

Informe de Proyecto de Evaluación 1: Procesamiento de Reportes Médicos con Arquitectura Dispatcher/Performer

Asignatura: Formación Profesional: RPA con UiPath

Profesor/a: Leximar Alcala

Fecha de Publicación: 10 de noviembre de 2025

Fecha Límite de Entrega: Viernes, 14 de noviembre de 2025, al final del día.

Formato de Entrega: Archivo .zip conteniendo el código fuente del proyecto único que implementa ambas fases.

1. Introducción y Objetivo del Proyecto

El objetivo de esta evaluación es implementar un proceso de RPA que aplique el patrón de diseño **Dispatcher/Performer** de manera modular dentro de un único proyecto. Los alumnos construirán una solución que se ejecuta en dos fases lógicas secuenciales:

1. **Fase Dispatcher:** Leerá un archivo de reportes de tratamiento y cargará masivamente los datos en una cola de UiPath Orchestrator.
2. **Fase Performer:** Recuperará cada transacción de la cola y la procesará en una aplicación web empresarial.

Este proyecto evaluará la habilidad para modularizar la lógica de negocio, construir un robot transaccional robusto y gestionar el flujo de trabajo a través de Orchestrator.

2. Prerrequisitos y Configuración Obligatoria

Antes de comenzar, cada alumno debe realizar la siguiente configuración:

1. **Registro Manual en ACME:** Cada alumno debe crear una cuenta de usuario personal y gratuita.
 - **Acción:** Navegar a la URL de registro: <https://acme-test.uipath.com/register>.
 - **Proceso:** Completar el formulario con su correo electrónico y una contraseña. Este paso es manual y es indispensable para que el robot pueda ejecutar el inicio de sesión posteriormente.
2. **Creación de la Cola en Orchestrator:** Dentro de la carpeta (Folder) "**RPA_Challenge**" que ya utilizamos en clase, deben crear una nueva cola con el siguiente nombre exacto: **TreatmentReportsQueue**.

3. Servicios y Aplicaciones a Utilizar

- **Aplicación Web:** ACME System 1 (<https://acme-test.uipath.com/login>).
- **URL Directa de Procesamiento:** <https://acme-test.uipath.com/medical/insurance>.

- **Fuente de Datos:** Un archivo CSV (Treatment_Report.csv) que será proporcionado por el profesor.
- **Orquestación:** UiPath Orchestrator.

4. Requisitos de Implementación y Conceptos a Aplicar

El proyecto consiste en la creación de un **único proceso de automatización** que contenga la lógica para ambas fases, separadas en flujos de trabajo (.xaml) distintos para demostrar modularidad.

Este flujo de trabajo es responsable de leer los datos y cargarlos eficientemente en la cola.

- **1.1. Lectura y Carga Masiva de Datos:**
 - El flujo de trabajo debe leer todos los registros del archivo Treatment_Report.csv y almacenarlos en una variable DataTable.
 - Inmediatamente después, utilizar la actividad Bulk Add Queue Items para cargar la DataTable completa directamente en la TreatmentReportsQueue.

Este flujo de trabajo procesará cada transacción recuperada de la cola.

- **2.1. Arquitectura Transaccional:** El flujo debe contener un bucle que procese los ítems de la cola hasta que no queden más. La estructura interna del bucle debe ser:
 - Utilizar la actividad Get Transaction Item para obtener un ítem de la TreatmentReportsQueue.
 - La lógica principal de procesamiento debe estar encapsulada en un bloque **Try Catch**.
 - Utilizar la actividad Set Transaction Status. En caso de éxito, el estado será Successful. Si ocurre un error en el bloque Catch, el estado debe ser Failed, el tipo de excepción Application, y se debe proporcionar un mensaje representativo en el campo Reason (ej. ex.Message).
- **2.2. Lógica de la Transacción:**
 - El robot debe iniciar sesión en <https://acme-test.uipath.com/login>.
 - **Navegación Directa:** Una vez iniciada la sesión, debe navegar directamente a la URL: <https://acme-test.uipath.com/medical/insurance>.
 - Extraer los datos del ítem de la cola (ej: transactionItem.SpecificContent("first_name").ToString).
 - Ingresar los datos en el formulario "Register Patient Insurance".
- **2.3. Robustez de UI (Aplicación de concepto de Clase 3):**

- **Requisito Crítico:** Se debe utilizar **Anclas (Anchors)** para todas las actividades Type Into. Cada campo de texto debe ser anclado a su etiqueta correspondiente.
- **Campos Opcionales (Bonus):** La automatización de los selectores desplegables (Gender, Care type) no es obligatoria y se pueden dejar con sus valores por defecto. Se anima a los alumnos que deseen un desafío adicional a intentar parametrizar la selección de estos campos, aunque no afectará a la calificación base.

Flujo Principal (Main.xaml):

El flujo de trabajo principal debe orquestar el proceso:

1. Invocar el flujo de trabajo Upload_Queue.xaml para poblar la cola.
2. Una vez completado, invocar el flujo de trabajo Process_Transactions.xaml para procesar todos los ítems de la cola.

5. Criterios de Evaluación

- **Modularidad y Arquitectura (40%):** Correcta separación de las lógicas en flujos de trabajo distintos (Upload_Queue.xaml, Process_Transactions.xaml) y correcta orquestación desde el Main.xaml.
- **Integración con Orchestrator (20%):** Uso adecuado de la actividad Bulk Add Queue Items en el Dispatcher y del ciclo Get Transaction Item / Set Transaction Status en el Performer.
- **Robustez de Selectores (30%):** Uso correcto y consistente de Anclas en el Performer.
- **Calidad General del Código (10%):** Claridad, organización y uso de nombres descriptivos.