Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**CLIENTE: T2C GROUP**

**Nome do processo**

**Projeto Integração**

DOCUMENTO DO DESENHO DA SOLUÇÃO

**ÍNDICE**

[1 OBJETIVO 3](#_Toc87537873)

[2 HISTÓRICO DOCUMENTAÇÃO 4](#_Toc87537874)

[3 DETALHES DO PROCESSO AUTOMATIZADO 5](#_Toc87537875)

[4 guia execução 6](#_Toc87537876)

[4.1 Estrutura arquitetônica do projeto 6](#_Toc87537877)

[4.2 Detalhes da Execução do Projeto 7](#_Toc87537878)

[4.3 Workflows do projeto 9](#_Toc87537879)

[4.4 Descrição dos argumentos dos Workflows 10](#_Toc87537880)

[4.5 Assets (Data\Config.xlsx) 11](#_Toc87537881)

[4.6 Settings (Data\Config.xlsx) 12](#_Toc87537882)

[4.7 Pacotes 13](#_Toc87537883)

[4.8 Scripts Utilizados 14](#_Toc87537884)

[4.9 Figuras 15](#_Toc87537885)

[4.9.1 Exemplo: Figura 1 – Planilha de execução dos processos 15](#_Toc87537886)

# OBJETIVO

Robô para extrair cidades do sul de Minas Gerais, obter mais informações de cada cidade, montar uma planilha bem estruturada para fácil entendimento e inserir os dados em um banco de dados.

* Buscar 100 cidades do sul de Minas Gerais no site do Correio
* Inserir em uma fila no orquestrador.
* Acessar o site do IBGE em todas as cidades na fila do orquestrador
* Extrair os dados de cada cidade
* Classifica-las pela quantidade de população
* Inserir no Banco de dados
* Gerar arquivo Excel e CSV

# HISTÓRICO DOCUMENTAÇÃO

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Nome | Comentários |
| 31/01/2022 | 1.0 | Projeto Integração SDD | Documentação Inicial |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# DETALHES DO PROCESSO AUTOMATIZADO

Os detalhes preenchidos devem refletir as informações reais do Projeto Mestre liberado para produção. A seguinte tabela será preenchida:

|  |  |
| --- | --- |
| **Item** | **Descrição** |
| **Nome do Projeto** | REFprj\_integracao\_Alison |
| **Tipo de Robô** | Atendido |
| **O processo é executado em Segundo plano?  (Sem interação com a UI)** | Não |
| **Usa o orquestrador?** | e.g. Sim |
| **Escalável?** | e.g. Não |
| **Versão UiPath** | 21.10.4.0 |

# guia execução

## Estrutura arquitetônica do projeto

Exibir a interação entre os componentes (pacote / robôs, filas do orquestrador e ordem de execução) em um diagrama. (Utilizar Tema Light/Claro)

Diagram, schematic

Description automatically generated

## Detalhes da Execução do Projeto

Descreve os detalhes do processo automatizado, preenchendo a tabela abaixo.

|  |  |
| --- | --- |
| **TÍTULO DO ITEM** | **DESCRIÇÃO**  *Preencha cada seção em negrito - campos vazios não são permitidos. Se a seção não se aplica à sua automação, marque como n / a.* |
| Detalhes do ambiente de produção | **Executa em máquina local.** |
| Pré-requisitos para a execução | **Arquivo config atualizado, conta no UiPath orquestrador.** |
| Dados de entrada | **Orquestrador.** |
| Saída esperada | **Leitura da fila do orquestrador.** |
| Como iniciar o processo automatizado | **Iniciar manual, a partir do projeto.** |
| Comunicando (relatórios de filas, Kibana, Excel ou outra plataforma) | **As informações serão inseridas no banco de dados e em um arquivo Excel e Csv.** |
| Como o orquestrador é usado? | **Armazena os dados em uma fila e executado quando solicitado.** |
| Políticas de senha (mencione quaisquer solicitações de conformidade específicas) | **n / a** |
| Credenciais armazenadas (Nunca use credenciais de hardcore no fluxo de trabalho!) | **n / a** |
| Lista de nomes de filas | **Fila orquestrador: ProjetoIntegração** |
| Detalhes da programação | **n / a** |
| Compatível com várias resoluções? (no caso de automação de imagem / Citrix e VDI) | **Não** |
| Resolução utilizada no desenvolvimento | **1920x1080** |
| Ambiente usado para desenvolvimento (nome, localização, detalhes de configuração, etc.) | **dellt2cn088\alison.santos** |
| Pré-requisitos de aplicações (Detalhes do sistema operacional, bibliotecas, aplicativos necessários) | **Windows, Google Chrome e Excel.** |
| Repositório para projeto (onde está armazenado o projeto desenvolvido) | **GitHub: https://github.com/AlisonLetos/prj\_integracao\_Alison** |
| Método de configuração (assets, excel file, Json file) | **Excel File.** |
| Lista de tabelas utilizadas | **Tabela Cidades** |
| Base de Dados onde estão as tabelas | **Landing** |
| Procedures utilizadas (Banco de dados) | **n / a** |
| Triggers utilizadas (Banco de dados) | **n / a** |

## Workflows do projeto

Para os arquivos de workflows definidos abaixo, especifique a descrição dos workflows.

|  |  |  |
| --- | --- | --- |
| Nome | Argumentos | Descrição |
| flw\_Send\_DadosIBGE | in\_TransactionItem  arg\_inIntTransactionNumber  arg\_inDtCidades | Acessa o site da IBGE, navega entre as cidades que estão no orquetrador e extrai os seus dados. |
| flw\_Send\_EnviandoBancoDeDados | in\_Config  in\_SystemException  arg\_inDtCidades | Remove a coluna não utilizada, classifica as cidades pela quantidade de população e insere no banco de dados |
| flw\_Send\_LeituraDadosCorreio | in\_Config  arg\_outDtCidades | Acessa o site do Correio, extrai os dados das cidades de MG, filtra somente as 100 primeiras e insere no orquestrador |
| Main | in\_OrchestratorQueueName  in\_OrchestratorQueueFolder  arg\_inDtCidades | Fluxo principal do projeto. |
| CloseAllApplications |  | Do the necessary procedures for ending the process (e.g., logout) and close the used applications. |
| GetTransactionData | in\_TransactionNumber  in\_Config  out\_TransactionItem  out\_TransactionField1  out\_TransactionField2  out\_TransactionID  io\_dt\_TransactionData | Get a transaction item from a specified source (e.g., Orchestrator queues, spreadsheets, databases, mailboxes or web APIs).  If there are no transaction items remaining, out\_TransactionItem is set to Nothing, which leads to the End Process state.  For cases in which there is only a single transaction (i.e., a linear process), use an If activity to check whether the argument in\_TransactionNumber has the value 1 (meaning it is the first and only transaction) and assign the transaction item to out\_TransactionItem. For any other value of in\_TransactionNumber, out\_TransactionItem should be set to Nothing.  If there are multiple transactions, use the argument in\_TransactionNumber as an index to retrieve the correct transaction to be processed. If there are no more transactions left, it is necessary to set out\_TransactionItem to Nothing, thus ending the process. |
| InitAllApplications | in\_Config | Open applications used in the process and do necessary initialization procedures (e.g., login). |
| InitAllSettings | in\_ConfigFile  in\_ConfigSheets  out\_Config | Initialize, populate and output a configuration Dictionary to be used throughout the project.  Settings and constants are read from the local configuration file, and assets are fetched from Orchestrator.  Asset values overwrite settings and constant values if they are defined with the same name. |
| KillAllProcesses |  | Use the Kill Process activity to force the termination of the Windows processes representing applications used in the business process being automated.  Note that killing processes might have undesirable outcomes, such as losing unsaved changes to files. |
| Process | in\_TransactionItem  in\_Config  arg\_inDtCidades  arg\_inIntTransactionNumber | Invoke major steps of the business process, which are usually implemented by multiple subworkflows.  If a BusinessRuleException is thrown, the transaction is skipped.  If another kind of exception occurs, the current transaction can be retried. |
| RetryCurrentTransaction | in\_Config  io\_RetryNumber  io\_TransactionNumber  in\_SystemException  in\_QueueRetry | Gerencia o mecanismo de repetição da estrutura e é invocado em SetTransactionStatus.xaml quando ocorre uma exceção do sistema. |
| SetTransactionStatus | in\_BusinessException  in\_TransactionField1  in\_TransactionField2  in\_TransactionID  in\_SystemException  in\_Config  in\_TransactionItem  io\_RetryNumber  io\_TransactionNumber  io\_ConsecutiveSystemExceptions | Defina e registre o status da transação junto com campos de log extras.  Pode haver três status possíveis: Sucesso, Exceção de Negócios e Exceção do Sistema.  A Exceção de Regra de Negócio caracteriza uma situação irregular de acordo com as regras do processo e impede que a transação seja processada. A transação não é repetida neste caso, pois o resultado será o mesmo até que o problema que causa a exceção seja resolvido.  Por exemplo, pode ser considerado um BusinessRuleException se um processo espera ler o anexo de um email, mas o remetente não anexou nenhum arquivo. Nesse caso, as tentativas imediatas da transação não produzirão um resultado diferente.  Por outro lado, as exceções do sistema são caracterizadas por exceções cujos tipos são diferentes de BusinessRuleException. Quando esse tipo de exceção ocorre, o item da transação pode ser tentado novamente após o fechamento e a reabertura das aplicações envolvidas no processo. A lógica por trás disso é que a exceção foi causada por um problema nos aplicativos, que pode ser resolvido reiniciando-os.  Se as filas do orquestrador forem a origem das transações, a atividade Definir status da transação será usada para atualizar o status. Além disso, o mecanismo de repetição também é implementado pelo Orchestrator.  Se as filas do orquestrador não forem usadas, o status poderá ser definido, por exemplo, gravando em uma coluna específica em uma planilha. Nesses casos, o mecanismo de novas tentativas é coberto pela estrutura e o número de tentativas é definido no arquivo de configuração.  Ao final, io\_TransactionNumber é incrementado, o que faz com que o framework obtenha a próxima transação a ser processada. |
| TakeScreenshot | in\_Folder  io\_FilePath | Capture a screenshot, log its name and location and save it with the PNG extension.  If no specific filepath is passed as argument, it saves the image in the folder specified by in\_Folder. |
| GeneralTestCase | in\_WorkflowFile  in\_ExpectedResult | GeneralTestCase.xaml is a data driven test case based on the Tests.xlsx, Tests sheet where the developer will write the workflow paths of the workflows to be tested and the expected exception - AppEx, BRE or Success. Running GeneralTestCase with data variations is going to be a clear result of comparing the expected result with the actual result after passing through this list of workflows as test data.  There are two possible statuses - PASS or FAIL for each workflow that was run. The status is PASS if the actual exception caught is the one previously defined in the Tests sheet and FAIL otherwise. |
| GetTransactionDataTestCase |  | Given the TransactionNumber, verify if GetTransactionData workflow works as expected.  Once a Transaction Item has been processed, its status will be In Progress.  Queue name should be configured in order for the test case to run. |
| InitAllApplicationsTestCase |  | Verify if the InitAllApplications workflow works as expected.  The verification should check if after opening the applications, the expected state is reached. |
| InitAllSettingsTestCase |  | Verify if the InitAllSettings workflow works as expected.  The verification should check if initalization of settings was successful: if the Config dictionary was created, if it contains a certain key etc. |
| MainTestCase |  | Verify if the Main workflow works as expected.  The verification should check whether the status file or report built after the process run is the expected one. |
| ProcessTestCase |  | Verify if the Process workflow works as expected.  The verification should check whether the output of the Process workflow is the expected one. |
| WorkflowTestCaseTemplate |  | Template workflow used to create tests for workflows in the process.  Create a new test workflow by copying and renaming this file. |

## Descrição dos argumentos dos Workflows

Especifique as funcionalidades dos argumentos definidos abaixo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nome | Direção | Tipo | Descrição | Workflow |
| in\_TransactionItem | In | QueueItem | Argumento de entrada para receber os itens do orquestrador | flw\_Send\_DadosIBGE |
| arg\_inIntTransactionNumber | In | Int32 | Argumento de entrada para receber o indice da fila do orquestrador. | flw\_Send\_DadosIBGE |
| arg\_inDtCidades | In | DataTable | Argumento de entrada para receber a tabelas com os dados do site do Correio | flw\_Send\_DadosIBGE |
| in\_Config | In | Dictionary(String, Object | Argumento de entrada para acessar os dados da planilha Config | flw\_Send\_EnviandoBancoDeDados |
| in\_SystemException | In | Exception | Argumento de entrada para armazenar quando ocorre uma exceção na fila do orquestrador. | flw\_Send\_EnviandoBancoDeDados |
| arg\_inDtCidades | In | DataTable | Argumento de entrada para acessar os dados extraidos do site do Correio e do IBGE | flw\_Send\_EnviandoBancoDeDados |
| in\_Config | In | Dictionary(String, Object | Argumento para receber os dados da planilha Config | flw\_Send\_LeituraDadosCorreio |
| arg\_outDtCidades | Out | DataTable | Argumento de saída para enviar a tabela com os dados extraidos do site do Correio | flw\_Send\_LeituraDadosCorreio |
| in\_OrchestratorQueueName | In | String | Allows the Orchestrator queue name to be passed as an argument, instead of only being defined in the configuration file. | Main |
| in\_OrchestratorQueueFolder | In | String | Allows the Orchestrator folder name where the queue is created to be passed as an argument, instead of only being defined in the configuration file. | Main |
| arg\_inDtCidades | In | DataTable | Argumento para receber a tabela com os dados extraidos no site do correio | Main |
| in\_TransactionNumber | In | Int32 | Sequential counter of transaction items. | GetTransactionData |
| in\_Config | In | Dictionary(String, Object | Dictionary structure to store configuration data of the process (settings, constants and assets). | GetTransactionData |
| out\_TransactionItem | Out | QueueItem | Transaction item to be processed. | GetTransactionData |
| out\_TransactionField1 | Out | String | Allow the optional addition of information about the transaction item. | GetTransactionData |
| out\_TransactionField2 | Out | String | Allow the optional addition of information about the transaction item. | GetTransactionData |
| out\_TransactionID | Out | String | Transaction ID used for information and logging purposes. Ideally, the ID should be unique for each transaction. | GetTransactionData |
| io\_dt\_TransactionData | InOut | DataTable | This variable can be used in case transactions are stored in a DataTable (for example, after being retrieved from a spreadsheet). | GetTransactionData |
| in\_Config | In | Dictionary(String, Object | Dictionary structure to store configuration data of the process (settings, constants and assets). | InitAllApplications |
| in\_ConfigFile | In | String | Path to the configuration file that defines settings, constants and assets. | InitAllSettings |
| in\_ConfigSheets | In | String[] | Names of the sheets corresponding to settings and constants in the configuration file. | InitAllSettings |
| out\_Config | Out | Dictionary(String, Object | Dictionary structure to store configuration data of the process (settings, constants and assets). | InitAllSettings |
| in\_TransactionItem | In | QueueItem | Transaction item to be processed. | Process |
| in\_Config | In | Dictionary(String, Object | Dictionary structure to store configuration data of the process (settings, constants and assets). | Process |
| arg\_inDtCidades | In | DataTable | Argumento de entrada para receber os dados extraidos no site do Correio | Process |
| arg\_inIntTransactionNumber | In | Int32 | Argumento de entrada para receber o indice da fila do orquestrador. | Process |
| in\_Config | In | Dictionary(String, Object | Dictionary structure to store configuration data of the process (settings, constants and assets). | RetryCurrentTransaction |
| io\_RetryNumber | InOut | Int32 | Used to control the number of attempts of retrying the transaction processing in case of system exceptions. | RetryCurrentTransaction |
| io\_TransactionNumber | InOut | Int32 | Sequential counter of transaction items. | RetryCurrentTransaction |
| in\_SystemException | In | Exception | Used during transitions between states to represent exceptions other than business exceptions. | RetryCurrentTransaction |
| in\_QueueRetry | In | Boolean | Used to indicate whether the retry procedure is managed by an Orchestrator queue. | RetryCurrentTransaction |
| in\_BusinessException | In | BusinessRuleException | Exception variable that is used during transitions between states and represents a situation that does not conform to the rules of the process being automated. | SetTransactionStatus |
| in\_TransactionField1 | In | String | Optionally used to include additional information about the transaction item. | SetTransactionStatus |
| in\_TransactionField2 | In | String | Optionally used to include additional information about the transaction item. | SetTransactionStatus |
| in\_TransactionID | In | String | Used for information and logging purposes. Ideally, the ID should be unique for each transaction. | SetTransactionStatus |
| in\_SystemException | In | Exception | Used during transitions between states to represent exceptions other than business exceptions. | SetTransactionStatus |
| in\_Config | In | Dictionary(String, Object | Dictionary structure to store configuration data of the process (settings, constants and assets). | SetTransactionStatus |
| in\_TransactionItem | In | QueueItem | Transaction item to be processed. | SetTransactionStatus |
| io\_RetryNumber | InOut | Int32 | Used to control the number of attempts of retrying the transaction processing in case of system exceptions. | SetTransactionStatus |
| io\_TransactionNumber | InOut | Int32 | Sequential counter of transaction items. | SetTransactionStatus |
| io\_ConsecutiveSystemExceptions | InOut | Int32 | Used to control the number of consecutive system exceptions. | SetTransactionStatus |
| in\_Folder | In | String | Path to the folder where the screenshot should be saved. | TakeScreenshot |
| io\_FilePath | InOut | String | Optional argument that specifies the path and the name of the screenshot to be taken. | TakeScreenshot |
| in\_WorkflowFile | In | String | Armazena o caminho do arquivo de fluxo de trabalho. | GeneralTestCase |
| in\_ExpectedResult | In | String | Armazena a sáida esperada. | GeneralTestCase |

## Assets (Data\Config.xlsx)

Inclua a lista de assets (encontrados na aba Assets do arquivo Data\Config.xlsx) e uma descrição de alto nível para cada um deles, para explicar sua finalidade:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Asset | OrchestratorAssetFolder | Description (Assets will always overwrite other config) |

## Settings (Data\Config.xlsx)

Inclua a lista de valores da aba settings do arquivo Data\Config.xlsx e uma descrição de alto nível para cada um deles, para explicar sua finalidade:

|  |  |  |
| --- | --- | --- |
| Name | Value | Description |
| OrchestratorQueueName | ProjetoIntegração | Orchestrator queue Name. The value must match with the queue name defined on Orchestrator. |
| OrchestratorQueueFolder | Default | Folder name. The value must match a folder defined in Orchestrator and queue specified as OrchestratorQueueName should be created in this folder. For classic folders leave the value field empty. |
|  |  |  |
| logF\_BusinessProcessName | Framework | Logging field which allows grouping of log data of two or more subprocesses under the same business process name |
|  |  |  |
| Correio\_URL | https://buscacepinter.correios.com.br/app/faixa\_cep\_uf\_localidade/index.php | URL site do Correio |

## Pacotes

Inclua a lista de pacotes e uma descrição de alto nível para cada um deles, para explicar sua finalidade:

|  |  |
| --- | --- |
| Nome do Pacote | Descricao |
| UiPath.Database.Activities : [1.5.0] | Pacote para conectar e manipular banco de dados |
| UiPath.Excel.Activities : [2.11.4] | Pacote para manipular planilhas no Excel |
| UiPath.System.Activities : [21.10.3] | Pacote com as principais atividades do UiPath, para manipular tabelas e coleções de dados. |
| UiPath.Testing.Activities : [1.4.4] | Atividades de teste que permitem aos usuários verificar valores de expressão e estados de controle em fluxos de trabalho de casos de teste. |
| UiPath.UIAutomation.Activities : [21.10.4] | O pacote contém atividades principais que permitem a automação de aplicativos de desktop, navegadores e máquinas virtuais. |
| UiPathTeam.NormalizeNumber.Activities : [1.0.0] | Essa atividade recebe uma entrada de string, um número não normalizado e gera sua conversão para Double. |

## Scripts Utilizados

Inclua os scripts utilizados, por exemplo códigos em python, requests de API, códigos em SQL, Javascript, HTML, etc.:

Text

Description automatically generated

## Figuras

Inclua a lista de figuras usadas como exemplo durante a explicação:

### Figura 1 – Extraindo dados do Correio

Graphical user interface, application

Description automatically generated

### Figura 2 – Extraindo dados do IBGE

Graphical user interface, application

Description automatically generated

### Figura 3 – Planilha com o resultado das extrações

Table

Description automatically generated with low confidence

### Figura 4 – Dados inseridos no banco de dados

Graphical user interface, table

Description automatically generated