

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 simples de gerenciamento de tarefas utilizando princípios básicos da Engenharia de Software. O sistema permite criar, listar, atualizar e apagar tarefas. Cada tarefa possui título, descrição, status e prioridade.

O foco é demonstrar:

- Organização do código
- Uso de metodologias ágeis (Kanban)
- Boas práticas de versionamento (commits)
- Testes automatizados
- Integração contínua (GitHub Actions)

O escopo inicial do sistema incluía:

- Criar tarefa
- Listar tarefas
- Atualizar tarefa
- Apagar tarefa

As tarefas deveriam conter apenas:

- Título
- Descrição
- Status

★ Metodologia ágil utilizada (ex: Scrum, Kanban ou híbrido)

Foi utilizada a metodologia Kanban, com 3 colunas:

- To do (A Fazer) — tudo o que precisa ser desenvolvido
- In progress (Em Progresso) — o que está sendo feito
- Done (Concluído) — o que já foi finalizado

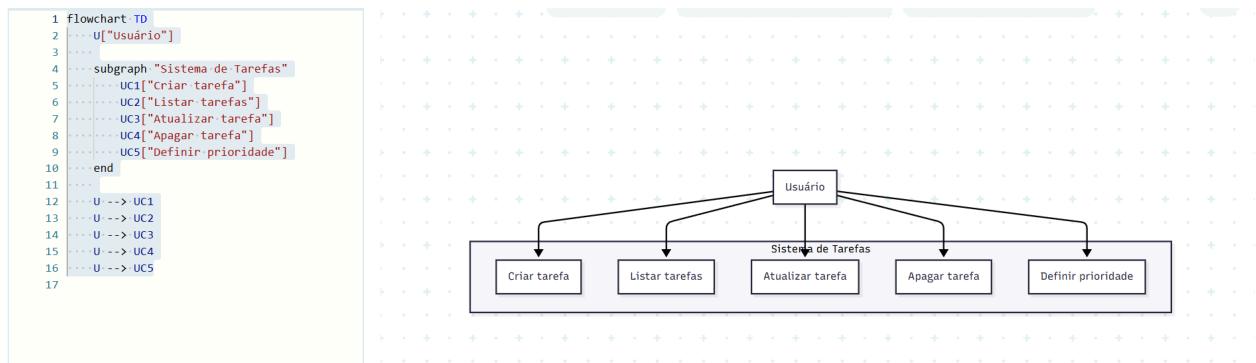
Cada funcionalidade virou um cartão no Kanban, permitindo organizar o fluxo de trabalho e acompanhar o progresso.

★ Explicação sobre a importância da modelagem na engenharia de software

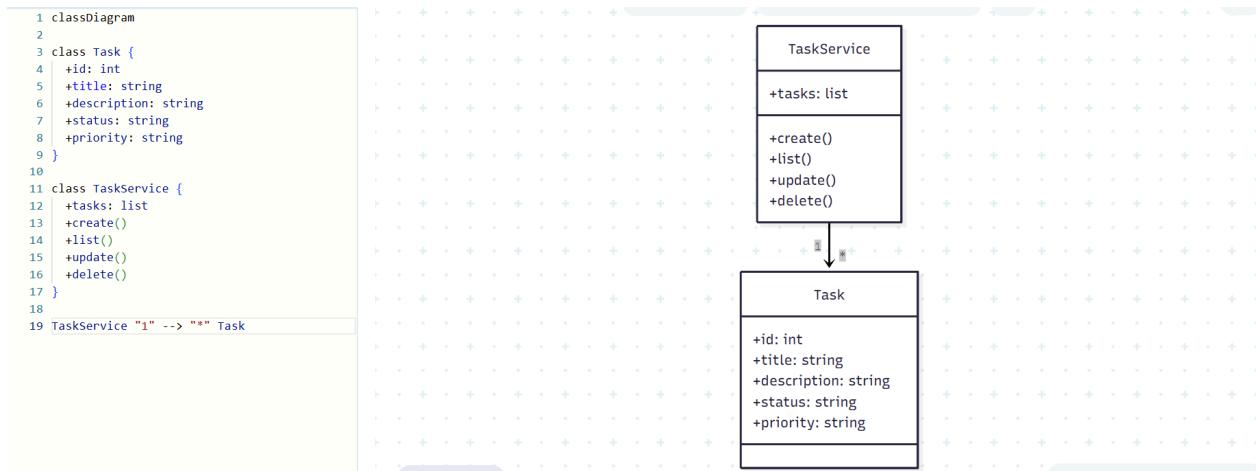
A modelagem permite visualizar e organizar o funcionamento do sistema antes da implementação. Diagramas UML ajudam a entender a estrutura das classes, os relacionamentos e como o usuário interage com o sistema. Isso reduz erros, melhora o entendimento e facilita futuras mudanças.

★ Diagramas UML:

1. Diagrama de Casos de Uso



2. Diagrama de Classes



★ Breve justificativa sobre a mudança de escopo feita ao longo do projeto

Após a finalização da primeira versão do sistema, foi solicitado um ajuste no escopo: *adicionar prioridade às tarefas*.

Nova funcionalidade: Prioridade (low, medium, high)

Justificativa: A startup precisava identificar rapidamente tarefas mais urgentes. A prioridade melhora o fluxo de trabalho, aumenta a eficiência e torna o sistema mais útil em um ambiente ágil.

★ Explicação sobre os testes automatizados utilizados

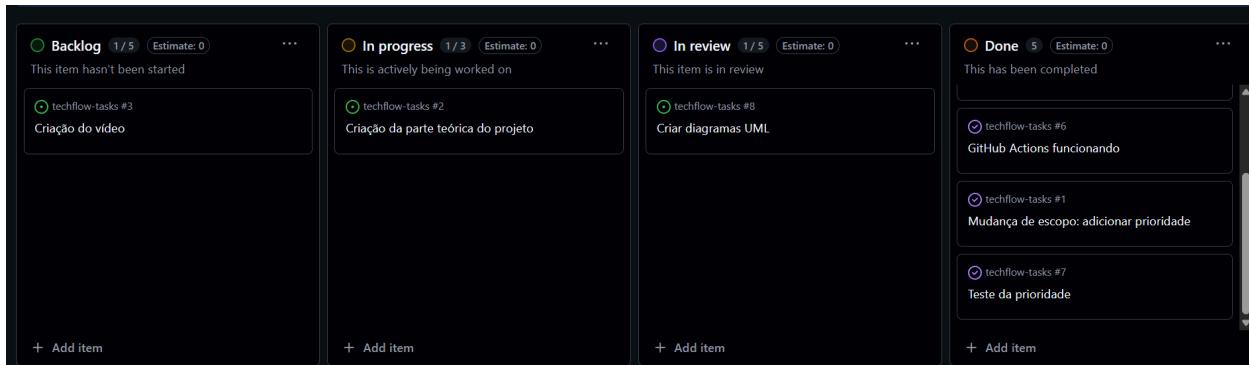
Foram desenvolvidos testes com Pytest para validar:

- Criação de tarefas
- Listagem
- Atualização
- Exclusão
- Prioridade

Todos os testes são executados automaticamente no GitHub Actions (CI), garantindo que cada mudança no código continue funcionando corretamente.

★ Prints comentados do GitHub mostrando:

- *Kanban com tarefas*



No GitHub, na seção “Projects”, organizei as atividades do trabalho usando um quadro Kanban. Cada funcionalidade do sistema foi registrada como um card. Conforme o desenvolvimento avançava, os cards foram sendo movidos pelas colunas: Backlog, In Progress, In Review e Done, refletindo exatamente o andamento do projeto.

- *Commits relevantes*



Implementado teste para o endpoint de prioridade da tarefa	...	Verified	3ec9a55		
<i>MarsInMarte authored 48 minutes ago · ✓ 1 / 1</i>					
Adicionado atualização de prioridade à modificação da tarefa	...	Verified	bd99539		
<i>MarsInMarte authored 48 minutes ago · ✓ 1 / 1</i>					
Adicionado prioridade à criação de tarefas.	...	Verified	870e291		
<i>MarsInMarte authored 50 minutes ago · ✓ 1 / 1</i>					
Implementado teste de exclusão de tarefas	...	Verified	7539482		
<i>MarsInMarte authored 54 minutes ago · ✓ 1 / 1</i>					
Adicionado o endpoint delete_task para remover tarefas.	...	Verified	717dd8b		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Adicionado teste para atualização de tarefa	...	Verified	a3d691c		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Implementação do endpoint de atualização de tarefa	...	Verified	bacf5f6		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Adicionado teste para listar tarefas em test_basico.py	...	Verified	bd6e576		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Adicionado endpoint para listar todas as tarefas	...	Verified	1b5ac81		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Teste para verificar o action.	...	Verified	53c0a4d		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					

Arrumando o caminho de importação para o aplicativo em test_basico.py.	...	Verified	31f2ee2		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Adicionado teste para criação de tarefas no aplicativo.	...	Verified	fd0aaa1		
<i>MarsInMarte authored 1 hour ago · ✘ 0 / 1</i>					
Implementado endpoint de criação de tarefa	...	Verified	62a3fc9		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Inicializado a lista de tarefas e a próxima variável de ID.	...	Verified	138e9bf		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Fluxo de trabalho de CI para testes com Python.	...	Verified	b1c84b5		
<i>MarsInMarte authored 1 hour ago · ✓ 1 / 1</i>					
Exemplo de teste básico para verificação.	...	Verified	443a3a1		
<i>MarsInMarte authored 1 hour ago</i>					
Adicionado Flask e Pytest aos requisitos.	...	Verified	0014797		
<i>MarsInMarte authored 1 hour ago</i>					
Inicialização do aplicativo Flask com uma rota inicial.	...	Verified	a4e854b		
<i>MarsInMarte authored 1 hour ago</i>					
Criação da pasta src e arquivo app.py	...	Verified	abbc15d		
<i>MarsInMarte authored 1 hour ago</i>					
Initial commit	...	Verified	b6887cf		
<i>MarsInMarte authored 1 hour ago</i>					

Durante o desenvolvimento, cada alteração feita no código foi registrada por meio de commits no GitHub. Cada commit recebeu um nome descritivo, indicando exatamente o que foi modificado — como criação de rotas, implementação de testes, ajustes no código e a mudança de escopo para adicionar prioridade. Isso permitiu acompanhar facilmente a evolução do projeto e manter um histórico organizado e transparente de tudo o que foi feito.

- *Pastas do projeto*

	MarsInMarte	Revisão da alteração de escopo para incluir o recurso de prioridad...	...	6d92fb8 · 48 minutes ago	22 Commits
	.github/workflows	Fluxo de trabalho de CI para testes com Python.		1 hour ago	
	src	Adicionado atualização de prioridade à modificação da tarefa		52 minutes ago	
	tests	Implementado teste para o endpoint de prioridade da tarefa		51 minutes ago	
	README.md	Revisão da alteração de escopo para incluir o recurso de pri...		48 minutes ago	
	requirements.txt	Adicionado Flask e Pytest aos requisitos.		1 hour ago	

A estrutura de pastas do projeto foi organizada para manter tudo separado e fácil de entender. Dentro da pasta `src`, ficou o arquivo principal da aplicação Flask. Já os arquivos de teste foram colocados em uma pasta dedicada, garantindo organização e facilitando a manutenção. Essa separação ajuda a manter o código limpo e permite localizar rapidamente cada parte do sistema.

- *Teste*

Os testes automatizados foram criados usando Pytest, cobrindo todas as funcionalidades principais: criar, listar, atualizar e deletar tarefas, além da prioridade. Eles servem para garantir que cada parte do sistema esteja funcionando corretamente. Sempre que um teste é executado, ele simula requisições reais à API e verifica se as respostas estão corretas. Isso deixa o projeto muito mais confiável.

- *Workflow de CI funcionando*

CI		ci.yml	Filter workflow runs	...		
17 workflow runs			Event ▾	Status ▾	Branch ▾	Actor ▾
✓	Revisão da alteração de escopo para incluir o recurso de priorid...	main	51 minutes ago	...	51s	...
CI #17:	Commit 6d92fb8 pushed by MarsInMarte					
✓	Aprimoração do arquivo README com detalhes e instruções do ...	main	52 minutes ago	...	12s	...
CI #16:	Commit d80d66a pushed by MarsInMarte					
✓	Implementado teste para o endpoint de prioridade da tarefa	main	54 minutes ago	...	15s	...
CI #15:	Commit 3ec9a55 pushed by MarsInMarte					
✓	Adicionado atualização de prioridade à modificação da tarefa	main	1 hour ago	...	13s	...
CI #14:	Commit bd99539 pushed by MarsInMarte					
✓	Adicionado prioridade à criação de tarefas.	main	1 hour ago	...	14s	...
CI #13:	Commit 870e291 pushed by MarsInMarte					

✓	Implementado teste de exclusão de tarefas	main	Today at 5:50 PM	...	13s	...
CI #12:	Commit 7539482 pushed by MarsInMarte					
✓	Adicionado o endpoint delete_task para remover tarefas.	main	Today at 5:50 PM	...	12s	...
CI #11:	Commit 717dd8b pushed by MarsInMarte					
✓	Adicionado teste para atualização de tarefa	main	Today at 5:48 PM	...	11s	...
CI #10:	Commit a3d691c pushed by MarsInMarte					
✓	Implementação do endpoint de atualização de tarefa	main	Today at 5:48 PM	...	10s	...
CI #9:	Commit bacf5f6 pushed by MarsInMarte					
✓	Adicionado teste para listar tarefas em test_basico.py	main	Today at 5:43 PM	...	14s	...
CI #8:	Commit bd6e576 pushed by MarsInMarte					
✓	Adicionado endpoint para listar todas as tarefas	main	Today at 5:42 PM	...	14s	...
CI #7:	Commit 1b5ac81 pushed by MarsInMarte					
✓	Teste para verificar o action.	main	Today at 5:34 PM	...	14s	...
CI #6:	Commit 53c0a4d pushed by MarsInMarte					
✓	Arrumando o caminho de importação para o aplicativo em test...	main	Today at 5:32 PM	...	13s	...
CI #5:	Commit 31f2ee2 pushed by MarsInMarte					

✖ Adicionado teste para criação de tarefas no aplicativo. CI #4: Commit fd0aaa1 pushed by MarsInMarte	main	Today at 5:28 PM 🕒 12s	...
✓ Implementado endpoint de criação de tarefa CI #3: Commit 62a3fc9 pushed by MarsInMarte	main	Today at 5:27 PM 🕒 14s	...
✓ Inicializado a lista de tarefas e a próxima variável de ID. CI #2: Commit 138e9bf pushed by MarsInMarte	main	Today at 5:26 PM 🕒 12s	...
✓ Fluxo de trabalho de CI para testes com Python. CI #1: Commit b1c84b5 pushed by MarsInMarte	main	Today at 5:25 PM 🕒 13s	...

Foi configurado um workflow de Integração Contínua (CI) no GitHub Actions. Ele executa automaticamente todos os testes sempre que um commit é enviado para o repositório. Se algo estiver errado, o workflow marca a execução como falha; se tudo estiver certo, fica verde. Isso garante que o código no repositório principal esteja sempre funcionando e que erros sejam detectados rapidamente.