



Universidade de Fortaleza

Curso: Análise e Desenvolvimento de Sistemas

Docência: Bruno Lopes Alcantara

Tutoria: Esther Rodrigues Ellery

Relatório - Atividade Parcial N704.25.2

Membros da Equipe:

Millene de Souza Júnior – 2326165

Herison Daniel Wanderley – 2315221

Talles de Lima Pereira – 2326201

João Eduardo Lúcio Araújo – 2913569

Proposta de Atividade – N704 Programação Funcional

Objetivo:

Aplicar os conceitos de Programação Funcional em Python através do desenvolvimento de um pequeno projeto que utilize funções lambda, list comprehensions, closures e funções de alta ordem.

Atribuição de Funções:

- **Millene Júnior** – Documentação, primeira fase de implementação, primeiros testes e relatório final.
- **Herison Daniel** – Segunda fase de implementação e integração das funcionalidades.
- **Talles de Lima** – Terceira fase de implementação, ajustes finais e codificação.
- **João Eduardo** – Segunda fase de testes, verificando a qualidade e funcionamento.

Requisitos Funcionais:

Requisito	Função no Código
Cálculo estatístico	sum_list, avg_list, max_list, min_list
Filtros personalizados	apply_filter
Uso de lambda	lambda x: x % 2 == 0
Histórico de cálculos (closure)	history_tracker
List comprehension	[x for x in lista if x % 2 == 0]
Função de alta ordem	map, filter ou reduce

Requisitos Não Funcionais:

- O sistema deve ser desenvolvido em Python 3.8+.
- O código deve estar disponível em um repositório GitHub.
- O sistema não deve apresentar erros em tempo de execução.
- O código deve estar organizado e comentado.

Casos de Teste:

Caso de Teste	Entrada	Saída Esperada
Soma de lista	[1, 2, 3]	6
Média da lista	[2, 4, 6]	4
Filtrar pares	[1, 2, 3, 4]	[2, 4]

Histórico de cálculo	último cálculo: média = 4	Retornar string
Lista vazia	[]	soma = 0, média = erro tratado, min/max = erro tratado
Valores negativos	[-5, -2, -3]	soma = -10, média = -3.33, min = -5, max = -2
Histórico múltiplos cálculos	soma([1,2]), média([2,4])	histórico armazena mais de um cálculo

Execução sem Erros:

O sistema foi executado em Python 3.8+ e todos os testes rodaram sem erros em tempo de execução.

Repositório GitHub:

Todas as instruções de uso, incluindo como baixar o arquivo ZIP, estão disponíveis no repositório do GitHub. O código-fonte e este relatório podem ser acessados em:

[\[https://github.com/millenebrj95/Projeto-Programa-Funcional-Lambda.git\]](https://github.com/millenebrj95/Projeto-Programa-Funcional-Lambda.git)

Projeto - Programação Funcional - Lambda

Em caso de dúvidas, entre em contato com o perfil que publicou o projeto no GitHub, ou com os demais colaboradores:

- [millenebrj95](#) – *Millene Júnior*
- [HerisonDaniel](#) – *Herison Daniel*
- [PereiraTalles](#) – *Talles de Lima*
- [DuDu0133](#) – *João Eduardo*

Observação:

Parte do desenvolvimento contou com apoio do ChatGPT, usado para estruturar o documento de requisitos, gerar exemplos de código e elaborar este relatório. Todas as respostas foram adaptadas pela equipe.