

Guia de Implementação UiPath - Desafio Banco do Brasil

Este documento serve como o "Blueprint" para a construção do robô no UiPath Studio. Como o formato `.xaml` é binário e complexo, siga este guia estrutural para montar o fluxo.

1. Configuração Inicial (Dependencies)

No UiPath Studio, certifique-se de que os seguintes pacotes estão instalados via **Manage Packages**:

- `UiPath.System.Activities`
- `UiPath.Excel.Activities`
- `UiPath.WebAPI.Activities` (Crucial para chamadas HTTP e JSON)

2. Variáveis Globais (Main Sequence)

Crie as seguintes variáveis no escopo principal:

Nome	Tipo	Descrição
<code>dt_Contas</code>	<code>DataTable</code>	Tabela lida do Excel de input
<code>str_Token</code>	<code>String</code>	Token Bearer recebido da Auth
<code>str_ResponseAPI</code>	<code>String</code>	Resposta crua (JSON string) da API
<code>json_Object</code>	<code>JObject</code>	Objeto JSON desserializado (necessita import <code>Newtonsoft.Json.Linq</code>)
<code>dt_ExtratoTemp</code>	<code>DataTable</code>	Tabela temporária para construir o extrato de uma conta
<code>dt_Log</code>	<code>DataTable</code>	Tabela para acumular status (Agencia, Conta, Status)
<code>clientID</code>	<code>String</code>	Seu Client ID
<code>clientSecret</code>	<code>String</code>	Seu Client Secret

3. Fluxo de Trabalho (Workflow)

Etapa 1: Inicialização e Leitura

1. Activity: Build Data Table -> Saída: `dt_Log`. Configure as colunas: "Agencia", "Conta", "Status".
2. Activity: Read Range Workbook
 - File: "contas_input.xlsx"
 - Output: `dt_Contas`

Etapa 2: Autenticação (Try/Catch)

Envolve em um Try Catch.

1. Activity: HTTP Request (Dentro do Try)

- Method: POST
- EndPoint: "https://oauth.sandbox.bb.com.br/oauth/token"
- Authentication: Basic (Insira ClientID e Secret nos campos Username/Password da activity ou monte o header manualmente).
- Parameters (Collection):
 - grant_type = "client_credentials"
 - scope = "extratos.leitura"
- Result (Response Content): str_ResponseAuth
- StatusCode: int_StatusAuth

2. Activity: If (int_StatusAuth = 200)

- Then:
 - Deserialize JSON: Input str_ResponseAuth , Output json_Auth .
 - Assign: str_Token = json_Auth("access_token").ToString
- Else:
 - Throw: New Exception("Falha na autenticação").

Etapa 3: Processamento (Iteração)

1. Activity: For Each Row in Data Table (Input: dt_Contas)

- Body:
 1. Assign: str_Agencia = CurrentRow("Agencia").ToString
 2. Assign: str_Contra = CurrentRow("Conta").ToString
 3. Try Catch (Para garantir que erro em uma conta não pare o robô)
 - Try:
 1. HTTP Request (Consultar Extrato)
 - Method: GET
 - URL: "https://api.hm.bb.com.br/contas/v1/extratos"
 - Headers:
 - Authorization = "Bearer " + str_Token
 - X-Developer-Application-Key = "SUA_KEY"
 - Parameters: agencia = str_Agencia , conta = str_Contra
 - Result: str_ResponseAPI
 2. Deserialize JSON: Input str_ResponseAPI , Output json_Object .
 3. Sequence (Parse JSON to DataTable):

- Use um loop sobre `json_Object("data")` ou use atividades de conversion se disponíveis.
- Popule `dt_ExtratoTemp` com as colunas: `dataLancamento`, `numeroDocumento`, `valorLancamento`, `textoDescricaoHistorico`.

4. Write Range Workbook (Excel Final)

- `File: "extrato_consolidado.xlsx"`
- `SheetName: str_Agencia + "-" + str_Conta`
- `DataTable: dt_ExtratoTemp`

5. Add Data Row (Sucesso)

- `ArrayRow: {str_Agencia, str_Conta, "OK"}`
- `DataTable: dt_Log`
- **Catches (System.Exception):**
 1. Add Data Row (Falha)
 - `ArrayRow: {str_Agencia, str_Conta, "ERRO: " + exception.Message}`
 - `DataTable: dt_Log`

Etapa 4: Finalização

1. Activity: Write Range Workbook

- `File: "extrato_consolidado.xlsx"`
- `SheetName: "Log_Execucao"`
- `DataTable: dt_Log`

Dicas de Boas Práticas (Implementadas acima)

- **Credential Assets:** Em um cenário real, nunca hardcode o ClientID/Secret. Use `Get Credential` do UiPath Orchestrator.
- **Timeout:** Configure o Timeout das requisições HTTP para 15000ms ou mais para evitar erros de rede.
- **Limpeza:** Use `Clear Data Table` no `dt_ExtratoTemp` no início de cada loop.