



neo

bio

PROVA MIS SENIOR

• • •



neo

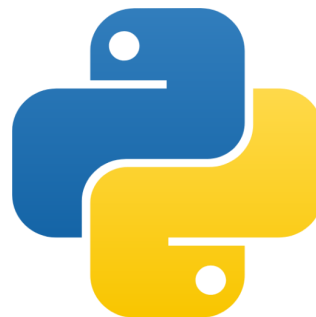
bpo

Instruções Gerais:

Prepare-se revisando os conceitos fundamentais e avançados nas áreas mencionadas.

Pratique escrevendo código limpo e eficiente, e familiarize-se com as funcionalidades avançadas do Power BI.

Pense em exemplos práticos que demonstrem sua capacidade de aplicar essas habilidades em cenários do mundo real.



Critérios de Avaliações:



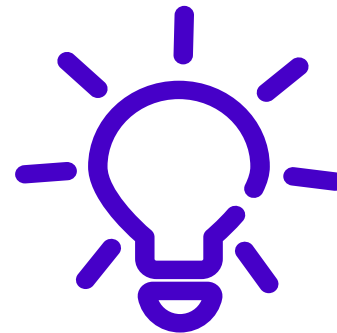
Qualidade de Código

Legibilidade, uso de boas práticas e documentação.



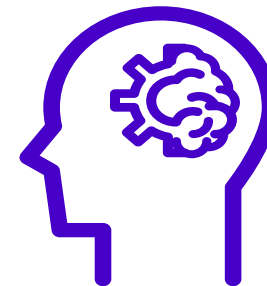
Eficiência da Soluções

Capacidade de otimizar o desempenho e gerenciar recursos.



Criatividade

Criatividade e inovação no design de soluções.



Análise

Profundidade de análise e capacidade de extrair insights significativos a partir dos dados.

Instruções Gerais para as Questões de Python

Para cada questão abaixo, você deverá desenvolver um script em Python que atenda aos requisitos especificados. Seu código será avaliado com base em:

Eficiência e Eficácia

Capacidade do script de executar a tarefa proposta de forma eficiente.

Flexibilidade e Escalabilidade

O script deve ser capaz de lidar com mudanças potenciais nos dados de entrada ou na estrutura das fontes de dados.

Tratamento de Erros

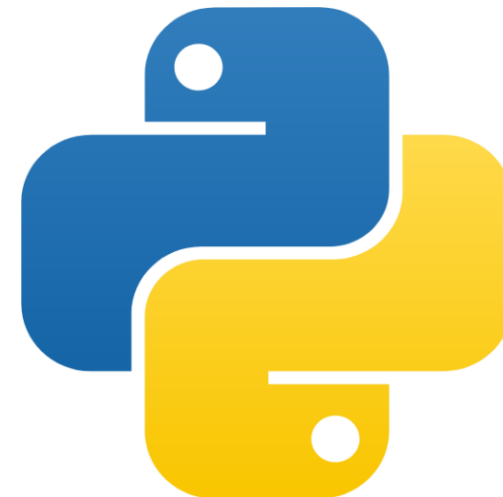
Implementação de tratamentos de erros robustos que permitam ao script lidar com situações inesperadas sem falhar.

Boas Práticas de Programação

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem

Documentação

Inclusão de comentários explicativos e documentação que torne o código acessível para outros desenvolvedores.



Questão 1: Automação para Download de Dados

Desenvolva um script Python que automatize o download dos seguintes arquivos de um repositório:

<https://www.kaggle.com/datasets/thomassimeo/contas-senior-neobpo>

`orcamento.xml despesas.xlsx centro_de_custo.pdf`

O script deve ser resiliente a mudanças na estrutura da página ou localização dos arquivos. Implemente lógica para identificar automaticamente os links de download desses arquivos específicos. Inclua tratamento de erros para lidar com possíveis interrupções na conexão ou mudanças no layout do site.

Tratamento Avançado de Erros: Implemente um tratamento de erros avançado para lidar com problemas comuns de API, como limites de taxa, timeouts de conexão, e dados mal formados. Use técnicas como tentativas de reconexão com backoff exponencial e validação de esquemas de dados.

Logging e Monitoramento: Adicione funcionalidades de logging ao script para registrar eventos críticos, erros, e informações de diagnóstico. Demonstre como você configuraria diferentes níveis de logging e como os logs poderiam ser usados para monitorar a saúde do processo de ETL.

Questão 2: Consumo e Transformação de Dados via API

Você precisa desenvolver um script Python para consumir dados de uma API REST. A API, representada pelo endpoint fictício <https://sheetdb.io/api/v1/xlc7njblhysnw>, retorna uma lista complexa de registros financeiros que precisam ser processados e transformados para análise.

`contas.csv`

Requisitos Detalhados:

Autenticação e Segurança: Implemente um mecanismo de autenticação no script, supondo que a API exija um token de acesso para consultas. Demonstre como você armazenaria e utilizaria este token de forma segura em seu script.

Consumo Condicional e Paginação: Suponha que a API suporte paginação e limite a quantidade de dados retornados em cada resposta. Modifique o script para detectar e navegar automaticamente pela paginação de dados. Além disso, implemente uma lógica para consumir dados condicionalmente, baseando-se em um critério específico (por exemplo, data da última atualização), para evitar o download desnecessário de dados já processados.

Questão 3: Avançado Processamento de XML e Reutilização de Código

Você tem o arquivo `orcamento.xml` da Questão 1, que contém dados financeiros detalhados. Este arquivo precisa ser convertido para um formato CSV para análise posterior. No entanto, o arquivo XML é conhecido por sua complexidade estrutural, incluindo aninhamentos profundos e variabilidade nos conjuntos de dados.

Requisitos Detalhados:

Análise e Mapeamento Complexo de XML: Realize uma análise detalhada do XML para identificar e documentar sua estrutura, incluindo elementos aninhados e atributos. Desenvolva uma lógica para mapear essa estrutura complexa para um formato tabular (CSV), garantindo que todas as informações relevantes sejam preservadas.

Tratamento de Variabilidades e Aninhamentos: Implemente um mecanismo flexível que possa lidar com variabilidades na estrutura do XML, como elementos faltantes em alguns registros, ou variações na profundidade de aninhamento. O script deve adaptar-se automaticamente a estas variações para evitar erros ou perda de dados.

Validação e Correção de Dados: Incorpore uma fase de validação de dados que verifique a integridade e a precisão dos dados extraídos antes de convertê-los em CSV. Implemente rotinas de correção automática para problemas comuns encontrados nos dados, como valores inválidos, formatos de data inconsistentes, ou referências quebradas entre elementos.

Questão 4: Extração de Dados de PDF para CSV

Você precisa extrair dados financeiros do arquivo CENTRO_DE_CUSTO.PDF obtido na Questão 1. Este documento PDF apresenta desafios significativos devido à sua estrutura complexa, que inclui tabelas com múltiplas colunas, aninhamento de informações, variação de formatos entre as páginas, e possíveis imagens contendo dados relevantes.

Requisitos Detalhados:

Análise Estrutural Complexa de PDF: Desenvolva um script que analise a estrutura do PDF para identificar diferentes layouts de página, incluindo a disposição de tabelas, texto corrido, e imagens. Esta análise deve automatizar a identificação de padrões de estrutura de dados para facilitar a extração precisa.

Extração de Dados de Tabelas e Texto: Implemente uma solução capaz de extrair dados de tabelas complexas, incluindo a identificação de cabeçalhos de colunas, linhas aninhadas e células que abrangem múltiplas colunas ou linhas. Além disso, o script deve ser capaz de extrair e correlacionar texto corrido que esteja associado a estas tabelas, como notas de rodapé ou legendas.

Entrega e Avaliação

Formato de Entrega

Scripts Python: Cada script desenvolvido para as questões propostas deve ser entregue em um arquivo separado, com a extensão `.py`, seguindo uma nomenclatura clara e consistente que facilite a identificação da questão a que se refere.

Dependências: Todas as bibliotecas externas utilizadas nos scripts devem ser listadas em um arquivo `requirements.txt`, que deve acompanhar os scripts. Este arquivo deve conter as versões exatas das bibliotecas utilizadas, para garantir a reprodução fiel do ambiente de desenvolvimento.

Arquivo Comprimido: Os arquivos `.py` e o `requirements.txt` devem ser compactados em um único arquivo `.zip`. O nome deste arquivo comprimido deve seguir o padrão `[Nome_Sobrenome]_python_neobpo.zip`, substituindo `[Nome_Sobrenome]` pelos seus respectivos dados.

Critérios de Avaliação

Funcionalidade: A corretude e eficácia dos scripts em realizar as tarefas propostas, incluindo a precisão dos resultados e a adequação às especificações de cada questão.

Estrutura do Código: A organização lógica do código, incluindo a clareza da estruturação, a modularização e a utilização eficiente de funções e classes para promover a reutilização de código.

Boas Práticas de Programação: A aderência a convenções de estilo reconhecidas (como PEP 8 para Python), o uso de nomes significativos para variáveis e funções, e a implementação de padrões de projeto quando aplicável.

Documentação: A presença de comentários explicativos e docstrings em pontos chave do código, facilitando a compreensão das decisões de design, a lógica implementada e o uso das funções. Uma documentação externa, se fornecida, deve ser clara e concisa, explicando a abordagem geral, a estrutura do projeto e instruções para execução.

Tratamento de Erros: A robustez do código em lidar com situações inesperadas, incluindo a validação de entradas, a gestão de exceções e a implementação de logs que auxiliem na identificação e correção de problemas.

Instruções Gerais para as Questões de SQL

Para cada questão abaixo, você deverá desenvolver um script em SQL que atenda aos requisitos especificados. Seu código será avaliado com base em:

Entendimento e Planejamento

Leia cada questão cuidadosamente para garantir um entendimento completo dos requisitos.

Planeje sua abordagem antes de começar a escrever o código SQL. Isto pode incluir esboçar o esquema de banco de dados, identificar as chaves primárias e estrangeiras, e determinar as funções e operações necessárias.

Tratamento de Erros

Implemente e documente o tratamento de erros em suas consultas para gerenciar situações excepcionais, garantindo que o banco de dados permaneça consistente e a integridade dos dados seja mantida.

Estruturação do Código

Organize seu código de maneira lógica e legível. Use indentação e espaçamento consistentes.

Nomeie tabelas, colunas, variáveis e qualquer objeto de banco de dados de forma clara e significativa.

Testes e Validações

Teste suas soluções para garantir que elas funcionem como esperado em diferentes cenários, incluindo casos de borda.

Valide a precisão dos resultados de suas consultas, especialmente quando realizar operações de agregação ou análise de dados.

Aplicação de Boas Práticas

Demonstre boas práticas de desenvolvimento, como a normalização de dados, a utilização eficiente de índices, e o uso de funções e procedimentos armazenados quando apropriado.

Garanta que suas soluções sejam eficientes em termos de desempenho, especialmente ao lidar com grandes volumes de dados.



Questão 1: Desenvolvimento e Otimização de Stored Procedures

Utilizando os dados extraídos e transformados dos arquivos CSV (ORCAMENTO, DESPESAS, CENTRO_DE_CUSTO, CONTAS), você foi designado para desenvolver uma stored procedure(s) complexa (s) no SQL Server. Esta(s) procedure(s) será responsável por inserir e atualizar dados na tabela DespesasDetalhadas, que consolida informações financeiras detalhadas de diversas categorias de despesas.

Desenvolva uma stored procedure chamada `sp_InserirAtualizarDespesasDetalhadas`. Esta procedure deve aceitar dados de entrada compatíveis com o formato e estrutura dos arquivos CSV DESPESAS.

A procedure deve inserir novos registros ou atualizar registros existentes baseando-se em uma chave única, como um identificador de despesa ou uma combinação de data e categoria.

Implementação de Transações:

Implemente transações para garantir a integridade dos dados. Cada inserção ou atualização de registro deve ser tratada como uma única operação atômica.

Inclua mecanismos de tratamento de erros dentro da transação para lidar com possíveis falhas, utilizando TRY...CATCH para reverter transações em caso de erro e registrar a ocorrência de falhas.

Questão 2: Resolução de Conflitos de Dados em Tempo Real

Contexto do Mercado: Em um sistema de e-commerce, produtos são vendidos em múltiplas plataformas simultaneamente. Ocorre um problema frequente de conflito de estoque, onde o mesmo produto é vendido em duas plataformas quase ao mesmo tempo, levando a vendas que não podem ser atendidas devido à falta de estoque.

Desafio: Implementar uma solução de banco de dados que minimize conflitos de estoque, garantindo que as atualizações de inventário sejam processadas em tempo real e de forma que vendas simultâneas não resultem em comprometimento do estoque disponível.

Crie as Tabelas e Dados:

Tabela de Produtos: Uma tabela contendo ID do Produto, Nome do Produto, Quantidade em Estoque. Essa tabela será utilizada para simular o estoque atual de produtos disponíveis para venda.

Tabela de Vendas: Uma tabela para registrar cada venda, contendo ID da Venda, ID do Produto, Quantidade Vendida, Timestamp da Venda. Essa tabela simula vendas ocorrendo em diferentes momentos.

Questão 3: Otimização de Consultas para Relatórios de Análise de Sentimento

Contexto do Mercado: Uma empresa monitora as redes sociais para análise de sentimento sobre seus produtos. O volume de dados é imenso, resultando em tempos de resposta lentos para consultas que agregam e analisam os sentimentos dos clientes ao longo do tempo.

Desafio: Otimizar as consultas usadas para gerar relatórios de análise de sentimento, de modo que possam ser executadas em tempo hábil.

Tabela de Posts: Uma tabela contendo ID do Post, Texto do Post, Data e Hora do Post, Sentimento (positivo, negativo, neutro), e ID do Produto (associando cada post a um produto específico). Essa tabela será usada para simular os dados coletados das redes sociais para análise de sentimento.

Entrega e Avaliação

Formato de Entrega

Arquivos: Submeter respostas como arquivos .sql individuais, e um documento README.md detalhando soluções e instruções.

Organização: Estruturar submissões em diretórios claros, compactando tudo em um arquivo .zip com o nome do candidato. O nome deste arquivo comprimido deve seguir o padrão [Nome_Sobrenome]_sql_neobpo.zip, substituindo [Nome_Sobrenome] pelos seus respectivos dados.

Critérios de Avaliação

1. Corretude Técnica: Precisão e adequação das soluções.
2. Eficiência e Otimização: Uso de técnicas para otimizar desempenho.
3. Clareza e Organização do Código: Legibilidade, estrutura, e estilo de codificação.
4. Documentação e Comentários: Qualidade da documentação e utilidade dos comentários no código.
5. Inovação e Criatividade: Aplicação de soluções inovadoras e técnicas criativas.
6. Adesão às Melhores Práticas: Uso de práticas recomendadas em design de banco de dados e escrita de consultas.

Instruções Gerais para as Questões de Power BI

Para cada questão abaixo, você deverá desenvolver de um Power BI que atenda aos requisitos especificados. Seu Power BI será avaliado com base em:

Fonte de Dados

Utilize as bases CONTAS.CSV, CENTRO_DE_CUSTO.CSV, DESPESAS.CSV, e ORCAMENTO.CSV como fonte de dados. Conversões para o formato Excel são permitidas se necessárias para otimizar a performance ou a usabilidade do dashboard

Indicadores

Desenvolva visualizações para os indicadores necessários e análises descritas, incluindo variações e desvios.

Funcionalidade e Interatividade

O dashboard deve ser interativo, com capacidade de filtragem por mês, conta e centro de custo. Deve incluir um menu de navegação que permita aos usuários alternar facilmente entre diferentes visualizações e análises..

Identidade Visual

O layout e os componentes visuais do dashboard devem aderir à identidade visual da NeoBPO, conforme diretrizes disponíveis no Guia de Identidade Visual. Cores, fontes, e logotipos devem ser consistentes com a marca.

<https://neobpocorp.sharepoint.com/sites/GuiadeIDVisual>



Questão 1: Indicadores Necessários

1.1 Valor Orçado - Consolidado, Por Mês, Por Conta, Por Centro de Custo

Desafio Adicional: Implemente medidas dinâmicas que ajustem automaticamente os valores orçados com base em taxas de câmbio ou ajustes de orçamento que possam ocorrer ao longo do ano. Use DAX para criar uma lógica condicional que considere essas variações.

1.2 Valor Realizado Consolidado - Consolidado, Por Mês, Por Conta, Por Centro de Custo

Desafio Adicional: Crie uma análise de tendência temporal que compare o valor realizado contra o valor orçado, incluindo a projeção para os próximos meses baseada no desempenho atual. Use funções DAX avançadas para modelar a projeção.

1.3 Desvio Real x Orçado - Consolidado, Por Mês, Por Conta, Por Centro de Custo

Desafio Adicional: Desenvolva um indicador de "alerta" que sinaliza desvios significativos do orçado que excedam um limite pré-definido. O indicador deve ser visualmente distinto e acionável, permitindo aos usuários investigar a causa dos desvios rapidamente.

Questão 2: Análise

2.1 Contas Ofensoras - Variação entre meses em p.p sobre o desvio

Desafio Adicional: Além de identificar contas ofensoras, desenvolva um modelo de pontuação que classifique as contas com base no impacto de seus desvios no desempenho geral. Integre uma visualização que mapeie contas ofensoras em um "mapa de calor" baseado na pontuação de impacto.

Questão 3: Disponibilize Gráfico - Realizado x Orçado do centro de custo TIME MIS

- Resposta
- Orçado
- Realizado
- Desvio
- % Desvio

Desafio Adicional: Crie um gráfico dinâmico que permita a seleção de múltiplos centros de custo para comparação simultânea. Implemente funcionalidades de drill-down que permitam analisar o desempenho mensal detalhado de cada centro de custo selecionado.

Questão 4: Mostre o resultado do indicador para o centro de custo TIME MIS no mês de Janeiro - CONTA: 260001

- Resposta
- Orçado
- Realizado
- Desvio
- % Desvio

Desafio Adicional: Automatize a geração de um relatório narrativo que descreva o desempenho do centro de custo TIME MIS, incluindo análises comparativas com outros centros de custo e recomendações baseadas em IA para melhoria

Questão 5: Disponibilize o print do relacionamento de tabelas final do modelo de dados realizado.

Desafio Adicional: Demonstre como você otimizou o modelo de dados para performance, incluindo a implementação de índices, agregações e técnicas de modelagem para reduzir a cardinalidade. Explique sua abordagem e o impacto na performance do dashboard.

Entrega e Avaliação

Formato de Entrega

Submeta um arquivo .pbix contendo o dashboard desenvolvido, junto com qualquer documentação de suporte que detalhe as decisões de design e técnicas utilizadas, e um documento README.md detalhando soluções e instruções.

Critérios de Avaliação

Os dashboards serão avaliados com base na precisão dos dados visualizados, aderência à identidade visual da NeoBPO, eficácia na comunicação das informações, uso de boas práticas de design de UI/UX, e inovação nas soluções de visualização e interatividade.

Inclua uma breve documentação que explique sua abordagem para a modelagem de dados, design do dashboard, escolha de visualizações, e qualquer funcionalidade customizada implementada. A documentação deve também incluir uma seção de reflexão sobre os desafios encontrados e como foram superados.

Entrega Final

Formato de Entrega

Preparação dos Arquivos: Certifique-se de que todos os scripts Python, arquivos de banco de dados SQL, e dashboards do Power BI estejam finalizados, testados e prontos para avaliação. Inclua também quaisquer documentos de apoio, como relatórios narrativos ou documentação do projeto, que detalhem suas análises e conclusões.

Organização dos Arquivos: Organize seus arquivos em pastas claramente nomeadas de acordo com cada seção do projeto (Python, SQL, Power BI). Dentro de cada pasta, nomeie os arquivos de forma intuitiva para refletir seu conteúdo ou objetivo específico.

Inclusão de requirements.txt para Python: Para a parte de Python do projeto, inclua um arquivo requirements.txt que liste todas as bibliotecas e suas versões usadas. Isso garante que o avaliador possa recriar seu ambiente de desenvolvimento sem problemas.

Compressão em Arquivo ZIP: Após organizar todos os arquivos necessários, comprima a pasta do projeto em um arquivo ZIP. Certifique-se de que o arquivo ZIP contenha todas as pastas e arquivos necessários para a completa avaliação do seu projeto.

Nomeação do Arquivo ZIP: Nomeie o arquivo ZIP seguindo o padrão **prova_nome_sobrenome_neobpo_dia_atual.zip**, substituindo nome_sobrenome pelos seus próprios e dia_atual pela data de envio. Essa convenção de nomenclatura ajuda na organização e garante que seu projeto seja facilmente identificável.

Envio: Envie o arquivo ZIP através do canal especificado pela organização do projeto. Certifique-se de seguir quaisquer instruções adicionais fornecidas para o envio.

Obrigado

neo

www.neobpo.com.br