

# RELATÓRIO DO PROJETO RPA

## Sistema Automatizado de Coleta de Conselhos

---

### INFORMAÇÕES DO ALUNO

- **Nome:** Jadson Oliveira Porto
  - **Disciplina:** RPA (Robotic Process Automation)
  - **Data de Entrega:** 28/11/2025
- 

### 1. INTRODUÇÃO

Este projeto foi desenvolvido como avaliação final da disciplina de RPA, com o objetivo de criar um sistema automatizado que integra coleta, armazenamento e processamento de dados de uma API pública.

### 2. API ESCOLHIDA E JUSTIFICATIVA

**API Utilizada:** Advice Slip API

**URL:** <https://api.adviceslip.com/>

#### Justificativa da Escolha:

- Fornece dados textuais (conselhos) interessantes para análise
- API simples e confiável
- Resposta em JSON de fácil manipulação
- Não requer chave de autenticação
- Dados variados e em inglês, permitindo análise de conteúdo

### 3. ETAPAS DO PROJETO

#### 3.1 Coleta de Dados da API

- Implementada na função `coletar_dados_api()`
- Requisição HTTP usando biblioteca `requests`
- Tratamento de erros e timeout
- Estruturação dos dados em dicionário Python

## Print da Execução:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\Jadson\Downloads\RPA-FINAL> python rpa_final.py
PROJETO RPA - SISTEMA DE CONSELHOS ALEATÓRIOS
=====

[1] Realizando requisição para a API...
[OK] Dados recebidos com sucesso!

Conselho recebido da API:
>>> Never let your Mother cut your hair.

[2] Criando/verificando banco de dados...
[OK] Banco de dados pronto!

[3] Salvando dados no banco...
[OK] Novo conselho inserido com sucesso!
[OK] Novo conselho inserido com sucesso!

[4] Processando dados...
[PROCESSAMENTO] Total de conselhos no banco: 1
[PROCESSAMENTO] Conselhos longos (>50 chars): 0
[PROCESSAMENTO] Conselhos curtos (≤50 chars): 1

[PROCESSAMENTO] Todos os conselhos armazenados:
1. 'Never let your Mother cut your hair.' - 2025-11-28 09:34:51

=====
[OK] Processo concluído com sucesso!
PS C:\Users\Jadson\Downloads\RPA-FINAL> |
```

## 3.2 Armazenamento em Banco de Dados

- Criação do banco SQLite projeto\_rpa.db
- Tabela conselhos com as colunas:
  - id (INTEGER PRIMARY KEY)
  - conselho (TEXT UNIQUE)
  - data\_coleta (TEXT)
- Controle de duplicatas usando constraint UNIQUE

## Print da Estrutura do Banco:

```
def criar_banco():
    print("\n[2] Criando/verificando banco de dados...")

    conn = sqlite3.connect('projeto_rpa.db')
    cursor = conn.cursor()

    cursor.execute('''
        CREATE TABLE IF NOT EXISTS conselhos (
            id INTEGER PRIMARY KEY,
            conselho TEXT UNIQUE,
            data_coleta TEXT
        )
    ''')

    conn.commit()
    conn.close()
```

### **3.3 Processamento de Dados**

- Análise estatística dos conselhos armazenados
- Contagem total de conselhos
- Classificação por tamanho (curtos/longos)
- Exibição de todos os conselhos armazenados

**Print do Processamento:**

### **4. FLUXO DO SISTEMA**

1. INICIAR PROGRAMA

2. COLETAR DADOS DA API

↓

3. CRIAR/VALIDAR BANCO DE DADOS

↓

4. SALVAR DADOS NO BANCO

↓

5. PROCESSAR E ANALISAR DADOS

↓

6. EXIBIR RELATÓRIO FINAL

### **5. RESULTADOS OBTIDOS**

O sistema demonstrou eficiência em:

- Coleta automática de conselhos da API
- Armazenamento seguro em banco local
- Prevenção de dados duplicados
- Análise estatística do conteúdo
- Execução completa e autonoma

**Exemplo de Saída:**

[PROCESSAMENTO] Total de conselhos no banco: 5

[PROCESSAMENTO] Conselhos longos (>50 chars): 3

[PROCESSAMENTO] Conselhos curtos ( $\leq 50$  chars): 2

## 6. DIFICULDADES ENCONTRADAS

1. **Problema com Schema do Banco:** Inicialmente a tabela tinha mais colunas que valores sendo inseridos
2. **Controle de Duplicatas:** Necessidade de implementar lógica para evitar conselhos repetidos
3. **Persistência de Dados:** O banco estava sendo recriado a cada execução, perdendo dados anteriores

## 7. APRENDIZADOS

- Integração de APIs com Python
- Manipulação de banco SQLite
- Controle de fluxo em automações
- Tratamento de erros e exceções
- Estruturação de projetos RPA
- Processamento e análise de dados textuais

## 8. CONCLUSÃO

O projeto foi desenvolvido com sucesso, atendendo a todos os requisitos solicitados. O sistema demonstra capacidade completa de automação desde a coleta de dados até a análise final, representando os conceitos fundamentais de RPA aprendidos na disciplina.

---

## ANEXOS

- Código fonte completo (arquivo .py)
- Print de execução
- Estrutura do banco de dados