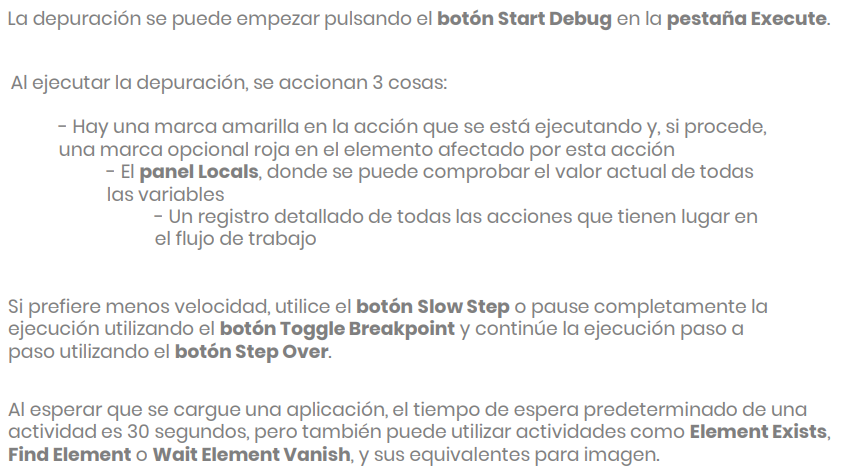
El mensaje “Object reference not set to an instance of an object al ejecutar Assign.

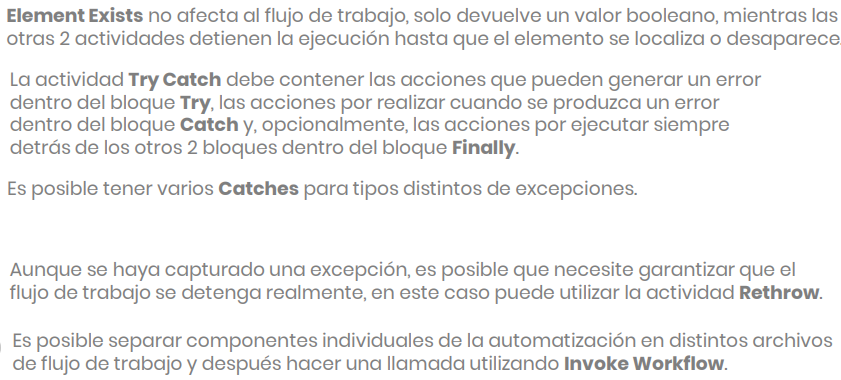
         ○      Se debe a que la variable *index* no está inicializada. Los tipos **Int32** y **String** se crean con valores predeterminados aunque no se indiquen de forma explícita, pero una variable de tipo **GenericValue** no se crea de forma implícita porque no es posible saber si contendrá un Integer (con valor predeterminado 0), un String (con valor predeterminado “”), etc.

        ○      Puede cambiar el tipo de variable a Int32, o definir Default value a 0.

El mensaje “Index was outside the bounds of the array.” (índice fuera de los límites de la matriz).

* + Se debe a que en ese momento, el valor de la variable de índice es 5, es decir, intenta acceder al índice/posición número 5 de filesArray y en este caso no existe. La matriz tiene 5 elementos, desde el índice 0 al índice 4. El índice 5 queda fuera de los límites de la matriz.
* Si comprobamos los registros, veremos una contradicción entre los mensajes del archivo 5.txt, uno que indica que el contenido no se añadirá y otro que dice que ha finalizado la anexión.
* Además, en los requisitos se indica que si falla la lectura del archivo (como en el caso del 5), la línea **Append** no se debe ejecutar.
* En tercer lugar, si analizamos el archivo combinado, el contenido es incorrecto porque hay varias líneas duplicadas.
  + Esto se debe a que la variable de texto tiene como ámbito **While**, por eso se mantiene de una iteración a otra. Esto significa que, si no se le asigna el valor del texto del archivo a causa de la excepción **Read Text File** (emitida por 5.txt), mantendrá el valor anterior (en este caso, el texto de 4.txt).
  + Esto se soluciona fácilmente cambiando el ámbito de la variable text de **While** a **Process** para un archivo, pero soluciona solo una parte del problema y sigue sin cumplir el requisito. De hecho, si solo cambiamos el ámbito, el robot escribirá el valor de text predeterminado, “DEFAULT\_TEXT” en la última línea.
* La forma más sencilla de arreglar la función es colocando el contenido de los dos bloques **Try Catch** en uno solo.
  + De este modo, si falla la primera actividad, la ejecución salta al bloque **Catch** de manejo de excepciones, y la segunda actividad ya no se vuelve a ejecutar.
  + Arrastre y suelte las actividades **Append Line** y **Log Message** al primer bloque **Try**, justo detrás de **Read Text File**. **Append Line** depende de **Read Text File** y por eso tienen que estar dentro del mismo bloque **Try Catch** para que funcione correctamente.
  + Elimine el segundo bloque **Try Catch**.





***¿Dónde se puede buscar más información acerca de este error que aparece cuando se carga un flujo de trabajo: “Activity could not be loaded because of errors in the XAML” (la actividad no se ha podido cargar por errores de XAML)? (Seleccione todas las respuestas que correspondan).***



En el panel Output



En el panel Outline



En el panel Properties



En el panel Library

**Single choice**

2)

***Si queremos saber si un elemento UI está disponible en la pantalla o no, ¿qué actividad debemos utilizar?***



Find Element



Wait Element Vanish



Element Exists

**Multiple Choice**

3)

***¿Es posible ejecutar el robot manualmente, paso a paso, para analizar el comportamiento del robot en determinadas condiciones?***

No, no es posible hacer esto

.Sí, utilizando Step Into y Step Over

.Sí, utilizando puntos de interrupción y ejecutando el flujo de trabajo normalmente

.Sí, utilizando puntos de interrupción y ejecutando el flujo de trabajo en modo Debug.

**Single choice**

4)

***¿Qué muestra el panel Locals cuando trabajamos en modo Debug?***



Los valores actuales de las variables.



No existe el panel Locals.



Los registros del flujo de trabajo.



Todas las actividades dentro del flujo de trabajo.

**Single choice**

5)

**Si queremos detener el flujo de trabajo hasta que un elemento UI haya desaparecido de la pantalla, ¿qué actividad debemos utilizar?**



Find Relative Element



Element Exists



Wait Element Vanish



Find Element

100%