

## Documentação do Projeto de Análise de Dados da Fórmula 1

Este documento detalha o processo de **ETL (Extração, Transformação e Carga)** e a estrutura de diretórios de um projeto avaliativo focado em dados de Fórmula 1, visando preparar um conjunto de dados para análises e visualizações.

### Objetivo do Projeto

O objetivo principal deste projeto é realizar um robusto processo de ETL utilizando dados abertos da Fórmula 1. A meta é criar um conjunto de dados limpo, estruturado e otimizado, que possa ser facilmente consumido por ferramentas de análise e visualização de dados, como Power BI, Excel e dashboards web, facilitando a obtenção de insights sobre o desempenho de pilotos e montadoras ao longo dos anos.

### Tecnologias Utilizadas

As seguintes tecnologias foram empregadas na execução do projeto:

- **Python 3:** Linguagem de programação principal para o desenvolvimento do ETL.
- **Pandas:** Biblioteca Python essencial para manipulação e análise de dados tabulares.
- **Openpyxl:** Biblioteca Python para leitura e escrita de arquivos .xlsx.
- **OS (módulo padrão):** Módulo Python para interação com o sistema operacional, útil para gerenciar caminhos de arquivos e diretórios.
- **Jupyter Notebook:** Ambiente interativo utilizado para desenvolver e documentar as etapas do processo ETL.

### Estrutura de Diretórios da Extração

A estrutura de diretórios para a fase de extração e saída dos dados é organizada da seguinte forma:

```
extracao/  
├── saída/  
│   ├── construtores.xlsx  
│   ├── corridas.xlsx  
│   ├── pilotos.xlsx  
│   └── resultados.xlsx
```

Dentro da pasta `extracao/saída/`, os arquivos Excel (.xlsx) representam o resultado final do processo de ETL, prontos para consumo.

## Etapas Detalhadas do Processo ETL

O processo ETL foi cuidadosamente dividido em quatro etapas principais para garantir a qualidade e a organização dos dados.

### 1. Extração

Os dados brutos foram extraídos diretamente de arquivos CSV hospedados no GitHub, aproveitando a capacidade do `pandas.read_csv()` de importar dados a partir de URLs.

Os arquivos extraídos incluem:

- `constructors.csv`
- `drivers.csv`
- `races.csv`
- `results.csv`

### 2. Transformação

Cada conjunto de dados passou por um processo de limpeza e padronização, que incluiu a remoção de colunas irrelevantes e a renomeação de outras para maior clareza e consistência.

#### a) Construtores (`constructors.csv`)

- **Colunas Removidas:** `constructorRef`, `url`
- **Colunas Renomeadas:**
  - `constructorId` → `montadora_id`
  - `name` → `nome`
  - `nationality` → `nacionalidade`

#### b) Pilotos (`drivers.csv`)

- **Coluna Criada:** `nome_completo` (resultante da concatenação de `forename` e `surname`)
- **Colunas Removidas:** `forename`, `surname`, `url`, `number`, `dob`, `code`, `driverRef`
- **Colunas Renomeadas:**
  - `driverId` → `piloto_id`
  - `nomeCompleto` → `nome_completo`
  - `nationality` → `nacionalidade`

### c) Corridas (races.csv)

- **Colunas Removidas:** time, url, fp1\_date, fp1\_time, fp2\_date, fp2\_time, fp3\_date, fp3\_time, quali\_date, quali\_time, sprint\_date, sprint\_time, round, circuitId
- **Colunas Renomeadas:**
  - raceId → corrida\_id
  - year → ano
  - name → nome
  - date → corrida\_data

### d) Resultados (results.csv)

- **Colunas Removidas:** number, grid, position, positionText, laps, time, milliseconds, fastestLap, rank, fastestLapSpeed, statusId
- **Colunas Renomeadas:**
  - resultId → resultado\_id
  - raceId → corrida\_id
  - driverId → piloto\_id
  - constructorId → montadora\_id
  - positionOrder → posicao\_ordem
  - points → pontos
  - fastestLapTime → volta\_mais\_rapida\_tempo

## 3. Validação dos Dados

Após a etapa de transformação, foram executadas validações cruciais para assegurar a integridade e a qualidade dos dados:

- **Verificação da Forma:** Utilização de .shape para confirmar as dimensões (linhas e colunas) dos DataFrames.
- **Análise das Colunas:** Uso de .info() para inspecionar tipos de dados e contagem de valores não nulos por coluna.
- **Valores Nulos:** Cálculo do percentual de valores nulos por coluna com .isnull().sum() para identificar e tratar dados ausentes.

## 4. Carga

Na fase final do ETL, os DataFrames transformados foram exportados para o formato .xlsx (Excel) utilizando a função pandas.to\_excel(). Os arquivos resultantes foram salvos no diretório extracao/saida/, conforme a estrutura de diretórios definida, prontos para serem utilizados em análises futuras.



## Possíveis Próximos Passos e Melhorias

Para expandir e aprimorar o projeto, as seguintes etapas podem ser consideradas:

- **Integração com Banco de Dados:** Carregar os dados tratados em sistemas de gerenciamento de banco de dados como PostgreSQL ou MySQL para maior escalabilidade e capacidade de consulta.
- **Automatização do Processo:** Implementar agendadores de tarefas (cron jobs, Apache Airflow) para automatizar a execução diária ou periódica do pipeline de ETL.
- **Visualizações Avançadas:** Desenvolver dashboards interativos e dinâmicos utilizando ferramentas como Dash, Power BI ou Tableau para apresentar insights de forma mais impactante.
- **Expansão dos Datasets:** Incorporar outros conjuntos de dados da Fórmula 1 (e.g., lap\_times.csv, pit\_stops.csv) para enriquecer as análises e obter uma visão mais completa.

## Estrutura de Diretórios do Projeto Completo

A organização geral do projeto foi meticulosamente planejada para segregar responsabilidades e otimizar o fluxo de trabalho.

### Projeto\_ETL\_F1/

- ├── 1 - TELAS/ # Imagens ou capturas de tela dos dashboards ou análises
- ├── 2 - PBIX/ # Arquivos do Power BI (.pbix)
- ├── 3 - ETL/ # Scripts e notebooks do processo de ETL
- ├── 4 - EXECUTAVEL ETL/ # Versão compilada do script ETL (ex: .exe gerado com PyInstaller)
- └── ETL\_F1\_Documentacao.md # Documentação do projeto em Markdown

### Explicação dos Diretórios:

- **1 - TELAS/:** Este diretório serve como um portfólio visual do projeto. Ele contém capturas de tela ou mockups dos dashboards e análises desenvolvidas, sendo ideal para apresentações e para uma validação visual rápida do trabalho final.
- **2 - PBIX/:** Aqui são armazenados os arquivos .pbix do Power BI. Estes arquivos contêm as visualizações e modelos de dados criados a partir dos dados limpos e transformados pelo processo de ETL, prontos para serem explorados interativamente.
- **3 - ETL/:** Este é o coração do processo de dados. Contém todo o código-fonte do ETL, incluindo Jupyter Notebooks (.ipynb) para experimentação e scripts Python (.py) para execução do pipeline de dados.
- **4 - EXECUTAVEL ETL/:** Destinado a abrigar a versão compilada do script ETL

(por exemplo, um .exe gerado com ferramentas como PyInstaller). Isso permite que o processo de ETL seja executado em ambientes onde o Python e suas dependências não estão instalados, facilitando a distribuição.

- **ETL\_F1\_Documentacao.md:** Este arquivo Markdown é a documentação central do projeto, descrevendo todas as etapas, desde a extração dos dados brutos até a exportação dos arquivos finais para consumo. É um guia essencial para entender a lógica e o fluxo do projeto.



## **Autor**

Paulo Vitor Jeronimo de Almeida

[https://br.linkedin.com/in/paulo-vitor-a17796175?trk=public\\_post\\_follow-view-profile](https://br.linkedin.com/in/paulo-vitor-a17796175?trk=public_post_follow-view-profile)