

GarroVision

Integrantes:

RM9904 - Thiago Garcia Tonato - 1TDSPY

RM555502 - Ian Madeira Gonçalves da Silva

RM555109 - Murilo Ribeiro Santos



Modelo do banco de dados:

RM99404.HISTORICO_ENERGIA		
P	ID_HISTORICO	NUMBER
F	ID_RESIDENCIA	NUMBER
	DATA_REGISTRO	DATE
	PRODUCAO	NUMBER (10,2)
	CONSUMO	NUMBER (10,2)
	SALDO_ENERGETICO	NUMBER (10,2)
HISTORICO_ENERGIA_PK (ID_HISTORICO)		
SYS_C004050197 (ID_RESIDENCIA)		



RM99404.RESIDENCIAS		
P	ID_RESIDENCIA	NUMBER
	NOME_RESPONSAVEL	VARCHAR2 (100 BYTE)
	ENDERECO	VARCHAR2 (200 BYTE)
	CAPACIDADE_GERACAO	NUMBER (10,2)
	TIPO_FONTE	VARCHAR2 (50 BYTE)
	LIMITE_CONSUMO	NUMBER (10,2)
	DATA_CADASTRO	DATE
RESIDENCIAS_PK (ID_RESIDENCIA)		

```
DROP TABLE residencias CASCADE CONSTRAINTS;
DROP TABLE historico_energia CASCADE CONSTRAINTS;
DROP SEQUENCE HISTORICO_SEQ;
DROP SEQUENCE RESIDENCIA_SEQ;

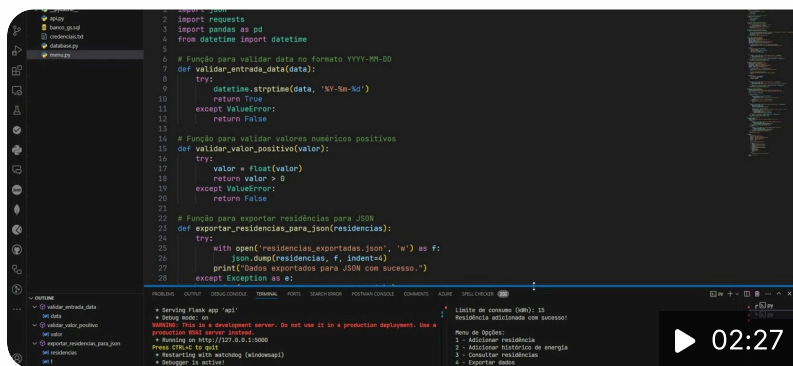
CREATE TABLE residencias (
    id_residencia      NUMBER PRIMARY KEY,
    nome_responsavel   VARCHAR2(100) NOT NULL,
    endereco           VARCHAR2(200) NOT NULL,
    capacidade_geracao  NUMBER(10, 2), -- Capacidade de geração em kWh
    tipo_fonte         VARCHAR2(50),  -- Ex: 'Solar', 'Eólica', etc.
    limite_consumo      NUMBER(10, 2), -- Limite de consumo em kWh
    data_cadastro       DATE DEFAULT SYSDATE
);

CREATE TABLE historico_energia (
    id_historico       NUMBER PRIMARY KEY,
    id_residencia       NUMBER REFERENCES residencias(id_residencia),
    data_registro       DATE NOT NULL,
    producao            NUMBER(10, 2) NOT NULL, -- Produção de energia em kWh
    consumo             NUMBER(10, 2) NOT NULL, -- Consumo de energia em kWh
    saldo_energetico    NUMBER(10, 2)          -- Produção - Consumo
);

CREATE SEQUENCE HISTORICO_SEQ
START WITH 0
INCREMENT BY 1
MINVALUE 0
NOCACHE;

CREATE SEQUENCE RESIDENCIA_SEQ
START WITH 0
INCREMENT BY 1
MINVALUE 0
NOCACHE;
```

Link para o vídeo da solução proposta:



The screenshot shows a video player interface with a dark theme. The main area displays Python code for a Flask API. The code includes imports for Flask, requests, pandas, and datetime. It defines three functions: `validar_data` for validating date format, `validar_valor_positivo` for validating positive numerical values, and `exportar_residencias_para_json` for exporting data to a JSON file. The video player controls at the bottom show a play button and a timestamp of 02:27.

```
1 import flask
2 import requests
3 import pandas as pd
4 from datetime import datetime
5
6 # Função para validar data no formato YYYY-MM-DD
7 def validar_data(data):
8     try:
9         datetime.strptime(data, '%Y-%m-%d')
10    except ValueError:
11        return False
12    return True
13
14 # Função para validar valores numéricos positivos
15 def validar_valor_positivo(valor):
16     try:
17         valor = float(valor)
18         return valor > 0
19     except ValueError:
20        return False
21
22 # Função para exportar residências para JSON
23 def exportar_residencias_para_json(residencias):
24     try:
25         with open('residencias_exportadas.json', 'w') as f:
26             json.dump(residencias, f, indent=4)
27         print("Dados exportados para JSON com sucesso.")
28     except Exception as e:
29         print(f"Erro ao exportar dados: {e}")
```



vídeo software garrovision

