

Process e Solution Design

# Case Técnico RPA

Módulo E – Consumo de API Pública

## 1. PDD – Process Definition Document

### 1.1. Visão Geral do Processo

Este documento descreve a automação do processo de **consumo de dados de mercado de criptomoedas** a partir da API pública do **CoinGecko**, com o objetivo de disponibilizar informações atualizadas e confiáveis para áreas como **Inteligência de Mercado, Riscos e Estratégia**.

O processo automatiza a coleta, validação, persistência e distribuição dos dados, eliminando consultas manuais à API ou a sites especializados, além de garantir padronização, rastreabilidade e escalabilidade por meio do uso de **fila no UiPath Orchestrator**.

### 1.2. Processo AS-IS

Antes da automação, a coleta de dados era realizada de forma manual ou semi-manual, conforme descrito abaixo:

1. Acesso manual ao site ou endpoints públicos do CoinGecko.
2. Consulta visual das principais criptomoedas por capitalização de mercado.
3. Cópia manual de valores (preço, volume, market cap).
4. Registro das informações em planilhas Excel.
5. Comparações manuais entre períodos.

**Tempo médio diário:** 30 a 45 minutos

#### **Principais erros identificados:**

- Erros de digitação e cópia;
- Dados desatualizados rapidamente;
- Falta de histórico estruturado;
- Ausência de métricas de execução;
- Baixa rastreabilidade e reprodutibilidade.

### **1.3. Processo TO-BE**

Com a automação implementada em UiPath, o processo passa a ocorrer de forma não assistida:

- Consumo automático da API do CoinGecko.
- Extração estruturada das 5 maiores criptomoedas.
- Registro dos dados em arquivo CSV.
- Publicação dos dados em fila no Orchestrator.
- Coleta de métricas técnicas da execução.

**Tempo estimado após automação:** inferior a 5 minutos por execução.

### **1.4. Escopo da Automação**

Incluído no escopo:

- Consumo da API pública CoinGecko;
- Extração dos principais indicadores de mercado;
- Geração de arquivo CSV diário;
- Publicação de itens em fila no Orchestrator;
- Implementação de retry e backoff;
- Registro de métricas de execução;
- Garantia de idempotência por execução.

Fora do escopo:

- Análise financeira ou recomendações de investimento;
- Integração direta com plataformas de trading;
- Persistência em banco de dados corporativo.

### **1.5. Benefícios Esperados**

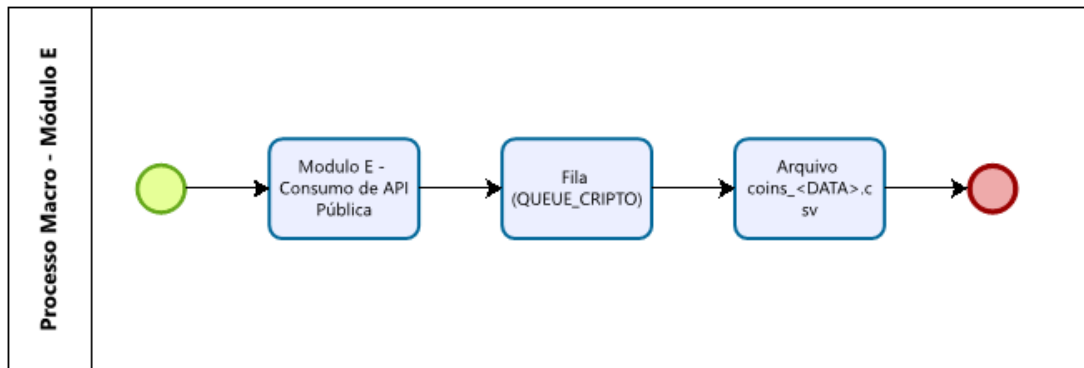
- Redução de tempo de até **85%**;
- Eliminação de erros manuais;
- Dados padronizados e reutilizáveis
- Histórico estruturado e rastreável

- Escalabilidade e fácil manutenção
- Execução automatizadas e confiável

## 2. SDD – Solution Design Document

### 2.1.Arquitetura da Solução

A solução foi desenvolvida em **UiPath**, com integração direta à API do CoinGecko e comunicação via **Queue no Orchestrator**.



### 2.2.Ferramentas e Componentes

- **RPA Tool:** UiPath Studio
- **API:** CoinGecko (pública)
- **Orquestração:** UiPath Orchestrator
- **Queue:** QUEUE\_CRIPTO
- **Configuração:** Assets (diretamente pelo Orchestrator)
- **Formato de saída:** CSV

### 2.3.Módulo E – Consumo de API Pública

#### Responsabilidades:

- Realizar chamada HTTP GET para o endpoint: */api/v3/coins/market*
- Parâmetros utilizados:
  - vs\_currency=usd

- order=market\_cap\_desc
  - per\_page=5
  - page=1
- Extrair os seguintes campos para cada criptomoeda:
  - id
  - symbol
  - name
  - current\_price
  - market\_cap
  - total\_volume
- Registrar métricas da execução:
  - Tempo de resposta da API;
  - Número de registros retornados;
  - Status da chamada HTTP.
- Persistir os dados no arquivo: *coins\_<DATA>.csv*
- Publicar cada moeda como um **Queue Item** na fila QUEUE\_CRIPTO

## 2.4. Idempotência e Controle de Duplicidade

- A automação garante que cada criptomoeda seja publicada apenas uma vez por execução.
- Validação baseada em:
  - Identificador da moeda (id);
  - Data da execução;
- Em caso de reprocessamento, os itens duplicados não são reenviados à fila.

## 2.5. Tratamento de Falhas e Retry

- Implementação de **retry com backoff exponencial** para falhas na chamada da API.
- Tratamento de exceções:
  - **Business Exceptions:** resposta vazia ou inconsistente;
  - **System Exceptions:** timeout, erro HTTP, indisponibilidade da API.

- Todas as exceções são tratadas conforme padrão do REFramework, com logging e retry configurável

## **2.6. Segurança e Governança**

- Parâmetros e URLs externalizados no Config.xlsx;
- Logs centralizados no Orchestrator;
- Monitoramento de execução e falhas via dashboards;
- Processo auditável por execução e item de fila.

## **3. Conclusão**

A automação do Módulo E — Consumo de API Pública estabelece um processo confiável e escalável para obtenção de dados de mercado de criptomoedas, eliminando atividades manuais e garantindo padronização, rastreabilidade e governança.