

Relatório Geral de Testes Estáticos de Código

O código foi analisado com o Pylint, revelando padrões comuns e pontos de atenção que afetam legibilidade, manutenção e aderência a boas práticas.

Principais Problemas Identificados

Formatação e estilo

- Linhas muito longas (>100 caracteres)
- Espaços em branco no final de linhas (trailing whitespaces)
- Indentação inconsistente
- Imports fora de ordem ou não no topo dos arquivos
- Final de arquivo sem nova linha

Padronização de nomes

- Variáveis, atributos e argumentos fora do padrão `snake_case`
- Constantes sem convenção consistente (ex.: `REQUIRED_COLS`)

Docstrings

- Falta de docstrings em módulos e funções na maioria dos arquivos

Boas práticas

- Captura de exceções genéricas (`Exception`)
- Uso de construções desnecessárias (ex.: `dict()` ao invés de `{}`)
- Variáveis não utilizadas
- Uso de f-strings sem interpolação
- Funções com muitos parâmetros locais

Arquivos impactados

main.py: nota Pylint mais baixa (6,44/10), com múltiplos problemas de estilo, exceções genéricas e variáveis não utilizadas.

graph_builder.py, graph_service.py, search.py: problemas de linhas longas, nomes de variáveis, imports e exceções genéricas.

processor.py: problemas menores, principalmente docstrings, nomes de atributos e imports; nota mais alta (9,26/10).

Sugestões Gerais

1. Corrigir linhas muito longas e remover trailing whitespaces.
2. Padronizar nomes de variáveis, atributos e argumentos em `snake_case`.
3. Adicionar docstrings em módulos e funções.
4. Evitar captura de exceções genéricas, utilizando exceções específicas.
5. Organizar imports: biblioteca padrão → terceiros → internos.
6. Revisar código para remover variáveis não utilizadas e otimizar f-strings.

Nota Média

8,05/10 (considerando todos os módulos analisados)

Indica que o código está funcional, porém melhorias de estilo e boas práticas são necessárias para aumentar legibilidade e manutenção.

Conclusão

O código apresenta **padrões de formatação inconsistentes** (linha longa, trailing whitespaces, nomes de variáveis), **captura de exceções genéricas** e **faltam docstrings** em vários módulos. Apesar disso, a maioria dos módulos ainda mantém notas relativamente altas (8+) exceto o `main.py` (6.44/10), indicando que, estruturalmente, o código está funcional, mas precisa de ajustes de estilo e boas práticas para manter a manutenção e legibilidade.