

Formación Profesional: RPA con UiPath

Guía de Práctica – Tercer Ejercicio

Tema: Procesamiento Transaccional Robusto y Manejo de Interfaz Dinámica

1. Objetivo Técnico

El objetivo de esta práctica es implementar un robot de alto rendimiento que simule el ingreso de un lote de datos a un sistema de negocios. El desafío principal es construir un proceso **robusto y escalable** que pueda:

1. Leer datos transaccionales de un archivo Excel.
2. Procesar cada registro como una transacción individual.
3. Superar los desafíos de una interfaz de usuario dinámica (campos que cambian de posición).
4. Implementar manejo de excepciones para garantizar que el robot complete el lote de 10 transacciones sin fallar.

2. Arquitectura del Proceso y Secuencia Lógica

El flujo de trabajo debe seguir una arquitectura secuencial-transaccional, donde cada fila del Excel representa una única transacción que debe ser tratada dentro de un bucle de control.

1. **Preparación de Activos:** Descargar y obtener la ruta del archivo challenge.xlsx.
2. **Lectura de Datos:** Leer el archivo Excel en una DataTable en memoria.
3. **Control Transaccional:** Utilizar un bucle For Each Row para iterar a través de cada registro.
4. **Automatización Robusta de UI:** Dentro del bucle, ingresar los 7 campos de datos del registro en el formulario web. **La solución debe emplear la técnica de Anclas (Anchors) para localizar los campos**, independientemente de su posición en la pantalla.
5. **Ejecución y Repetición:** Hacer clic en "Submit" y repetir el proceso hasta completar los 10 registros.

3. Activos de Entrada

- **Aplicación Objetivo: RPA Challenge** (<https://www.rpachallenge.com/>)
- **Archivo de Datos:** challenge.xlsx (disponible para descarga en el sitio web).

4. Implementación Paso a Paso

Paso 4.1: Preparación y Lectura de Datos

1. **Crear el Proceso:** Cree un nuevo proceso llamado EJ3-Procesador de Datos.

2. **Lectura de Excel:** Arrastre la actividad **Read Range Workbook**.
 - o Workbook path: Especifique la ruta del archivo challenge.xlsx.
 - o Sheet name: "Sheet1"
 - o Guarde la salida en una DataTable llamada dt_ChallengeData.

Paso 4.2: Inicio y Bucle Transaccional

1. **Inicio de la Sesión:** Arrastre la actividad **Use Application/Browser** y apunte a la URL objetivo.
2. **Preparación de la Interfaz:** Dentro del contenedor, implemente una actividad **Click** para hacer clic en el botón **START** antes de comenzar el bucle.
3. **Bucle Transaccional:** Arrastre el bucle **For Each Row in Data Table** y configúrelo para iterar sobre dt_ChallengeData. El ítem de la iteración se llamará filaTransaccion.

Paso 4.3: Automatización de UI Robusta (Técnica de Anclas)

Para resolver el problema de la interfaz dinámica, la automatización de cada campo debe utilizar la técnica de Anclas.

1. Dentro del bucle, para el campo **First Name (Nombre):**
 - o Arrastre una actividad **Type Into**.
 - o En lugar de solo apuntar al campo de texto, utilice la opción **Ancla (Anchor)**.
 - o **Ancla (Referencia):** Seleccione la etiqueta de texto "**First Name**".
 - o **Destino (Target):** Seleccione el campo de texto de entrada vacío junto a él.
 - o **Value:** Ingrese la expresión para obtener el dato de la tabla:
filaTransaccion("First Name").ToString
2. **Generalización:** Repita el proceso para los **siete (7) campos restantes** del formulario (Last Name, Company Name, etc.). Para cada Type Into, su respectiva etiqueta de texto debe ser configurada como el Ancla.

Paso 4.4: Finalización de la Transacción

1. Despues de ingresar los 7 campos de datos, arrastre una actividad **Click** y apúntela al botón **SUBMIT**.
2. **Consideración:** El Click debe ser robusto y podría requerir la misma técnica de Ancla que los Type Into, si la posición del botón también varía.

5. Desafío de Ingeniería: Implementación de Control de Errores

Un robot de producción no puede fallar solo porque un dato está mal. El robot debe ser capaz de manejar un fallo y continuar con el siguiente registro.

Requisito de Robustez (Paso Opcional, pero muy recomendado):

1. Modifique la arquitectura del proceso para incorporar un bloque **Try Catch** dentro del bucle **For Each Row**.
2. La lógica de ingreso de datos y el Click del formulario deben residir en el bloque **Try**.
3. El bloque **Catch** debe quedar vacío (o simplemente utilizar un Log Message para registrar el error), asegurando que si la transacción actual falla por cualquier motivo (ej. un selector no se encuentra), el bucle simplemente continúe a la siguiente filaTransaccion sin que el flujo de trabajo principal se interrumpa.