

Relatório de Desempenho

Período: até 18 de Outubro de 2024

Responsável: Mewerton de Melo Silva

1. Otimização de Query para Folha de Pagamento

Revisamos e organizamos uma query SQL para a folha de pagamento do estado, com foco na identificação de possíveis irregularidades, como acúmulo de função, salários acima do permitido, múltiplas matrículas e pagamento indevido de verbas. Selecionamos colunas relevantes para gestores, como funções efetivas, níveis salariais, verbas financeiras, dependentes e admissões. Esse processo facilitou a análise e o cruzamento de dados no dashboard.

2. Implementação e Integração com o Banco de Dados

Conectamos com sucesso ao banco de dados ElogicaRH utilizando PyODBC, executando queries para extrair dados relevantes de servidores, como CPF, função, vínculo e valores calculados. Os dados foram organizados e armazenados inicialmente em CSV (com delimitação por pipe), e posteriormente foram convertidos para o formato Parquet utilizando PyArrow, otimizando o tamanho dos arquivos e melhorando o desempenho.

3. Tratamento e Solução de Problemas

Identificamos e resolvemos problemas no carregamento e visualização de arquivos grandes, especialmente no VS Code. Além disso, corrigimos inconsistências nos nomes das colunas ao gerar os arquivos Parquet diretamente dos DataFrames do Pandas, evitando a necessidade de conversão intermediária para CSV.

4. Desenvolvimento e Ajustes no Dashboard de Servidores

Criamos o novo dashboard de Servidores, seguindo a estrutura de Despesas e Diárias, com filtros por UG e SIGLA_UG. Foi implementada uma lógica para garantir que o valor de "TOTAL VANTAGENS" seja priorizado ao exibir o salário dos servidores, tanto no dashboard quanto nas consultas por CPF feitas pelo chatbot.

5. Integração e Desenvolvimento do Chatbot Carly

Implementamos o chatbot Carly no Painel do Gestor, integrando-o a todos os painéis, como Despesas, Diárias, Contratos e Servidores. Carly foi configurada para interagir com os usuários, respondendo perguntas relacionadas aos dados da folha de pagamento e consultas de CPF.

Ações:

- Integração inicial com a API da Groq para fornecer respostas de forma amigável.
 - Implementação da função `buscar_dados_por_cpf`, que permite ao chatbot buscar informações específicas sobre servidores com base no CPF fornecido.
 - Configuração para manter o histórico de conversas ao alternar entre páginas do painel.
-

6. Ajustes no Sidebar e Modificações nos Painéis

Atualizamos todos os painéis (Despesas, Diárias, Contratos e Servidores) para incluir a chamada do `render_chatbot()` no sidebar, permitindo que Carly esteja sempre disponível para interação. Além disso, padronizamos o uso da função centralizada `load_sidebar` para garantir consistência no carregamento de filtros.

7. Aprimoramento das Regras de Conversa no Chatbot

Definimos regras claras para a interação do chatbot Carly, como manter respostas simples e evitar discussões fora do escopo de atendimento. A função `buscar_dados_por_cpf` foi

configurada para ser acionada automaticamente quando o CPF é fornecido na conversa, permitindo que o chatbot realize consultas específicas de forma eficiente.

8. Solução de Inconsistências entre o Painel e o Chatbot

Foi identificado que Carly não estava retornando os mesmos valores de salário que o painel de Servidores exibia. Após análise, a lógica de filtragem foi ajustada no chatbot para garantir que o valor correto de "TOTAL VANTAGENS" seja exibido tanto no painel quanto nas respostas de Carly.

9. Tratamento de Salário Bruto no Painel de Servidores

Ajustamos a lógica de duplicação no painel de Servidores para garantir que a linha contendo o valor de "TOTAL VANTAGENS" seja preservada. Esse valor foi unificado com outros dados de vínculo do servidor, garantindo a exibição correta do salário bruto no painel.

10. Melhorias na Tabela de Servidores

Adicionamos funcionalidades de pesquisa e sumarização na tabela de servidores, permitindo a filtragem por nome ou CPF. Também implementamos a soma total dos valores da coluna `Financ_Valor_Calculado`, exibindo apenas valores associados à verba "TOTAL VANTAGENS".

11. Otimização do Carregamento de Dados

O arquivo `data_loader.py` foi revisado e otimizado para garantir o carregamento eficiente dos dados de servidores, contratos e despesas. Utilizamos cache para evitar carregamentos desnecessários, resultando em uma melhora significativa na performance do painel.