

Grupo 1: Cobra Coders

Autores

- André Carioca
- Diego Teixeira
- Marcelo Zilotti
- Mirra Bernardo
- Tales Honorio
- William Lopes

Visão Geral

Este documento apresenta uma análise crítica e detalhada do projeto submetido no módulo 1 da formação da Globo. O objetivo é avaliar a implementação do sistema de análise de interações com conteúdos, considerando boas práticas de programação, clareza lógica, uso adequado dos recursos da linguagem Python e cumprimento dos requisitos propostos no enunciado.

CONFORMIDADE COM OS REQUISITOS DO PROJETO

Leitura e Estruturação Inicial:

O código realiza a leitura do CSV com uso de DictReader e organiza os dados em lista de dicionários.

Há uma preocupação com portabilidade (caminho_arquivo) e salvamento do arquivo tratado.

Limpeza e Validação de Dados:

A função tratar_linha trata campos de texto, lida com valores ausentes e aplica try/except.

Usa filter() e map() para aplicar validações e transformações funcionalmente.

Estruturação de Dados:

A função agrupar_por_conteudo cria a estrutura sugerida no enunciado com justificativa implícita.

CÁLCULO DE MÉTRICAS

Todas as métricas estão presentes:

1. Total de Interações por Conteúdo

2. Contagem por Tipo de Interação
3. Tempo Total de Visualização
4. Média de Visualização
5. Listagem de Comentários
6. Top-5 mais visualizados

Todas estão bem organizadas em funções separadas com uso de defaultdict e compreensão de listas.

USO DOS RECURSOS ABORDADOS EM AULA

- Listas e Dicionários: Extensivo
- Compreensão de Listas: Parcial
- Programação funcional (map, filter): Bem usado
- Exceções (try-except): Rigoroso
- Funções: Bem organizadas
- Comentários: Claros e explicativos

Criatividade e Boas Práticas

- **Diferenciais notáveis:**

- Uso de separação clara entre camadas (leitura, tratamento, análise).
- Uso de `os.path` para tornar o código portátil.
- Uso de abstrações elegantes como `linha_valida()` e `limpar_none_keys()`.

- **Sugestões de melhoria:**

- Poderia haver uma função que imprime todas as métricas de forma consolidada.
- A estrutura do projeto poderia ser explicada via docstring inicial ou README.

Considerações finais.

- Entendimento profundo de manipulação de dados com listas e dicionários: as funções desenvolvidas fazem uso eficaz dessas estruturas, permitindo organizar e agrupar informações de forma clara e eficiente. O uso de ``defaultdict``, compreensão de listas e iteração sobre dicionários evidencia esse domínio.
- Aplicação de boas práticas de modularização: o código está bem segmentado em funções com responsabilidades distintas. Funções como ``tratar_linha``, ``agrupar_por_conteudo``, ``calcular_interacoes_por_conteudo`` e ``listar_comentarios_por_conteudo`` seguem princípios de coesão, facilitando testes e manutenção.
- Clareza de raciocínio e lógica robusta para tratamento de exceções e limpeza de dados: as etapas de tratamento de campos numéricos e textuais estão encapsuladas em funções específicas, com blocos ``try-except`` bem utilizados. As validações implementadas garantem que dados inconsistentes não interrompam o fluxo da aplicação.
- De forma geral, o trabalho reflete um forte entendimento dos conceitos abordados nesse primeiro módulo. O grupo demonstrou organização, domínio técnico e maturidade no desenvolvimento de um pipeline que cumpre todas as etapas propostas: carregamento, limpeza, estruturação e cálculo de métricas. Excelente trabalho!