**Parte Teórica - Construindo um Projeto Ágil no GitHub: Da Gestão ao Controle de Qualidade**

**Descrição do Projeto e seu Escopo Inicial**

O projeto consiste no desenvolvimento de um sistema de gerenciamento de tarefas para a TechFlow Solutions, uma empresa fictícia contratada por uma startup de logística. O escopo inicial visa criar uma aplicação web que permita acompanhar o fluxo de trabalho em tempo real, priorizar tarefas críticas e monitorar o desempenho da equipe, utilizando metodologias ágeis e o GitHub como plataforma de controle. O sistema implementado inclui funcionalidades básicas de CRUD (Create, Read, Update, Delete) para gerenciamento de tarefas, conforme visto no código em Flask.

**Metodologia Ágil Utilizada**

Foi adotada a metodologia Kanban, implementada por meio da aba Projects do GitHub, com colunas "A Fazer", "Em Progresso" e "Concluído". Essa escolha reflete a necessidade de uma gestão visual e flexível, permitindo a adaptação contínua do fluxo de trabalho e a priorização dinâmica das tarefas, alinhando-se às práticas ágeis estudadas.

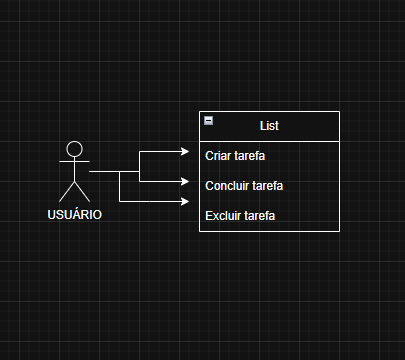
**Importância da Modelagem na Engenharia de Software**

A modelagem é fundamental para estruturar o projeto, clarificar requisitos e facilitar a comunicação entre a equipe e os stakeholders. No contexto deste trabalho, ela garante que o sistema seja desenvolvido de forma organizada, reduzindo erros e servindo como base para a implementação do código, especialmente ao definir as interações e classes do sistema.

**Diagramas UML Obrigatórios**

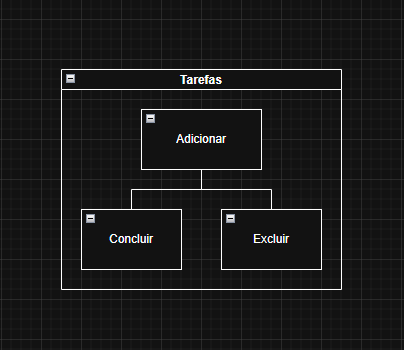
**Diagrama de Casos de Uso**

* **Atores**: Usuário (Gerente, Membro da Equipe)
* **Casos de Uso**:
  + Adicionar Tarefa
  + Marcar Tarefa como Concluída
  + Excluir Tarefa
  + Visualizar Lista de Tarefas
* **Descrição**: O sistema permite que o gerente adicione e gerencie tarefas, enquanto os membros da equipe atualizam o status e consultam a lista.



**Diagrama de Classes**

* **Classes**:
  + Tarefa (atributos: título, descrição, completada)
  + Usuário (atributos: nome, cargo)
* **Relações**:
  + Um Usuário gerencia várias Tarefa (associação 1:N)



**Breve Justificativa sobre a Mudança de Escopo**

Durante o desenvolvimento, decidiu-se adicionar uma funcionalidade de notificação de tarefas pendentes por mais de 7 dias, justificando-se pela necessidade de aumentar a proatividade da equipe. Essa mudança foi documentada no README.md e refletida no quadro Kanban com um novo card.

**Explicação sobre os Testes Automatizados Utilizados**

Os testes automatizados foram implementados com PyTest, focando na validação de entradas (ex.: título não vazio) e na funcionalidade do CRUD. Essa prática, integrada ao pipeline de GitHub Actions, assegura a qualidade do código, detectando erros precocemente e promovendo uma entrega confiável.

**Prints Comentados do GitHub**

* **Kanban com Tarefas**: Imagem mostrando as colunas "A Fazer", "Em Progresso" e "Concluído" com pelo menos 10 cards distribuídos.
* **Commits Relevantes**: Captura de tela exibindo commits como "Inicializar projeto com Flask" e "Adicionar funcionalidade de notificação".