

```

# Databricks notebook source
# MAGIC %md
# MAGIC ## Tabela: refined.d_embarcacao_farol
# MAGIC
# MAGIC ### Objetivo:
# MAGIC Tabela da dimensÃo de pessoas do indicador Farol.
# MAGIC
# MAGIC ### Fontes de Dados
# MAGIC
# MAGIC | Origem |
# MAGIC |-----|
# MAGIC | DescritÃo |
# MAGIC |-----|
# MAGIC | trusted.tb_tab_pessoa_farol | Tabela com dados cadastrais de pessoas fÃsicas e jurÃdicas. |
# MAGIC
# MAGIC ### HistÃrico de alteraÃÃes
# MAGIC
# MAGIC | Data | Desenvolvido por | ModificaÃÃes |
# MAGIC |-----|-----|-----|
# MAGIC | 21/05/2025 | Michel Santana | CriaÃÃo do notebook |

# COMMAND -----

# Importa e executa o notebook `ingestion_function`, localizado em `../00_config/`.
#
# O comando `%run` carrega todas as funÃÃes, variÃveis e configuraÃÃes definidas no notebook referenciado
# para o ambiente atual. Isso permite reutilizar lÃgica comum, como funÃÃes de ingestÃo de dados, sem duplicaÃÃo de cÃdigo.
#
# Ãstil para centralizar rotinas reutilizÃveis e manter notebooks modulares e organizados.

# COMMAND -----

# MAGIC %run ../00_config/ingestion_function

# COMMAND -----

container_target = 'refined'
directory = 'farol'
table_name = 'd_embarcacao_farol'
delta_table_name = f'{environment}.{container_target}.{table_name}'
delta_file = f"abfss://{container_target}@{data_lake_name}.dfs.core.windows.net/{directory}/{table_name}/"
comment_delta_table = 'Tabela de dimensÃo de Pessoas...'

print(f'delta_table_name = {table_name}')
print(f'delta_file = {delta_file}')

# COMMAND -----

"""
Cria um widget interativo chamado `reprocessar` para controle da carga da dimensÃo.

```

- Exibe um dropdown com as opções "True" e "False".
- A variável `reprocessar` será `True` apenas se o usuário selecionar essa opção no notebook.

Objetivo: permitir que o usuário escolha, de forma interativa, se a carga será completa (`overwrite`) ou incremental (`merge`).

```
dbutils.widgets.dropdown("reprocessar", "False", ["True", "False"], "Reprocessar dimensão?")
reprocessar = dbutils.widgets.get("reprocessar") == "True"

# COMMAND -----

spark.sql(f"""
CREATE TABLE IF NOT EXISTS {delta_table_name} (
    sk_embarcacao BIGINT GENERATED ALWAYS AS IDENTITY (START WITH 1
INCREMENT BY 1)
    ,sequencia_embarcacao INT
    ,descricao_embracacao STRING
    ,numero_registro_capitania STRING
    ,numero_registro_mpa STRING
    ,informacao_embarcacao_1 STRING
    ,informacao_embarcacao_2 STRING
    ,insert_date TIMESTAMP
    ,update_date TIMESTAMP
)
USING DELTA
LOCATION '{delta_file}'
COMMENT '{comment_delta_table}';
""")

# COMMAND -----

spark.sql(f"""

    select
        seq_embarcacao as sequencia_embarcacao,
        des_embarcacao as descricao_embracacao,
        num_registro_capitania as numero_registro_capitania,
        num_registro_mpa as numero_registro_mpa,
        inf_embarcacao_1 as informacao_embarcacao_1,
        inf_embarcacao_2 as informacao_embarcacao_2
    from {environment}.trusted.tb_tab_pessoa_embarcacao_farol
    where l=1

""").createOrReplaceTempView("source_df")

# COMMAND -----

if reprocessar:
    source_df = source_df.withColumn("insert_date",
lit(current_timestamp() ) )
    source_df.write.format("delta")\
        .mode("overwrite")\
        .saveAsTable(f"{delta_table_name}", path=f"{delta_file}")
    print(f"Carga overwrite realizada com sucesso! {delta_table_name}")
else:
    print("Realizando carga em Merge..")
    spark.sql(f"""
        MERGE INTO {delta_table_name} AS target
        USING source_df AS source
        ON target.sequencia_embarcacao = source.sequencia_embarcacao
```

```

        WHEN MATCHED THEN
            UPDATE SET
                target.descricao_embracacao =
source.descricao_embracacao
                ,target.numero_registro_capitania =
source.numero_registro_capitania
                ,target.numero_registro_mpa =
source.numero_registro_mpa
                ,target.informacao_embarcacao_1 =
source.informacao_embarcacao_1
                ,target.informacao_embarcacao_2 =
source.informacao_embarcacao_2
                ,target.update_date = current_timestamp()
        WHEN NOT MATCHED THEN
            INSERT (
                sequencia_embarcacao
                ,descricao_embracacao
                ,numero_registro_capitania
                ,numero_registro_mpa
                ,informacao_embarcacao_1
                ,informacao_embarcacao_2
                ,insert_date
            )
            VALUES (
                source.sequencia_embarcacao
                ,source.descricao_embracacao
                ,source.numero_registro_capitania
                ,source.numero_registro_mpa
                ,source.informacao_embarcacao_1
                ,source.informacao_embarcacao_2
                ,current_timestamp()
            );
    """)

```