**Desafio Python (RPA)**

**Construir um Gerenciador de Recursos**

**Objetivo:**

Neste desafio, você será responsável por desenvolver um script em Python para automatizar um processo específico de negócio. O objetivo é criar um bot que seja capaz de realizar tarefas repetitivas em um cenário fictício de processamento de pedidos em uma loja online. O desafio consiste em demonstrar suas habilidades de desenvolvimento em Python, bem como seu entendimento dos conceitos de automação de processos.

**Tarefas:**

**1 - Descrição do processo:**

* O processo consiste em receber, processar e registrar pedidos de clientes em uma loja online fictícia.

R: Neste projeto foi utilizado o site para treinamento RPA da AutomationAnywhere, considerando os detalhes, foi desenvolvido um projeto para o desafio Online Grocery Ordering. Antes da execução ler o arquivo README.md para entendimento melhor do fluxo e instalação de dependências necessárias para a execução.

* Descreva em detalhes o fluxo do processo, desde o momento em que um pedido é recebido até a sua conclusão.

R: O processo inicia quando um pedido é recebido, geralmente em forma de um arquivo CSV contendo detalhes sobre os itens que precisam ser processados. Esse arquivo pode ser obtido via requisições HTTP utilizando a classe RequestManager.

Inicialização do Sistema:

Ao entrar no estado INITIALIZATION, o sistema configura o ambiente. Isso inclui a inicialização do logger para registrar informações sobre o fluxo do programa, a configuração do WebDriver para interações com a interface do usuário e a realização do login no site onde os pedidos serão feitos. O sistema verifica se é a primeira execução, inicializando as configurações necessárias, como desativar notificações do navegador e maximizar a janela.

Leitura do Arquivo de Pedidos:

Após a inicialização, o sistema entra no estado GET TRANSACTION, onde lê o arquivo CSV. A leitura é feita de forma a capturar cada item pelo seu ID, permitindo uma manipulação eficiente e a possibilidade de paralelismo no processamento de pedidos.

Processamento do Pedido:

O sistema verifica o status de cada item. Se o status for "pendente", o pedido é preparado para processamento. O status é atualizado para "Processando" no arquivo CSV, e a ordem de comida é inserida na interface do usuário através da classe PageMain, que utiliza o Selenium para automação.

Submissão do Pedido:

Após a inserção, o sistema aguarda a confirmação do pedido. Ele captura dados relevantes, como o tempo de processamento e a acurácia da operação, que são essenciais para análises futuras.

Tratamento de Erros:

Caso ocorra um erro durante o processamento, o sistema atualiza o status do item para "Falha", captura uma imagem da tela para documentação e registra o erro no log. Isso garante que todos os problemas sejam devidamente registrados e tratados.

Finalização do Processo:

Uma vez que todos os itens tenham sido processados, o sistema entra no estado END. Aqui, o pedido é submetido formalmente, e uma captura de tela de sucesso é feita para a documentação. O arquivo processado é movido para um diretório designado, organizando os dados de forma que possam ser facilmente acessados posteriormente. O logger registra informações sobre o tempo de processamento e a acurácia do preenchimento.

Encerramento:

O sistema finaliza o processo fechando o driver do Selenium e liberando todos os recursos utilizados. O loop do programa é desativado, sinalizando que todos os pedidos foram processados com sucesso.

* Identifique as etapas do processo que podem ser automatizadas e quais tarefas são executadas manualmente.

R: Todas as etapas do processo podem ser automatizadas, incluindo a requisição do arquivo via HTTP, eliminando a necessidade de intervenção manual, como a inserção do arquivo em uma pasta. O robô gerencia todo o fluxo de pedidos automaticamente. Somente a análise pós processamento deve ser feita por um usuário com experiência no processo.

**2 - Automação das tarefas:**

* Utilizando as bibliotecas e ferramentas adequadas, desenvolva um script Python que automatize as tarefas do processo de pedidos. R: Sim, o código todo está no arquivo zip, o main é o arquivo principal e na pasta src é onde se encontram os módulos desenvolvidos.
* O bot deve ser capaz de receber informações sobre um novo pedido, processar os dados necessários, gerar relatórios e registrar as informações relevantes. R: Sim, o robô realiza isso com a geração de logs, screenshots da tela para contingência e documentação, e realiza o processamento dos dados contidos no arquivo.
* Certifique-se de que o bot seja capaz de lidar com diferentes cenários, como pedidos com quantidades variadas de produtos, informações de clientes incompletas, entre outros.

R: Sim, o bot é projetado para lidar com diferentes cenários por meio do uso de uma máquina de estados, permitindo uma leitura flexível da planilha e manipulação dinâmica dos dados. Ele consegue gerenciar pedidos com quantidades variadas de produtos e lidar com informações incompletas graças às tratativas de erros, assim, garantindo um fluxo robusto e adaptável.

**3 - Manipulação de dados:**

* Implemente funcionalidades no seu script para ler, processar e armazenar os dados relevantes dos pedidos.

R: Sim, o robô possui funcionalidades para tratar, armazenar, alterar, excluir, ler e inserir dados no sistema, através do uso da biblioteca Pandas e Selenium.

* Utilize bibliotecas adequadas para realizar tarefas como leitura e escrita de arquivos, consulta a bancos de dados ou uso de APIs externas, se aplicável.

R: No meu projeto foi utilizado a biblioteca requests para realizar requisições HTTP e à APIs externas e Pandas para leitura e manipulação de dados, poderia ter sido incluída também a utilização da biblioteca SQLALchemy para inserção, alteração, leitura, exclusão e consultas em banco de dados, como neste projeto tinha um arquivo não houve necessidade. Mas é uma boa prática de desenvolvimento o utilizar para armazenar estados e dados além de somente no arquivo utilizado.

**4 - Testes e escalabilidade:**

* Verifique se o seu script funciona corretamente para o fluxo de pedidos descrito.

R: Sim, foram realizados testes funcionais de forma unitária e de ponta a ponta.

* Pense em como o script poderia ser escalável para lidar com um volume maior de pedidos ou processos adicionais.

R: O script está preparado para escalar e lidar com um volume maior de pedidos graças à implementação da máquina de estados. Essa abordagem permite o processamento eficiente de quantos itens forem necessários, facilitando a manipulação de dados e a execução de tratativas de erro. Além disso, a estrutura inspirada no ReFramework garante robustez e segurança, tornando o sistema flexível para adicionar novos processos ou funcionalidades conforme a demanda aumenta.

* Considere a possibilidade de implementar técnicas como paralelismo, otimização de consultas de banco de dados ou integração com outras ferramentas para otimizar a automação do processo.

R: O script está preparado para paralelismo, permitindo o processamento simultâneo de múltiplos pedidos através da máquina de estados, que gerencia as transações em diferentes threads. A integração com Pandas facilita a leitura e escrita de arquivos CSV, permitindo manipulações rápidas e eficientes de grandes volumes de dados. A biblioteca Requests é utilizada para baixar automaticamente arquivos de pedidos via HTTP, eliminando a necessidade de intervenção manual. A manipulação de diretórios e arquivos é gerenciada por uma classe dedicada, garantindo que os dados processados sejam organizados de forma estruturada. Essas técnicas otimizam o fluxo de trabalho, tornando o sistema escalável e robusto para lidar com um aumento na demanda.

**Requisitos:**

* Utilize a linguagem Python.
* Utilize bibliotecas e ferramentas adequadas para a automação do processos.
* Certifique-se de que o código seja organizado, legível e bem comentado.
* Inclua instruções claras de como executar o programa e qualquer dependência necessária.

**Observações:**

* Este é um cenário fictício, portanto, não é necessário realizar interações reais com sistemas ou APIs de lojas online.
* Você pode simular as interações e manipulação de dados utilizando estruturas de dados em memória, arquivos locais ou um banco de dados local, se preferir.
* Não se esqueça de incluir o código-fonte e qualquer instrução ou documentação adicional quando enviar sua solução.

Ao finalizar o desafio, envie para [people@somosmesha.com](mailto:people@somosmesha.com).

Boa sorte! Estou à disposição para qualquer dúvida.