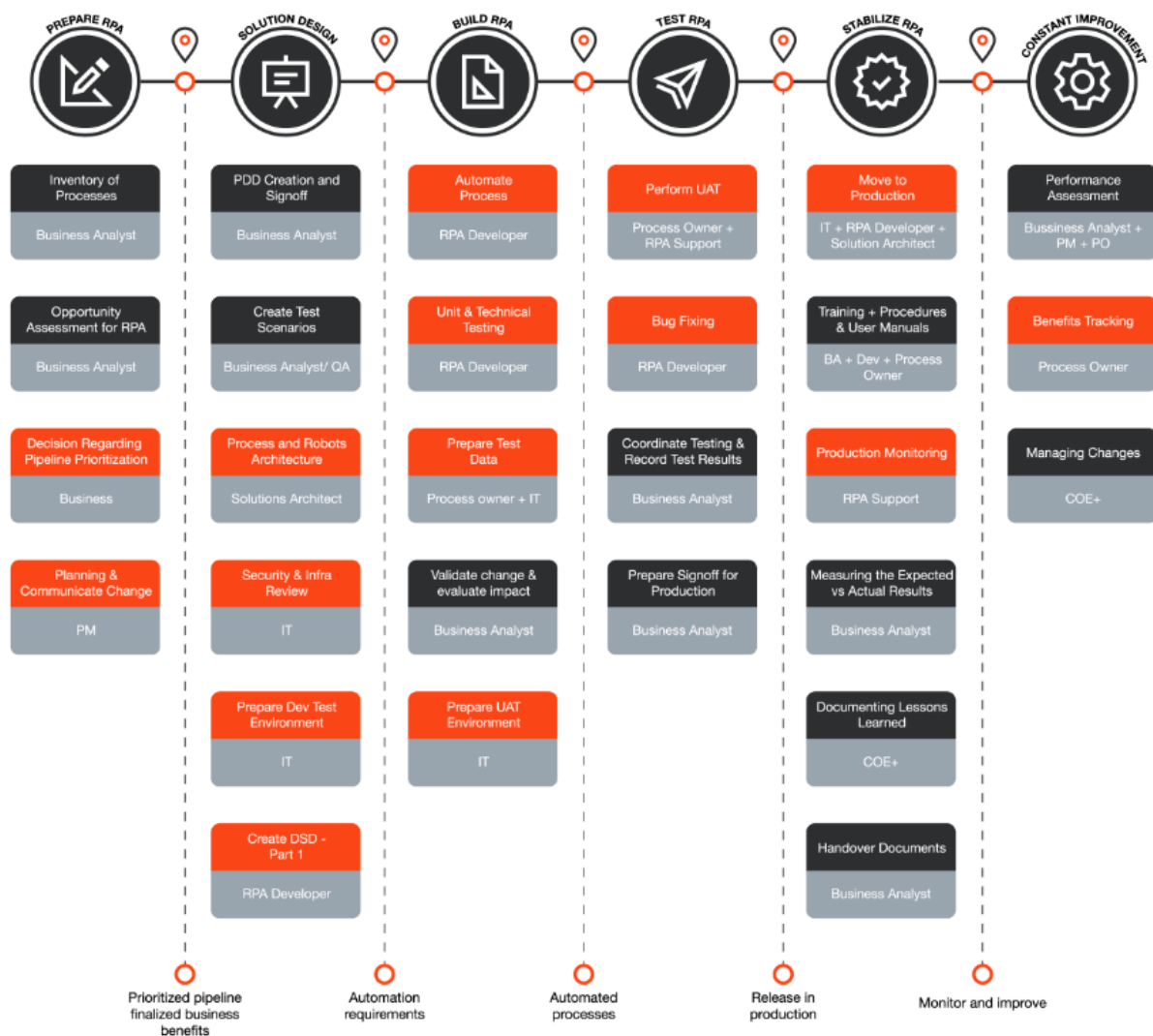


INTRODUCCIÓN AL ROL DE DESARROLLADOR DE RPA

Muy a menudo, encontraremos seis etapas en las implementaciones de RPA:

1. **Preparar RPA:** los procesos se definen, evalúan y priorizan y se planifica la implementación.
2. **Diseño de soluciones:** cada proceso que se va a automatizar se documenta ("as is" y "to be"), se crea y revisa la arquitectura, se preparan los escenarios y los entornos de prueba y se crea y se documenta el diseño de la solución para cada proceso.
3. **Crear RPA:** los procesos se automatizan, se prueba y valida el flujo de trabajo y se prepara la UAT.
4. **Probar RPA:** se realiza la UAT, se depura el flujo de trabajo y se cierra la sesión del proceso.
5. **Estabilizar RPA:** se prepara la puesta en funcionamiento, el proceso se traslada a producción, se supervisa, se mide y las lecciones aprendidas se documentan.
6. **Mejora constante:** se evalúa el rendimiento de la automatización del proceso, se realiza un seguimiento de los beneficios y se gestionan los cambios.



Arquitecto de soluciones: se encarga de definir la arquitectura de la solución de RPA. El arquitecto de soluciones traduce los requisitos capturados por los analistas funcionales, creando la arquitectura y los artefactos de diseño. Lideran, asesoran y son responsables de la entrega del equipo de los desarrolladores.

Analista empresarial: responsable de asignar los procesos "AS IS" y "TO BE". Los analistas de negocios poseen conocimiento del proceso de negocio que se automatiza, la teoría general de procesos de negocio y las

capacidades de RPA. Son responsables de enumerar los requisitos del proceso para la automatización, aclarar las entradas y las salidas esperadas, crear documentación de RPA (documentos de diseño de procesos y mapas de procesos).

Gestor de Implementación / Gestor de proyectos: forma y gestiona el equipo de RPA, realiza la planificación de recursos y la disponibilidad de los equipos, con el fin de alcanzar los objetivos de automatización. La mayoría de las veces, el PM es el punto único de contacto (SPOC) para preguntas, iniciativas de RPA o proyectos de productos de RPA paralelos.

Desarrolladores de RPA: en proyectos complejos, varios desarrolladores colaborarán para automatizar todos los procesos.

Administrador de infraestructura y seguridad de TI: con buenas habilidades técnicas y de seguridad, son responsables de configurar y mantener los recursos de hardware y software para las instalaciones de productos UiPath. Configuran cuentas para todos los desarrolladores, usuarios finales y robots.

Propietario del proceso: se trata de la parte interesada clave y los beneficiarios de la solución RPA. Por lo general, un alto directivo, con más de 10 o 15 años de experiencia, posiblemente dividido entre dominios. Varias personas pueden tener este rol, según el departamento (Finanzas, TI, RR. HH., etc.).

Soporte de RPA: gestiona los robots después de que los procesos se trasladen a la producción, con el apoyo de los desarrolladores originales de RPA que han realizado la automatización.

CREACIÓN DE VARIABLES

Hay 3 formas de crear variables en UiPath:

Desde el panel Variables: abra el panel Variables, seleccione la opción 'Crear nueva variable' y rellene los campos según sea necesario. Cuando lo necesite, proporcione su nombre en el panel Diseñador o en el campo Propiedades deseado.

Desde el panel Diseñador: arrastre una actividad con un campo de variable visible (es decir, 'Asignar') y presione Ctrl+K. Póngale un nombre y, a continuación, compruebe sus propiedades en el panel Variables.

En el panel Propiedades: en el panel Propiedades de la actividad, coloque el cursor en el campo en el que se necesita la variable (es decir, Salida) y presione Ctrl+K. Póngale un nombre y, a continuación, compruebe sus propiedades en el panel Variables

MANIPULACIÓN DE DATOS

1. Cadenas (strings)
2. Listas
3. Diccionarios
4. Expresiones regulares (Regex)

STRINGS

¿Qué son?

Las cadenas son el tipo de datos correspondiente al texto. Es difícil imaginar un escenario de automatización que no implique el uso de cadenas.

Cada vez que un texto debe capturarse, procesarse, enviarse entre aplicaciones o mostrarse, la cadena es útil (a menos que los datos estén estructurados, como una tabla).

¿Cuáles son algunos escenarios empresariales en los que probablemente encontrará diccionarios?

- Obtener el estado de una operación
- Extraer la pieza relevante de una porción de texto más grande
- Mostrar información al usuario humano

Metodos de cadena - Strings Methods:

All String Methods:

<https://docs.microsoft.com/en-us/dotnet/api/system.string?view=netframework-4.8#methods>

- **Directory.GetFiles** -> coge los archivos de un directorio
- **fileName.Replace**(".pdf", now.ToShortDateString) -> cambiar el nombre añadiendo la fecha de hoy
- **FileInfo Clase** -> Proporciona propiedades y métodos de instancia para crear, copiar, eliminar, mover y abrir archivos y contribuye a la creación de objetos FileStream. Esta clase no puede heredarse.
- **Concatena (Concat):**
 - Concatena las representaciones de cadena de dos objetos especificados
 - Expresión: String.Concat (VarName1, VarName2)
- **Contiene (Contains):**
 - Comprueba si una subcadena especificada ocurre dentro de una cadena. Devuelve verdadero o falso
 - Expresión: VarName.Contains ("texto")
- **Formato (Format):**
 - Convierte el valor de los objetos en cadenas (y las inserta en otro texto)
 - Expresión: String.Format ("{0} es {1}", VarName1, VarName2)
- **Índice de (IndexOf):**
 - Devuelve el índice de base cero de la primera aparición de un carácter en una cadena
 - Expresión: VarName1.IndexOf ("a")
- **Entrar (Join):**
 - Concatena los elementos de una colección y los muestra como String
 - Expresión: String.Join ("|", CollVarName1)
- **Reemplazar (Replace):**
 - Reemplaza todas las ocurrencias de una subcadena en una cadena
 - Expresión: VarName.Replace ("original", "reemplazado")
- **Separar (Split):**
 - Divide una cadena en subcadenas usando un separador dado
 - Expresión: VarName.Split ("|" c) (índice)
- **Subcadena (Substring):**
 - Extrae una subcadena de una cadena usando el índice inicial y la longitud
 - Expresión: VarName1.Substring (startIndex, longitud)

LISTAS

¿Qué son?

Las listas (o List <T>, como las encontrarás) son estructuras de datos que consisten en objetos del mismo tipo de datos (por ejemplo, cadena o entero). Cada objeto tiene una posición fija en la lista; por lo tanto, se puede acceder a él por índice. Si bien las matrices son estructuras de tamaño fijo para almacenar múltiples objetos, las listas nos permiten agregar, insertar y eliminar elementos.

Las listas pueden almacenar una gran cantidad de elementos: nombres, números, coordenadas de tiempo y muchos otros. Las listas proporcionan métodos específicos de manipulación, como:

- Agregar y quitar elementos
- Buscando un elemento
- Recorrer los elementos (y realizar determinadas acciones en cada uno)
- Clasificando los objetos
- Extraer elementos y convertirlos a otros tipos de datos.

¿Cuáles son algunos de los escenarios comerciales en los que probablemente encontrará listas?

- Almacenar los nombres de computadora de los miembros de un equipo de proyecto para una determinada configuración que debe realizarse
- Recopilar y almacenar la cantidad de facturas que cumplen con ciertos criterios.
- Realizar un seguimiento de los números de ticket creados en un período determinado sobre un tema determinado;

Métodos UiPath aplicables a colecciones

La manipulación de listas se puede realizar utilizando métodos .NET o utilizando los métodos de recopilación que ofrece UiPath Studio:

Añadir a la colección

- Agrega un elemento a una colección específica. Es equivalente a **List.Add ()**. Se puede utilizar, por ejemplo, para agregar un nuevo nombre a una lista de nombres de empresas.

Quitar de la colección

- Elimina un elemento de una colección específica y puede generar una variable booleana que confirma el éxito de la operación de eliminación. Esta actividad se puede utilizar, por ejemplo, para eliminar un número de factura de una lista de facturas a procesar.

Existe en la colección

- Indica si un elemento determinado está presente en una colección determinada generando un booleano como resultado. Podemos utilizar esta actividad para comprobar si una lista de clientes contiene un nombre específico.

Limpiar colección

- Borra una colección específica de todos los elementos. Un uso posible es vaciar una colección antes de comenzar una nueva fase de un proceso que la poblará nuevamente.

DICCIONARIOS

¿Qué son?

Los diccionarios (o Dictionary <TKey, TValue>, como los encontrará) son colecciones de pares (clave, valor), en las que las claves son únicas. Piense en la libreta de direcciones de su teléfono móvil, donde cada nombre tiene los datos correspondientes (número (s) de teléfono, correo electrónico).

Los tipos de datos para claves y valores deben elegirse cuando se crea una instancia de la variable. Los tipos de datos en los diccionarios pueden ser cualquiera de las variables admitidas (incluidos los diccionarios, por ejemplo).

Las operaciones que se asocian con mayor frecuencia con los diccionarios son:

- Agregar y eliminar pares (clave, valor)
- Recuperar el valor asociado con una clave
- Reasignación de nuevos valores a claves existentes

¿Cuáles son algunos escenarios empresariales en los que probablemente encontrará diccionarios?

- Almacenar detalles de configuración u otra información a la que se debe acceder a lo largo de un proceso
- Almacenar los títulos de trabajo u otra información relevante de los empleados.
- Almacenar las cuentas bancarias de los proveedores

Métodos para trabajar con diccionarios

Inicialización

- Al igual que en el ejemplo de Listas, los diccionarios deben inicializarse. En el siguiente ejemplo, la inicialización se realiza dentro de una actividad "Asignar". Sin embargo, como recordará del capítulo Listas, puede hacerlo desde el Panel de variables.

Añadiendo

- VarName.Add (clave, valor): agrega un elemento a un diccionario existente. Dado que Agregar no devuelve un valor, utilice la actividad Invocar código.

Eliminando

- VarName.Remove (Key): elimina un elemento del Diccionario. Se puede utilizar en una actividad de "Asignar".

Recuperando

- VarName.Item (Key): devuelve el elemento del diccionario por su clave
- VarName.Count: devuelve un valor Int32 del número de elementos del diccionario
- VarName.ContainsKey (Key): comprueba si el elemento con la clave dada existe en el diccionario y devuelve un resultado booleano
- VarName.TryGetValue (clave, valor): comprueba si un elemento con una clave determinada existe en el diccionario y devuelve un resultado booleano y el valor si se encuentra

GENERADOR DE EXPRESIONES REGULARES

¿Qué es?

La expresión regular (REGEX o regexp) es un patrón de búsqueda específico que se puede usar para hacer coincidir, ubicar y administrar texto fácilmente. Sin embargo, crear expresiones RegEx puede resultar complicado.

UiPath Studio contiene un constructor de RegEx que simplifica la creación de expresiones regulares.

Los usos típicos de RegEx incluyen:

- Validación de entrada
- Análisis de cadenas
- Raspado de datos
- Manipulación de cuerdas

¿Cuáles son algunos escenarios comerciales en los que usará RegEx?

Recuperar fragmentos de texto que siguen un patrón determinado, por ejemplo:

- extraer números de teléfono que comienzan con un determinado dígito;
- recopilar todos los nombres de las calles de un texto masivo, incluso si no siguen un patrón específico; algunos de ellos contienen "Calle", otros "Rd.", etc.

Se necesitaría mucho más tiempo para construir la misma expresión usando los métodos de cadena regulares; por ejemplo, RegEx tiene una expresión predefinida para ubicar todas las URL en una cadena.

Métodos en UiPath que usan el constructor RegEx:

Matches

- Busca una cadena de entrada para todas las apariciones y devuelve todas las coincidencias correctas.

IsMatch

- Indica si la expresión regular especificada encuentra una coincidencia en la cadena de entrada especificada.

Replace

- Reemplaza cadenas que coinciden con un patrón de expresión regular con una cadena de reemplazo especificada.

Documentación – RegEx -> <https://docs.uipath.com/activities/docs/regex-builder-wizard>

EXCEL AND DATA TABLE

✓	≡ Cell	The hiring date of an employee
✓	≡ Column	The dates of birth of all employees
✓	≡ Row	All the data of one employee
✓	≡ Worksheet	All the data of all the employees of one company in the group
✓	≡ Workbook	All the data of all the employees of all the companies in the group

¿Qué son las tablas de datos?

DataTable es el tipo de variable que puede almacenar datos como una hoja de cálculo simple con filas y columnas, de modo que cada dato pueda identificarse en función de sus coordenadas únicas de columna y fila. Piense en ello como la representación de memoria de una hoja de cálculo de Excel.

En DataTables, se aplica la convención habitual de identificar las columnas y las filas: las columnas se identifican mediante letras mayúsculas y las filas mediante números.

¿Cuál es la diferencia entre una hoja de trabajo y una DataTable?

Un DataTable es una representación en memoria de una única tabla de base de datos que tiene una colección de filas y columnas.

Una Hoja de cálculo de Excel es una representación visual de datos con diferentes opciones de visualización y un amplio uso de la interfaz gráfica de usuario.

¿Cómo se crean las tablas de datos?

Las formas más comunes de crear tablas de datos son:

CONSTRUIR TABLA DE DATOS

- Al usar esta actividad, puede elegir el número de columnas y el tipo de datos de cada una de ellas. Además, puede configurar cada columna con opciones específicas como permitir valores nulos, valores únicos, incremento automático (para números), valor predeterminado y longitud (para cadenas).

LEER RANGO

- Esta actividad obtiene el contenido de una hoja de trabajo (o una selección de esa hoja de trabajo) y lo almacena en una variable DataTable, que se puede crear desde el panel Propiedades usando Ctrl + K.

LEER CSV

- Esta actividad captura el contenido de un archivo CSV y lo almacena en una variable DataTable. Aunque ya no se usan comúnmente, todavía existen aplicaciones heredadas o internas que funcionan con este tipo de documentos.

RASPADO DE DATOS

- Esta funcionalidad de UiPath Studio le permite extraer datos estructurados de su navegador, aplicación o documento a un DataTable.

ACTIVIDADES DE DATATABLE

UiPath ofrece una amplia gama de actividades que se pueden utilizar para trabajar con variables de DataTable:

1. Agregar columna de datos (Add Data Column)
 - a. Agrega una columna a una variable DataTable existente. Los datos de entrada pueden ser del tipo DataColumn o la columna se puede agregar vacía, especificando el tipo de dato y configurando las opciones (permitiendo valores nulos, solicitando valores únicos, auto-incremento, valor por defecto y longitud máxima).
2. Agregar fila de datos (Add Data Row)
 - a. Agrega una nueva fila a una variable DataTable existente. Los datos de entrada pueden ser de tipo DataRow o pueden ingresarse como Array Row, haciendo coincidir cada objeto con el tipo de datos de cada columna.
3. Crear tabla de datos (Build Data Table)
 - a. Se usa para crear un DataTable usando una ventana dedicada. Esta actividad permite la personalización del número de columnas y el tipo de datos para cada columna.
4. Borrar tabla de datos (Clear Data Table)
 - a. Borra todos los datos de una variable DataTable existente.
5. Filtrar tabla de datos (Filter Data Table)
 - a. Permite filtrar un DataTable a través de un Asistente de filtro, usando varias condiciones. Esta actividad se puede configurar para crear una nueva DataTable para la salida de la actividad, o para mantener la existente y filtrar (eliminar) las entradas que no coinciden con las condiciones de filtrado.
6. Para cada fila (For Each Row)
 - a. Se utiliza para realizar una determinada actividad para cada fila de un DataTable (similar a un bucle For Each).
7. Generar tabla de datos (Generate Data Table)

- a. Se puede utilizar para crear una tabla de datos a partir de datos no estructurados, al permitir que el usuario indique los separadores de filas y columnas.
- 8. Unir tablas de datos (Join Data Tables)
 - a. Combina filas de dos tablas utilizando valores comunes entre sí, de acuerdo con una regla de unión que responde a la pregunta "¿Qué hacer con los datos que no coinciden?". Es una de las actividades más útiles en escenarios empresariales, donde trabajar con más de una tabla de datos es muy común. Es por eso que lo cubriremos con más profundidad a continuación.
- 9. Tabla de datos de búsqueda (Lookup Data Table)
 - a. Es similar a vLookup en Excel, ya que permite buscar un valor proporcionado en un DataTable especificado y devuelve el RowIndex en el que se encontró, o puede configurarse para devolver el valor de una celda con coordenadas dadas (RowIndex y Target Column).
- 10. Combinar tabla de datos (Merge Data Table)
 - a. Se usa para agregar un DataTable especificado al DataTable actual. La operación es más simple que la actividad Unir tipos de datos, ya que tiene 4 acciones predefinidas para realizar sobre el esquema que falta.
- 11. Tabla de datos de salida (Output Data Table)
 - a. Escribe un DataTable en una cadena utilizando el formato CSV.
- 12. Eliminar columna de datos (Remove Data Column)
 - a. Quita una determinada columna de un DataTable especificado. La entrada puede consistir en el índice de la columna, el nombre de la columna o una variable de la columna de datos.
- 13. Eliminar fila de datos (Remove Data Row)
 - a. Elimina una fila de un DataTable especificado. La entrada puede consistir en el índice de fila o una variable de fila de datos.
- 14. Eliminar filas duplicadas (Remove Duplicate Rows)
 - a. Elimina las filas duplicadas de una variable DataTable especificada, manteniendo solo la primera aparición.
- 15. Ordenar tabla de datos (Sort Data Table)
 - a. Puede ordenar un DataTable de forma ascendente o descendente según los valores de una columna específica.

UNIR TABLAS DE DATOS

¿Cómo funciona?

1. Deben especificarse 3 variables de la tabla de datos: 2 tablas de datos de entrada y 1 tabla de datos de salida. Tenga en cuenta que el orden de los 2 primeros es muy importante, ya que hay una opción que mantiene los valores de la Tabla de datos 1 y no se puede cambiar.
2. Se debe elegir el tipo de unión; hay 3 opciones:
 - Interior: mantenga todas las filas de ambas tablas que cumplan con la regla de unión. Las filas que no cumplen con la regla se eliminan de la tabla resultante.
 - Izquierda: mantenga todas las filas de DataTable1 y solo los valores de DataTable2 que cumplan con la regla de unión. Los valores nulos se insertan en la columna para las filas de DataTable1 que no tienen una coincidencia en las filas de DataTable2.
 - Completo: mantenga todas las filas de DataTable1 y DataTable2, independientemente de si se cumple la condición de combinación. Los valores nulos se agregan a las filas de ambas tablas que no tienen una coincidencia.

3. Las reglas de unión deben configurarse (puede haber una o más reglas):

- Una columna de cada DataTable debe especificarse por sus nombres (String), por su índice (Int32) o por variables ExcelColumn
- Se debe elegir el operador: = (Igual a), != (No igual a), > (Mayor que), < (Menor que), >= (Mayor o igual que), <= (Menor o igual que a)

¿Cuáles son algunos escenarios comerciales en los que usaré tablas de datos de unión?

Unir tabla de datos proporciona una de las formas más fáciles de traer datos de dos fuentes en un solo lugar:

- Reuniendo 2 bases de datos de empleados extraídas de 2 aplicaciones
- Comprobación de cuáles de los clientes (base de datos 1) han sido contactados en una campaña de marketing (base de datos 2)
- Comprobación de cuáles de los proveedores de una empresa (base de datos interna) han solicitado ayudas públicas (base de datos pública)

LIBROS DE TRABAJO Y ACTIVIDADES COMUNES

En muchos escenarios comerciales, las bases de datos se almacenan en libros de trabajo (generalmente conocidos como archivos de Excel u hojas de cálculo). A partir de ahí, pueden ingresarse en DataTables y procesarse aún más utilizando los métodos presentados en el capítulo anterior, así como otros métodos y herramientas disponibles. Es hora de ver cómo RPA maneja los libros de trabajo.

UiPath ofrece 2 formas distintas de acceder y manipular libros de trabajo, cada una de ellas con ventajas y limitaciones:

NIVEL DE ACCESO AL ARCHIVO

- Todas las actividades del libro de trabajo se ejecutarán en segundo plano.
- (+) No requiere la instalación de Microsoft Excel, puede ser más rápido y confiable para algunas operaciones simplemente no abriendo el archivo;
- (!) Funciona solo para archivos .xlsx.

INTEGRACIÓN DE LA APLICACIÓN EXCEL

- UiPath abrirá Excel como lo haría un humano.
- (+) Funciona con .xls y .xlsm, y tiene algunas actividades específicas para trabajar con .csv. Todas las actividades se pueden configurar para que sean visibles para el usuario o se ejecuten en segundo plano;
- (!) Se debe instalar Microsoft Excel, incluso cuando la casilla 'Visible' no está marcada. Si el archivo no está abierto, se abrirá, guardará y cerrará para cada actividad.

Ambos niveles de acceso comparten algunas actividades, y la integración de aplicaciones de Excel tiene varias más. Tenga en cuenta que en UiPath hay dos actividades para cada método que se presenta a continuación: una en 'Integración de aplicaciones> Excel' y la otra en 'Sistema> Archivo> Libro de trabajo'.

Comencemos con las actividades comunes:

1. Agregar rango
 - a. Agrega la información de un DataTable al final de una hoja de cálculo de Excel especificada. Si la hoja no existe, la crea.

2. Obtener rango de tabla
 - a. Localiza y extrae el rango de una tabla de Excel de una hoja de cálculo especificada utilizando el nombre de la tabla como entrada.
3. Leer celda
 - a. Lee el contenido de una celda determinada y se almacena como String.
4. Leer fórmula de celda
 - a. Lee la fórmula de una celda determinada y la almacena como String.
5. Leer columna
 - a. Lee una columna que comienza con una celda ingresada por el usuario y la almacena como una variable IEnumerable <object>.
6. Rango de lectura
 - a. Lee un rango especificado y lo almacena en un DataTable. Si 'Usar filtro' está marcado en la actividad Rango de lectura en 'Alcance de la aplicación de Excel', solo leerá los datos filtrados. Esta opción no existe para la actividad de rango de lectura en 'Libro de trabajo'.
7. Leer fila
 - a. Lee una fila que comienza con una celda ingresada por el usuario y la almacena como una variable IEnumerable <object>.
8. Escribir celda
 - a. Escribe un valor en una celda especificada. Si la celda contiene datos, la actividad los sobrescribirá. Si la hoja especificada no existe, se creará.
9. Rango de escritura
 - a. Escribe los datos de una variable DataTable en una hoja de cálculo comenzando con la celda indicada en el campo StartingCell.

Documentación

Guía de actividades de UiPath - Actividades del libro de trabajo (nivel de acceso a archivos).

<https://docs.uipath.com/activities/docs/append-range>

Guía de actividades de UiPath - Actividades de Excel (integración de la aplicación de Excel)

<https://docs.uipath.com/activities/docs/append-csv-file>

ALCANCE DE LA APLICACIÓN DE EXCEL Y ACTIVIDADES ESPECÍFICAS

Alcance de la aplicación de Excel

La integración con Excel se habilita mediante el uso de una actividad de Ámbito de aplicación de Excel. De hecho, es un contenedor y todas las demás actividades utilizadas para trabajar con el archivo de Excel especificado deben colocarse dentro del contenedor. Básicamente, abre un libro de trabajo de Excel y proporciona un alcance para las actividades de Excel. Cuando finaliza la ejecución, el libro de trabajo especificado y la aplicación de Excel se cierran.

El ámbito de aplicación de Excel se puede configurar para escribir el resultado de las actividades en el contenedor en un archivo diferente.

Nota importante: si el mismo flujo de trabajo trata con información de dos o más archivos de Excel, se debe utilizar un ámbito de aplicación de Excel para cada archivo.

Actividades específicas de integración de aplicaciones de Excel

CSV

Estas actividades pueden leer y escribir en archivos CSV, utilizando variables DataTable. Aunque se encuentran en Integración de aplicaciones de Excel, funcionan incluso si no se colocan dentro de un contenedor de Alcance de aplicación de Excel.

- Agregar a CSV: agregue la información de un DataTable a un archivo CSV, creándolo si no existe. La actividad no sobrescribe los datos existentes.
- Leer CSV: lee todas las entradas de un archivo CSV y las almacena en un DataTable
- Escribir CSV: sobrescribe un CSV con la información de un DataTable

RANGO

Estas actividades pueden leer datos, insertar y eliminar filas y columnas, e incluso copiar / pegar rangos completos. Son similares a las actividades correspondientes en DataTable, pero funcionan directamente en el archivo de Excel.

- Eliminar columna: elimina una columna de un archivo de Excel según el nombre.
- Insertar columna: inserta una columna en blanco en un archivo de Excel, en una posición determinada.
- Insertar / Eliminar columnas: agrega columnas en blanco o elimina columnas existentes, según el tipo de cambio especificado.
- Insertar / Eliminar filas: agrega filas en blanco o elimina filas existentes, según el tipo de cambio especificado.
- Seleccionar rango: selecciona un rango específico en un archivo de Excel. En general, se empareja con otra actividad que realiza una cierta manipulación sobre los datos seleccionados.
- Obtener rango seleccionado: genera un rango dado como Cadena.
- Eliminar rango: elimina un rango especificado de un archivo de Excel.
- Rango de llenado automático: aplica una fórmula determinada sobre un rango determinado en un archivo de Excel.
- Copiar y pegar rango: copia y pega un rango completo (valores, fórmulas y formato) desde una hoja de origen a una hoja de destino.
- Rango de búsqueda: busca un valor en todas las celdas de un rango determinado.
- Eliminar rango duplicado: elimina todas las filas duplicadas de un rango determinado.

TABLA

Estas actividades crean, filtran y clasifican tablas directamente en archivos de Excel.

- Filtro de tabla: aplica un filtro a todos los valores de una columna en una tabla dentro de un archivo de Excel. Una vez que se guarda el archivo, solo se mostrarán las filas que cumplan con el filtro. Tenga en cuenta que esta actividad no elimina las filas que no cumplen con los criterios, sino que solo las oculta. Un buen uso de este método implica usar una actividad de rango de lectura después de esta, con la casilla 'Usar filtros' marcada. La salida será un DataTable que contendrá solo las entradas que cumplieron con los criterios dados.
- Ordenar tabla: ordena una tabla en un archivo de Excel según los valores de una columna determinada.
- Crear tabla: crea una tabla (con nombre) en un rango especificado en el panel Propiedades.

ARCHIVO

Estas actividades funcionan directamente con los archivos de Excel, ya sea guardándolos o cerrándolos.

- Cerrar libro de trabajo
- Guardar libro de trabajo

COLOR DE CELDA

Estas actividades pueden capturar y modificar el color de fondo de las celdas en archivos de Excel.

- Obtener color de celda: lee el color de fondo o una celda determinada en un archivo de Excel y lo almacena como salida de variable de color.
- Establecer color de rango: cambia el color de fondo de todas las celdas en un rango determinado. La entrada es una variable de color.

HOJA

Estas actividades pueden realizar varias acciones sobre las hojas de un archivo de Excel.

- Obtener hoja del libro de trabajo: lee el nombre de una hoja por su índice.
- Obtener hojas de libro de trabajo: extrae los nombres de las hojas y las almacena ordenadas por índice.
- Copiar hoja: copia una hoja en un archivo de Excel y la pega en el mismo archivo de Excel o en uno diferente especificado.

TABLA DINAMICA

Estas actividades facilitan el trabajo con tablas dinámicas en archivos de Excel.

- Actualizar tabla dinámica: actualiza una tabla dinámica en un archivo de Excel. Esto es útil cuando cambian los datos de origen de la tabla dinámica, ya que la actualización no es automática.
- Crear tabla dinámica: crea una tabla dinámica usando una hoja específica y parámetros dados.

MACRO

Estas actividades pueden ejecutar macros que ya estaban definidas en el archivo de Excel, o pueden invocar macros de otros archivos. Tenga en cuenta que estas actividades funcionan con archivos .xslm.

- Ejecutar macro
- Invocar VBA: macro desde otro archivo

Recursos de aprendizaje

Tablas de datos

Guía de UiPath Studio - Variables de DataTable

<https://docs.uipath.com/studio/docs/data-table-variables>

Guía de actividades de UiPath - Métodos de DataTable

<https://docs.uipath.com/activities/docs/add-data-column>

Actividades del libro de trabajo

Guía de actividades de UiPath - Actividades del libro de trabajo (nivel de acceso a archivos)

<https://docs.uipath.com/activities/docs/append-range>

<https://docs.uipath.com/activities/docs/append-csv-file>

Examen Teórico

¿Qué actividad puede utilizar si desea agregar datos a un documento .xlsx existente sin sobrescribir los datos existentes?

- Rango anexado de Excel
- Rango de anexo de libro de trabajo

¿Qué expresión usaría para crear una instancia de una variable de diccionario que empareje nombres (clave) con edades (valor)?

- Nuevo diccionario (de cadena, Int32)

¿Qué sucede si usa una actividad de rango de escritura e intenta escribir datos en un archivo .xlsx que no existe?

- Creará ese archivo por usted y escribirá los datos en él.

¿Qué actividad sería su primera opción para recorrer todas las filas de una tabla de datos?

- Por cada fila

Desde el punto de vista del potencial de automatización, un proceso en el que los cambios son frecuentes, el entorno del sistema es volátil y se requieren múltiples acciones manuales (incluso no digitales) pertenece al ...

- Sin categoría RPA

Este tipo de proyecto se recomienda para flujos de trabajo simples y lineales sin múltiples nodos de decisión.

- Secuencia

¿Se pueden ejecutar actividades relacionadas con Excel en una máquina que no tiene instalada la aplicación Excel?

- Sí, pero solo para archivos .xlsx

¿Qué actividad se puede utilizar para leer una hoja completa de un archivo de Excel?

- Rango de lectura

¿Qué actividades puede utilizar para crear una tabla de datos a partir de datos no estructurados?

- Generar tabla de datos

Ha incluido una actividad Ordenar tabla de datos en su flujo de trabajo. La variable DataTable se llama dt_Usernames. El campo Input DataTable se establece en la variable dt_Usernames y el campo Output DataTable también se establece en dt_Usernames.

La actividad ordenará los valores de la tabla de datos y:

- Almacenarlos en la misma variable

AUTOMATIZACIÓN DE LA INTERFAZ DE USUARIO

INTRODUCCION A LA AUTOMATIZACIÓN DE LA INTERFAZ DE USUARIO

Descripción general del módulo

Este módulo está destinado a presentar a los desarrolladores de RPA todos los conceptos básicos relacionados con la automatización de la interfaz de usuario.

Qué aprenderá en este módulo

Al final de este módulo, debería poder:

1. Explique qué es la automatización de la interfaz de usuario;
2. Explicar los elementos clave de la automatización de la interfaz de usuario con UiPath Studio;
3. Explicar la diferencia entre las experiencias de diseño moderno y clásico en Studio;
4. Habilite la experiencia de diseño moderno en Studio;
5. Establezca una experiencia de diseño predeterminada para todos los nuevos proyectos de automatización.

¿Qué es la automatización de la interfaz de usuario?

Interfaz de usuario o interfaz gráfica de usuario Automatización (Automatización de UI) se refiere al proceso de interactuar con elementos gráficos dentro de aplicaciones mediante la simulación de controles de mouse y teclado.

Así como los usuarios humanos usan aplicaciones a través de actividades como hacer clic, escribir o seleccionar, los desarrolladores de RPA pueden configurar UiPath Robots a través de UiPath Studio para realizar las mismas tareas de manera más rápida y sin errores.

A través de una profunda integración entre UiPath y todos los principales marcos de GUI, desde Win32 hasta WPF, HTML y Java, los desarrolladores de RPA pueden automatizar fácilmente todo tipo de aplicaciones de escritorio.

UiPath reconoce los objetos gráficos por sus propios atributos específicos de la plataforma y proporciona una API de automatización GUI unificada que funciona de la misma manera en todas partes.

¿Cómo debería ver la automatización de la interfaz de usuario?

Teniendo en cuenta que los robots realizarán acciones en las interfaces de usuario (UI) de la aplicación tal como lo harían los humanos, los desarrolladores de RPA deberán configurar cada acción, en una ruta lógica y secuencial.

Por ejemplo, consideremos que nuestro objetivo es automatizar una tarea simple que ingresa datos en la aplicación UI Demo, hace clic en el botón Aceptar y obtiene el número de transacción. Para hacer esto, debemos considerar el estado de la aplicación desde la que queremos comenzar, cómo un usuario humano ejecutaría los pasos y luego dividir la tarea en cada paso.

El camino humano

Estado de inicio: UIDemo está abierto en la pantalla Depósito.

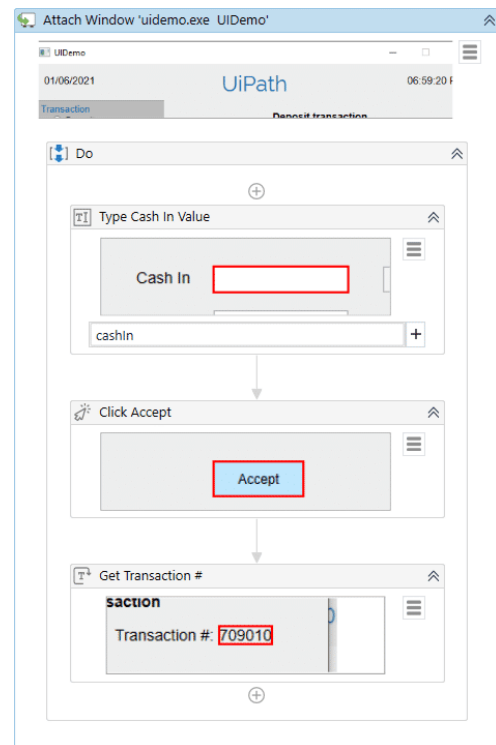
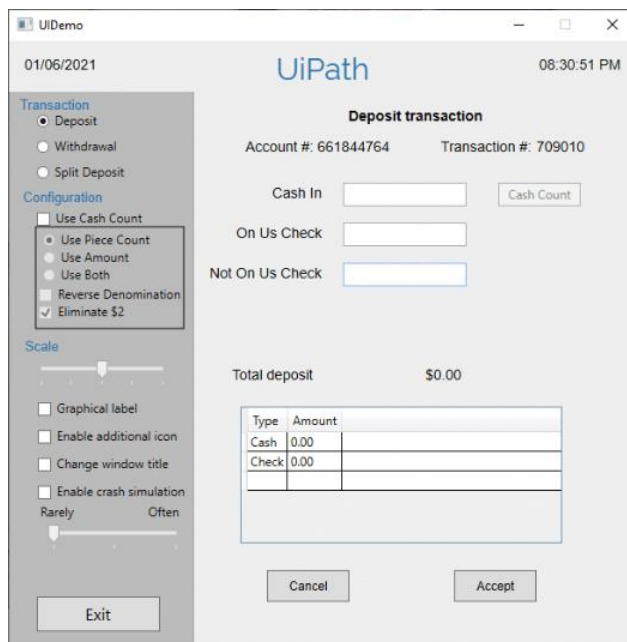
Pasos humanos:

1. **Seleccione** la aplicación de demostración de la interfaz de usuario;
2. **Escriba** el valor en el campo Efectivo;
3. Haga **clic** en Aceptar;
4. **Obtenga** el número de transacción.

Bastante simple, ¿verdad? Una vez que se identifican los pasos que realizaría un usuario humano, podemos traducirlos en actividades de UiPath que se pueden configurar para lograr el mismo resultado.

El camino del robot

Estado de inicio: UIDemo está abierto en la pantalla Depósito.



Pasos del robot:

1. La actividad **Adjuntar ventana** identifica la aplicación de demostración de la interfaz de usuario dentro de la cual se deben realizar los pasos;
2. La actividad **Type Into** identifica el campo Cash In e ingresa el valor en la variable de cadena cashIn;
3. Una actividad de **clic** identifica el botón Aceptar y hace clic en él;
4. Una actividad **Obtener texto** identifica el valor de la transacción #, lo obtiene y lo almacena en una variable.

Independientemente de la complejidad de la aplicación o el proceso, UiPath Studio proporciona Actividades universales o especialmente diseñadas para que los desarrolladores de RPA puedan crear soluciones de automatización de UI sólidas y confiables para todas las aplicaciones de escritorio.

Recapitulemos

Principales conclusiones

- UI Automation es el proceso de automatizar las interfaces de usuario de la aplicación mediante la simulación de acciones de entrada y salida humanas a través de actividades específicas de UI;

- El primer paso para automatizar las IU es comprender la secuencia lógica de pasos que tomaría un usuario humano. El segundo paso consiste en traducir estos pasos en actividades de UiPath Studio y configurarlas;
- La complejidad de la interfaz de usuario o de la aplicación es irrelevante. Todas las aplicaciones de escritorio se pueden automatizar mediante el uso de actividades universales o adaptadas a la aplicación.

¿Cuáles de las siguientes afirmaciones sobre la automatización de la interfaz de usuario son verdaderas?

- La automatización de la interfaz de usuario se logra imitando las interacciones reales del usuario humano.

¿Como funciona la Automatización de usuario?

Al final de esta lección, debería poder **explicar** los **elementos clave de la automatización de la interfaz de usuario** con UiPath Studio;

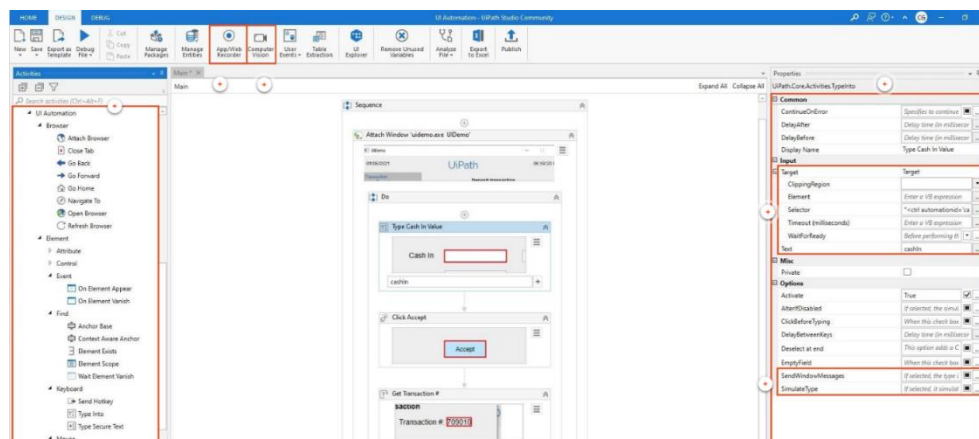
Automatización de interfaces de usuario con UiPath Studio

En la lección anterior, vimos un ejemplo de automatización de la interfaz de usuario muy simple. Es posible que se haya quedado con bastantes preguntas como: "¿Qué tipo de actividades puedo utilizar?", "¿Cómo sabe el robot a qué elementos apuntar?" o "¿Qué opciones tengo para controlar la ejecución de una actividad?". En esta lección, exploraremos los conceptos clave relacionados con la automatización de la interfaz de usuario utilizando UiPath Studio.

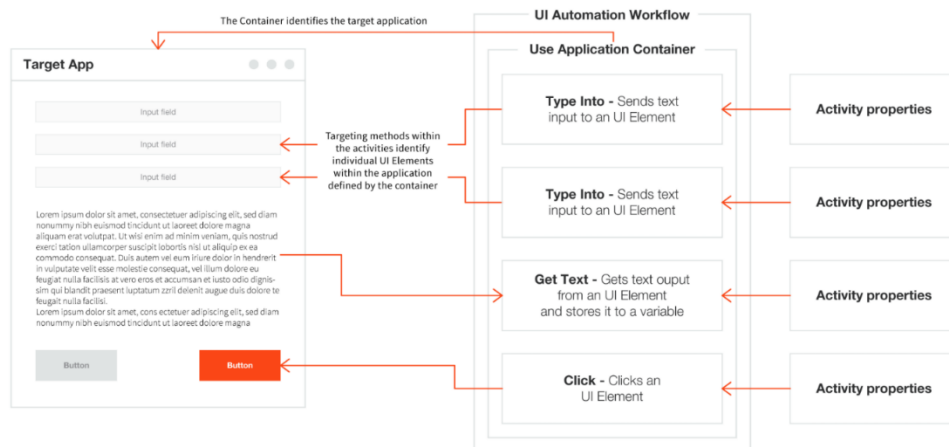
Al comenzar, hay siete conceptos centrales relacionados con la automatización de la interfaz de usuario que debe conocer. Ellos son:

- Actividades de automatización de la interfaz de usuario
- Propiedades
- Métodos de orientación
- Métodos de entrada y salida
- Grabadores y magos
- El repositorio de objetos
- Visión por computadora con IA

Consulte los marcadores a continuación para obtener más información sobre estos siete conceptos y ver dónde puede encontrarlos en Studio.



Ahora veamos este diagrama para comprender mejor cómo encaja todo. Aprenderemos sobre cada componente con más profundidad más adelante.

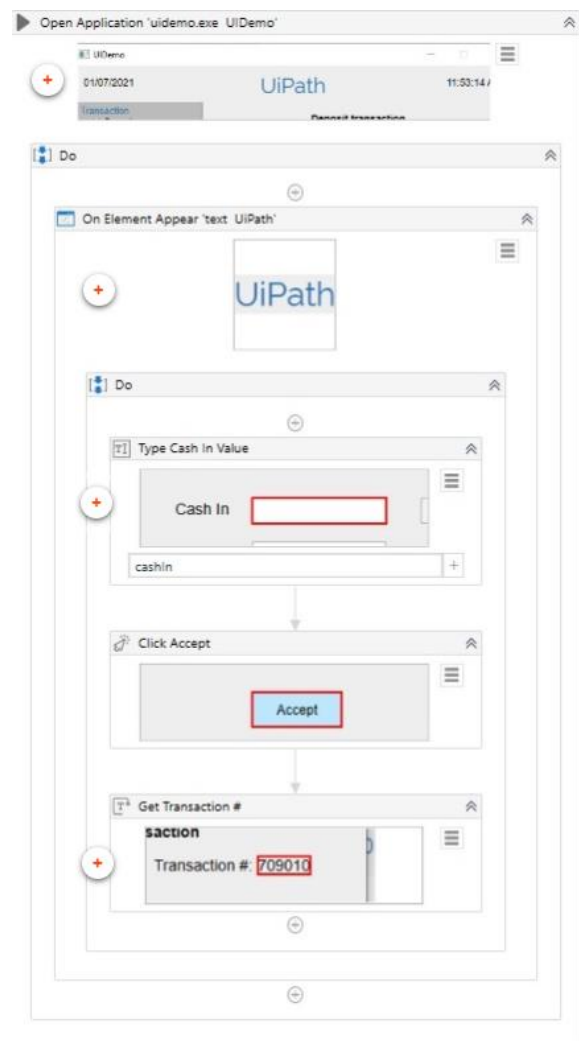


Actividades de automatización de la interfaz de usuario

Las actividades de automatización de la interfaz de usuario permiten a los desarrolladores escribir rápidamente instrucciones para que el robot interactúe con las interfaces de varias aplicaciones. Las actividades se pueden dividir en:

- **Contenedores:** son actividades que identifican los navegadores o aplicaciones con los que el proceso necesita interactuar. Todas las actividades incluidas dentro de un contenedor se ejecutarán en la misma aplicación. Algunos ejemplos son: Abrir navegador, Adjuntar navegador, Abrir aplicación o Usar aplicación / navegador.
- **Actividades de entrada:** estas actividades envían información a los elementos de la interfaz de usuario. Se pueden usar para hacer clic, verificar, escribir, enviar teclas de acceso rápido, etc.
- **Actividades de salida:** estas actividades obtienen información de los elementos de la GUI. Pueden indicarle al Robot que obtenga texto mediante el uso de varios métodos, obtener datos estructurados u obtener UIElements que contengan imágenes.
- **Actividades de sincronización:** lo ayudan a crear activadores basados en el comportamiento de la interfaz de usuario, lo que permite que el robot ejecute ciertas acciones cuando ocurren eventos específicos en una máquina.

Veamos un ejemplo sencillo a continuación. Haga clic en los marcadores para identificar cada tipo de actividad.



Propiedades de la actividad

Al igual que para otros tipos de actividades, las propiedades determinan cómo el robot realiza una acción. Se pueden encontrar en el panel Propiedades. Las actividades de UI Automation tienen conjuntos específicos de propiedades según el tipo de actividad. A continuación, se muestran algunos ejemplos de propiedades comunes:

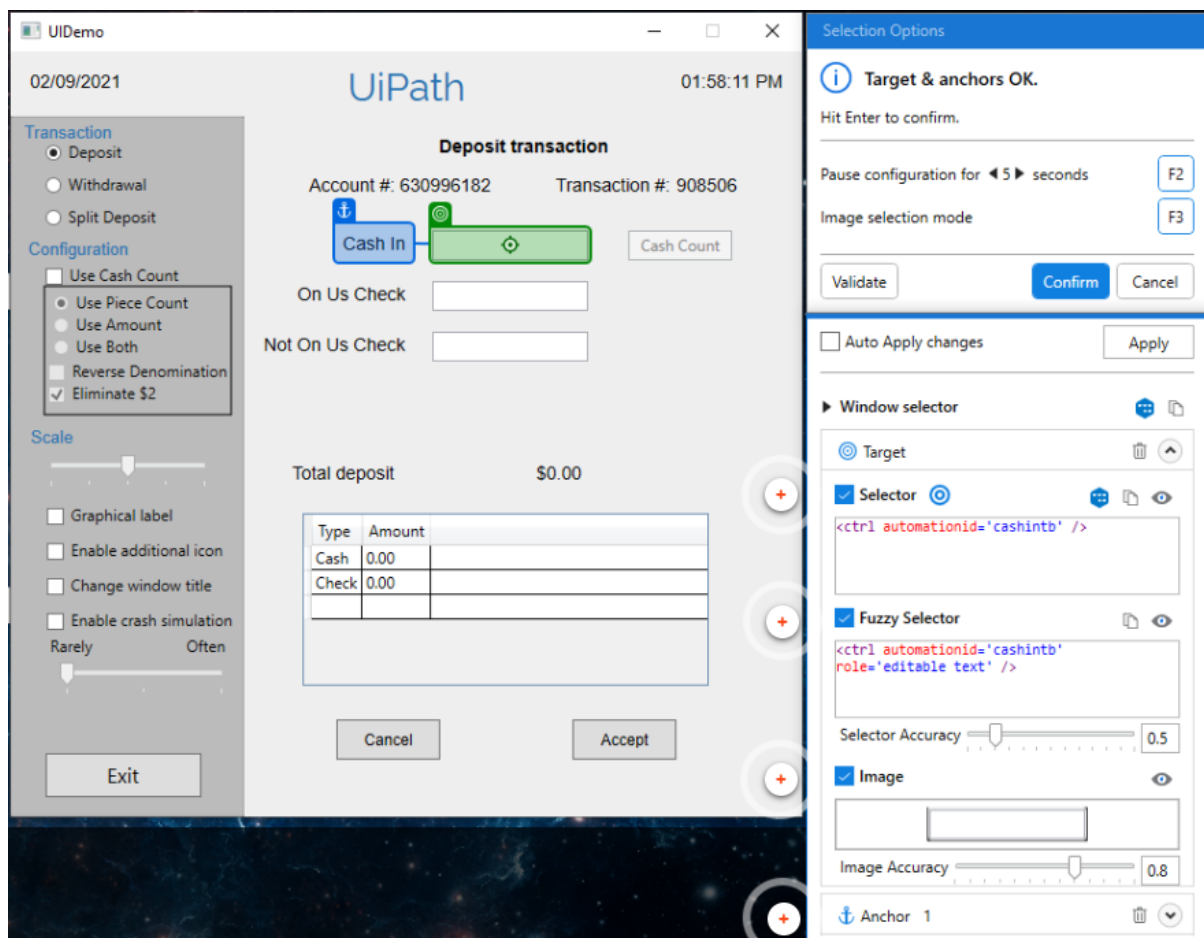
Propiedad	Que hace
DelayAfter / DelayBefore	Cuántos milisegundos espera el robot antes o después de ejecutar la actividad.
ContinueOnError	¿Se lanzará o ignorará una excepción encontrada al ejecutar la actividad. Por ejemplo, si no se encuentra el elemento, ¿el Robot arrojará un error y detendrá la ejecución o lo ignorará y continuará?)
Destino	Proporciona varias propiedades relacionadas con la identificación del elemento de la interfaz de usuario de destino.
Tiempo de espera (milisegundos)	¿Cuántos milisegundos intentará el robot realizar una acción en un elemento de la interfaz de usuario?
SendWindowsMessages / Simulate Type	¿Qué método de entrada usamos para las actividades de entrada?
Salida	¿Qué variable usamos para almacenar la salida de la actividad?

Métodos de orientación

Los métodos de orientación son un subconjunto de Propiedades. Proporcionan varias formas de identificar el elemento de la interfaz de usuario con el que interactuará el robot. Los métodos de focalización clave son:

- Selectores
- Selectores difusos
- Imagen
- Anclas

El método de orientación más utilizado es el **selector**. Haga clic en los marcadores a continuación para obtener más información sobre cada método de orientación:



Métodos de entrada y salida

Como ya hemos comentado, cada vez que insertamos datos en una aplicación, o enviamos un comando a un sistema para producir un cambio (o una continuación), realizamos una acción de entrada. Las acciones de salida se utilizan en UiPath para extraer datos (en general, como texto) de un elemento de la interfaz de usuario. Los métodos de salida son los que permiten que las acciones de salida extraigan datos de los elementos de la interfaz de usuario.

UiPath proporciona varios métodos para realizar acciones de entrada y salida. Estos métodos controlan varios aspectos importantes de nuestra automatización:

- ¿Qué tan rápido se realizará la acción?
- ¿Puede la acción ejecutarse en segundo plano o en una ventana oculta?
- ¿Interfiere con el usuario?
- ¿Es compatible con todas las aplicaciones?

Hay diferentes métodos disponibles para las actividades de entrada y salida.

Métodos de entrada

- **Eventos de hardware** - Clics: el cursor del mouse se mueve por la pantalla. Escritura: el controlador del teclado se utiliza para escribir caracteres individuales. No funciona en segundo plano. El usuario asistido no puede tocar el mouse o el teclado durante la automatización.
- **Enviar mensaje de ventana**: reproduce los mensajes de ventana que recibe la aplicación de destino cuando se usa el mouse / teclado. Funciona en segundo plano. Los usuarios pueden trabajar en otras actividades durante la ejecución de los procesos automatizados.
- **Simular**: utiliza la tecnología de la aplicación de destino (el nivel de API) para enviar instrucciones. Funciona en segundo plano. Los usuarios pueden trabajar en otras actividades durante la ejecución de los procesos automatizados.

Métodos de salida

- **Texto completo**: el método Texto completo es el método predeterminado y lo suficientemente bueno en la mayoría de los casos. Es el más rápido, puede extraer texto oculto, tiene una precisión del 100% y puede funcionar en segundo plano.
- **Nativo**: el método nativo es compatible con aplicaciones que utilizan Graphics Design Interface (GDI), la API de Microsoft utilizada para representar objetos gráficos. No extrae texto oculto y no puede funcionar en segundo plano; y al igual que FullText, no es compatible con entornos virtuales.
- **OCR** - OCR (o reconocimiento óptico de caracteres) es el único método de salida que funciona con entornos virtuales y con texto de "lectura" de imágenes. Su tecnología se basa en reconocer a cada personaje y su posición. Por otro lado, no puede funcionar en segundo plano, no puede extraer texto oculto y su velocidad es, con mucho, la más baja.

Grabadores

La grabación es una parte importante de UiPath Studio, que puede ayudarlo a ahorrar mucho tiempo al automatizar sus procesos comerciales. Esta funcionalidad le permite capturar fácilmente las acciones de un usuario en la pantalla y traducirlas en secuencias. Estos proyectos se pueden modificar y parametrizar para que pueda reproducirlos y reutilizarlos fácilmente en tantos otros procesos como necesite.

El repositorio de objetos

El repositorio de objetos garantiza la gestión, la reutilización y la fiabilidad de los elementos de la interfaz de usuario al capturarlos como objetos en un repositorio, que se pueden compartir entre proyectos. Permite crear y reutilizar taxonomías de IU dentro y entre proyectos de automatización. Esta es una función más avanzada de la que aprenderemos más adelante.

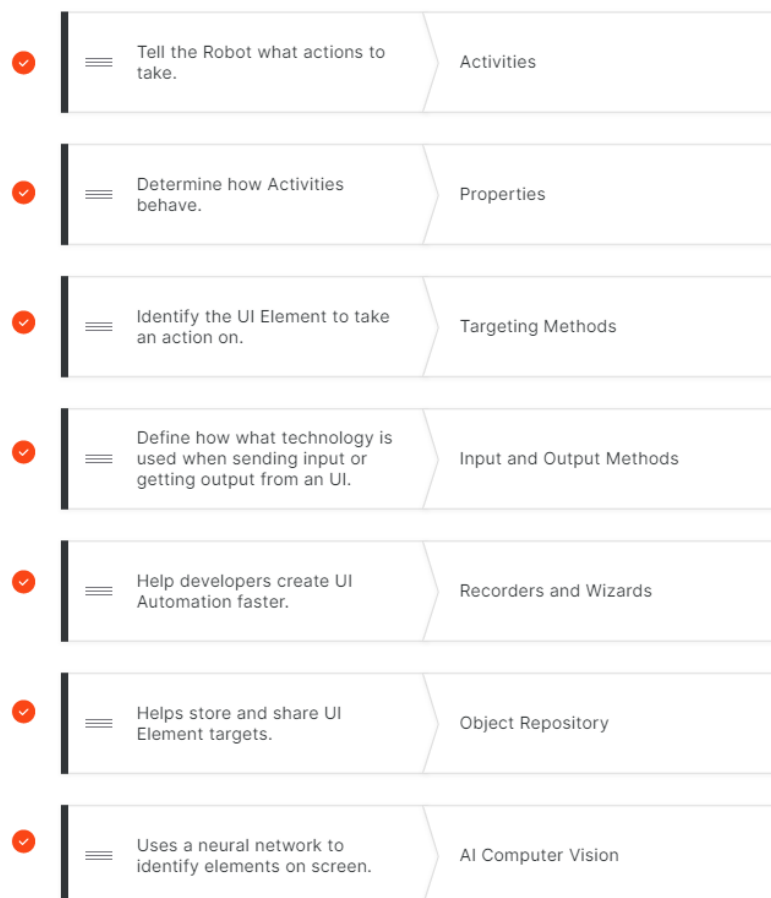
Visión por computadora con IA

El paquete AI Computer Vision contiene actividades fundamentales refactorizadas de UIAutomation como Click, Type Into o Get Text. La principal diferencia entre las actividades de CV y sus contrapartes es el uso de la red neuronal Computer Vision desarrollada internamente por nuestro departamento de Machine Learning. La red neuronal puede identificar elementos de la interfaz de usuario como botones, campos de entrada de texto o casillas de verificación sin el uso de selectores. Esta es una función más avanzada de la que aprenderemos más adelante.

RESUMIENDO

Principales conclusiones

- Los conceptos clave de la automatización de la interfaz de usuario en UiPath son: actividades de automatización de la interfaz de usuario, propiedades de la actividad, métodos de orientación, métodos de entrada y salida, registradores, el repositorio de objetos y visión por computadora.
- **Actividades de automatización de la interfaz de usuario:** las actividades pueden ser contenedores, actividades de entrada, actividades de salida o actividades de sincronización.
- **Propiedades de la actividad:** las propiedades determinan cómo el robot realiza una acción. Se pueden encontrar en el panel Propiedades.
- **Métodos de orientación:** proporcionan varias formas de identificar el elemento de la interfaz de usuario con el que interactuará el robot. Con mucho, los más comunes son los selectores.
- **Métodos de entrada y salida:** defina el tipo de tecnología utilizada para interactuar con los elementos de la interfaz de usuario. La tecnología utilizada por los métodos de entrada es diferente a la utilizada en los métodos de salida;
- **Grabadores:** esta funcionalidad le permite capturar fácilmente las acciones de un usuario en la pantalla y traducirlas en secuencias.
- **El repositorio de objetos:** el repositorio de objetos garantiza la administración, la reutilización y la confiabilidad de los elementos de la interfaz de usuario al capturarlos como objetos en un repositorio, que se pueden compartir entre proyectos.
- **AI Computer Vision:** puede identificar elementos de la interfaz de usuario como botones, campos de entrada de texto o casillas de verificación sin el uso de selectores.



¿Cuál es el propósito de un selector?

- Contiene detalles de elementos de la GUI en forma de fragmentos XML que ayudan a identificar los elementos de la interfaz de usuario.

Experiencia de diseño clásico y moderno

Al final de esta lección, debería poder explicar la diferencia entre las experiencias de diseño moderno y clásico en Studio;

¿Qué son las experiencias de diseño?

Hasta ahora, hemos aprendido sobre la automatización de la interfaz de usuario en general. Hay una última cosa que debe comprender antes de utilizar estas funciones. UiPath Studio ofrece dos formas de desarrollar la automatización de la interfaz de usuario: la experiencia de diseño clásico y la experiencia de diseño moderno. Ayudan a los Desarrolladores a adaptarse a diferentes contextos y las diferencias más importantes entre ellos se basan en los conceptos clave:

- Actividades de automatización de la interfaz de usuario
- Propiedades
- Métodos de orientación
- Registradores y asistentes de raspado
- El repositorio de objetos

La experiencia de diseño predeterminada se establece a nivel de proyecto desde Configuración del proyecto o para todos los proyectos nuevos desde Vista Backstage> Configuración> Diseño. Al comenzar, la experiencia predeterminada es Clásica.

¿Cuál es la diferencia entre las experiencias de diseño moderno y clásico?

Veamos cómo difieren las dos experiencias en los conceptos clave.

Actividades de automatización de la interfaz de usuario

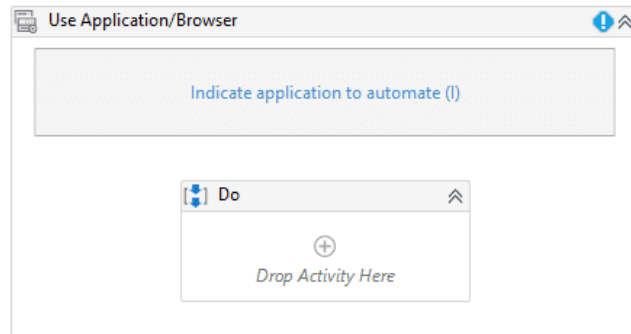
Un conjunto separado de actividades de la interfaz de usuario está disponible de forma predeterminada en cada experiencia. Independientemente de la experiencia seleccionada de un proyecto de automatización, moderno o clásico, se puede habilitar el otro conjunto de actividades.

Si la experiencia predeterminada es Clásica, para ver las Actividades modernas, simplemente seleccione el filtro "Mostrar modernas" en el panel Actividades. Si la experiencia predeterminada es moderna, puede seleccionar "Mostrar clásico".

Un gran ejemplo sería el:

Usar la actividad moderna de la aplicación / navegador que reemplaza las siguientes actividades clásicas:

- Aplicación abierta;
- Navegador abierto;
- Adjuntar ventana;
- Adjuntar navegador;
- Alcance del elemento;
- Cerrar ventana;
- Iniciar proceso.



The Use Application/Browser card available in Modern Experience.

Tenga en cuenta que las actividades modernas solo funcionan dentro de contenedores, mientras que las actividades clásicas también se pueden configurar para que funcionen sin contenedores.

Propiedades de la actividad

Las actividades clásicas y modernas tienen propiedades ligeramente diferentes debido a los diferentes métodos de orientación y comportamientos de actividad. Aprenderemos más sobre esto en las siguientes lecciones.

Métodos de orientación

Cuando se usan **actividades clásicas**, el selector es el método de orientación más común para elementos de la interfaz de usuario (99% del tiempo) y en escenarios donde no es confiable, tiene la posibilidad de cambiar manualmente la actualización de la actividad para usar selectores difusos o anclas.

Las **actividades modernas recorrerán las opciones** apiladas **Selector**, **Selector difuso**, **Ancla** e **Imagen** para determinar la opción más confiable en el orden en que las acabamos de presentar. En caso de que el método anterior no sea confiable, automáticamente elegirá el siguiente. Esta tecnología de focalización se denomina **objetivo unificado**.

Registradores y asistentes de raspado

Modern y Classic ofrecen dos juegos diferentes de grabadoras:

- **Experiencia moderna:** las grabadoras App / Web y Computer Vision están disponibles.
- **Experiencia clásica:** las grabadoras Basic, Desktop, Web, Image, Native Citrix y Computer Vision están disponibles.

Cuando se trata de Scraping Wizards:

- **Experiencia moderna:** el asistente de extracción de tablas está disponible para la extracción de datos.
- **Experiencia clásica:** están disponibles los asistentes para el raspado de datos y el raspado de pantalla.

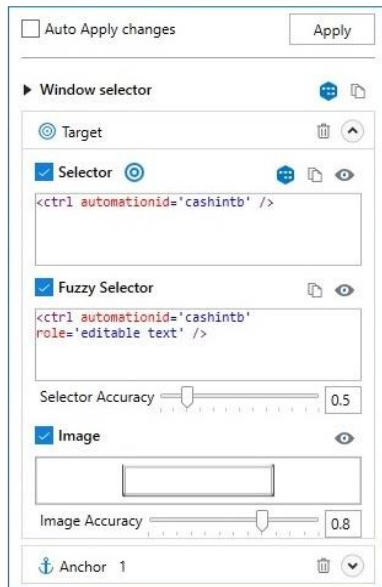
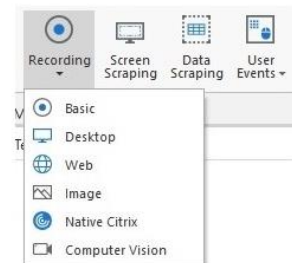
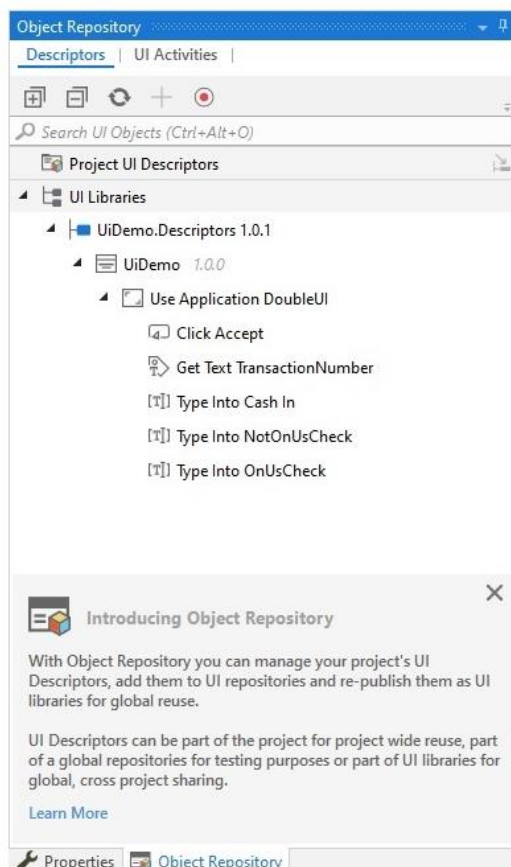


Image from the Modern Experience Capture Elements wizard which shows the targeting methods stacked.



Modern Experience Recorders and wizards are depicted on top, Classic Experience below.



Snip of the Object Repository panel, only available in the Modern Experience.

Repositorio de objetos

El repositorio de objetos garantiza la **gestión**, la **reutilización** y la **fiabilidad de los elementos** de la interfaz de usuario al capturarlos como objetos en un **repositorio**, que se pueden **compartir entre proyectos**. El repositorio de objetos **solo** está **disponible** en la **experiencia de diseño moderno**.

No se preocupe por recordar todo esto. Es útil saber cuáles son las áreas clave de diferencia y puede consultar este curso más adelante, si es necesario.

RESUMIENDO

- Hay dos experiencias de diseño de UI Automation en UiPath Studio: Modern y Classic
- La experiencia predeterminada es Clásica. Esto se puede cambiar a nivel de proyecto o para todos los proyectos nuevos.
- Las dos experiencias difieren en:
 - **Actividades:** ambas experiencias proporcionan actividades diferentes, pero puede filtrar el panel Actividades para ver las de la experiencia no predeterminada.
 - **Propiedades de la actividad:** las propiedades son ligeramente diferentes entre las actividades modernas y clásicas.
 - **Métodos de orientación:** Classic usa principalmente selectores, mientras que Modern usa un objetivo unificado (una pila de selector, selector difuso, ancla e imagen)
 - **Grabadoras:** Modern ofrece una solución de grabación simplificada con dos grabadoras, mientras que Classic ofrece seis grabadoras
 - **Asistentes de raspado:** Modern ofrece el asistente de extracción de tablas, mientras que Classic ofrece los asistentes de raspado de datos y raspado de pantalla.
 - **Repositorio de objetos:** el repositorio de objetos solo está disponible para la experiencia de diseño moderno.

Habilitación de la experiencia de diseño moderno

Al final de esta lección, debería poder:

- Habilite la experiencia de diseño moderno en Studio;
- Establezca una experiencia de diseño predeterminada para todos los nuevos proyectos de automatización.

Descripción general

En esta lección, aprenderemos a cambiar entre actividades modernas y clásicas, cómo configurar sus nuevos proyectos de automatización en una experiencia de diseño predeterminada y cómo cambiar la experiencia de diseño a nivel de proyecto.

RESUMIENDO

Principales conclusiones

- Independientemente de la experiencia de diseño utilizada, los desarrolladores de RPA tienen la posibilidad de utilizar actividades modernas y clásicas en los mismos proyectos de automatización. Además, las actividades clásicas se pueden utilizar como entrada para las actividades modernas y viceversa;
- Para ver las actividades desde una experiencia de diseño diferente a la seleccionada actualmente, haga clic en el botón Filtro del panel de actividades y seleccione "Mostrar moderno" o "Mostrar clásico". Las actividades clásicas que no tienen una contraparte en la experiencia de diseño moderno serán visibles independientemente del filtrado de actividades;
- Las actividades de interacción de la interfaz de usuario dentro de los "contenedores Do" solo apuntarán a elementos presentes en la aplicación de destino;
- El objetivo unificado que se encuentra en Experiencias de diseño moderno utiliza una pila de métodos de orientación compuestos por selectores, selectores difusos, anclajes e imágenes en un orden determinado. Si uno falla, pasa al siguiente;
- Cuando se utiliza Modern Recorder, los selectores se pueden editar directamente desde el UI Explorer.

Pasos para ver actividades modernas y clásicas en el panel de actividades

1. Navegar al panel de actividades
2. Haga clic en Filtro
3. Seleccione "Mostrar moderno" o "Mostrar clásico" según la experiencia de diseño actual

Examen Teórico

Los desarrolladores de RPA solo pueden usar actividades clásicas en proyectos creados con versiones de Studio antes de la actualización 20.10.

- Falso

¿Qué métodos de orientación utiliza Unified Target?

Elija todas las opciones que apliquen

- Selectores
- Selectores difusos
- Anclas
- Imagen

Al crear proyectos de automatización en las versiones 20.10 y posteriores de UiPath Studio, puede:

Elija todas las opciones que apliquen

- Utilice las actividades modernas como entrada para las actividades clásicas
- Utilice las actividades clásicas como entrada para las actividades modernas
- Cambie la experiencia de diseño de proyectos antiguos de clásico a moderno

Práctica

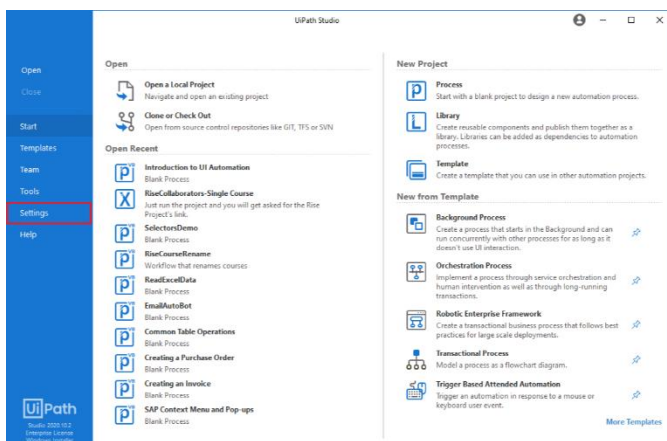
Habilitación de la experiencia de diseño moderno

Resolvamos un desafío muy simple para esta sesión de práctica:

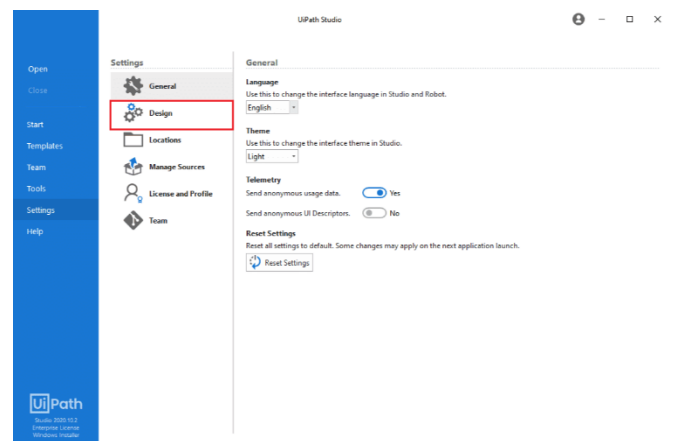
1. Cree un nuevo proyecto de Studio y habilite la experiencia de diseño moderno.
2. Establezca la Experiencia de diseño moderno como predeterminada para todos los proyectos nuevos.

Si ya habilitó la experiencia de diseño moderno en la lección anterior, puede omitir este ejercicio.

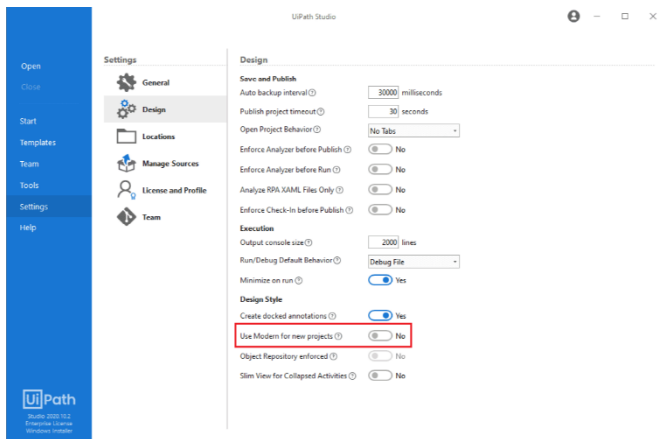
1



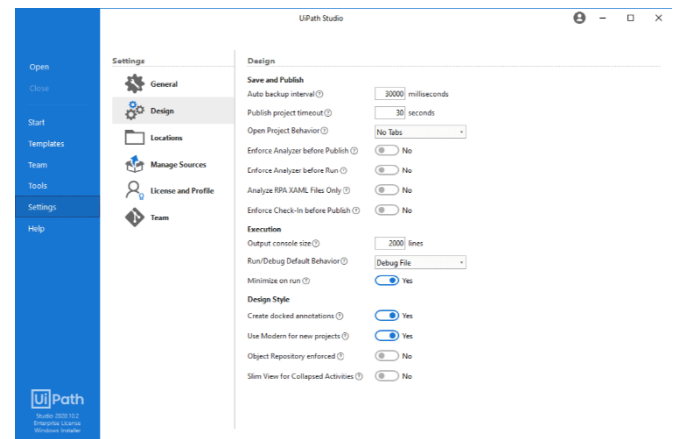
2



3

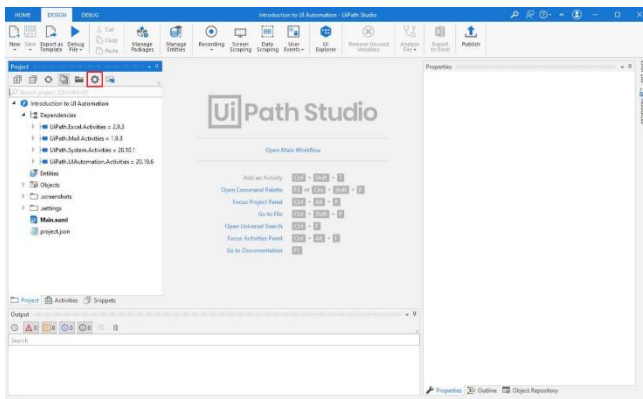


4

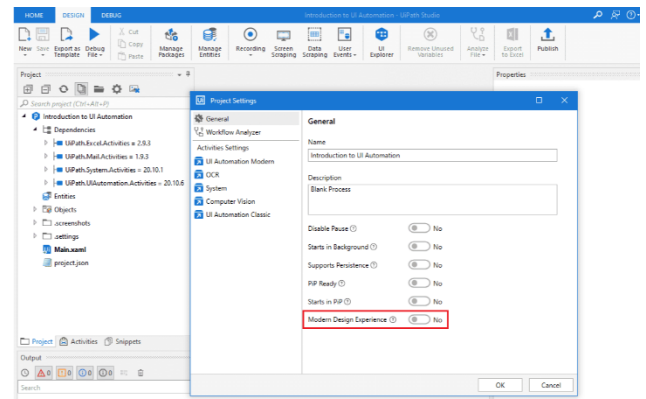


Establecer la experiencia de diseño moderno como predeterminada para todos los proyectos nuevos

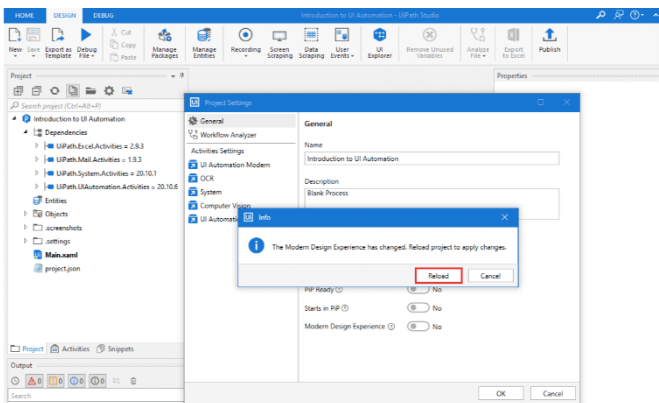
1



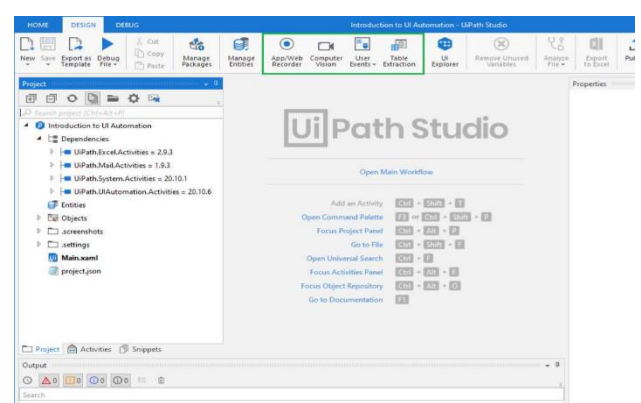
2



3



4



Examen Teórico

Acaba de crear un nuevo proyecto. ¿Cómo establecería la Experiencia de diseño moderno como predeterminada solo para este proyecto específico?

- Configuración del proyecto> General

Antes de crear una ruta de robot que defina un proyecto de automatización de la interfaz de usuario, los desarrolladores de RPA primero deben:

- Identifique el Camino Humano que logra el mismo objetivo al desglosar cada paso individual.

¿Cuáles de las siguientes afirmaciones son verdaderas?

- Los desarrolladores pueden utilizar actividades modernas y clásicas independientemente de la experiencia de diseño del proyecto de automatización.
- El repositorio de objetos solo está disponible en la experiencia de diseño moderno.
- Modern Activities utiliza una tecnología de orientación denominada objetivo unificado que recorre todos los métodos de orientación disponibles.

¿Cuál de las siguientes opciones se debe considerar en caso de que un Selector regular deje de ser confiable durante el desarrollo de la automatización?

- Intente utilizar los selectores difusos.
- Uso de anclajes para señalar el elemento de la interfaz de usuario.
- Usando el método de orientación por imágenes.

Está ejecutando la experiencia de diseño clásico. ¿Qué ruta utilizaría para habilitar la experiencia de diseño moderno para todos los proyectos nuevos?

- Paso 1: Acceder a la vista Backstage de Studio
- Paso 2: Haga clic en Configuración
- Paso 3: Haga clic en Diseño
- Paso 4: Habilitar el uso moderno para nuevos proyectos

AUTOMATIZACIÓN DE LA INTERFAZ DE USUARIO CON LA EXPERIENCIA MODERNA

AUTOMATIZACIÓN DE LA INTERFAZ DE USUARIO CON LA EXPERIENCIA CLASICA

INTERACCIONES DE IU

Buscar elemento: Espera a que el elemento de IU especificado aparezca en la pantalla (para estar en primer plano) y lo devuelve como una variable UiElement. Esto es útil, por ejemplo, cuando una determinada acción tiene que ejecutarse en el elemento de IU encontrado.

Elemento existente: Le permite verificar si existe un elemento de IU, incluso si no es visible. Devuelve una variable boolean, lo que la hace muy útil en las actividades "Instrucción Si", por ejemplo.

Esperar a fuga en elemento: Espera a que el elemento de IU especificado desaparezca de la pantalla. Es una alternativa a Buscar elemento, por ejemplo, cuando la desaparición de un elemento (un signo de carga) es más confiable que la aparición de otro elemento.

Apariencia en elemento: Un contenedor que espera que aparezca un elemento de la IU y le permite realizar varias acciones dentro de él.

Fuga en elemento: Un contenedor que le permite realizar una o varias acciones después de que desaparezca un elemento de la IU específico.

Texto existente: Comprueba si un texto se encuentra en un elemento de IU determinado. Hay una versión alternativa de esta que usa la tecnología OCR para comprobar si hay un elemento de IU determinado. Esto resulta útil cuando los elementos de IU no son accesibles más que como imágenes

Actividad Obtener texto: Extrae un valor de texto de un elemento de la IU especificado.

Actividad Obtener texto completo: Extrae una cadena y su información de un elemento indicado de la IU mediante el método de extracción de pantalla Texto completo. Por lo tanto, el texto oculto también se captura de forma predeterminada (aunque ofrece la opción de ignorar el texto oculto). Esta actividad se genera automáticamente cuando se hace la extracción de pantalla con el método Texto completo, junto con un contenedor.

Actividad Obtener texto visible: Extrae una cadena y su información desde un elemento de IU indicado mediante el método de extracción de pantalla nativo. Esta actividad se genera automáticamente cuando se hace la extracción de pantalla con el método nativo seleccionado, junto con un contenedor.

SELECTORES

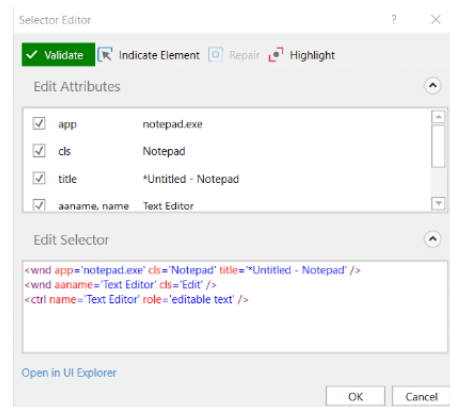
- wnd → (ventana)
- html → (página web)
- ctrl → (control)
- webctrl → (control de página web)
- java → (control de aplicación Java)

Los marcos de IU:

1. **PREDETERMINADO:** Este es el método propietario que por lo general funciona correctamente con todos los tipos de interfaces de usuario.
2. **ACCESIBILIDAD ACTIVA:** Esto representa una solución anterior de Microsoft que hace que las aplicaciones sean accesibles. Se recomienda cuando se usa software heredado, si el marco predeterminado no funciona como se esperaba.
3. **AUTOMATIZACIÓN DE IU:** Este es el modelo de accesibilidad mejorado de Microsoft que se recomienda cuando se usan aplicaciones más recientes en caso de que el marco predeterminado no funcione como se esperaba.

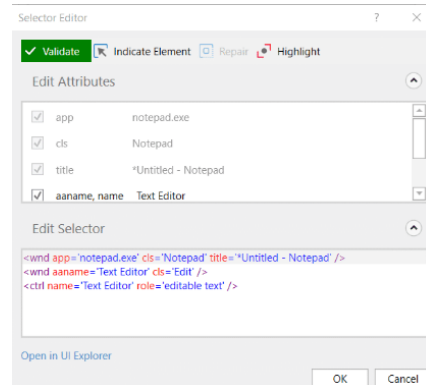
SELECTORES COMPLETOS

- Contienen todas las etiquetas y atributos necesarios para identificar un elemento de IU, incluida la ventana de nivel superior
- Generados por la Grabadora básica
- Más adecuados cuando las acciones realizadas requieren cambiar entre varias ventanas.



SELECTORES PARCIALES

- No contienen las etiquetas y atributos de la ventana de nivel superior, por lo que las actividades con selectores parciales tienen que incluirse en contenedores
- Generados por la Grabadora de escritorio
- Más adecuados para realizar varias acciones en la misma ventana.



¿CUÁNDO SE USAN LOS SELECTORES PARCIALES O COMPLETOS?

El mejor ejemplo de uso de un selector parcial sería una automatización simple donde el flujo de trabajo implementado solo realiza acciones en la misma aplicación sin cambiar entre varias ventanas como un simple CRM.

Por otro lado, si el flujo de trabajo realmente tuviera que interactuar con varias ventanas como el mismo CRM y un documento, lo que haría que los elementos de IU necesarios en este ejemplo en particular se dispersaran en varias ventanas, se requeriría un selector completo

¿CUÁNDO SE TIENEN QUE AJUSTAR LOS SELECTORES?

Selectores generados dinámicamente Como sucede con algunos sitios web, los valores de los atributos cambian con cada visita.

Selectores demasiado específicos Algunos selectores se generan de forma automática con el nombre del archivo o con un valor que cambia. Aquí, los marcadores de posición son muy útiles.

Cambios del sistema Algunos selectores contienen la versión de la aplicación u otro elemento que cambia cuando se actualiza la aplicación.

Selectores que usan IDX El IDX es el índice del elemento actual en un contenedor con varios elementos similares. Esto puede cambiar cuando un nuevo elemento aparece en el mismo contenedor.

GESTIÓN DE SITUACIONES DIFÍCILES

Base de anclaje

Esto es muy útil en los casos en los que los valores de atributo no son confiables (por ejemplo, se generan en cada ejecución), pero hay un elemento de IU que es estable y está vinculado al elemento de IU de destino.

La actividad Base de anclaje tiene dos partes, una para ubicar el elemento de IU de anclaje (como "Buscar elemento") y la segunda para realizar la actividad deseada

Selector relativo Esta actividad básicamente incorporará la información sobre el selector del anclaje en el selector del elemento de IU de destino. Sin embargo, el nuevo selector probablemente deberá editarse de nuevo, ya que algunos nodos del primer selector todavía estarán en el nuevo. La solución es quitar esa parte (como un ID dinámico) y el selector se estabilizará con el selector del anclaje.

Jerarquía de árbol visual La jerarquía del árbol visual puede mejorar la fiabilidad de un selector al incluir las etiquetas y los atributos del elemento anterior en la jerarquía.

Esto es muy útil cuando el selector del elemento de IU de destino no es confiable, pero sí lo es el selector del elemento de IU justo encima en la jerarquía. Sin embargo, de nuevo, el selector necesita más edición y validación, ya que la parte dinámica tiene que eliminarse y, al mismo tiempo, debe asegurarse de que el elemento de destino se pueda identificar con un atributo único.

Buscar elementos secundarios Esta actividad puede identificar todos los elementos secundarios de un elemento que es más estable. Dado que su salida es la colección de elementos secundarios, tendrá que crear un mecanismo para identificar solo el elemento de IU de destino (mediante uno de sus atributos que hace que sea único entre los elementos secundarios, pero no sería suficiente para identificarlo universalmente).

WILDCARDS

Wildcards are symbols that enable you to replace zero or multiple characters in a string. These can be quite useful when dealing with dynamically-changing attributes in a selector.

- Asterisk (*) – replaces zero or more characters
- Question mark (?) – replaces a single character

ORGANIZACIÓN DE PROYECTOS

SECUENCIA

¿Cuándo usarlo?

Cuando hay una sucesión clara de pasos sin demasiadas condiciones (por ejemplo, una automatización de IU).

Normalmente, las secuencias se usan para anidar flujos de trabajo y la lógica de alto nivel se controla a través de diagramas de flujo o de máquinas de estados.

¿Qué ventajas ofrece?

Fácil de entender y de seguir gracias al enfoque de arriba a abajo.

Ideal para la lógica simple, como la búsqueda de un elemento en Internet.

¿Cuáles son las desventajas?

Anidar demasiadas condiciones en la misma secuencia dificulta la lectura del proceso.

No es adecuado para flujos continuos.

DIAGRAMA DE FLUJO

¿Cuándo usarlo?

Cuando tiene un flujo complejo con varias condiciones, el diagrama de flujo es, al menos visualmente, mucho más fácil de entender y de seguir.

Cuando necesita un flujo que se ejecute continuamente o que termine solamente en ciertas condiciones.

¿Qué ventajas ofrece?

Fácil de entender, ya que es similar a los diagramas lógicos en la computación de software.

Se puede utilizar para flujos de trabajo continuos.

¿Cuáles son las desventajas?

Los diagramas de flujo solamente se pueden utilizar como flujo de trabajo general (con secuencias anidadas en el interior), no para partes individuales de proyectos (anidadas dentro de otros flujos de trabajo).

MÁQUINA DE ESTADOS

¿Cuándo usarlo?

En primer lugar, vamos a ver qué es una máquina de estados. Es una máquina abstracta que consiste en un número finito de estados predefinidos y en transiciones entre estos estados. En cada momento, basándose en las entradas externas y en las condiciones verificadas, solamente puede estar en uno de los estados.

Las máquinas de estados pueden pasar por un número finito de estados claros y estables. Algunos ejemplos de su funcionamiento habitual incluyen máquinas expendedoras, ascensores o semáforos

¿Qué ventajas ofrece?

Se pueden utilizar para flujos de trabajo continuos más complejos.

Las transiciones entre estados se pueden definir fácilmente y ofrecen flexibilidad.

Puede alojar procesos más complejos y no se pueden capturar mediante bucles simples ni instrucciones Si.

Es más fácil cubrir todos los casos/transiciones posibles con máquinas de estados.

¿Cuáles son las desventajas?

Tiempo de desarrollo más largo debido a su complejidad: dividir el proceso en "estados" lógicos, averiguar las transiciones, etc.

Nota: Las máquinas de estados no se deben utilizar excesivamente, deben definir solamente el esqueleto del proyecto.

Notas importantes sobre las máquinas de estados

1. Es muy importante darles nombres a los estados en una máquina de estados para facilitar el mantenimiento y el desarrollo futuro.
2. En UiPath Studio, el orden de las transiciones mostradas en cada estado es muy importante, ya que se corresponde con el orden en que son evaluadas.

GESTIÓN DE DATOS

Entrada: si queremos usar el valor del argumento dentro del flujo de trabajo invocado.

Salida: si queremos pasar el argumento fuera del flujo de trabajo invocado.

Entrada/Salida: si queremos pasar una variable del flujo de trabajo principal al flujo de trabajo invocado, modificarla en el flujo de trabajo invocado y, a continuación, pasarla de nuevo a la carpeta principal.

¿CÓMO GESTIONAR DIFERENTES VERSIONES DE UN MISMO PROYECTO?

Los sistemas de control de origen les permiten a los usuarios de diferentes equipos y ubicaciones acceder a los mismos recursos y trabajar en los mismos campos de proyecto. También se utilizan para versionar el código y mantener el historial de todos los cambios realizados durante el desarrollo.

A través de la página Equipo en la vista backstage, Studio admite los siguientes sistemas de control de origen:

1. **GIT:** Con la integración Git puede:
 - Clonar un repositorio remoto.
 - Añadir un proyecto.
 - Confirmar e insertar
 - Copiar un proyecto en Git.
 - Crear y administrar ramas.
 - Resolver conflictos con la opción Diferencia de archivo
2. **TFS**
 - Se admiten las siguientes versiones:
 - a. 2012
 - b. 2013
 - c. 2015
 - d. Express 2012
 - e. Express 2013
 - f. Express 2015
 - En primer lugar, debe configurar TFS en Studio.
 - A continuación, puede abrir un proyecto o agregar un nuevo proyecto a TFS.
3. **SVN:** Con la integración SVN puede:
 - Abrir un proyecto desde SVN.
 - Añadir un proyecto a SVN.

GESTIÓN DE ERRORES Y EXCEPCIONES

EXCEPCIONES COMUNES

- **NullReferenceException**: este error suele ocurrir cuando se usa una variable sin valor establecido (no inicializada).
- **IndexOutOfRangeException**: ocurre cuando el índice de un objeto está fuera de los límites de la colección.
- **ArgumentException**: se lanza cuando un método se invoca y, en por lo menos uno de los argumentos pasados, no cumple las especificaciones del método llamado.
- **SelectorNotFoundException**: se lanza cuando el robot no encuentra el selector designado a una actividad en la aplicación de destino dentro del periodo de tiempo de espera.
- **ImageOperationException**: ocurre cuando no se encuentra una imagen dentro del periodo de tiempo de espera.
- **TextNotFoundException**: ocurre cuando el texto indicado no se encuentra dentro del periodo de tiempo de espera.
- **ApplicationException**: describe un error anclado en un problema técnico, como por ejemplo una aplicación que no responde.

INTENTAR CAPTURAR / TRY CATCH

Esta actividad captura un tipo de excepción especificado en una secuencia o actividad y muestra una notificación de error o lo descarta y continúa la ejecución.

Como mecanismo, Intentar capturar ejecuta las actividades en el bloque Intentar y, si se produce un error, ejecuta las actividades en el bloque Capturar. El bloque Final solo se ejecuta cuando no se lanza ninguna excepción o cuando se captura una excepción y se controla en el bloque Capturar (sin que se vuelva a lanzar).

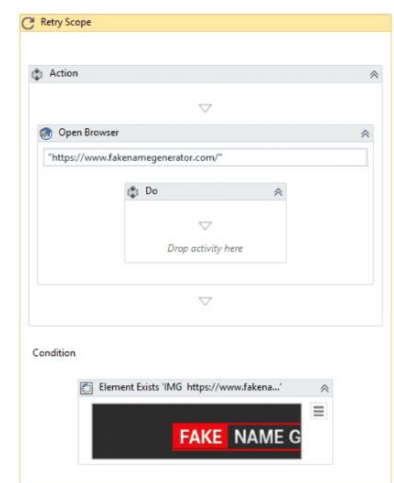
1. **TRY**: Las actividades llevadas a cabo que tienen la posibilidad de arrojar un error.
2. **CATCH**: La actividad o el conjunto de actividades que se deben llevar a cabo cuando se produce un error. Tenga en cuenta que en este bloque se pueden añadir varios errores y las actividades correspondientes.
3. **END**: La actividad o el conjunto de actividades que se deben llevar a cabo después del bloque Intentar Capturar. Esta sección solamente se ejecuta cuando no se lanza ninguna excepción o cuando se produce un error y se captura en la sección Capturar.

RETRY

La actividad Reintentar vuelve a intentar las actividades contenidas siempre que no se cumpla la condición o se produzca un error.

La actividad Reintentar se usa para detectar y gestionar un error, por lo que es similar a Intentar capturar. La diferencia es que esta actividad simplemente vuelve a intentar la ejecución en lugar de proporcionar un mecanismo de gestión más complejo.

Se puede utilizar sin una condición de terminación, en cuyo caso volverá a intentar las actividades hasta que no se produzca ninguna excepción (o se supere el número de intentos proporcionado).



Propiedades adicionales

- **NumberOfRetries:** el número de veces que se va a volver a intentar la secuencia.
- **RetryInterval:** especifica la cantidad de tiempo (en segundos) entre cada reintento.

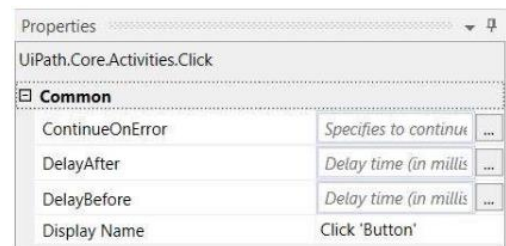
PROPIEDAD CONTINUEONERROR

Continuar en caso de error es una propiedad que especifica si la ejecución debe continuar cuando la actividad lanza un error.

Recuerde que si **ContinueOnError** está establecida en Verdadero en una actividad que tiene un ámbito (como por ejemplo Asociar ventana o Asociar navegador), también se van a ignorar todos los errores que se produzcan en otras actividades dentro de ese ámbito.

No es recomendable tener esta propiedad establecida en verdadero en todos los casos, pero en algunos tiene sentido, como por ejemplo:

cuando utilizamos la extracción de datos: para que la actividad no lance un error en la última página (cuando no se encuentra el selector del botón "Siguiente"); si no queremos capturar el error, sino simplemente ejecutar la actividad.



Este campo solamente admite valores booleanos (Verdadero, Falso). El valor predeterminado es **Falso**, por lo tanto, si el campo está en blanco y se produce un error, la ejecución del proyecto se detiene. Si el valor está establecido en **Verdadero**, la ejecución del proyecto continúa sin tener en cuenta ningún error.

CONTROLADOR GLOBAL DE EXCEPCIONES

El controlador de excepciones global es un tipo de flujo de trabajo diseñado para determinar el comportamiento cuando se encuentra un error de ejecución a nivel de proyecto. Esta es la razón por la que un controlador de excepciones global **solamente se puede establecer por proyecto de automatización**.

Al controlador de excepciones llegarán solamente las excepciones no detectadas. Si se produce una excepción dentro de una actividad Intentar Capturar y se detecta y trata correctamente dentro del bloque Captura (y no se vuelve a producir), no llega al controlador de excepciones global.

Un controlador de excepciones global se puede crear iniciando un proyecto nuevo con este tipo o estableciendo un proyecto existente como controlador de excepciones global desde el panel Proyecto.

¿Cómo funciona?

El controlador de excepciones global cuenta con 2 argumentos predefinidos que no se deben eliminar:

1. **errorInfo**, con dirección de entrada: contiene el error que se produjo y el flujo de trabajo que produjo un error, simplemente registra el error. El desarrollador puede elegir el nivel de registro: Grave, Error, Advertencia, Info, etc.
2. **result**, con dirección de salida: almacenará el siguiente comportamiento del proceso cuando encuentre el error. el desarrollador puede elegir la acción que se debe realizar cuando se encuentra un error durante la ejecución:
 - Continuar: se vuelve a lanzar la excepción.
 - Ignorar: se ignora la excepción y la ejecución continúa a partir de la siguiente actividad.
 - Reintentar: se vuelve a intentar la actividad que produjo la excepción.
 - Anular: la ejecución se detiene después de ejecutar el controlador actual.

DEPURACIÓN / DEBUGGING

- Ahora, las opciones Depurar archivo o Depurar proyecto, Ejecutar archivo o Ejecutar proyecto son estados del mismo botón en la cinta.
- El **panel Locales** muestra los valores de los argumentos, las propiedades de la actividad actual y las de la ejecutada previamente. Los valores de las variables y de los argumentos se pueden modificar desde este panel. Se ha mejorado el diseño del panel.
- Puede hacer clic con el botón derecho del ratón en las variables y en los argumentos en el panel Locales para añadirlos al **panel de Inspección**.
- En el panel Locales se muestran los detalles sobre la excepción lanzada.
- El **panel Inmediato** le permite evaluar las expresiones que defina durante la depuración para inspeccionar los datos.
- Puede supervisar la ejecución de las actividades, los contenedores y los archivos de proyecto usando el **panel Pila de llamadas**. Este panel le permite acceder fácilmente a la actividad que lanzó una excepción.
- Al trabajar con puntos de interrupción, ahora puede añadir condiciones, un recuento y registrar un mensaje cuando los encuentre.
- Ahora puede probar las actividades tanto dentro del flujo de trabajo, usando la opción del menú de contexto Probar actividad o desde el panel Actividades, creando un banco de pruebas.
- Las opciones del menú de contexto **Ejecutar en esta actividad** y **Ejecutar desde esta actividad** empiezan a depurar el flujo de trabajo en y desde una actividad determinada.
- Utilice la opción de la cinta **Reiniciar** para reiniciar el proceso de depuración desde la primera actividad del proceso o la actividad desde la cual se desencadenó la opción Ejecutar desde esta actividad.
- La nueva acción "**Paso a paso para salir**" finaliza la ejecución de la actividad actual y vuelve al nivel del contenedor deteniendo el proceso de depuración.
- Cada vez que se encuentra una **excepción durante la depuración**, puede Interrumpir, Ignorar el error y Continuar. La opción Reintentar vuelve a ejecutar la actividad previa y lanza la excepción si se vuelve a encontrar. Ahora estas acciones están disponibles en la pestaña Depurar.

AUTOMATIZACIÓN DE PDF

AUTOMATIZACIÓN DE CORREO ELECTRÓNICO

SMTP es la abreviatura de Simple Mail Transfer Protocol (Protocolo simple de transferencia de correo) y es un protocolo básico utilizado solo para enviar mensajes. **POP3** significa Post Office Protocol (Protocolo de Oficina de Correos) y es un protocolo antiguo y casi obsoleto para leer mensajes, pero la mayoría de los servidores de correo electrónico lo admiten. **IMAP** o Internet Message Access Protocol (Protocolo de acceso a mensajes de Internet) solo se utiliza para recibir mensajes, pero ofrece algunas funciones útiles para marcar los mensajes como leídos o moverlos entre carpetas. **Exchange** es la solución de correo electrónico de empresa de Microsoft, que UiPath integra perfectamente, ofreciendo actividades para enviar y recibir mensajes, mover correos electrónicos entre carpetas y eliminarlos. Las actividades de **Outlook** son ligeramente diferentes. Funcionan con la API de la aplicación de escritorio, por lo que las acciones ya tienen un contexto. No es necesario configurar servidores, usuarios y otros detalles técnicos: estas actividades están diseñadas para usar las cuentas de Outlook ya existentes. Por último, hay dos actividades de correo electrónico genéricas disponibles para guardar mensajes de correo electrónico y archivos adjuntos de correo electrónico en la unidad local.

CONSULTAR EL CORREO ELECTRÓNICO

Hay cuatro actividades para Obtener correo: POP3, Outlook, IMAP y Exchange. Todas ellas ofrecen la misma funcionalidad de proporcionar mensajes de correo electrónico; naturalmente, también comparten otras opciones. Recuperar el correo electrónico de una carpeta de buzón de correo determinada, obtener solo los mensajes no leídos y marcarlos como leídos para evitar el reprocesamiento, limitando el número de correos electrónicos entrantes. Para configurar una conexión de servidor, la dirección del servidor y el puerto generalmente son necesarios junto con un correo electrónico y una contraseña del usuario. Tenga en cuenta que IMAP y POP3 muestran advertencias de validación, mientras que las versiones de Outlook y Exchange no. Esto se debe a que mientras que IMAP y POP3 requieren parámetros de conexión, Outlook y Exchange funcionan con valores predeterminados y detección automática, lo que requiere menos esfuerzo cuando se usan mediante flujos de trabajo.

ENVÍO DE CORREOS ELECTRÓNICOS

Tenemos tres opciones muy similares: SMTP (abreviatura de Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo), Outlook y Exchange

Outlook y Exchange no requieren parámetros de conexión y Outlook no necesita usuario y contraseña.

ORCHESTRATOR PARA DESARROLLADORES

Orchestrator es el componente de UiPath Suite a través del cual se publican los flujos de trabajo de automatización desarrollados en Studio, se asignan a robots y se ejecutan. Viene en forma de una aplicación web que permite la gestión de robots, paquetes de actividad, datos a procesar, horarios de ejecución y otros activos.

Orchestrator es ideal para grandes implementaciones de robots que cubren procesos complejos, pero también se puede implementar en escenarios que se ocupan de procesos cortos y repetitivos y menos robots.

¿Cuáles son las capacidades de Orchestrator?

- **Aprovisionamiento:** crea y mantiene la conexión con los robots.
- **Implementación:** garantiza la entrega de los flujos de trabajo para su ejecución, ya sea de forma inmediata o mediante programaciones.
- **Configuración:** permite la creación, configuración y mantenimiento de grupos de robots y la ejecución de tareas.
- **Colas:** los datos que deben procesarse se desglosan en operaciones indivisibles denominadas transacciones. Las colas pueden almacenar cualquier número de transacciones y facilitan su distribución, ejecución y supervisión.
- **Supervisión:** realiza un seguimiento de los datos de identificación del robot y mantiene los permisos de usuario.
- **Registro:** almacena e indexa los registros en una base de datos SQL y/o ElasticSearch.
- **Interconectividad:** actúa como el punto de comunicación centralizado para aplicaciones o soluciones de terceros y se puede usar para almacenar paquetes de actividad, bibliotecas y otros activos (como credenciales).

¿Cuáles son las ventajas de Orchestrator?

Accesibilidad y control de versiones: Los flujos de trabajo se pueden publicar como paquetes y almacenarse en Orchestrator, a nivel de versión y con las notas de la versión del desarrollador. Se pueden distribuir diferentes versiones a los robots para su ejecución.

Las bibliotecas también se pueden publicar y almacenar en Orchestrator. Desde allí, se puede acceder a ellas en cualquier momento y utilizarse durante el desarrollo y distribuirse a los robots.

Procesamiento transaccional: Las colas permiten la asignación eficiente de la carga de trabajo entre robots, la ejecución continua y la supervisión a nivel de transacción. Si una determinada transacción en la cola no se puede ejecutar por ciertos motivos, no detendrá ni retrasará la ejecución general.

Planificación y ejecución eficientes: Orchestrator permite la ejecución de tareas de acuerdo con programaciones que pueden incluir varios escenarios (como días no laborables), elegir destinos de ejecución de los robots disponibles e incluso crear bucles de ejecución continua fuera del horario comercial.

Supervisión: Los robots, los procesos y la ejecución de tareas se pueden supervisar a través de Orchestrator, lo que puede permitir una reacción rápida en caso de cualquier error, y también proporciona los medios para la notificación precisa y la auditoría del trabajo del robot.

CONCEPTOS RELACIONADOS CON ORCHESTRATOR

Robot: El robot es el agente de ejecución de UiPath que le permite ejecutar flujos de trabajo integrados en Studio. Puede ser de varios tipos, como verá en el siguiente capítulo.

Entorno: Un entorno es un grupo de robots configurados en Orchestrator. Los procesos se pueden asignar a robots individuales, pero es más eficaz asignarlos a entornos. Un robot puede formar parte de más de un entorno, siempre que esté en el mismo servicio.

Unidad de organización: En general, es una entidad de Orchestrator que corresponde a una unidad de negocio. La misma instancia de Orchestrator puede tener varias unidades de organización configuradas, cada una de ellas con robots, entornos, colas y activos independientes, etc.

El plan Community Edition no permitirá la creación de otras unidades de organización distintas de la predeterminada creada.

Paquete: Un proyecto desarrollado en UiPath Studio que se publica en Orchestrator. Se pueden almacenar y utilizar varias versiones del mismo proyecto.

Proceso: Es un paquete que se ha asignado a un entorno determinado.

Trabajo: Un trabajo es un proceso que se ha enviado para su ejecución a algunos o a todos los robots del entorno.

Programación: Proceso configurado para una ejecución que no es inmediata, pero según una programación. Es posible realizar múltiples configuraciones cuando se trata de programaciones.

Activo: Un activo son datos almacenados en Orchestrator para el uso de robots. Existen cuatro tipos de activos:

- Texto: almacena solo cadenas (no es necesario agregar comillas);
- Booleano: admite valores true o false;
- Número entero: almacena solo números enteros;
- Credencial: contiene nombres de usuario y contraseñas que el robot requiere para ejecutar procesos particulares, como los detalles de inicio de sesión para los sistemas ERP.

Cola: Una cola es una secuencia de transacciones que se compila en Orchestrator y, a continuación, se utiliza para enviar a robots para su procesamiento.

TIPOS DE ROBOTS

1. **Robot atendido:** Este tipo de robot es activado por eventos de usuario y funciona junto a un humano en la misma estación de trabajo. Los robots atendidos se utilizan con Orchestrator para realizar una implementación de procesos y para un medio de registros centralizados.
2. **Robot desatendido:** Los robots se ejecutan desatendidos en entornos virtuales y ejecutan un número indefinido de procesos. Además de las capacidades de los robots atendidos, Orchestrator permite la ejecución, la supervisión, la programación y la compatibilidad con las colas de trabajo.
3. **Robot de desarrollo:** Tiene las características de un robot desatendido, pero debe utilizarse solo para conectar Studio a Orchestrator, con fines de desarrollo.
4. **Robot de no producción:** Son similares a los robots desatendidos, pero deben usarse solo con fines de desarrollo y para la realización de pruebas.

Según la interacción de robot/máquina, hay dos tipos de robots

1. **Robot estándar:** Funciona solamente en una sola máquina estándar. Es una buena opción cuando la máquina en la que se ejecuta el robot es conocida y nunca cambiará.
2. **Robot flotante:** Funciona en cualquier máquina definida en Orchestrator. Es una buena opción cuando los usuarios humanos trabajan en turnos en la misma máquina o en máquinas diferentes y cuando las máquinas virtuales se regeneran con frecuencia.

PROCESOS

Un proceso representa la asociación entre un paquete y un entorno. Se puede acceder a los procesos desde el menú de la izquierda. Para crear un nuevo proceso, simplemente haga clic en el signo "+", seleccione un paquete disponible en Orchestrator, seleccione el entorno y asígnele una descripción.

Si el main.xaml del proceso tiene argumentos In, Out o In/Out, se mostrarán en la pestaña Parámetros de la ventana Ver procesos. En Orchestrator, se convierten en parámetros de entrada y salida y se pueden configurar desde la pestaña Parámetros.

TRABAJOS Y PROGRAMACIONES

Una vez creado un proceso, se puede desencadenar su ejecución. Hay 3 maneras de hacerlo:

1. Uso de trabajos (inmediatamente)

Para iniciar un trabajo, simplemente diríjase a Trabajos desde el menú de la izquierda, elija un proceso y seleccione los robots que desea que ejecuten el trabajo desde el entorno o asigne el trabajo dinámicamente. Un trabajo asignado directamente a ciertos robots tendrá prioridad sobre los trabajos asignados dinámicamente, pero los trabajos asignados dinámicamente se ejecutan inmediatamente cuando cualquier robot del entorno pasa a estar disponible.

Al iniciar un trabajo, tiene la opción de establecer parámetros de entrada o mostrar los parámetros de salida si existen como argumentos en el archivo Main.xaml del paquete.

Hay 2 opciones para detener un trabajo:

- mediante Detener trabajo, que lo detendrá cuando se encuentre una actividad "Debe detenerse" en el flujo de trabajo
- mediante Cancelar trabajo, que lo detendrá inmediatamente.

2. Uso de programaciones (planificado)

La programación ofrece varias opciones de configuración, tales como:

- elegir un intervalo regular para ejecutar el proceso
- configurar parámetros (argumentos de Main.xaml en Studio)
- seleccionar los robots para ejecutar (de forma similar a Trabajos): todos los robots, robots seleccionados o dinámicamente
- configurar días no laborables y otros parámetros.

Para programar una ejecución de proceso, vaya a la página Programaciones, cree una nueva programación y configure sus parámetros. Una vez que haya terminado, haga clic en "Crear".

3. Desde la bandeja de robot

La ejecución de procesos desde la bandeja de robots puede ser muy útil para fines de prueba o automatizaciones simples creadas para partes de su trabajo o asuntos personales.

Para ejecutar un proceso directamente desde la bandeja del robot:

- asegúrese de que el paquete esté asignado al entorno del que forma parte el robot atendido;
- abra el robot;
- si el paquete tiene una actualización disponible, verá una flecha hacia arriba desde la que puede actualizarlo en su robot;
- una vez hecho esto, puede ejecutar el proceso haciendo clic en el "icono de reproducción".

¿CÓMO SE CREAN Y UTILIZAN LOS ACTIVOS?

En Orchestrator:

- Los activos se pueden crear desde el área específica;
- Debe proporcionarse el nombre y el tipo de datos. Mediante el uso de las pestañas, los activos se pueden configurar de la siguiente manera:
 - Valor único: todos los robots pueden acceder a él y utilizarlo;
 - Valor por robot: a cada valor proporcionado solo puede acceder el robot indicado.
- Los activos se pueden modificar o eliminar desde el mismo menú.

En Studio:

- Para credenciales, se debe usar la actividad "Obtener credencial";
- Para todos los demás tipos de activos, se utiliza la actividad "Obtener activo".

TRABAJAR CON COLAS

Las colas se crean fácilmente en Orchestrator. Cuando se crean, se encuentran vacías, pero hay actividades específicas en UiPath Studio para hacer que los robots las rellenen. La carga masiva también se admite directamente en Orchestrator, desde archivos .csv.

Al crear una cola, se establece el número máximo de reintentos (el número de veces que desea que se vuelva a intentar un artículo en cola) y el campo Referencia única (seleccione Sí si desea que las referencias de las transacciones sean únicas). Una vez creada una cola, esta configuración no se puede modificar

Las colas son muy importantes para el modelo Distribuidor y Ejecutor, en el que se separan las dos etapas principales de un proceso que implica colas:

- la etapa en la que los datos se toman y se introducen en una cola en Orchestrator, desde donde los robots pueden tomarlos y procesarlos. Esto se denomina Distribuidor;
- la etapa en la que se procesan los datos, se denomina Ejecutor.

ESTABLECER ESTADO DE LA TRANSACCIÓN

Cambia el estado del elemento de transacción a Error (con una aplicación o una excepción empresarial) o a Correcto. Como enfoque general, se volverá a intentar una transacción fallida debido a excepciones de aplicación y no se volverán a intentar las transacciones fallidas debido a excepciones empresariales.

Un artículo en cola puede tener uno de los siguientes estados:

- Nuevo: se acaba de agregar a la cola mediante Agregar artículo en cola (o el elemento se pospuso o se agregó una fecha límite).
- En curso: el elemento se procesó con la actividad Obtener elemento de transacción o Agregar artículo en cola;
- Error: el elemento no cumplía un requisito de negocio o aplicación dentro del proyecto;
- Correcto: el elemento se ha procesado;
- Abandonado: el elemento permaneció en el estado En curso durante un largo período de tiempo (aprox. 24 horas) sin ser procesado;
- Reintento: el elemento falló con una excepción de aplicación y se volvió a intentar (al final del proceso reintentado, el estado se actualizará a uno final: Correcto o Error)
- Eliminado: el elemento se ha eliminado manualmente de la página Transacciones

INTRODUCCIÓN A ROBOTIC ENTERPRISE FRAMEWORK

¿Qué es una transacción?

Una transacción representa la cantidad mínima (atómica) de datos y los pasos necesarios para procesar los datos, para cumplir una sección de un proceso empresarial. Un ejemplo típico sería un proceso que lee un único correo electrónico de un buzón de correo y extrae datos de él.

Llamamos a los datos atómicos porque, una vez procesados, la suposición es que ya no los necesitaremos en el futuro con el proceso empresarial.

Teniendo en cuenta los pasos de un proceso empresarial y cómo se repiten, podemos dividir los procesos empresariales en **tres categorías**:

1. Lineal:

Los pasos del proceso se realizan una sola vez y, si existe la necesidad de procesar datos diferentes, la automatización debe ejecutarse de nuevo. Por ejemplo, si volvemos al ejemplo de correo electrónico de la introducción de este capítulo y llega un nuevo correo electrónico, la automatización debe ejecutarse de nuevo para procesarlo.

Los procesos lineales suelen ser simples y fáciles de implementar, pero no muy adecuados para situaciones que requieren repeticiones de pasos utilizando datos diferentes.



2. Iterativo:

Los pasos del proceso se realizan varias veces, pero cada vez se utilizan elementos de datos diferentes. Por ejemplo, en lugar de leer un solo correo electrónico en cada ejecución, la automatización puede recuperar varios correos electrónicos y recorrerlos en iteración realizando los mismos pasos.

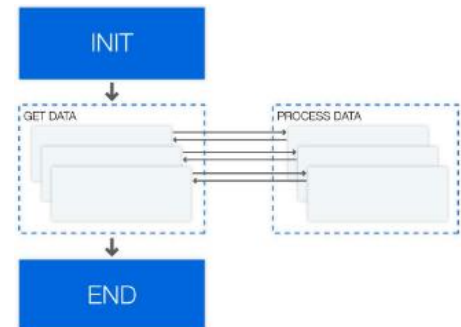
Este tipo de proceso se puede implementar con un bucle simple, pero tiene la desventaja de que, si se produce un problema al procesar un elemento, se interrumpe todo el proceso y los demás elementos permanecen sin procesar.



3. Transaccional:

De forma similar a los procesos iterativos, los pasos de los procesos transaccionales se repiten varias veces en diferentes elementos de datos. Sin embargo, la automatización está diseñada para que cada pieza repetible se procese de forma independiente.

Estas partes repetibles se denominan transacciones. Las transacciones son independientes entre sí porque no comparten ningún dato ni tienen ninguna orden en particular para procesar.



REFRAMEWORK

En términos generales, un marco es una plantilla que le ayuda a diseñar procesos de automatización. Como mínimo, un marco debe ofrecer una manera de almacenar, leer y modificar fácilmente los datos de configuración del proyecto, un esquema de control de excepciones sólido, registro de eventos para todas las excepciones e información de transacciones relevante.

REFramework se implementa como una máquina de estados, que es un tipo de flujo de trabajo que tiene dos características muy útiles:

- Estados que definen las acciones a realizar de acuerdo con la entrada especificada
- Transiciones que mueven la ejecución entre estados dependiendo de los resultados de los propios estados.

REFramework tiene 4 estados principales que suelen ser comunes para los procesos empresariales:

1. **Estado inicial:** Aquí es donde comienza el proceso. Es una operación en la que el proceso inicializa la configuración y realiza comprobaciones de la aplicación para asegurarse de que se cumplen todos los requisitos previos para el inicio del proceso.
2. **Estado Obtener datos de la transacción:** Obtiene el siguiente elemento de transacción. Esto puede ser un artículo en cola o cualquier elemento de una colección. De forma predeterminada, los elementos de transacción son artículos en cola, pero esto se puede cambiar fácilmente para adaptarse a sus necesidades. Este es también el estado en el que el desarrollador debe configurar la condición para salir de este estado cuando no hay elementos para procesar.
3. **Estado Procesar transacción:** Realiza acciones/aplica lógicas en varias aplicaciones para el elemento de transacción obtenido en el paso anterior. Una vez que se procesa un elemento de transacción, el proceso continúa con el siguiente elemento de transacción disponible.
4. **Estado Finalizar proceso:** El proceso finaliza (y las aplicaciones abiertas durante la automatización deberían cerrarse correctamente).

EXCEPCIÓN EMPRESARIAL Y DE APLICACIÓN

Si la causa del problema se puede solucionar **reiniciando las aplicaciones**, el marco lo hace automáticamente e **intenta procesar de nuevo la misma transacción**. Estas excepciones se denominan ***excepciones de aplicación***.

Si el problema está relacionado con los datos en sí o con un requisito empresarial subyacente (por ejemplo, no procesar facturas con un valor superior a un importe determinado), esa transacción **se omite y el marco pasa a la siguiente transacción**. Estas excepciones se denominan ***excepciones de reglas empresariales***.

DISTRIBUIDOR Y EJECUTOR

Distribuidor

El distribuidor es un proceso que se utiliza para insertar elementos de transacción en una cola de Orchestrator. Extrae datos de uno o varios orígenes y los utiliza para crear elementos de cola para que los procesen los robots Ejecutores.

La información se inserta en una o varias colas, lo que permite al distribuidor utilizar un formato común para todos los datos almacenados en elementos de cola.

La principal ventaja de usar un patrón de distribuidor es que puede dividir el procesamiento de los elementos entre varios robots.

Ejecutor

El ejecutor es un proceso que se utiliza para extraer elementos de transacción de una cola de Orchestrator y procesarlos según sea necesario en la empresa. Los elementos de cola se procesan de uno en uno.

Utiliza mecanismos de gestión de errores y reintentos para cada elemento procesado.

Una ventaja importante del ejecutor es su escalabilidad (se pueden utilizar varios ejecutores con una sola cola)

Ventajas del modelo Distribuidor y Ejecutor:

1. Mejor separación de procesos (entre Distribuidor y Ejecutor)
2. Mejor separación y distinción entre arquitectura y capas de proceso
3. Mejor mecanismo de gestión de errores y reintentos
4. Posibilidad de ejecutar procesos en varias máquinas (disponibilidad)
5. Mejor reutilización dentro de los componentes creados por su proyecto
6. Configuración integrada mejorada e integración en Orchestrator
7. Los flujos de trabajo anteriores creados sin REFramework se pueden adaptar e implementar fácilmente para utilizar REFramework y el modelo Distribuidor/Ejecutor