

GUIA DE CONFIGURAÇÃO E DESENVOLVIMENTO

SUMÁRIO

Introdução.....	
Tecnologias utilizadas	
Configurações do script Python	
Configuração do Metabase	
Serviços da AWS.....	
Versões.....	
Possíveis modificações no projeto	
Considerações Finais.....	

INTRODUÇÃO

Este guia abrangente visa fornecer instruções detalhadas para a configuração e desenvolvimento de um projeto de cubo de dados em colaboração com a Integration, uma empresa de consultoria. Este guia destaca as etapas cruciais necessárias para a implementação bem-sucedida do projeto, garantindo que a análise de dados seja realizada de maneira eficiente e eficaz.

TECNOLOGIAS UTILIZADAS

No decorrer da parceria com a Integration, a implementação do projeto de cubo de dados foi enriquecida pela adoção estratégica de diversas tecnologias, fundamentais para a criação de uma infraestrutura sólida e eficiente. A seleção meticulosa dessas tecnologias foi guiada pelas necessidades específicas do projeto, alinhadas com as melhores práticas do setor.

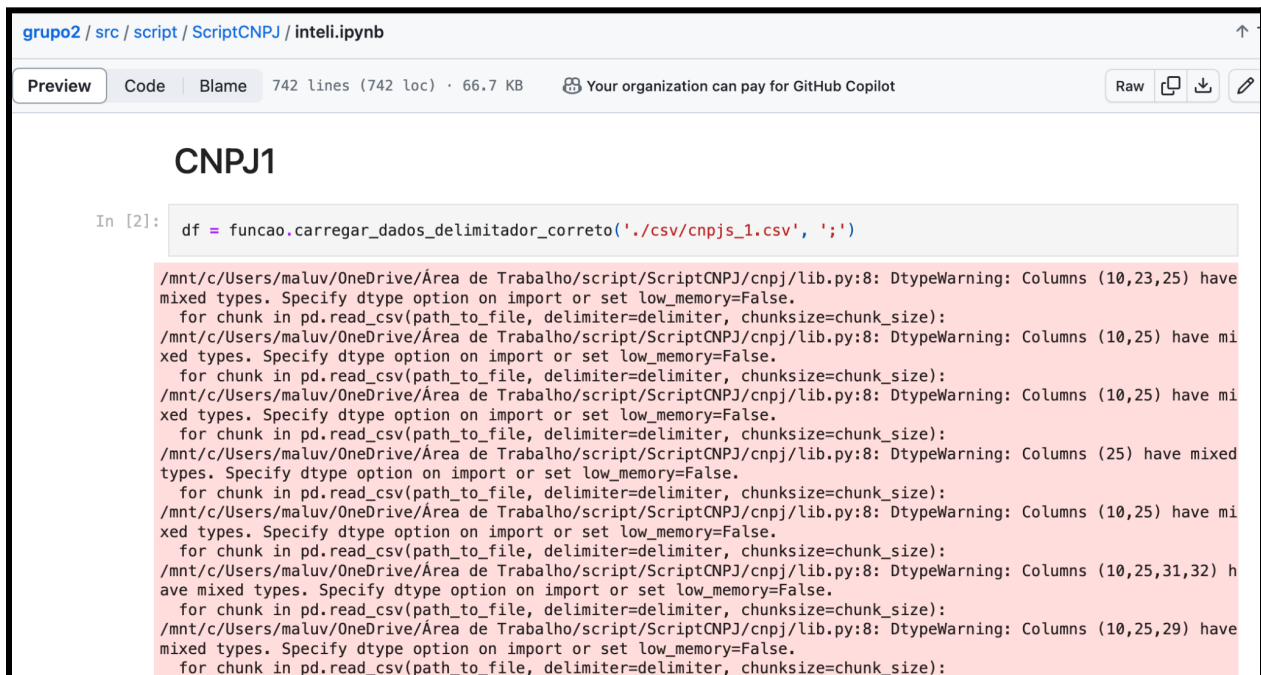
No âmbito do desenvolvimento, scripts em Python foram empregados com precisão para o pré-processamento dos dados, demonstrando a flexibilidade e versatilidade da linguagem. A visualização dos insights extraídos foi simplificada através do Metabase, que ofereceu uma abordagem intuitiva na criação de infográficos e representações visuais.

A integração estratégica com o Amazon Redshift desempenhou um papel central na criação de um ambiente OLAP eficaz. A aplicação de SQL no Redshift permitiu manipulações analíticas complexas, enquanto a exploração do Google BigQuery como alternativa destacou-se por sua escalabilidade, proporcionando flexibilidade adicional.

A arquitetura do Redshift, baseada em armazenamento colunar e paralelismo massivo, trouxe otimizações notáveis para consultas analíticas, consolidando eficiência no gerenciamento e análise de grandes volumes de dados. Este projeto resultou em uma sinergia entre agilidade e eficiência, impulsionada pela harmoniosa integração de tecnologias cuidadosamente selecionadas.

CONFIGURAÇÕES DO SCRIPT PYTHON

Python para Pré-processamento de Dados: Para lidar com a variedade de arquivos de dados recebidos, optou-se por desenvolver scripts em Python. Essa escolha se justifica pela flexibilidade oferecida pela linguagem, sua vasta biblioteca para manipulação de dados e sua ampla aceitação na comunidade de análise de dados.



The screenshot shows a Jupyter Notebook interface with the title 'CNPJ1'. The code cell contains the following Python code:

```
In [2]: df = funcao.carregar_dados_delimitador_correto('./csv/cnpps_1.csv', ';')
```

The output of the code cell is a series of warnings from pandas, indicating that the CSV file has mixed data types in several columns. The warnings are as follows:

- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (10,23,25) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):
- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (10,25) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):
- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (10,25) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):
- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (25) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):
- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (10,25) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):
- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (10,25,31,32) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):
- /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptCNPJ/cnpj/lib.py:8: DtypeWarning: Columns (10,25,29) have mixed types. Specify dtype option on import or set low_memory=False.
- for chunk in pd.read_csv(path_to_file, delimiter=delimiter, chunksize=chunk_size):

Imagem 1: Exemplo de script em Python com o intuito de fazer o processo de pré-processamento dos dados

```

  ▾ GINI Geral

[ ] df_gini_geral = funcao.load_data_with_correct_delimiter("./csv/gini_geral.csv")
    df_gini_geral_clean = funcao.clean_data(df_gini_geral)
    gini_geral_s3 = funcao.ajustar_amazon_s3(df_gini_geral_clean, './csv_s3/gini_geral_s3.csv')

/mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptIBGE/ibge/lib.py:29: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
df = df.applymap(lambda x: x.replace(',', ';') if isinstance(x, str) else x)

  ▾ GINI Industria

▶ df_gini_industria = funcao.load_data_with_correct_delimiter("./csv/gini_industria.csv")
  df_gini_industria_clean = funcao.clean_data(df_gini_industria)
  gini_industria_s3 = funcao.ajustar_amazon_s3(df_gini_industria_clean, './csv_s3/gini_industria_s3.csv')

▶ /mnt/c/Users/maluv/OneDrive/Área de Trabalho/script/ScriptIBGE/ibge/lib.py:29: FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
df = df.applymap(lambda x: x.replace(',', ';') if isinstance(x, str) else x)

```

Imagem 2: Pré-processamento e tratamento dos dados

1.1 Arquivos do pacote

Este é um breve guia sobre os arquivos contidos neste pacote Python.

- **.env.tmpl**: Arquivo de configuração de ambiente. Contém as keys da AWS Lab.
- **.ipynb_checkpoints**: Diretório que armazena checkpoints automáticos de notebooks Jupyter.
- **.pytest_cache**: Diretório que armazena cache e dados temporários do Pytest.
- **build**: Diretório gerado durante o processo de construção do pacote. Contém arquivos intermediários.
- **csv**: Diretório para armazenar arquivos CSV relacionados ao projeto.
- **inteli.ipynb**: Jupyter Notebook principal do projeto, que contém todo o processo de tratamento dos dados CSV.
- **nome_do_script**: Pacote Python principal, aqui cada script tem o seu nome específico, seguindo o nome do dado.
 - **init.py**: Arquivo necessário para que o diretório seja tratado como um pacote Python.
 - **lib.py**: Módulo Python contendo funcionalidades específicas. Aqui são definidas todas as funções de processamento utilizadas no `inteli.ipynb`.
- **nome_do_script.egg-info**: Informações sobre o pacote, geradas durante a construção, aqui cada script tem o seu nome específico, seguindo o nome do dado.
- **send_s3.ipynb**: Jupyter Notebook onde os arquivos CSV, após o processamento, são encaminhados aos buckets da AWS S3.
- **setup.py**: Script de configuração para instalar o pacote.
- **tests**: Diretório contendo testes para o pacote.

Imagem 3: Arquivos do package do script python

CONFIGURAÇÃO DO METABASE

A criação de infográficos e representações visuais dos dados foi realizada utilizando o Metabase. Esta ferramenta proporciona uma interface intuitiva e eficiente para a geração de visualizações interativas, facilitando a interpretação e comunicação dos insights derivados dos dados.

A configuração inicial do Metabase envolve alguns passos essenciais para integrar a ferramenta ao ambiente de desenvolvimento. Ao acessar a página é necessário configurar um banco de dados para armazenar os metadados do Metabase, podendo ser uma opção embutida ou um banco de dados externo, como MySQL ou PostgreSQL.

Com o Metabase em execução, o próximo passo é conectar as fontes de dados ao projeto. Dessa forma, a criação de visualizações é simplificada pela interface gráfica do Metabase. Criando tipos de visualizações, como gráficos de barras, pizza, linhas, mapas, entre outros.

Ao integrar o Metabase com o Amazon Redshift, é crucial configurar a conexão corretamente, inserindo informações como endereço do banco de dados, permitindo a exploração para gerar consultas otimizadas.

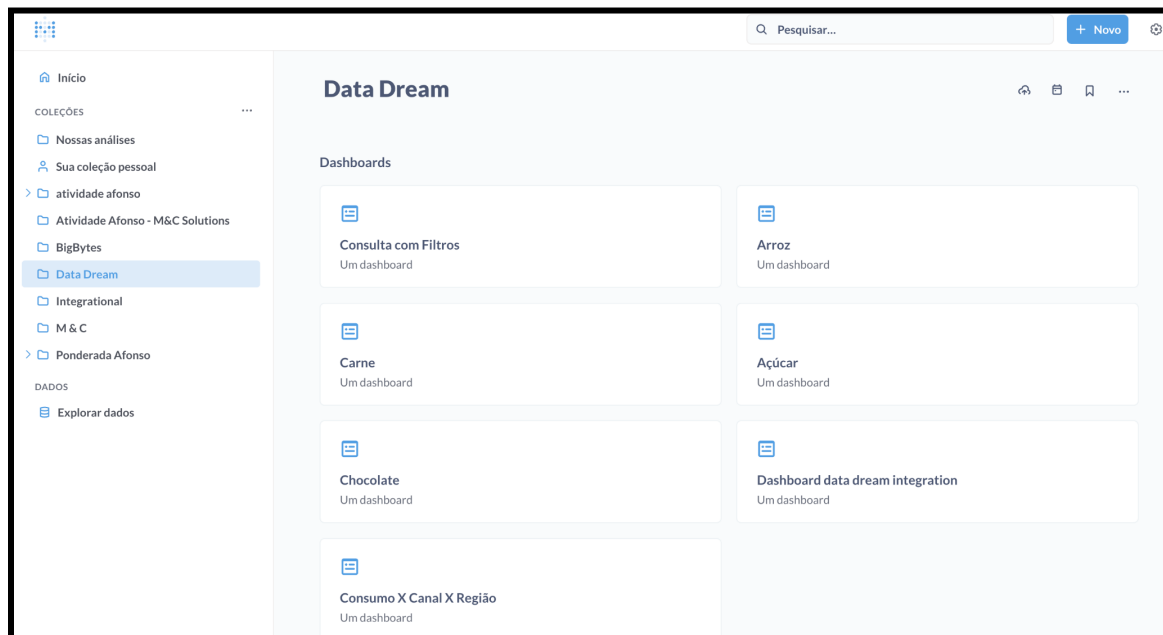


Imagem 4: Aba de início do Metabase - Grupo Data Dream

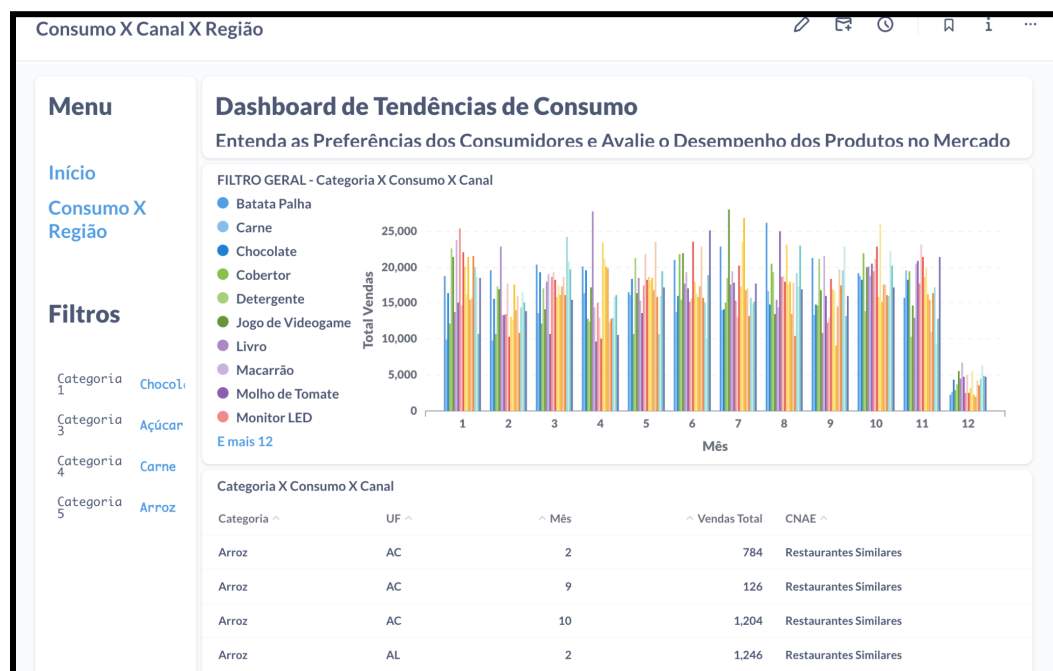


Imagem 5: Gráfico desenvolvido no Metabase - Dashboard de Tendências de Consumo

SERVIÇOS DA AWS

A base do nosso projeto foi solidamente construída na Amazon Web Services (AWS), uma plataforma líder em serviços de computação em nuvem. Ao utilizar serviços como EC2, Redshift e participar ativamente nos laboratórios de aprendizagem da AWS Academy, conseguimos estabelecer uma infraestrutura robusta e escalável, essencial para o sucesso do nosso projeto.

A AWS oferece uma ampla gama de serviços em nuvem, cobrindo desde hospedagem de servidores até serviços de armazenamento, processamento e análise de dados. Utilizamos um login e usuário compartilhado no contexto de um grupo, o que permitiu acesso facilitado aos serviços AWS necessários para implementar e gerenciar os dados em buckets, otimizando o compartilhamento eficiente de recursos e garantindo segurança na gestão de permissões.

O Redshift, um serviço de data warehouse gerenciado pela AWS, foi um elemento crucial para nossa arquitetura de dados. Sua capacidade de armazenamento massivo, escalabilidade e desempenho ágil foram fundamentais para consolidar e

analisar grandes volumes de dados. A participação nos laboratórios da AWS Academy aprimorou ainda mais nossas habilidades, permitindo-nos explorar as melhores práticas e recursos avançados da AWS para otimizar nossa abordagem analítica.

A escolha da AWS como base para nossa infraestrutura na nuvem proporcionou não apenas uma solução técnica eficaz, mas também uma eficiência operacional notável. A capacidade de compartilhar facilmente credenciais e recursos dentro do grupo, aliada à flexibilidade e desempenho dos serviços contribuiu para uma implementação bem-sucedida e eficiente do nosso projeto de cubo de dados.

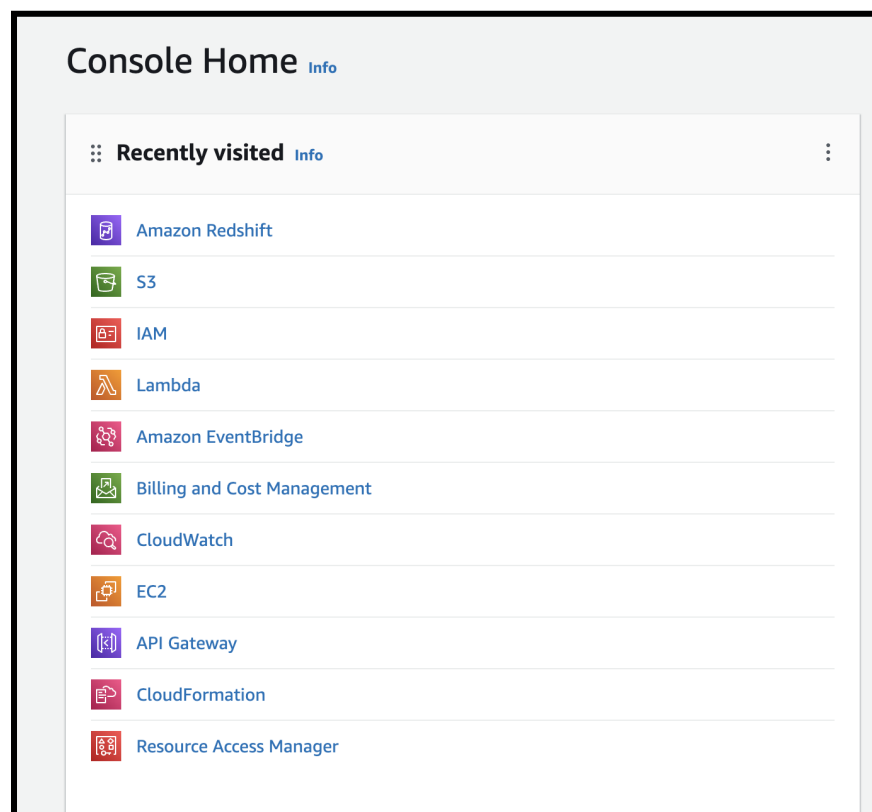


Imagem 6: Sistemas e tecnologias da AWS utilizadas no projeto

VERSÕES

- Versões Python : 3.11
- Versão Metabase : 0.47.9

MODIFICAÇÕES NO PROJETO NO FUTURO

À medida que um projeto de Big Data evolui, diversas áreas se destacam como pontos de possível aprimoramento. A escalabilidade permanente é crucial, especialmente ao considerar o aumento na quantidade e complexidade dos dados. Monitorar e ajustar recursos, como adicionar mais nós ao cluster Redshift, garantirá um desempenho contínuo e eficiente.

A expansão de fontes de dados é outra consideração vital. À medida que novas fontes se tornam relevantes, a integração desses dados pode aprimorar ainda mais a análise. Explorar serviços adicionais da AWS para diferentes tipos de dados é uma extensão lógica para enriquecer a variedade de informações disponíveis.

No âmbito da análise, otimizar consultas SQL no Redshift e considerar a criação de índices relevantes são ações que podem resultar em melhorias significativas no processamento analítico. Além disso, a constante evolução no campo de ferramentas de BI e visualização oferece oportunidades para explorar novas soluções, garantindo uma interpretação mais eficaz dos dados. À medida que o projeto avança, implementar práticas adicionais de segurança, integrar novas tecnologias emergentes e reforçar políticas de backup e recuperação são elementos-chave para manter a eficiência e a integridade do sistema. Essas considerações proporcionam uma visão holística para garantir que o projeto permaneça adaptável e alinhado às demandas dinâmicas do ambiente analítico.

CONSIDERAÇÕES FINAIS

Este guia visa fornecer uma referência abrangente para configuração e desenvolvimento, oferecendo insights para garantir que o projeto de cubo de dados permaneça adaptável e alinhado às demandas dinâmicas do ambiente analítico. A parceria com a Integration e a utilização estratégica das tecnologias apresentadas proporcionaram uma implementação bem-sucedida e prepararam para futuras melhorias e inovações.