Contexto:

Uma empresa utiliza um sistema CRM (*Customer Relationship Management*) para gerir os seus clientes e vendas. Após vários anos de utilização, os dados ficaram inconsistentes e com erros. Como analista de dados, a tua tarefa é avaliar e melhorar a qualidade dos dados antes de integrares as tabelas num *Data Warehouse*.

Objetivos:

- Identificar e corrigir problemas de qualidade de dados (valores nulos, duplicados, tipos incorretos, idades inválidas).
- Validar regras de integridade referencial entre tabelas.
- Aplicar técnicas de limpeza e normalização.
- Gerar ficheiros limpos para análise posterior.

Ficheiros de trabalho:

- clientes_T5.csv
- vendas_T5.csv

Tarefas:

1. *Leitura dos dados*
   - Lê os ficheiros clientes_T5.csv e vendas_T5.csv.

2. *Deteção de problemas*
   - Verifica valores nulos, duplicados e idades fora do intervalo 18-100.
   - Procura emails mal formatados (sem @ ou domínio).
   - Verifica se todos os id_cliente das vendas existem na tabela de clientes.

3. *Tratamento de dados*
   - Remove duplicados e registos inválidos.
   - Corrige emails e preenche valores nulos.
   - Elimina vendas sem cliente associado.

4. *Normalização*
   - Formata nomes (str.title()).
   - Coloca emails em minúsculas.
   - Ordena clientes por id_cliente.

5. *Exportação*
   - Gera os ficheiros clientes_final.csv e vendas_validas.csv.

Nota (extra não obrigatória):

Implementar validação automática usando expressões regulares para:

- Emails
- Nomes (apenas letras e espaços)

Proposta de Resolução:

```python
import pandas as pd
import numpy as np
import re

# ---------------------- 1. Leitura dos Dados ----------------------

clientes = pd.read_csv("clientes_T5.csv")
vendas = pd.read_csv("vendas_T5.csv")

print("\n➡ Dados lidos com sucesso!")
```

```python
# ---------------------- 2. Deteção de Problemas ----------------------

print("\n--- PROBLEMAS ENCONTRADOS ---\n")

# Valores nulos
print("Valores nulos por coluna (clientes):")
print(clientes.isnull().sum())

print("\nValores nulos por coluna (vendas):")
print(vendas.isnull().sum())

# Duplicados
duplicados = clientes.duplicated(subset=["id_cliente", "email"], keep="first")
print(f"\nClientes duplicados: {duplicados.sum()}")

# Idades fora dos limites
idades_invalidas = clientes[(clientes["idade"] < 18) | (clientes["idade"] > 100)]
print("\nIdades inválidas:")
print(idades_invalidas)

# Emails mal formatados (simples: precisa conter "@", e pelo menos algo depois)
def email_valido(email):
    if pd.isna(email):
        return False
    return bool(re.match(r"^[^@\s]+@[^@\s]+\.[^@\s]+$", email))

clientes["email_valido"] = clientes["email"].apply(email_valido)
emails_invalidos = clientes[clientes["email_valido"] == False]
print("\nEmails inválidos:")
print(emails_invalidos[["id_cliente", "email"]])

# Verificação integridade referencial
clientes_ids = set(clientes["id_cliente"])
vendas_ids_invalidos = vendas[~vendas["id_cliente"].isin(clientes_ids)]
print("\nVendas com id_cliente inexistente:")
print(vendas_ids_invalidos)
```

```python
# ---------------------- 3. Tratamento de Dados ----------------------

# Remover duplicados
clientes = clientes.drop_duplicates(subset=["id_cliente", "email"], keep="first")

# Corrigir nulos: nome recebe "Desconhecido", idade recebe média válida
clientes["nome"] = clientes["nome"].fillna("Desconhecido")

media_idade = clientes["idade"].dropna()
media_idade = media_idade[(media_idade >= 18) & (media_idade <= 100)].mean()
clientes["idade"] = clientes["idade"].apply(lambda x: media_idade if (pd.isna(x) or x < 18 or x > 100) else x)

# Corrigir emails: nulos = "email_desconhecido@empresa.pt"
clientes["email"] = clientes["email"].fillna("email_desconhecido@empresa.pt")

# Revalidar emails e substituir inválidos
clientes["email_valido"] = clientes["email"].apply(email_valido)
clientes.loc[clientes["email_valido"] == False, "email"] = "correção_necessaria@empresa.pt"

# Eliminar vendas com id_cliente inexistente
vendas_validas = vendas[vendas["id_cliente"].isin(clientes["id_cliente"])]

# Eliminar vendas com valor nulo
vendas_validas["valor"] = vendas_validas["valor"].fillna(0)
```

```python
# ---------------------- 4. Normalização ----------------------

# Nomes formatados
clientes["nome"] = clientes["nome"].str.strip()
clientes["nome"] = clients["nome"].str.title()

# Emails minúsculas
clientes["email"] = clientes["email"].str.lower()

# Ordenar por id_cliente
clientes = clientes.sort_values(by="id_cliente")

# Remover coluna auxiliar
clientes = clientes.drop(columns=["email_valido"], errors="ignore")
```

```python
# ---------------------- 5. Exportação ----------------------

clientes.to_csv("clientes_final.csv", index=False)
vendas_validas.to_csv("vendas_validas.csv", index=False)

print("\n✅ Limpeza concluída.")
print("➡ clientes_final.csv criado.")
print("➡ vendas_validas.csv criado.")
```