



**GRADUAÇÃO 4D**  
**PUCPR**



■ Projeto da Disciplina

## AI Factory: Use Case Analysis



## ETAPA 2

### Codificação de soluções com apoio de IA generativa

\*\*\* Entrega semana 10.

#### Descrição da etapa:

Na etapa anterior você trabalhou com ferramentas de IA generativa no apoio a tarefas diárias de profissionais e times de vários departamentos da empresa. O foco agora é exclusivamente na solução de problemas computacionais da empresa, que exigem a codificação de *scripts* em determinada linguagem de programação, neste caso, mais especificamente o Python. A seguir, passaremos por todos os detalhes e especificações requeridas para o desenvolvimento da etapa, e ao final, todos os entregáveis serão resumidos em uma listagem única.

Você deve selecionar qualquer uma das ferramentas de IA generativa disponíveis para a geração dos códigos para o seu projeto. Confira algumas possibilidades: **ChatGPT**, **Microsoft Copilot**, **Github Copilot**, **Bing**, **Bard**, **ChatSonic**, **JasperChat**, **YouChat**, **PerplexityAI**, **Claude**.

#### Desenvolvimento de rotina de Integração dos dados de vendas

O seu cliente vende parte de seus produtos/serviços *on-line*, em uma plataforma de *e-commerce*. A plataforma por contrato protege o acesso ao seu banco de dados, porém, permite exportar diariamente as vendas realizadas em um formato de texto simples (TXT), em que cada linha representa uma venda, incluindo informações como ID do produto/serviço, nome do produto/serviço, quantidade vendida, preço unitário, e data da venda. Recentemente, a empresa decidiu integrar seu sistema de vendas (*e-commerce*) com um novo software de gerenciamento de estoque que aceita dados no formato JSON e XML.

O seu desafio é desenvolver um *script* em Python que leia o arquivo de vendas em formato TXT, processe esses dados, e os converta para os dois formatos desejados (JSON e XML). Você deve utilizar uma ferramenta de IA generativa para contextualizar o problema e pedir que um *script* Python seja gerado. A seguir um exemplo de como é o arquivo de entrada, e como seriam os arquivos de saída.

vendas.txt	vendas.json	vendas.xml
101,Produto 1,3,1200.00,2024-01-10  102,Serviço 1,5,800.00,2024-01-11  103,Produto 2,2,2500.00,2024-01-12	[ { "id_produto": "101", "nome_produto": "Produto 1", "quantidade": 3, "valor": 1200.00}, {"id_produto": "102", "nome_produto": "Serviço 1", "quantidade": 1,5, "valor": 800.00}, {"id_produto": "103", "nome_produto": "Produto 2", "quantidade": 2,2, "valor": 2500.00}]	<vendas>  <venda>  <id_produto>101</id_produto>  <nome_produto>Produto 1</nome_produto>  <quantidade>3</quantidade>  <venda>  <id_produto>102</id_produto>  <nome_produto>Serviço 1</nome_produto>  <quantidade>1,5</quantidade>  <venda>  <id_produto>103</id_produto>  <nome_produto>Produto 2</nome_produto>  <quantidade>2,2</quantidade>

104, Produto 3,4,600.00,2024-01-13	<pre>         "preco_unitario": 1200.00,         "data_venda":         "10/01/2024",         "data_importacao":         "30/01/2024"       },       {         "id_produto": "102",         "nome_produto": "Serviço 1",         "quantidade": 5,         "preco_unitario": 800.00,         "data_venda":         "11/01/2024",         "data_importacao":         "30/01/2024"       },       {         "id_produto": "103",         "nome_produto": "Produto 2",         "quantidade": 2,         "preco_unitario": 2500.00,         "data_venda":         "12/01/2024",         "data_importacao":         "30/01/2024"       },       {         "id_produto": "104",         "nome_produto": "Produto 3",         "quantidade": 4,         "preco_unitario": 600.00,       }     </pre>	<pre> &lt;preco_unitario&gt;1200.00&lt;/preco_unitario&gt; &lt;data_venda&gt;10/01/2024&lt;/data_venda&gt; &lt;data_importacao&gt;30/01/2024&lt;/data_importacao&gt;  &lt;/venda&gt; &lt;venda&gt; &lt;id_produto&gt;102&lt;/id_produto&gt; &lt;nome_produto&gt;Serviço 1&lt;/nome_produto&gt; &lt;quantidade&gt;5&lt;/quantidade&gt; &lt;preco_unitario&gt;800.00&lt;/preco_unitario&gt; &lt;data_venda&gt;11/01/2024&lt;/data_venda&gt; &lt;data_importacao&gt;30/01/2024&lt;/data_importacao&gt;  &lt;/venda&gt; &lt;venda&gt; &lt;id_produto&gt;103&lt;/id_produto&gt; &lt;nome_produto&gt;Produto 2&lt;/nome_produto&gt; &lt;quantidade&gt;2&lt;/quantidade&gt; &lt;preco_unitario&gt;2500.00&lt;/preco_unitario&gt; &lt;data_venda&gt;12/01/2024&lt;/data_venda&gt; &lt;data_importacao&gt;30/01/2024&lt;/data_importacao&gt;  &lt;/venda&gt; &lt;venda&gt; &lt;id_produto&gt;104&lt;/id_produto&gt; &lt;nome_produto&gt;Produto 3&lt;/nome_produto&gt; &lt;quantidade&gt;4&lt;/quantidade&gt; &lt;preco_unitario&gt;600.00&lt;/preco_unitario&gt; </pre>
------------------------------------	--	---

	<pre>     "data_venda":      "13/01/2024",        "data_importacao":      "30/01/2024"        },        {          "id_produto": "105",          "nome_produto": "Serviço        2",          "quantidade": 10,          "preco_unitario": 150.00,          "data_venda":        "14/01/2024",          "data_importacao":        "30/01/2024"        }   ] </pre>	<pre> &lt;data_venda&gt;13/01/2024&lt;/data_venda&gt;    &lt;data_importacao&gt;30/01/2024&lt;/data_&gt;  importacao    &lt;/venda&gt;    &lt;venda&gt;    &lt;id_produto&gt;105&lt;/id_produto&gt;    &lt;nome_produto&gt;Serviço  2&lt;/nome_produto&gt;    &lt;quantidade&gt;10&lt;/quantidade&gt;    &lt;preco_unitario&gt;150.00&lt;/preco_unitario&gt;    &lt;data_venda&gt;14/01/2024&lt;/data_venda&gt;    &lt;data_importacao&gt;30/01/2024&lt;/data_&gt;  importacao    &lt;/venda&gt;    &lt;/vendas&gt; </pre>
--	--	--

Um detalhe importante é que no sistema de estoque o formato de datas aceito é o brasileiro, onde utilizamos o formato dia/mês/ano. Além disso, tanto o arquivo JSON quanto o XML devem conter um campo extra chamado “data\_importacao” para todos os registros, contendo a data em que foi executado o *script* de importação.

Além do desenvolvimento do *script*, você deve também construir um pequeno parecer técnico, a ser entregue ao responsável pelo setor de infraestrutura da empresa, recomendando qual tipo de arquivo (JSON ou XML) seria o mais recomendado para rodar uma operação mais econômica em termos de trânsito de dados.

#### ***Script de organização dos arquivos de mídia (e lá vem o departamento de marketing de novo...)***

O departamento de *marketing* tem enfrentado problemas na organização de seus arquivos. Eles lidam com uma grande quantidade de arquivos de mídia, incluindo imagens, vídeos, *slides* e documentos de texto. Estes arquivos que são salvos em um diretório compartilhado na rede, são frequentemente desorganizados e têm nomes não padronizados, o que dificulta a localização e o uso eficiente dos recursos. A empresa deseja automatizar o processo de renomeação e organização desses arquivos para melhorar a eficiência operacional.

O desafio é desenvolver um *script* em Python que automatize a tarefa de renomear e organizar arquivos em diretórios específicos com base em seus tipos e datas de criação. O *script* deve identificar arquivos em um diretório específico e ser capaz de reconhecer diferentes tipos de arquivos (*e.g.*, .jpg, .mp4, .docx, .pptx, .pdf).

Além disso, deve implementar uma funcionalidade para renomear os arquivos com um formato padrão que inclua a data de criação do arquivo, o tipo de mídia e um identificador único. Os arquivos devem ser automaticamente movidos para diretórios baseados em seu tipo. Por exemplo, imagens vão para a pasta "Imagens", vídeos para "Vídeos", documentos para "Documentos" e apresentações para "Apresentações".

Exemplo de formato de nome: 2024-01-30\_Imagen\_001.jpg, 2024-01-30\_Video\_002.mp4.

O script deve manter um registro (*log*) de todas as ações realizadas, incluindo nomes de arquivos originais, novos nomes e possíveis erros em um arquivo chamado "registro.log". Inclua tratamento de erros para lidar com situações como permissões de arquivo, nomes de arquivos duplicados e erros de leitura/escrita.

### Interpretação e atualização de script de dados

O recém-lançado departamento de ciência de dados da empresa, perdeu seu principal funcionário sênior, que era responsável por enviar mensalmente um relatório que apresentava uma análise de perfil dos clientes diretamente para a diretoria. Como agora ficaram apenas os funcionários de nível Junior da equipe, eles precisam de ajuda para compreender como este relatório era gerado, e como eles poderiam adaptar este script para realizar a leitura dos dois arquivos diferentes, um CSV e outro JSON, e juntar ambos em um único data frame para gerar as visualizações.

#### script.py

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats

df = pd.read_json('input.json')

summary_stats = df.describe()

plt.figure(figsize=(10, 6))
sns.histplot(df['Idade'], bins=10, kde=True)
plt.title('Distribuição')
plt.xlabel('Idade')
plt.ylabel('Contagem')
plt.show()

plt.figure(figsize=(10, 6))
```

```
sns.scatterplot(data=df,      x='Idade',      y='Pontuação_Satisfação',      hue='Salário',  
palette='coolwarm')  
plt.title('Dispersão')  
plt.xlabel('Idade')  
plt.ylabel('Índice')  
plt.show()  
  
correlation, p_value = stats.pearsonr(df['Idade'], df['Pontuação_Satisfação'])  
  
print(f'Correlação: {correlation:.2f}')  
print(f'Valor-p: {p_value:.4f}')  
  
plt.figure(figsize=(10, 6))  
sns.boxplot(data=df, x='Idade', y='Pontuação_Satisfação')  
plt.title('Box Plot')  
plt.xlabel('Idade')  
plt.ylabel('Índice')  
plt.show()
```

Você deve gerar um relatório, explicando linha a linha, o que o script acima faz, e apresentar uma versão atualizada do script, que contemple a leitura de dois diferentes arquivos e a sua junção para geração das visualizações. Lembre-se, caso você queira visualizar o *output do script* você também pode pedir a ferramenta de IA gerar um arquivo de exemplo que sirva como entrada ao seu script.

### Criação de *script* para outros departamentos

Para esta parte do projeto você irá escolher um departamento da empresa, e elaborar uma situação-problema que possa ser resolvida com um *script* em Python. A situação deve contemplar um caso que envolva obrigatoriamente as características específicas que você definiu sobre a empresa (*e.g.*, comércio de alimentos, venda de imóveis etc.), e obrigatoriamente deve manipular, transformar e apresentar dados de arquivos. Você deve utilizar as ferramentas de IA generativa para produção dos códigos-fonte em Python, sempre utilizando *prompts* adequados e contextualizados.



### RESUMO DOS ENTREGÁVEIS DA ETAPA 2:

- *Link* para vídeo hospedado no YouTube (ou outra plataforma aberta) com uma apresentação rápida de todos os programas desenvolvidos em funcionamento (máx. cinco minutos).
- Arquivo .py contendo o *script* para integração dos dados de vendas.
- Documento Word ou PDF contendo o parecer técnico sobre os arquivos JSON e XML (máx. uma página).
- Arquivo .py contendo o *script* para automatização e organização dos arquivos de mídia.
- Documento Word ou PDF contendo o relatório que explica o funcionamento do *script* do departamento de dados (máx. uma página).
- Arquivo .py contendo o *script* do departamento de dados atualizado para abertura de dois arquivos.
- Arquivo .py contendo em comentários a descrição da situação-problema elaborada, juntamente ao *script* gerado para a solução.
- Documento Word ou PDF contendo todo histórico de interações com a ferramenta de IA generativa durante a construção de todas as etapas da atividade.

**Todos os arquivos devem ser compactados e entregues em um único arquivo .zip.**

**\*\*\* Entrega na semana 10.**

 Observe na tabela abaixo os critérios pelos quais você será avaliado nesta etapa 2.



## RUBRICA DA ETAPA 2 (critérios de avaliação)

CRITÉRIOS	Autônomo (100% a 90%)	Capaz (89% a 70%)	Aprendiz (69% a 40%)	Em desenvolvimento (39% a 0)
Desenvolve rotina de integração de vendas. (2,0 pontos)	Desenvolve a rotina de integração de maneira <b>correta</b> , prevendo <b>todas</b> as regras pedidas, seguindo as boas práticas na utilização de ferramentas de IA generativa.	Desenvolve a rotina de integração de maneira <b>correta</b> , prevendo <b>alguma</b> regra pedida, seguindo as boas práticas na utilização de ferramentas de IA generativa.	Desenvolve a rotina de integração de maneira correta, prevendo <b>alguma regra pedida, infringindo ou não seguindo</b> alguma das boas práticas na utilização de ferramentas de IA generativa.	Desenvolve a rotina de integração <b>incorrectamente, e/ou infringindo</b> as boas práticas na utilização de ferramentas de IA generativa.
Constrói <i>script</i> de organização de arquivos de mídia. (2,0 pontos)	Constrói <i>script</i> de organização de arquivos de mídia <b>corretamente, seguindo</b> a nomenclatura, organização e <i>logs</i> pedidos, <b>segundo</b> as boas práticas na utilização de ferramentas de IA generativa.	Constrói <i>script</i> de organização de arquivos de mídia <b>seguindo parcialmente</b> a nomenclatura, organização e <i>logs</i> pedidos. <b>Segue</b> as boas práticas na utilização de ferramentas de IA generativa.	Constrói <i>script</i> de organização de arquivos de mídia <b>seguindo parcialmente</b> a nomenclatura, organização e <i>logs</i> pedidos. <b>Segue parcialmente</b> as boas práticas na utilização de ferramentas de IA generativa.	Constrói <i>script</i> de organização de arquivos de mídia <b>incorrectamente</b> . <b>Não segue</b> as boas práticas na utilização de ferramentas de IA generativa.
Descreve relatório explicação de <i>script</i> de dados e atualiza-o. (2,0 pontos)	Descreve relatório de explicação de <i>script</i> <b>corretamente</b> . Apresenta <i>script</i> atualizado de <b>maneira correta</b> . <b>Segue</b> as boas práticas na utilização de ferramentas de IA generativa.	Descreve relatório de explicação de <i>script</i> <b>corretamente</b> . Apresenta <i>script</i> atualizado de <b>maneira parcialmente correta</b> . <b>Segue</b> as boas práticas na utilização de ferramentas de IA generativa.	Descreve relatório de explicação de <i>script</i> <b>corretamente</b> . Apresenta <i>script</i> atualizado de <b>maneira parcialmente correta</b> . <b>Segue</b> <b>parcialmente</b> as boas práticas na utilização de ferramentas de IA generativa.	Descreve relatório de explicação de <i>script</i> <b>incorrectamente</b> . Apresenta <i>script</i> atualizado de <b>maneira parcialmente correta</b> . <b>Segue parcialmente</b> as boas práticas na utilização de ferramentas de IA generativa.
Cria <i>script</i> com solução para situação-problema gerada para outro departamento. (4,0 pontos)	Descreve situação-problema <b>corretamente</b> relacionada às características da empresa. Apresenta <i>script</i> de solução <b>corretamente e seguindo</b> as boas práticas na utilização de ferramentas de IA generativa.	Descreve situação-problema <b>corretamente</b> relacionada às características da empresa. Apresenta <i>script</i> de solução <b>parcialmente correto e seguindo</b> as boas práticas na utilização de ferramentas de IA generativa.	Descreve situação-problema <b>inconsistente</b> às características da empresa. Apresenta <i>script</i> de solução <b>parcialmente correto e seguindo</b> as boas práticas na utilização de ferramentas de IA generativa.	Descreve situação-problema <b>inconsistente</b> às características da empresa. Apresenta <i>script</i> de solução <b>parcialmente correto e seguindo parcialmente</b> as boas práticas na utilização de ferramentas de IA generativa.



GRADUAÇÃO 4D  
**PUCPR**

