Case Tecnico – GoParts (Clara)

Data: 27/08/2025

Objetivo

Avaliar sua capacidade de fuçar (autonomia, curiosidade e persistencia) para resolver problemas reais do dia a dia na GoParts, alem de habilidades basicas de Python.

Entregaveis

- Script(s) Python bem organizado(s), com funcoes e instrucoes de execucao.
- Um CSV limpo normalizado a partir do arquivo fornecido.
- Relatorio curto (README.md) explicando suas decisoes (como tratou encoding, parsing de preco, validacoes e como testou retries).

Parte 1 – Normalizacao de CSV

1. Voce recebeu um arquivo CSV com 500 linhas (produtos de autopecas). Colunas: nome_produto, codigo, preco, estoque.
2. Problemas propositais: precos com formatos mistos (100,50 / 100.50 / R$100,50 / R$ 100,50), campos vazios e valores invalidos no estoque (-1, 'None', 'n/a'), espacos extras e nomes com acentuacao.
3. Tarefa: normalizar o arquivo para que: (a) preco vire float padrao (ponto como separador decimal), (b) estoque invalido ou vazio se torne 0, (c) nomes sejam stripados (sem espacos sobrando). Salve um novo CSV limpo.

Parte 2 – Encoding (acentos quebrados)

O arquivo fornecido esta propositalmente em ISO-8859-1 (Latin-1). Descubra e ajuste o encoding ao ler e/ou ao gravar, garantindo que acentos aparecam corretamente (UTF-8).

Parte 3 – Integracao com API (com falhas intermitentes)

1. Crie um script que leia o CSV limpo e faca POST de cada produto para uma API REST (pode ser uma API simples sua em Flask/FastAPI ou usar httpbin.org simulando respostas).
2. Implemente retries com backoff (ex.: exponencial simples), tratando excecoes como timeouts e HTTP 5xx, ate um limite de tentativas.
3. Registre logs do resultado de cada envio (sucesso/erro) e um sumario ao final.

Requisitos Tecnicos

- Python 3.9+ (sua escolha de libs padrao; requests permitido).
- Codigo organizado em funcoes/modulos, com docstrings e comentarios claros.
- Tratamento robusto de erros (try/except), logs e prints informativos.
- Nao precisa de interface grafica; scripts por linha de comando bastam.

Criterios de Avaliacao

- Persistencia e autonomia: como voce investigou encoding, formatos e erros da API.

- Qualidade do codigo: clareza, organizacao, legibilidade e testes manuais.
- Capacidade de comunicacao: README conciso explicando decisoes e instrucoes para executar.
- Resultado: CSV limpo correto e rotina de envio com retries funcionando.

Dicas (opcionais)

- Para encoding: tente abrir o arquivo especificando 'encoding' e, se necessario, detectar com chardet/charset-normalizer.
- Para normalizar preco: remova 'R$' e espacos, troque virgula por ponto quando apropriado e converta para float.
- Para retries: use tempo de espera crescente (ex.: 1s, 2s, 4s...) e pare apos N tentativas.
- Para testes locais de API: crie um endpoint que falhe aleatoriamente (ex.: retorna 500 em 30% das requisicoes).

Como enviar

- Suba tudo em um repositorio privado (GitHub/GitLab) ou compacte em .zip e envie por e-mail.
- Inclua instrucoes claras de execucao no README.md (comandos e dependencias).

Arquivos fornecidos

• produtos_bagunçados_latin1.csv – CSV com 500 linhas e encoding Latin-1 (ISO-8859-1).