

TRABALHO - N704

Neil Tran
Industrial Design
2025

Disciplina: Programação Funcional
Atividade: Atividade Parcial - N704
Tema do Projeto: Gerenciador Funcional
de Tarefas de Alunos (Task Analyzer)
Data de Entrega: 29/09/2025

Repositório:

[Repositório no GitHub](#)

Integrantes:

Paulo Ricardo de Castro Sousa – Matrícula: 2326189
Rangel Amaral Ferreira – Matrícula: 2327067
João Pedro Pereira Alves – Matrícula: 2326205
Moacir Cadmiel Silva Dos Santos – Matrícula: 2318038

Definição de Papéis na Equipe

Documentação e Requisitos:

Membro Designado: Membro 1

Responsabilidade Principal: Elaboração do Documento, RNF, Mapeamento e Relatório Final

Implementação Principal

Membro Designado: Membro 2

Responsabilidade Principal: Codificação, foco nas funções de Alta Ordem e Closure.

Implementação Funcional e GitHub

Membro Designado: Membro 3

Responsabilidade Principal: Codificação (Foco em Lambda e List Comprehension), commit e upload para o GitHub.

Testes e Qualidade

Membro Designado: Membro 4

Responsabilidade Principal: Elaboração e execução dos Casos de Teste e garantia de execução sem erros.

Requisitos Funcionais (RF) e Não-Funcionais (RNF)

RF1

O sistema deve listar os IDs das tarefas que ainda estão pendentes (não concluídas).

Função Mapeada no Código: `listar_ids_pendentes`

RF2

O sistema deve permitir ordenar tarefas por qualquer campo (ex: prioridade ou prazo).

Função Mapeada no Código: `ordenar_tarefas`

RF3

O sistema deve permitir a aplicação de um filtro baseado em um prazo limite definido dinamicamente.

Função Mapeada no Código: `criar_filtro_por_prazo`

RF4

O sistema deve permitir aplicar uma função de processamento customizada a todas as tarefas da lista.

Função Mapeada no Código: `processar_tarefas`

RF5

O sistema deve adicionar um campo de urgência (ALTA_URGENCIA) em tarefas com prioridade ≥ 4 .

Função Mapeada no Código: `adicionar_campo_urgencia` (Usada pelo RF4)

RF6

O sistema deve permitir marcar uma tarefa específica como concluída, gerando um novo estado.

Função Mapeada no Código: `marcar_como_concluida`

Requisitos Funcionais (RF) e Não-Funcionais (RNF)

RNF1

O sistema deve ser desenvolvido utilizando a linguagem Python, que possui suporte ao paradigma funcional.

Função Mapeada no Código: Todas as funções

RNF2

O sistema deve simular a imutabilidade de dados, garantindo que as operações não alterem a base de dados (TAREFAS_DB) original.

Função Mapeada no Código: marcar_como_concluida

RNF3

O código deve garantir a manutenibilidade, separando o processamento da definição dos dados.

Função Mapeada no Código: processar_tarefas

RNF4

O programa deve ser executado corretamente, sem erros ou exceptions.

Função Mapeada no Código: executar_testes

Utilização dos Conceitos de Programação Funcional

Função Lambda

Função Mapeada: ordenar_tarefas

Justificativa no Código: Passada como argumento `key=lambda t: t.get(...)` para a função `sorted`.

List Comprehension

Função Mapeada: listar_ids_pendentes

Justificativa no Código: Utilizada para filtrar IDs de tarefas não concluídas de forma declarativa.

Closure

Função Mapeada: criar_filtro_por_prazo

Justificativa no Código: A função interna `filtro_prazo` fecha sobre e utiliza a variável `prazo_limite` (escopo externo).

Função de Alta Ordem

Função Mapeada: processar_tarefas

Justificativa no Código: Recebe uma função (processador) como argumento e a aplica à coleção.

Casos de Teste

T1 (Lambda)

Objetivo: Ordenação por prioridade

Resultado Esperado (Validado no Terminal): Ordenação correta por prioridade. Tarefas 101 e 104 (Prioridade 5) devem vir primeiro.

Status: OK

T3 (Closure)

Objetivo: Filtro por prazo (prazo_limite: '2025-09-27').

Resultado Esperado (Validado no Terminal): Apenas a Tarefa 104 deve ser retornada (pendente e prazo antes ou igual ao limite).

Status: OK

T4 (HO Function)

Objetivo: Aplicação do status ALTA_URGENCIA.

Resultado Esperado (Validado no Terminal): Tarefas 101, 103 e 104 devem ter o novo status.

Status: OK

T5 (Imutabilidade)

Objetivo: Verificação de alteração de estado (concluida).

Resultado Esperado (Validado no Terminal): Tarefa 102 deve ser True na nova lista, e a lista original (TAREFAS_DB) permanece inalterada.

Status: OK

USO DE I.A

O grupo utilizou ferramentas de IA, Gemini, como um auxílio à consulta, otimização e validação de conceitos, conforme previsto nas orientações da atividade.

O uso da IA concentrou-se nas seguintes áreas:

1. Otimização do código: A IA foi utilizada para gerar sugestões do código inicial das funções de Programação Funcional (PF). Esse código serviu como ponto de partida para conferência, absorção de conhecimento e posterior otimização pela equipe, garantindo que a implementação final fosse mais limpa e didática.
2. Estruturação do Relatório: A IA foi utilizada para estruturar o template do Documento de Requisitos, garantindo que todos os requisitos de avaliação, incluindo a separação de RF, RNF e o Mapeamento de PF, fossem cobertos. Foi de extrema importância, pois com ela, podemos ver se estávamos dentro de todos os requisitos da atividade ou não.

O código final foi integralmente revisado, adaptado e validado pela equipe, garantindo que a lógica e a execução reflitam o entendimento dos conceitos de Programação Funcional.