

# Requisitos e Detalhes do Projeto de Impressão de Etiquetas

## Projeto de Impressão de Etiquetas

### Descrição do Código

O código tem como objetivo processar arquivos de dados (CSV ou Excel), permitir ao usuário seleccionar uma coluna de interesse e gerar arquivos ZPL correspondentes para impressão de etiquetas.

### Funcionalidades do Código

1. Detectar a codificação do arquivo de entrada.
2. Permitir a leitura de arquivos CSV ou Excel.
3. Exibir ao usuário as colunas disponíveis no arquivo e permitir a escolha de uma coluna para processamento.
4. Iterar sobre os valores da coluna escolhida e criar arquivos ZPL individuais para cada registro.
5. Perguntar ao usuário se deseja imprimir cada etiqueta ou pular o registro.
6. Gerar arquivos ZPL na pasta especificada (/Applications/MAMP/htdocs/ZPL\_estudos/etiquetas).
7. Perguntar ao usuário se deseja carregar outro arquivo ou finalizar o programa.

### Pontos Fortes

- Detecção automática de codificação, minimizando erros de leitura.
- Flexibilidade para lidar com diferentes formatos de arquivos (CSV e Excel).
- Geração de arquivos ZPL para impressão.

### Limitações Observadas

## Requisitos e Detalhes do Projeto de Impressão de Etiquetas

1. Interação Linear: O sistema processa os registros um por um, o que pode ser ineficiente para arquivos grandes.
2. Formatos de Etiqueta Limitados: Gera apenas um formato fixo de etiqueta.
3. Controle de Registros: Não permite seleção avançada de registros (como intervalos ou múltiplos registros específicos).
4. Feedback ao Usuário: Poderia fornecer informações mais detalhadas sobre erros.
5. Falta de Modularidade: Algumas funções poderiam ser separadas para facilitar a manutenção e expansão.

### Código

```
import os
import pandas as pd
import chardet

def detecta_codificacao(arquivo):
    with open(arquivo, 'rb') as f:
        resultado = chardet.detect(f.read())
    return resultado['encoding']

def processa_arquivo():
    while True:
        caminho = input("Digite o caminho completo do arquivo (CSV ou Excel): ").strip()
        if not os.path.isfile(caminho):
            print("Erro: O arquivo especificado não foi encontrado.")
            continue

        try:
            codificacao = detecta_codificacao(caminho)
            print(f"Codificação detectada: {codificacao}")
        except Exception as e:
            print(f"Erro ao detectar a codificação do arquivo: {e}")
            continue

        try:
            if caminho.endswith(".csv"):
                df = pd.read_csv(caminho, encoding=codificacao, on_bad_lines="skip")
            elif caminho.endswith((".xls", ".xlsx")):
                df = pd.read_excel(caminho)
```

## Requisitos e Detalhes do Projeto de Impressão de Etiquetas

```
else:
    print("Erro: O arquivo deve ser CSV ou Excel.")
    continue

# Mostra as colunas detectadas no arquivo
print("Colunas detectadas no arquivo:")
print(df.columns.tolist())

# Solicita ao usuário para selecionar uma coluna
coluna_selecionada = input("Digite o nome da coluna que deseja processar: ").strip()
if coluna_selecionada not in df.columns:
    print("Erro: Coluna selecionada não encontrada no arquivo.")
    continue

except Exception as e:
    print(f"Erro ao carregar o arquivo: {e}")
    continue

# Processa os dados
for _, row in df.iterrows():
    try:
        valor = row[coluna_selecionada]
        print(f"Processando valor: {valor}")
    except KeyError:
        print(f"Erro: Coluna '{coluna_selecionada}' não encontrada.")
        break

resposta = input("Deseja imprimir esta etiqueta? (s/n): ").lower()
if resposta == "s":
    nome_arquivo = f"{valor}.zpl"
    caminho_zpl = os.path.join(
        "/Applications/MAMP/htdocs/ZPL_estudos/etiquetas", nome_arquivo
    )
    try:
        with open(caminho_zpl, "w") as f:
            f.write(f"^XA
^FO50,50^ADN,36,20^FD{valor}^FS
^XZ")

        print(f"Arquivo criado: {caminho_zpl}")
    except Exception as e:
        print(f"Erro ao criar o arquivo .zpl: {e}")
    elif resposta == "n":
        print("Impressão cancelada para esta etiqueta.")
    else:
        print("Resposta inválida. Pulei para o próximo valor.")

print("\nTodos os registros foram processados.")
nova_execucao = input("Deseja carregar outro arquivo de banco de dados? (s/n): ").lower()
if nova_execucao != "s":
```

## Requisitos e Detalhes do Projeto de Impressão de Etiquetas

```
        print("Finalizando o programa.")
        break

if __name__ == "__main__":
    os.makedirs("/Applications/MAMP/htdocs/ZPL_estudos/etiquetas", exist_ok=True)
    processa_arquivo()
```

### Planos de Melhoria

- Incorporar suporte para múltiplas colunas e formatos personalizados de etiquetas.
- Adicionar opções de seleção de registros (todos, intervalos ou específicos).
- Oferecer uma interface para selecionar formatos de etiquetas e unidades de medida (mm ou polegadas).
- Otimizar o processo para arquivos grandes, com feedback contínuo ao usuário.
- Modularizar o código para facilitar adições futuras.