

Nome: Tiago Ferreira Saraiva

RGM: 02494649-4

Documentação:

Objetivo: Carregar um banco de dados, extrair, limpar e transformar as informações, carregando-as em um serviço de nuvem, criar uma conexão local com o serviço de nuvem para manipular as tabelas de dados de forma mais ágil, utilizar os dados para criar um painel interativo em Power BI que possa apresentar os dados de forma dinâmica, com informações acessíveis ao usuário.

O projeto envolve a extração dos dados, que estão em formato sqlite3 e a análise e transformação para uso após tratamento utilizando Jupyter Notebook. Com a transformação das tabelas é realizada a ingestão dos dados ao serviço de nuvem Google Cloud, na solução BigQuery.

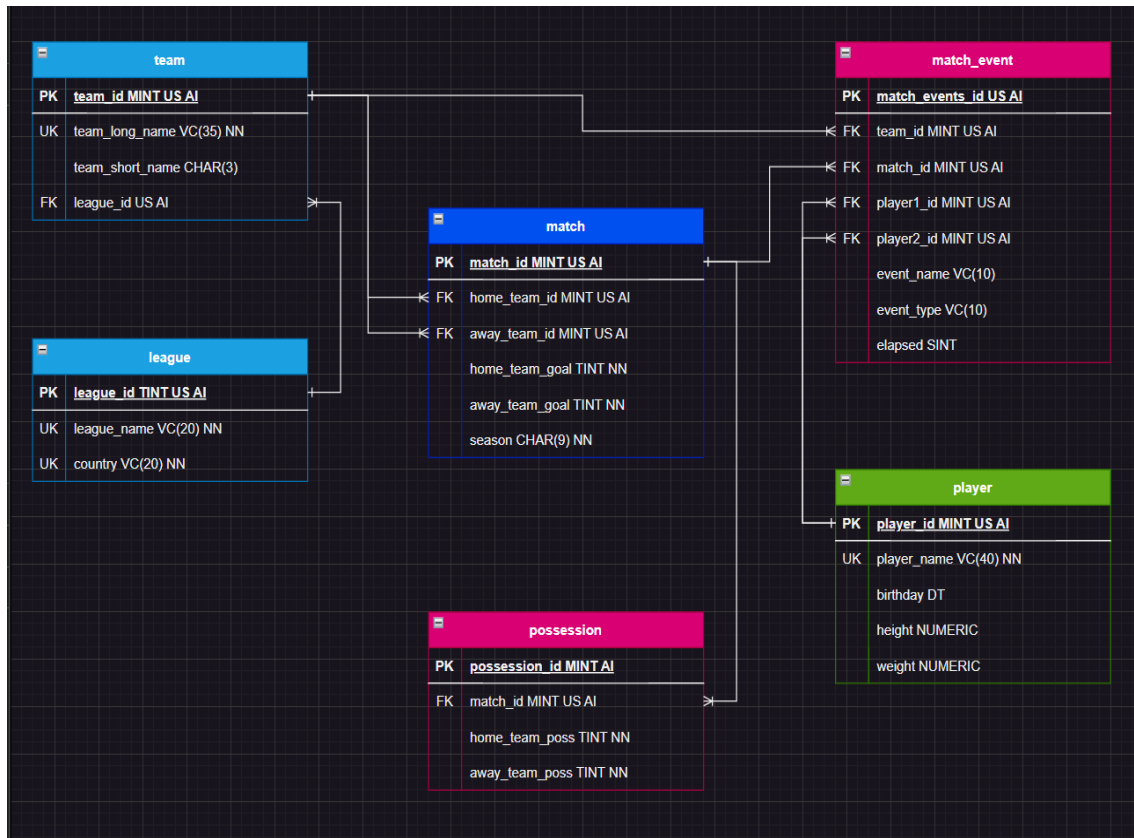
Após a ingestão dos dados é feito um acesso local para manipulação das tabelas utilizando SQL Server, conectando com o DBeaver -- qualquer outra ferramenta de preferência. A conexão permite que sejam realizados procedimentos nas tabelas para criação ou exclusão de tabelas, inclusão ou retirada de dados.

Com os dados tratados e disponíveis em serviço de nuvem é o utilizado o Power BI para criação de um dashboard interativo onde o usuário pode consultar informações sobre os dados, de forma acessível.

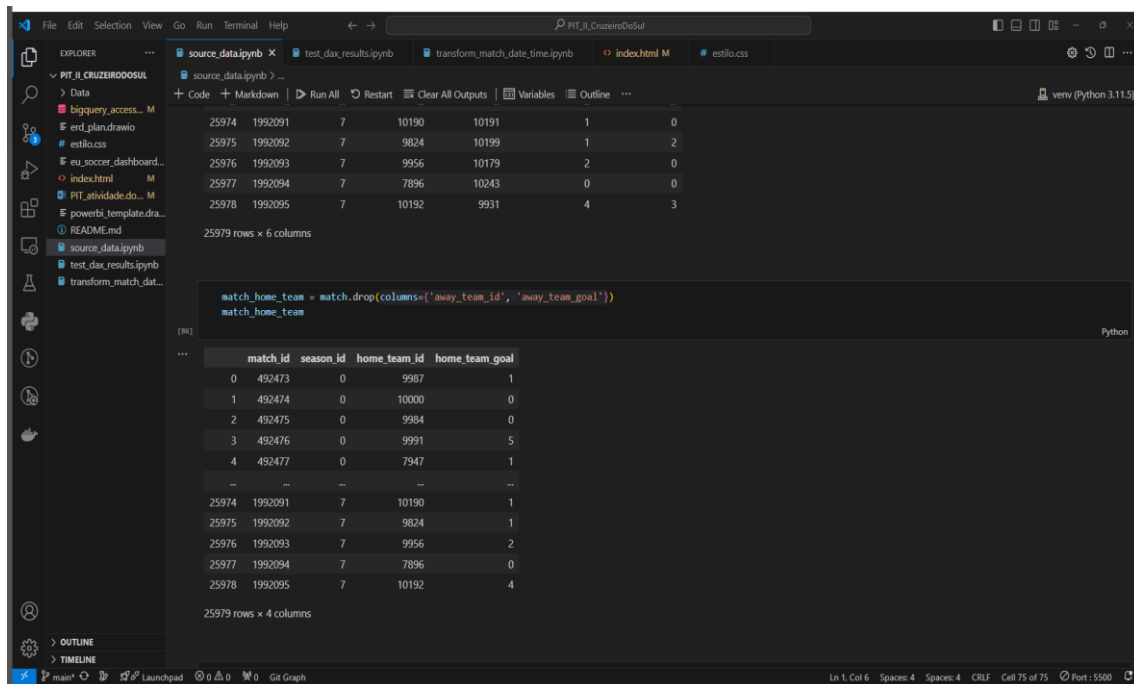
Os dados estão disponíveis na plataforma Kaggle, conforme site abaixo, e possuem alguns problemas, nem todas as ligas têm informações completas para utilização. As tabelas devem ficar conforme o ERD, anexo e disponível no Github.

Foram solucionados alguns problemas nos dados, também foram realizados alguns testes para utilização no Power BI, alguns eficazes, outros ineficazes. A construção do Power BI trouxe vários desafios e a projeção inicial dos painéis foi alterada devido limitações impostas.

Diagrama Entidade Relacionamento – ERD:



Extração, limpeza e transformação dos dados:



```

source_data.ipynb
test_dax_results.ipynb
transform_match_date_time.ipynb
index.html
estilo.css

source_data.ipynb > ...
+ Code + Markdown | Run All | Restart | Clear All Outputs | Variables | Outline ...
venv (Python 3.11.5)

25974 1992091 7 10190 10191 1 0
25975 1992092 7 9824 10199 1 2
25976 1992093 7 9956 10179 2 0
25977 1992094 7 7896 10243 0 0
25978 1992095 7 10192 9931 4 3

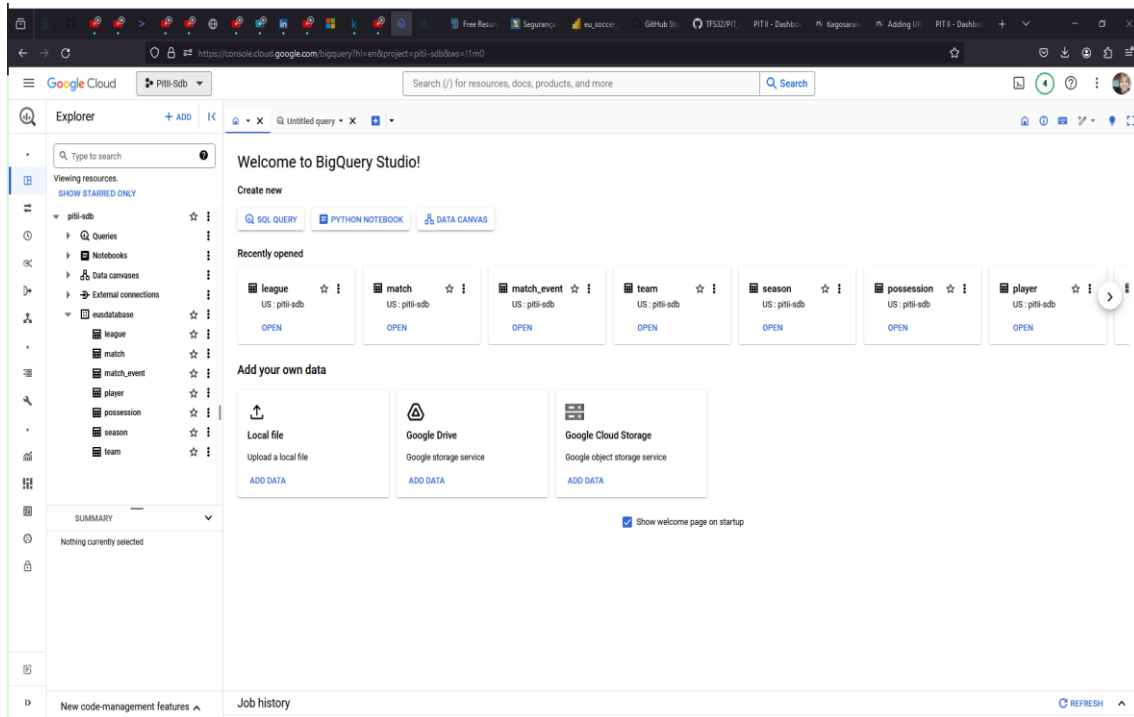
25979 rows x 6 columns

match_home_team = match.drop(columns=['away_team_id', 'away_team_goal'])
match_home_team

[86]
...
   match_id  season_id  home_team_id  home_team_goal
0  492473         0         9987             1
1  492474         0        10000             0
2  492475         0         9984             0
3  492476         0         9991             5
4  492477         0         7947             1
...      ...      ...      ...
25974 1992091         7        10190             1
25975 1992092         7         9824             1
25976 1992093         7         9956             2
25977 1992094         7         7896             0
25978 1992095         7        10192             4

25979 rows x 4 columns
    
```

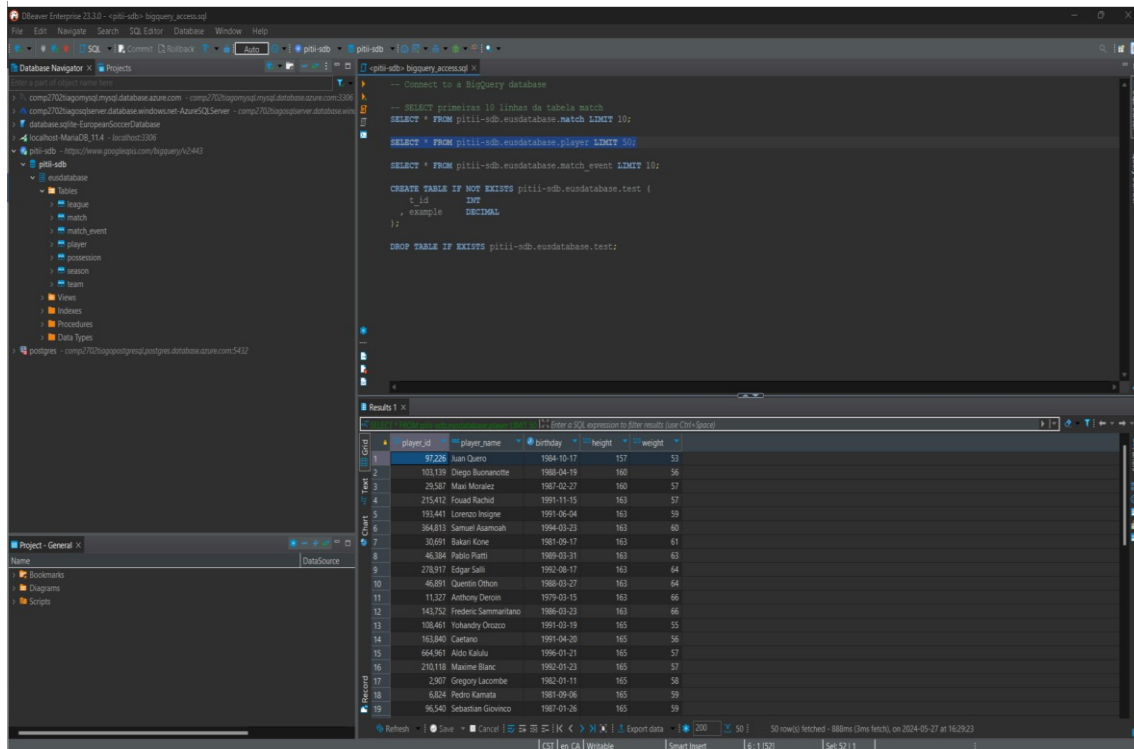
Google Cloud – BigQuery:



The screenshot shows the Google Cloud BigQuery Studio interface. The top navigation bar includes the Google Cloud logo, a search bar, and various icons. The main content area is divided into several sections:

- Explorer:** A sidebar on the left showing a tree view of resources. It includes sections for "Viewing resources" (SHOW STARRED ONLY), "Recently opened" (listing files like league, match, match_event, team, season, possession, player), and "Add your own data" (with options for Local file, Google Drive, and Google Cloud Storage).
- Welcome to BigQuery Studio!**: A central panel with buttons for "Create new" (SQL QUERY, PYTHON NOTEBOOK, DATA CANVAS) and "Recently opened" (listing files like league, match, match_event, team, season, possession, player).
- Job history:** A section at the bottom right showing a list of jobs.

Conexão framework local:



The screenshot shows the DBeaver Enterprise interface. The left sidebar displays a tree view of databases, including "piti-sdb" and "eudatabase". The main window shows a SQL query being executed against the "piti-sdb" database. The query is as follows:

```

-- Connect to a BigQuery database
-- SELECT primeira id linha da tabela match
SELECT * FROM piti-sdb.eudatabase.match LIMIT 10;

SELECT * FROM piti-sdb.eudatabase.player LIMIT 50;

SELECT * FROM piti-sdb.eudatabase.match_event LIMIT 10;

CREATE TABLE IF NOT EXISTS piti-sdb.eudatabase.test (
  id INT,
  example DECIMAL
);

DROP TABLE IF EXISTS piti-sdb.eudatabase.test;

```

The results of the query are displayed in a table with the following columns: player_id, player_name, birthday, height, and weight. The table contains 19 rows of data, including players like Juan Quintero, Diego Buonanotte, and Maxi Moralez.



Tabela de Construção (Desenvolvimento Ágil)

ID	História do Usuário	Estimativa em Pontos	Prioridade
0	Escolha do banco de dados para uso no projeto	4	A
1	Análise dos dados componentes do banco, normalização e projeção do diagrama de entidade e relacionamento	8	B
4	Extração e transformação do banco de dados	16	A
2	Escolha da aplicação de nuvem e carregamento dos dados para nuvem	4	C
3	Conexão local para manipulação dos arquivos em nuvem	2	D
5	Painel de apresentação desenvolvido em Power BI, após recebimento dos dados tratados.	20	A
6	Criação de uma página da web para acoplar o painel criado em Power BI	5	C

ID: 0

Título: Banco de dados

Requerente: Product Owner

Ação: Toda a equipe escolhe as fontes do banco de dados para ser utilizado no projeto.
Comentários: Observar quais serão as fontes, qual tipo de arquivo será recebido, como será o tratamento.
Critérios de aceitação: Confiabilidade e integridade.
Regras de negócio:
Requisito não funcional:
Prioridade: <input checked="" type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D
Pontos de história: 4

ID: 1
Título: Diagrama Entidade Relacionamento (ERD)
Requerente: Product Owner
Ação: Ambas equipes definem como será a construção do relacionamento entre as entidades para que seja realizada a transformação de acordo com os critérios de construção da visualização.
Comentários: Normalizar as tabelas, evitando redundância.
Critérios de aceitação: Design simples que atenda aos requisitos de construção do painel ao final.
Regras de negócio:
Requisito não funcional:
Prioridade: <input type="checkbox"/> A <input checked="" type="checkbox"/> B

☐ C

☐ D

Pontos de história: 8

ID: 2

Título: Serviço de Nuvem

Requerente: Product Owner

Ação: A equipe de engenharia de dados escolhe o serviço de nuvem adequado para hospedar o banco de dados resultante do tratamento dos dados.

Comentários: Atenção para os critérios de segurança e replicabilidade.

Critérios de aceitação: Conformidade com o código de segurança e com o backup.

Regras de negócio:

Requisito não funcional:

Prioridade:

☐ A

☐ B

☒ C

☐ D

Pontos de história: 4

ID: 3

Título: Conexão Nuvem

Requerente: Product Owner

Ação: Equipe de engenharia de dados cria o fluxo de informação para que os dados em nuvem sejam manipulados diretamente em framework local.

Comentários:

Critérios de aceitação: Funcionamento.



Regras de negócio:
Requisito não funcional:
Prioridade: <input type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input checked="" type="checkbox"/> D
Pontos de história: 4

ID: 4
Título: ETL
Requerente: Product Owner
Ação: Equipe de engenharia de dados realiza a extração e transformação dos dados conforme os critérios estabelecidos, verificando a integridade das informações e tratando os problemas que o banco apresentou.
Comentários:
Critérios de aceitação: Garantir que os critérios do ERD sejam atendidos.
Regras de negócio:
Requisito não funcional:
Prioridade: <input checked="" type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D
Pontos de história: 16

ID: 5
Título: Power BI

Requerente: Product Owner
Ação: A equipe de visualização desenha os painéis de Power BI conforme o projeto.
Comentários: Tentar atender aos critérios do planejamento.
Critérios de aceitação: Utilizar todas as informações que o banco possui.
Regras de negócio:
Requisito não funcional:
Prioridade: <input checked="" type="checkbox"/> A <input type="checkbox"/> B <input type="checkbox"/> C <input type="checkbox"/> D
Pontos de história: 20

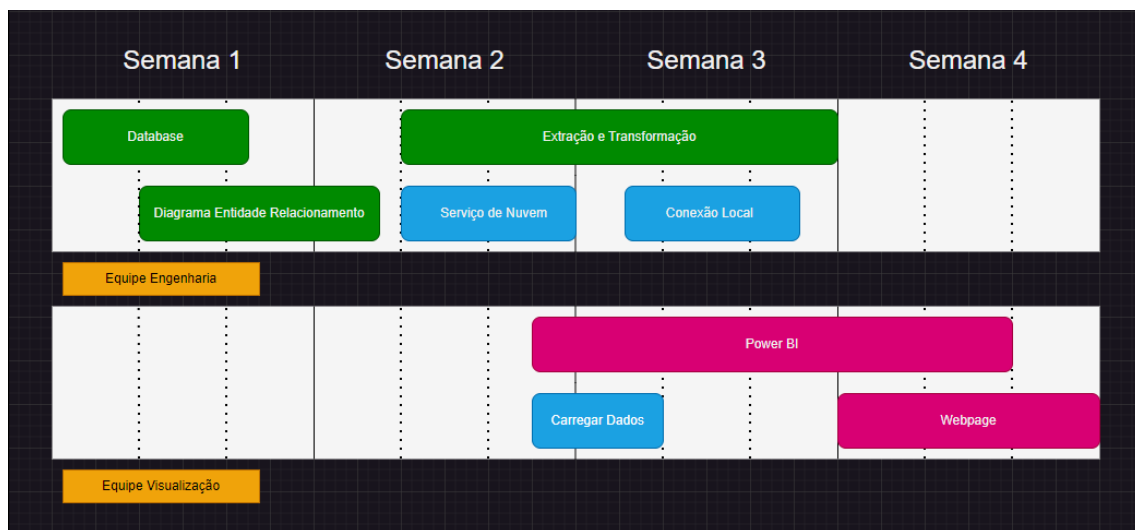
ID: 6
Título: Website
Requerente: Product Owner
Ação: Equipe de visualização desenvolve um site para receber o Power BI facilitando o acesso dos usuários internos a painel.
Comentários:
Critérios de aceitação:
Regras de negócio:
Requisito não funcional:
Prioridade: <input type="checkbox"/> A <input type="checkbox"/> B <input checked="" type="checkbox"/> C <input type="checkbox"/> D

Pontos de história: 5

Fluxo de apresentação



Fluxo de Trabalho



Codificação:

Na Tabela a seguir insira as informações referentes ao desenvolvimento do código do processo.

Linguagem	Python, SQL, DAX
Banco de Dados	https://www.kaggle.com/datasets/hugomathien/soccer/code?datasetId=63
Hospedagem	Git Hub
Plataforma	Power BI
Modo de Codificação	(X) Tradicional () Low-code
Link do repositório no GitHub com os códigos abertos	https://github.com/TFS32/PIT_II_CruzeiroDoSul
Link da solução em funcionamento	https://tiagosaraiva.works/
Link do vídeo narrado (no mínimo 5 min)	https://www.youtube.com/watch?v=7F1IT4sS_Lw

Testes da Solução

Escolha 5 colegas para testar sua aplicação, disponibilize o *link* de acesso ou os recursos necessários para que testem como usuários. Preencha a Tabela a seguir com as informações obtidas:

Nome: Diego Goncalves Silva	Data do teste: 27/05/24
O que testou e funcionou: Todas as funcionalidades, botões das listas.	
O que testou e não funcionou – O que deve ser corrigido: N/A	
Funcionalidade não testada (faltou ou não foi implementada): N/A	

Nome: Rodrigo Waltmann Nascimento	Data do teste: 28/05/2024
O que testou e funcionou: -	
O que testou e não funcionou – O que deve ser corrigido: relatou que não conseguiu acessar o site.	
Funcionalidade não testada (faltou ou não foi implementada): -	

Nome: Rafael da Silva Camargo	Data do teste: 28/05/2024
O que testou e funcionou: acessei o site, acessei Power BI, fiz seleções usando as opções de botões, ok.	
O que testou e não funcionou – O que deve ser corrigido: os dados de time fora de casa apresentam os mesmos dados de time da casa. Painel do Power BI tem delay. Os dados de gol ou cartões apresentam as vezes informações do jogador em branco. O botão de selecionar temporada no painel do clube não funciona.	
Funcionalidade não testada (faltou ou não foi implementada):	

Nome: Giovanna Albuquerque Riello	Data do teste: 28/05/2024
O que testou e funcionou: site funcionando, Power BI funcionando, é possível fazer seleções.	
O que testou e não funcionou – O que deve ser corrigido: o site é muito simples não tem aparência legal.	
Funcionalidade não testada (faltou ou não foi implementada): não há identidade visual no projeto.	

Nome: Fernando Alves de Oliveira	Data do teste: 30/05/2024
O que testou e funcionou: site ok, acesso ao Power BI ok.	
O que testou e não funcionou – O que deve ser corrigido: as vezes aparecem dados em branco nas tabelas com nomes de jogadores. Quando o dado de ocorrências na partida é zero, aparece como (blank).	
Funcionalidade não testada (faltou ou não foi implementada):	

Vídeo da Solução atualizada

Após levantar os *feedbacks* e executar as correções necessárias e pertinentes, grave um vídeo de **até 5 minutos** apresentando as modificações realizadas no sistema.

Link para o vídeo	https://youtu.be/ei_MSyhf7lY
-------------------	---