

# Documentação dos Planos de Carga

Para as tabelas dimensões foram usados os mesmos steps para todas as cargas: Microsoft Excel Input, Select Values e Table Output.

## Fluxo no Apache Hop para Criação das Dimensões

### Passo 1: Microsoft Excel Input

- **Objetivo:** Ler os dados de entrada a partir de um arquivo Excel.
- **Configurações:**
  - **Nome do Arquivo:** Caminho para o arquivo Excel que contém os dados das dimensões.
  - **Planilha:** Nome da planilha específica dentro do arquivo Excel.
  - **Campos a Ler:** Especifica quais colunas do Excel devem ser lidas para o processo de ETL.

### Passo 2: Select Values

- **Objetivo:** Selecionar e transformar os dados conforme necessário antes da carga na tabela de dimensão.
- **Transformações Aplicadas:**
  - **Renomear Campos:** Se necessário, renomeia os campos para padrões específicos.
  - **Filtrar Dados:** Pode-se filtrar linhas com base em critérios específicos, se necessário.
  - **Converter Tipos de Dados:** Converte tipos de dados conforme requerido pelas especificações da tabela de dimensão.
  - **Aplicar Expressões:** Calcula novos campos ou aplica expressões para preparar os dados para a carga.

### Passo 3: Table Output

- **Objetivo:** Carregar os dados processados na tabela de dimensão no banco de dados.
- **Configurações:**
  - **Conexão de Banco de Dados:** Define a conexão com o banco de dados PostgreSQL onde a tabela de dimensão será criada.
  - **Tabela de Destino:** Nome da tabela de dimensão no banco de dados onde os dados serão carregados.
  - **Ação:** Define se o step deve inserir novos registros, atualizar existentes ou ambas as operações.

## Documentação Completa do Fluxo ETL

Este fluxo utiliza o Apache Hop para extrair dados de um arquivo Excel, transformá-los usando o step `Select Values` e carregá-los em uma tabela de dimensão no PostgreSQL

através do step `Table Output`. Cada passo é configurado para garantir que os dados sejam lidos corretamente, transformados conforme necessário e carregados de maneira eficiente na tabela de destino.

## Processos das Cargas das Dimensões



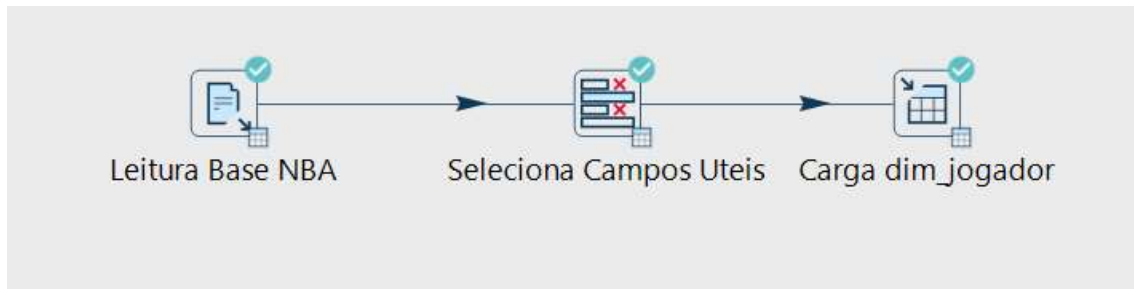
### Carga dim\_equipe



### Carga dim\_tempo



### Carga dim\_oponente



Carga dim\_jogador

## Fluxo no Apache Hop para Criação da Tabela Fato *fat\_performance*

### Passo 1: Microsoft Excel Input

- **Objetivo:** Ler os dados de entrada a partir de um arquivo Excel que contém métricas de desempenho de jogadores.
- **Configurações:**
  - **Nome do Arquivo:** Caminho para o arquivo Excel que contém os dados de desempenho.
  - **Planilha:** Nome da planilha específica dentro do arquivo Excel.
  - **Campos a Ler:** Especifica quais colunas do Excel devem ser lidas para o processo de ETL.

### Passo 2: Dimension Lookup (para cada dimensão)

- **Objetivo:** Realizar um lookup nas tabelas de dimensão (*dim\_jogador*, *dim\_equipe*, *dim\_oponente*, *dim\_tempo*) para obter as chaves correspondentes aos nomes presentes nos dados de desempenho.
- **Configurações:**
  - **Tabela de Lookup:** Define a tabela de dimensão onde ocorrerá a busca.
  - **Campo de Pesquisa:** Especifica o campo na tabela de dimensão usado para buscar o valor correspondente ao nome do jogador, equipe, oponente ou data do jogo.

### Passo 3: Select Values

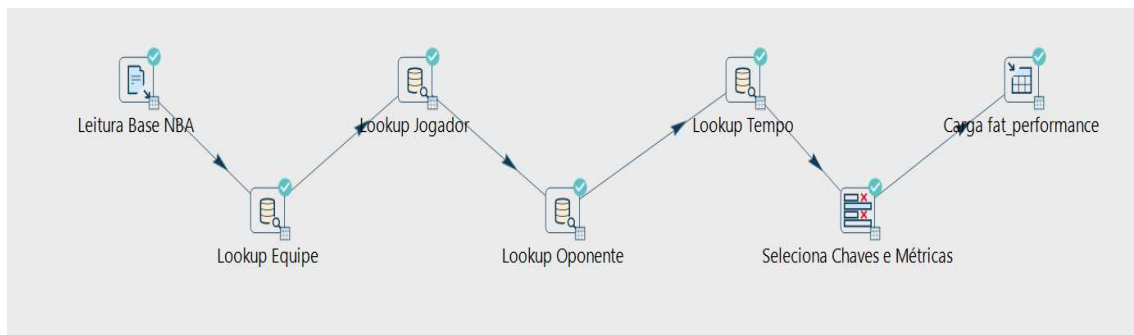
- **Objetivo:** Selecionar e transformar os dados conforme necessário antes da carga na tabela fato *fat\_performance*.
- **Transformações Aplicadas:**
  - **Substituir Campos por Chaves:** Substitui os nomes dos jogadores, equipes, oponentes e datas pelos respectivos identificadores (chaves) obtidos no passo de lookup.
  - **Selecionar Métricas:** Escolhe as métricas de desempenho relevantes que serão carregadas na tabela fato.
  - **Converter Tipos de Dados:** Converte tipos de dados conforme necessário para correspondência com os tipos na tabela fato.

## Passo 4: Table Output

- **Objetivo:** Carregar os dados processados na tabela fato `fat_performance` no banco de dados PostgreSQL.
- **Configurações:**
  - **Conexão de Banco de Dados:** Define a conexão com o banco de dados PostgreSQL onde a tabela fato será criada.
  - **Tabela de Destino:** Nome da tabela fato `fat_performance` no banco de dados onde os dados serão carregados.
  - **Ação:** Define se o step deve inserir novos registros, atualizar existentes ou ambas as operações, dependendo da estratégia de carga.

## Documentação Completa do Fluxo ETL

Este fluxo no Apache Hop é projetado para extrair dados de desempenho de jogadores de um arquivo Excel, realizar lookups nas dimensões para obter chaves correspondentes, selecionar métricas relevantes usando o `Select Values` e carregar os dados na tabela fato `fat_performance` no PostgreSQL usando `Table Output`. Cada passo é configurado para garantir que os dados sejam integrados de maneira eficiente e precisa no data warehouse, mantendo a integridade referencial com as tabelas de dimensão.



Carga `fat_performance`

## Descrição do Job no Apache Hop

### Objetivo

O job foi criado para orquestrar a execução de múltiplos pipelines de transformação de dados de forma sequencial ou paralela, utilizando o step `Pipeline` para chamar cada pipeline individualmente.

### Configurações do Job

- **Nome do Job:** Nome do job no Apache Hop.
- **Descrição:** Breve descrição do propósito do job, detalhando sua função no fluxo de ETL.

### Passos do Job

### 1. Pipeline (para cada pipeline a ser executado):

- **Objetivo:** Executar um pipeline específico dentro do job.
- **Configurações:**
  - **Nome do Pipeline:** Nome do pipeline que será executado pelo job.
  - **Parâmetros:** Se necessário, pode passar parâmetros para o pipeline sendo chamado, para personalizar o processamento de dados conforme necessário.

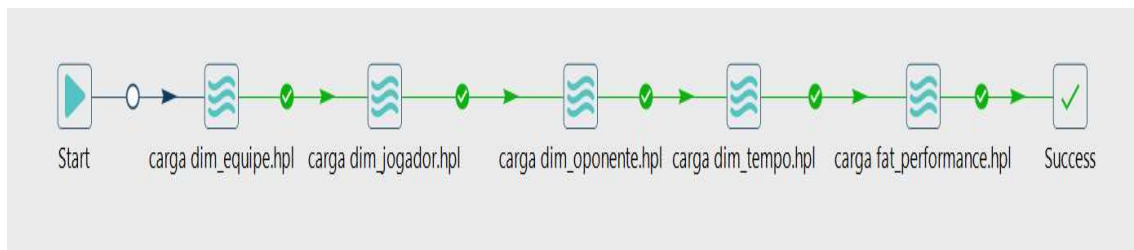
### Funcionamento

O job utiliza o step `Pipeline` para chamar cada pipeline individualmente. Isso permite modularizar o processo de ETL em etapas menores e mais gerenciáveis, facilitando a manutenção e a reutilização de lógicas de transformação de dados em diferentes contextos.

### Documentação Completa do Job

Este job no Apache Hop é responsável por coordenar a execução de pipelines de transformação de dados. Cada instanciação do step `Pipeline` dentro do job representa a execução de um pipeline específico, permitindo uma abordagem modular e flexível para construir processos de ETL complexos e escaláveis.

Cada pipeline chamado dentro do job pode ser personalizado com parâmetros, se necessário, para atender requisitos específicos de transformação ou integração de dados. Isso proporciona uma estrutura robusta para implementar workflows de ETL eficientes e organizados dentro do ambiente Apache Hop.



Job Atualiza\_Carga