

[FRAMEWORK] Fluxograma | Fluxograma de Funcionamento do Framework de Desenvolvimento de Scripts de Automação

Este Fluxograma representa o workflow de funcionamento do Framework de Desenvolvimento automatizado, porém, este é um workflow de exemplo de execução, onde cada etapa é bem definida por suas etapas, normalmente, no log, de cada script.

Initialize Environment

O Fluxograma se inicia no estado de Initialize Environment, seu objetivo é fazer com todo tipo de configuração, relacionado ao ambiente de execução do script, organização de pastas e demais utilidades, inicialização de variáveis de globais e tabelas para coleta de informações de relatório.

Descrição de funções

- `read_config_file`: esta função lê o json configurado para montar as variáveis de config e seu output deve ser uma variável global com essas dados para uso em outras etapas do Framework
- `create_queue_item`: esta função tem como objetivo ler o json de queue item e retornar uma variável com esses itens e suas informações para uso posterior (importante lembrar que no config teremos uma variável que será manipulada como um número inteiro que determinará a quantidade de vezes que um item deve ser lido, logo, cada item deve receber uma característica de contador de tentativas)
- `clear_queue_item`: esta função limpa pastas de um temporário da automação ou cria caso não existam e final é que as pastas criadas sejam todas dentro da estrutura do projeto, nunca pastas de rede etc.)
- `create_global_variables`: esta função é responsável por criar variáveis globais para uso em outros scripts, tais como variáveis de logs, variáveis de coleta de informações e etc.

Initialize Applications

Com o sucesso de estado de Initialize Environment a etapa de Initialize Applications tem como objetivo iniciar quaisquer recursos relacionados às aplicações utilizadas no script.

Descrição de funções

- `init_all_applications`: esta função monta todas as processos de aplicações que serão utilizadas na automação (processos montados desde o final de estado)
- `initiate_all_applications`: esta função inicializa quaisquer aplicações utilizadas no processo, retornando uma variável que aponta para o estado, assim que o estado deve ser considerado "sucesso", assim a execução do SMT por exemplo, sua função é deixar o terminal o mais preparado possível para o estado de process

Get Queue Items

Este estado primeiramente verifica uma variável do processo que define se um item deve ser repetido, caso seja a primeira execução ou não, se não tem para ser repetido, ele parte para a função `get_queue_item` (esta função trata a queue item e segmenta eles para o processo de processamento, em caso de ainda existirem queue items o destino é o processo main, caso não tenham mais items no file, o destino é o final state)

Caso um item deve ser repetido, ele primeiramente executa uma função chamada `replicate_item` que é responsável por replicar o item e ser repetido no final da lista de items, para depois voltar para função de `get_queue_item`.

Descrição de funções

- `get_queue_item`: esta função lê o json de items de file e se segmenta através de uma lista de items que estão aguardando processamento
- `replicate_item`: verifica a requisição da função `replicate_item` presente no processo, caso esta variável venha com alguma informação, é entendido que um item falhou e que deve ser repetido, logo, será replicado no file de items (items preparados para que cada item tem um contador de repetições, logo, esse contador deve ser incrementado e verificado para saber se a variável global que determinará a quantidade máxima de tentativas não foi atingida)

Process

Este estado é responsável por contemplar o passo a passo do processo que se deseja automatizar. São 10 os possíveis estados de um processo:

- `Success`
- `SE - Exceção de Sistema - Problema com aplicações ou sistema`
- `BRE - Business Rule Exception - Problema com input de dados errôneos`

Descrição de funções

- `process`: esta função é responsável por contemplar o passo a passo do processo que se deseja automatizar
- `consolidate_report`: consolidar as tabelas de captação de informações de relatório da automação em excel para ser enviado pelo e-mail (é importante lembrar que além das tabelas partíveis do framework, esta função deve receber uma variável booleana que indica a primeira de tabelas criada pelo dev, caso positivo, para cada variável declarada em uma lista, um log deve existir criando novas variáveis, assim, assim para contemplar cada tabela criada e mais pelo dev)
- `send_email_to_user`: envia o e-mail ao(s) usuário(s) da automação com o excel consolidado pela função anterior
- `update_queue_item_status_system`: esta função atualiza o status do item como SE, passando informações de erro source e error message

Final State

Este estado é responsável por finalizar o processo, enviando items ao data lake e e-mails as partes envolvidas contendo um relatório do processo.

Descrição de funções

- `send_items_to_data_lake`: esta função é responsável por enviar a data base no dados da execução do job e dos queue items processados, bem como suas informações e status (esse conexão é feita via 10 times do processo (ou 100 vezes) e um ID único do job e o queue item serão enviados (com seus próprios IDs únicos) ao ID do job)
- `update_queue_item_status_success`: esta função atualiza o status do item como sucesso
- `update_queue_item_status_failure`: esta função atualiza o status do item como BRE, passando informações de erro source e error message
- `send_email_to_dev_team`: envia um e-mail para o time de dev com informações de exceção, relatório, informações sobre o processo

