simples de gerenciamento de tarefas, com as seguintes funcionalidades:
 - Adicionar uma tarefa
 - Listar as tarefas existentes
 - Marcar uma tarefa como concluída
 - Remover uma tarefa
- Conteúdo nos próximos slides

Programa

1. Crie um projeto Python usando sua ferramenta preferida (como PyCharm ou VSCode).
2. Crie uma classe **Task** com os seguintes atributos: **id**, **description** e **done** (feito ou não feito).
3. Crie uma classe **TaskManager** que será responsável por gerenciar as tarefas. Essa classe deve ter os seguintes métodos:
 - **add_task(task: Task)**: Adiciona uma nova tarefa na lista de tarefas.
 - **list_tasks() -> List[Task]**: Retorna a lista de tarefas existentes.
 - **mark_task_done(task_id: int)**: Marca uma tarefa como concluída com base no seu id.
 - **remove_task(task_id: int)**: Remove uma tarefa com base no seu id.

Testes Unitários

- Crie testes unitários com pytest para as classes **Task** e **TaskManager** que cubram os seguintes cenários:
 - Criar uma tarefa com sucesso
 - Verificar se uma tarefa foi criada corretamente
 - Adicionar uma tarefa na lista de tarefas
 - Listar as tarefas existentes
 - Marcar uma tarefa como concluída
 - Remover uma tarefa da lista

Testes Integrados

- Crie testes de integração com o pytest para verificar se a classe **TaskManager** está funcionando corretamente. Alguns cenários a serem testados:
 - Adicionar uma tarefa e verificar se ela foi adicionada corretamente na lista de tarefas
 - Listar as tarefas existentes e verificar se a lista está correta
 - Marcar uma tarefa como concluída e verificar se ela foi atualizada corretamente na lista de tarefas
 - Remover uma tarefa e verificar se ela foi removida corretamente da lista de tarefas

Execução e Cobertura

- Execute os testes e verifique se todos passaram com sucesso. Caso contrário, corrija os erros e execute novamente.
- Execute também o Pytest Coverage que vimos em aula para gerar um **HTML** da cobertura de testes
- Com isso pronto, gere o vídeo explicativo:
 - Demonstre os códigos (programa e testes)
 - Execute os testes para demonstrar os resultados (primeiro os unitários e depois os integrados)
 - Execute o Pytest Coverage
 - Demonstre o HTML gerado
 - Se não estiver em 100% de cobertura, explique o porquê.

- Divirta-se!