

RELATÓRIO DO PROJETO RPA

Coleta de Dados Meteorológicos

INFORMAÇÕES DO PROJETO

Nome do Aluno:	Ryan Ditkowski
Disciplina:	RPA - Robotic Process Automation
Projeto:	Sistema de Coleta de Dados Meteorológicos
Data:	26/11/2025
Arquivo Principal:	main.py
Banco de Dados:	projeto_rpa.db

API ESCOLHIDA E JUSTIFICATIVA

API Selecionada: OpenWeatherMap API

URL: <https://openweathermap.org/api>

Justificativa da Escolha:

- Gratuita: Oferece plano gratuito com até 1000 requisições/dia
- Documentação completa: API bem documentada e fácil de usar
- Dados em tempo real: Informações meteorológicas atualizadas
- Dados estruturados: Retorna dados JSON bem organizados
- Suporte ao português: Localização em português brasileiro
- Dados relevantes: Temperatura, umidade, pressão, vento, etc.
- Múltiplas cidades: Consulta dados de diversas localidades

ETAPAS EXECUTADAS NO PROJETO

1. Configuração do Ambiente

- Criação do ambiente virtual Python
- Instalação da biblioteca requests
- Configuração da API key diretamente no código
- Estruturação do projeto em funções modulares

2. Integração com a API

- Implementação da função obter_dados_clima()
- API key integrada diretamente no código para facilitar correção
- Tratamento de erros de requisição (timeout, conexão)
- Validação de resposta da API (status codes)
- Processamento de dados JSON retornados

3. Estruturação de Dados

- Criação da função estruturar_dados()
- Extração de campos relevantes da resposta JSON

- Formatação de dados para inserção no banco
- Adição de timestamp local para controle

4. Banco de Dados SQLite

- Criação do banco projeto_rpa.db
- Definição da tabela dados_clima com 14 campos
- Implementação da função inserir_dados_banco()
- Criação de função para consultar registros salvos

5. Interface e Exibição

- Formatação de output no console
- Exibição de dados meteorológicos organizados
- Feedback de progresso durante execução
- Relatório final com últimos registros

PRINTS DA EXECUÇÃO DO PROJETO

Print 1: Início da Execução e Configuração

■■ SISTEMA DE COLETA DE DADOS METEOROLÓGICOS - RPA

=====

✓ API Key carregada com sucesso!

✓ Banco de dados 'projeto_rpa.db' criado/verificado com sucesso!

Print 2: Coleta de Dados - Londres

--- Processando: London ---

Realizando requisição para a cidade: London...

✓ Dados obtidos com sucesso! (Status: 200)

=====

DADOS METEOROLÓGICOS - London, GB

=====

Temperatura: 7.71°C

Sensação Térmica: 5.13°C

Mín/Máx: 6.95°C / 8.47°C

Umidade: 83%

Pressão: 1019 hPa

Descrição: Chuva leve

Velocidade do Vento: 4.12 m/s

Nebulosidade: 40%

Data/Hora da Consulta: 2025-11-26 17:51:59

=====

✓ Dados da cidade 'London' inseridos no banco com sucesso!

Print 3: Relatório Final

■ Últimos 10 registros no banco de dados:

- Paris - 4.09°C - céu limpo - 2025-11-26 17:52:02
- Tokyo - 9.8°C - algumas nuvens - 2025-11-26 17:52:01
- New York - 15.16°C - nublado - 2025-11-26 17:52:00
- São Paulo - 20.8°C - algumas nuvens - 2025-11-26 17:52:00
- London - 7.71°C - chuva leve - 2025-11-26 17:51:59

✓ Processo concluído com sucesso!

✓ Banco de dados: projeto_rpa.db

ESTRUTURA DO BANCO DE DADOS

Nome do Banco: projeto_rpa.db

Total de Registros: 15

Tabela Principal: dados_clima

Dependências: Apenas biblioteca 'requests'

Configuração: API key integrada no código

#	Nome da Coluna	Tipo	Descrição
1	id	INTEGER	Chave primária
2	cidade	TEXT	Nome da cidade
3	pais	TEXT	Código do país
4	temperatura	REAL	Temperatura atual °C
5	sensacao_termica	REAL	Sensação térmica °C
6	temp_minima	REAL	Temperatura mín. °C
7	temp_maxima	REAL	Temperatura máx. °C
8	pressao	INTEGER	Pressão hPa

CONCLUSÃO DO PROJETO

Objetivos Alcançados:

- ✓ Integração bem-sucedida com API externa (OpenWeatherMap)
- ✓ Coleta automática de dados meteorológicos de múltiplas cidades
- ✓ Estruturação e normalização dos dados coletados
- ✓ Armazenamento eficiente em banco de dados SQLite
- ✓ Interface de usuário clara e informativa
- ✓ Tratamento adequado de erros e exceções
- ✓ Código modular e bem documentado

Principais Dificuldades:

- Tratamento de diferentes tipos de erro da API
- Estruturação de dados JSON complexos e aninhados
- Formatação adequada para o formato brasileiro
- Simplificação do código para entrega como arquivo único

Principais Aprendizados:

- Consumo de APIs REST e tratamento de respostas
- Operações com banco de dados SQLite em Python
- Normalização e estruturação de dados externos
- Boas práticas de programação e tratamento de exceções
- Criação de processos automatizados end-to-end
- Desenvolvimento de código independente para fácil distribuição

Conclusão Final:

O projeto foi desenvolvido com sucesso, atendendo a todos os requisitos da prova. O sistema demonstra um processo RPA completo e funcional, desde a coleta automática de dados até o armazenamento estruturado. O código foi otimizado para facilitar a correção, sendo totalmente independente e precisando apenas da biblioteca 'requests'.

A experiência proporcionou conhecimentos valiosos sobre integração de APIs, manipulação de dados e desenvolvimento de soluções automatizadas.

Status: ✓ PROJETO COMPLETO, FUNCIONAL E PRONTO PARA ENTREGA

Facilidade de Correção: ✓ CÓDIGO INDEPENDENTE (apenas main.py)

