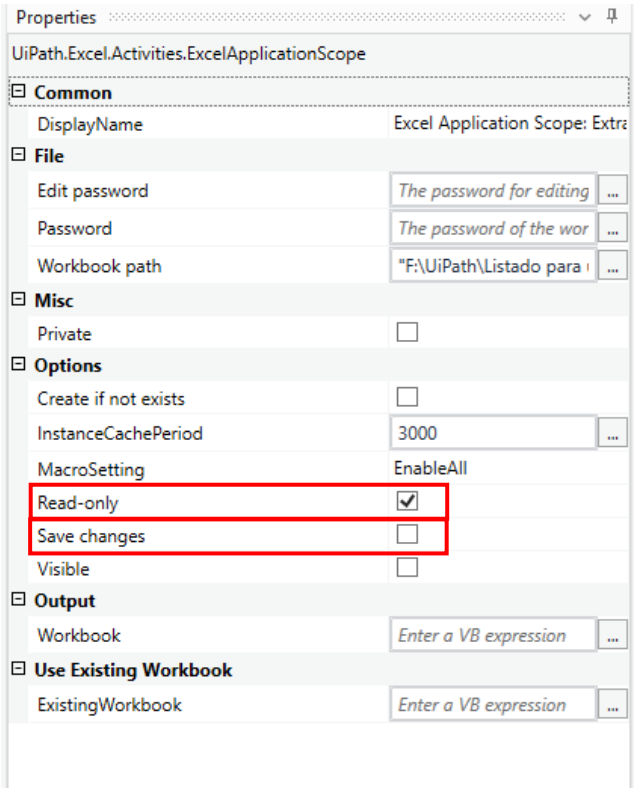
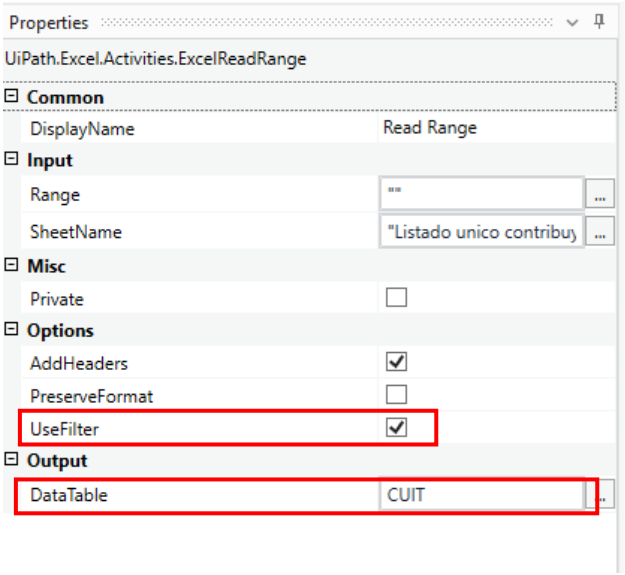
Resumen de Actividades más usadas en UiPath

Abrir excels que no se quieren modificar



Solo tildar el **Read-only** y destildar **Save Changes.** Se puede también destildar la opción de Visible para no abrir una ventana de Excel.

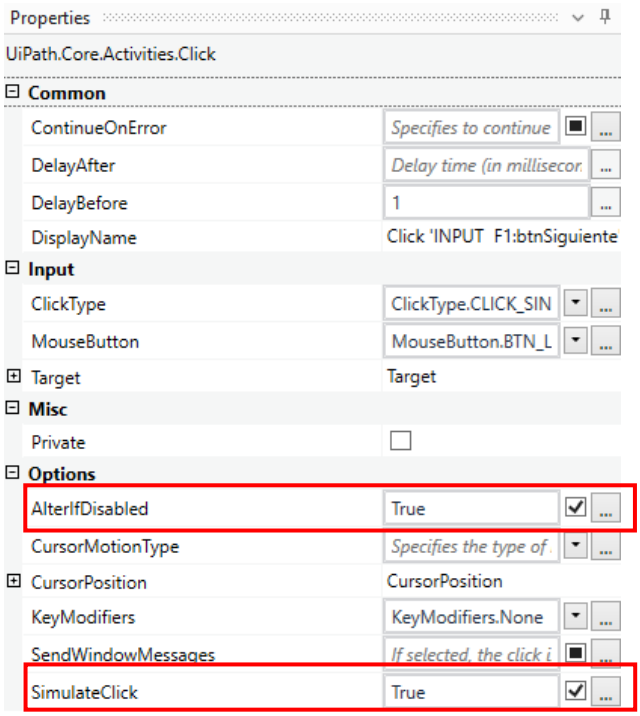
Usar filtro en Excels leídos



En propiedades tildar el **UseFilter** y **AddHeaders**

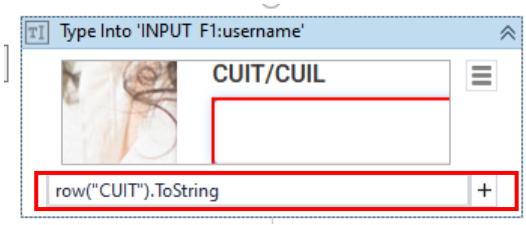
Asignar la base de datos a una variable (con **control + K**) o escribir el nombre de la variable creada previamente

Permitir clicks en segundo plano y habilitar click en botones deshabilitados



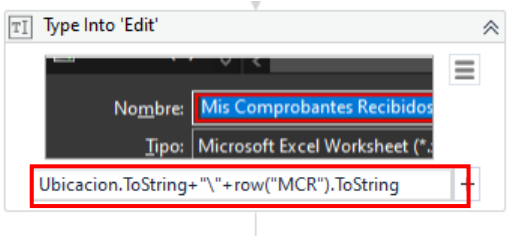
En propiedades tildar el **SimulateClick** y en **AlterIfDisabled**

Tipear Texto que vienen de columnas de una base de datos



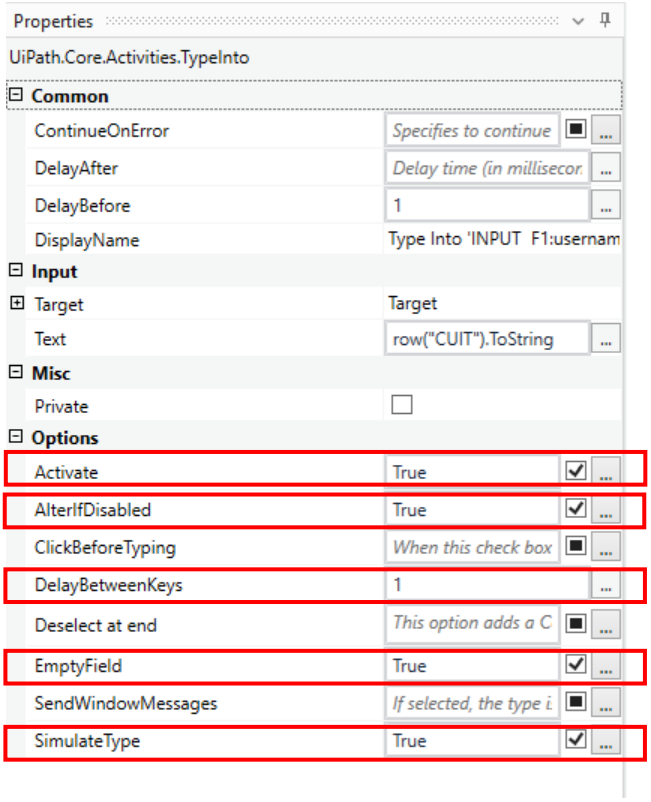
Escribir **Row(“Nombre de la columna”).ToString**

Concatenado para una ubicación



Utilizar la **variable de la ubicación** mas el **.ToString** agregar un **“\”** y agregar la **columna** con la que se quiere renombrar al archivo **.ToString** . También se puede agregar un **+”.pdf”** o **+”.xlsx”** si se quiere guardar el archivo como un PDF o un Archivo de Excel.

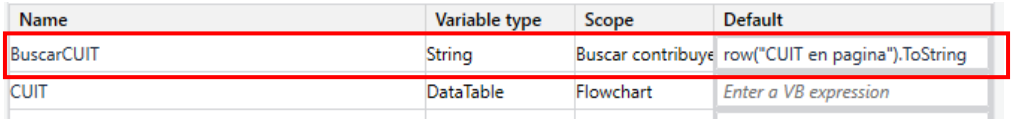
Permitir Type into en segundo plano y habilitar rango en botones deshabilitados



Para evitar errores en propiedades tildar en **Activate**, **Alter If Disabled**, **Delay Between Keys 1**, **Empty Field** y **Simulate Keys**

Seleccionar/Clickear un item basado en una variable

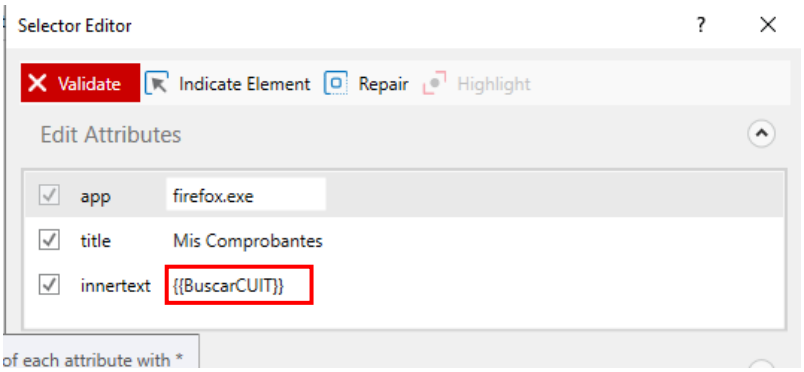
1. Crear una variable



Name	Variable type	Scope	Default
BuscarCUIT	String	Buscar contribuye	row("CUIT en pagina").ToString
CUIT	DataTable	Flowchart	*Enter a VB expression*

Crear una variable con la expresión **row(“Nombre de columna que posee el dato que se quiera buscar”).ToString** y disminuir el nivel del **Scope** al proceso en el que se quiere utilizar

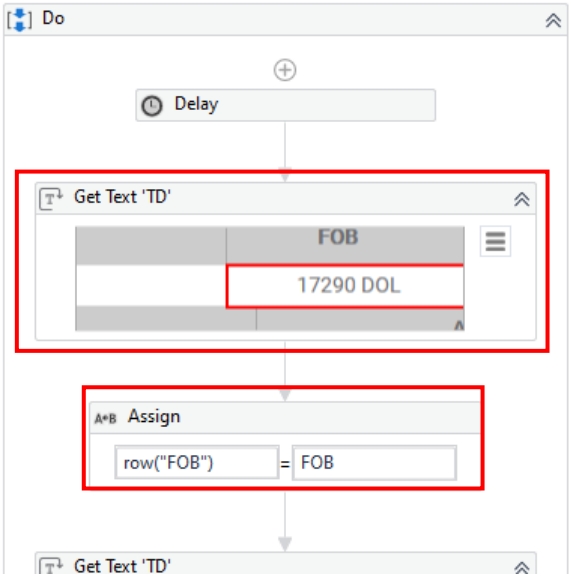
2. Editar el selector



En el Selector elegir la **variable** con click derecho (automáticamente se pone entre **{}**)

También se puede agregar ***** (**asteriscos**) antes y después para omitir el texto que hay antes y después

Obtener texto y asignarlo a una columna

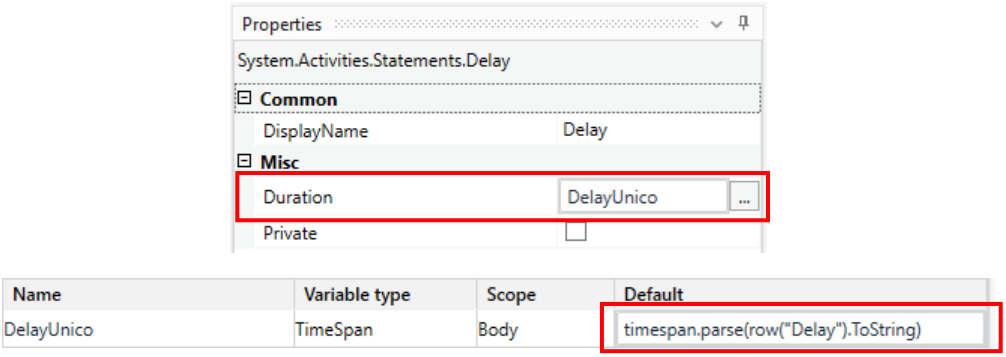


Utilizar la actividad **Get Text** y asignarlo a una columna con la actividad **Assign** con **row(“Nombre de la columna a asignar”)**.

Luego hay que utilizar la actividad **Write Range** para que esos datos extraídos se vuelquen a un Excel

Delay con tiempo basado en una columna

En vez de elegir el tiempo se lo debe asignar a una variable



Utilizar expresión **timespan.parse(row("Nombre de la columna que posee el tiempo").ToString)**

El tiempo en el Excel tiene que tener el formato 00:00:00 (**HH:MM:SS**)

Copiar/Pegar un mismo rango en varios Excels (No se me ocurrió todavía pero seria interesante hacerlo, por ejemplo agregar el mismo rango de un Excel a otro Rango para automatizar un papel de trabajo) Se me ocurre hacerlo con un Read range → for each file in a folder → Write range in each file

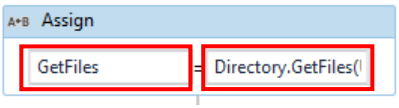
1. Copiar el rango de un Excel y asignarlo a una Base de Datos

2. Obtener la dirección de los archivos a los cuales se les va a pegar el rango

3. Pegar los Rangos de la Base de Datos en todos los archivos

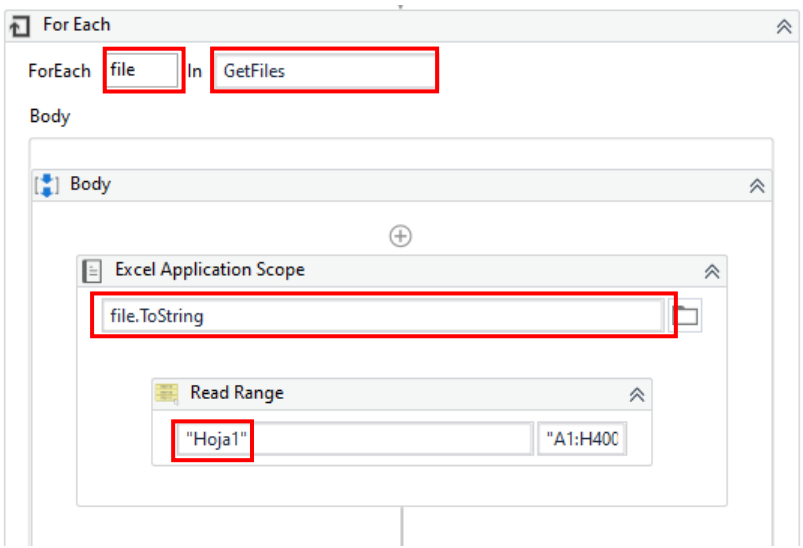
Copiar/Pegar Varios rangos a un mismo Excel y ejecutar una macro

1. Obtener la dirección de todos los archivos de una carpeta



Utilizar la actividad **Assign** con las expresiones **Get Files** y **Directory.GetFiles(“Ubicación”)**

2. Crear un loop con los Archivos Base y el Papel de Trabajo

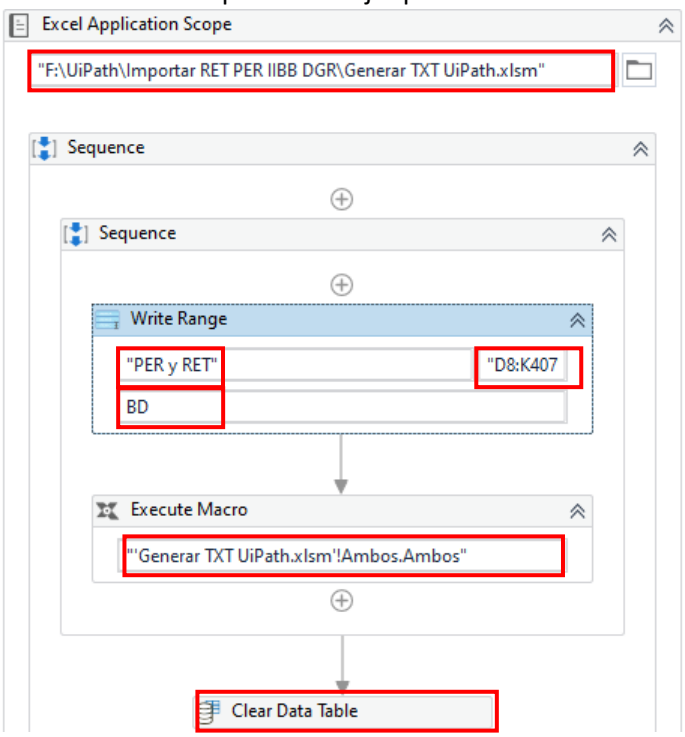


Utilizar la actividad **For Each** con las expresiones **file** y **GetFiles**.

Para que seleccione los excels de los cuales extraer la información utilizar la actividad **Excel Application Scope** y en el archivo seleccionado utilizar la expresión **file.ToString.** (en propiedades tildar solo en **Read-only** para no modificar los archivos base).

En la actividad **Read Range** Seleccionar la **hoja** de la cual se va a extraer la información. (en propiedades asegurarse que **no esté tildado PreserveFormat** para mejorar el rendimiento del robot, además de **destildar** las opciones de **AddHeaders** y **UseFilter**) y de **Output** tiene que definirse una **BaseDeDatos**

3. Pegar los Archivos base en el Papel de trabajo que contiene macros



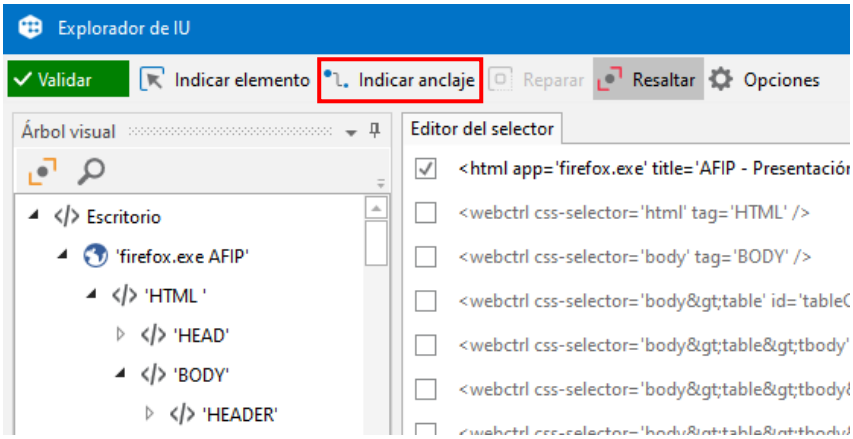
Utilizar la actividad **Excel Application Scope** y en el archivo seleccionado utilizar el **Papel de Trabajo con Macros integradas.** (en propiedades asegurarse que en **MacroSetting** este **EnableAll** y que estén **destildados: Read-only** (para poder "editar" dentro del papel de trabajo), **Save Changes** (para no modificar el Papel de Trabajo) , y **visible** (para que no se abra una ventana de Excel))**.**

En la actividad **Write Range** hay que definir la **hoja** del Papel de Trabajo donde se va a pegar la info de los Archivos Base, el **Rango** donde se va a pegar y la **BaseDeDatos** definida en el paso anterior. (en propiedades destildar la opción **AddHeaders**).

En la actividad **Excecute Macro** definir la macro a ejecutar (lo ideal es que la macro no abra ventanas y devuelva mensajes).

Luego de ejecutar las Macros utilizar la Actividad **Clear Data Table** para Reiniciar la **BaseDeDatos.** (en propiedades hay que seleccionar la Tabla de datos a eliminar)

Selector con anclaje en otra columna

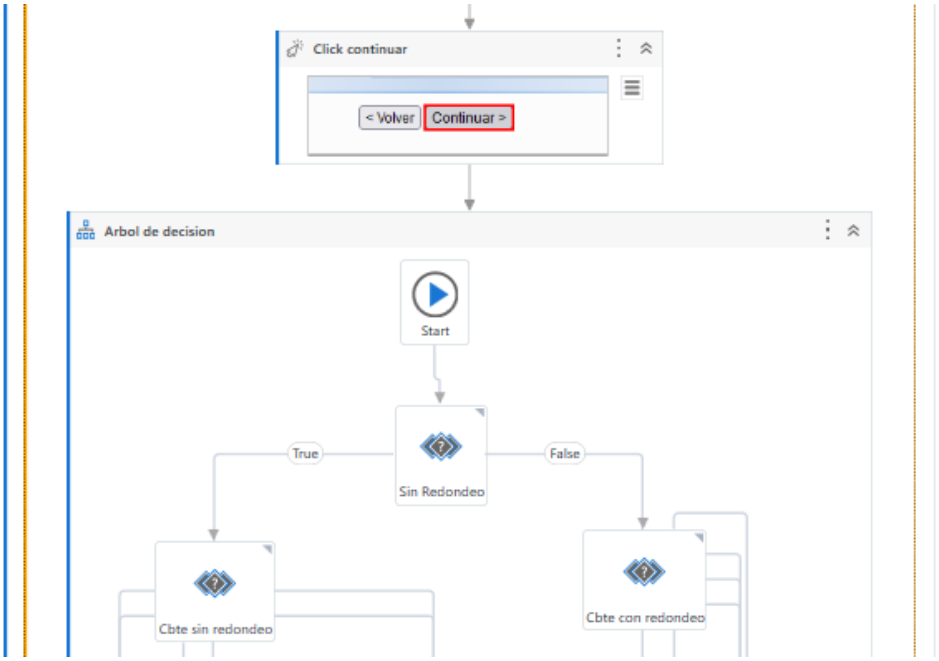



```
<webctrl isleaf='1' tableCol='3' tag='TD' innertext='*GANANCIAS*' />
```

Se debe indicar un anclaje y luego analizar la línea agregada, por ejemplo en este caso sabemos que la columna a buscar en es en la 3 y el texto interno contiene GANANCIAS (en este caso podemos poner lo que queremos o hasta relacionarlo con una variable, lo cual se puede hacer con {{}} o con click derecho en el selector.

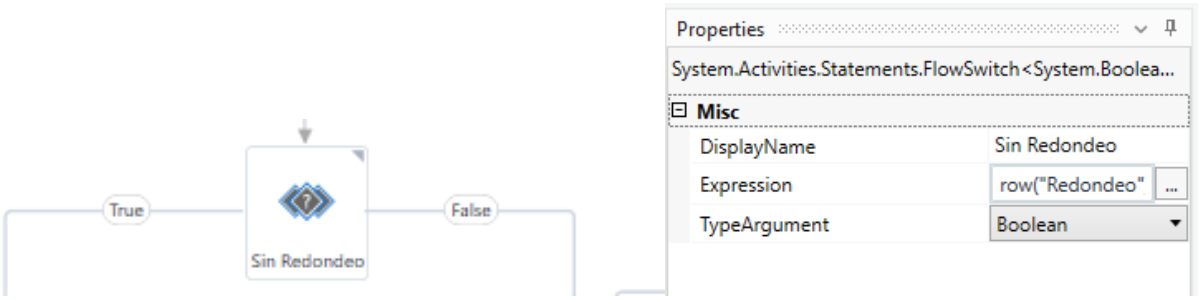
Diagrama de Decisiones (flowswitch)

Para crear una diagrama de decisiones se debe utilizar la actividad Flowswitch. La misma solo puede ser utilizada dentro de un Flowchart, es decir si tenemos una secuencia y dentro de una secuencia queremos hacer un diagrama de decisiones primero tenemos que agregar un Flowchart dentro de la Sequence y dentro del Flowchart iría el Flowswitch. Ejemplo a continuación:

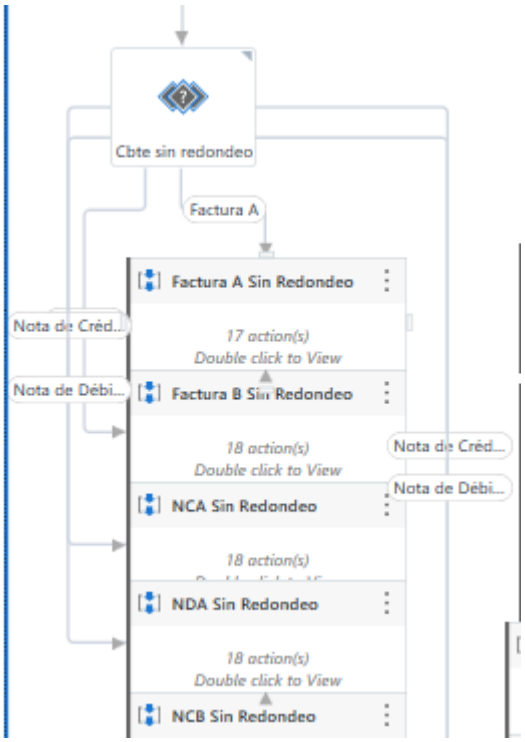


Dentro de las propiedades podemos asignar distintos tipos de Flowswitch. Pero siempre hay que definir una condición:

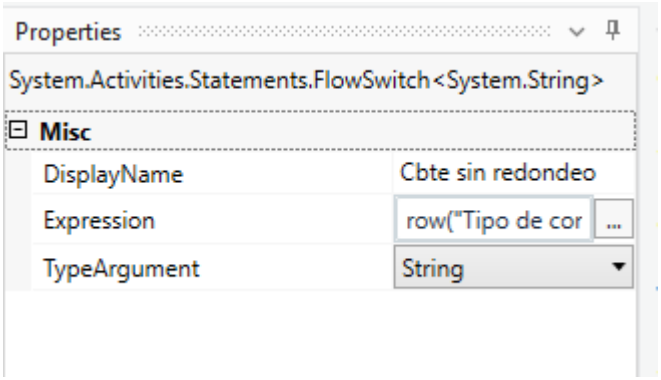
Ejemplo de una condición booleana [row("Redondeo").ToString = "0.00"], si se cumple la considicon tendría dos posibles caminos, que sea Verdadeo o Falso, además lo defino en el tipo de argumento



Ejemplo con múltiples opciones row("Tipo de comprobante").ToString, es decir que me convierte al dato de una columna en Tipo String. Para este caso debemos crear todos los pasos para cada uno de los posibles outputs de nuestra condición.



Propiedades de Actividad



Propiedades del Caso (se pueden definir casos por default, es decir si no se cumple la condición)