**Enunciados ejercicios vistos en clase**

**Ejercicio 1**

1. Para ejemplificar cómo utilizar un argumento en una automatización con una actividad **Invoke Workflow File**, se van a crear dos secuencias separadas. Una muy sencilla en la que se asigna un valor a un argumento, y una segunda que lo invoca y muestra el valor en un Message Box.
2. Crear una nueva secuencia.
3. Dentro del panel **Argumentos**, crea un argumento, out\_StoreValue.
4. Desde la lista **Dirección**, selecciona In/Out y no cambies el **Tipo de argumento** de **Cadena**.
5. Agrega una actividad de **Assign** en el panel del **Diseñador**.
6. En el panel **Propiedades**, en el campo **To**, agrega el argumento out\_StoreValue.
7. En el campo **Value**, escribe una cadena como, por ejemplo, "¿Cómo se utiliza un argumento?".
8. Crea una nueva secuencia.
9. Crea una variable de tipo cadena, FinalValue.
10. Agrega una actividad **Invoke Workflow File** al panel **Diseñador**.
11. En la actividad, haz clic en el botón **Examinar (...)** y busca la secuencia creada anteriormente.
12. Haz clic en **Importar argumentos**. Se muestra la ventana **Argumentos de flujo de trabajo invocado**. Observa que aquí se muestra el argumento de la primera secuencia.
13. En el campo **Value**, agrega variable FinalValue y haz clic en **Aceptar**. El argumento es importado y el valor del mismo es guardado en el proyecto actual a través de la variable FinalValue.
14. Agrega una actividad de **Message box** en la sección **Invoke Workflow File.**
15. En el panel **Propiedades** en el **Texto**, escribe FinalValue + por ejemplo, "De esta manera".  
    Presiona Ctrl + F6 en la segunda secuencia. La automatización se ejecuta de forma correcta y el cuadro de mensajes muestra el texto deseado.
16. Empezar un nuevo proyecto y crear una secuencia inicial.
17. Crear una variable de tipo String llamada nombre.
18. Crear una variable de tipo Int32 llamada edad.
19. Crear una actividad en UiPath llamada “Input Dialog” para Ingresar el nombre del usuario.
20. Almacenar en la variable nombre el resultado.
21. Crear una actividad en UiPath llamada “Input Dialog” para ingresar la edad.
22. Almacenar en la variable edad el resultado.
23. Crear una nueva secuencia individual que se llame “ImprimirResultados”
24. Crear dos argumentos en la secuencia “ImprimirResultados” (Donde posteriormente vamos a guardar los valores de nombre y edad)
25. En la secuencia principal, crear la actividad “Invoke Workflow File” y llamar a la secuencia “imprimirResultados”
26. Importar argumentos. Pasar los valores de nombre y edad a los argumentos de la secuencia “ImprimirResultados”
27. En la secuencia ImprimirResultados, imprimir los valores de los argumentos.

**Ejercicio 2**

Vamos a crear una automatización que pregunte al usuario tanto su nombre y apellido como su edad, almacene la información en una matriz y luego mostrarla en un message box.

1. Crea una nueva secuencia.
2. Crea tres variables de cadena, FirstName, LastName y Age, en las que se almacene la información obtenida del usuario.
3. Crea una variable de matriz de cadenas llamada NameAge.
4. Agrega una actividad de **Input Dialog** al panel **Diseñador**.
5. En el panel de **Propiedades**, rellena los campos **Etiqueta** y **Título** para solicitar el nombre del usuario.
6. En el campo **Resultado**, escribe la variable FirstName. Esta variable almacena el nombre del usuario.
7. Agregamos dos actividades Input Dialog más, una para solicitarle al usuario el apellido y otra para la edad.

### 📘

**Nota:**

Usamos una variable de cadena y no un entero para almacenar la edad, de manera que no tengamos que convertirla después, cuando la agreguemos a la variable de matriz de cadenas.

1. Agrega una actividad **Assign**.
2. En el panel de **Propiedades**, en el campo **To**, escribe la variable NameAge.
3. En el campo **Value**, escribe {FirstName,LastName,Age}. Esta actividad **Assign** permite almacenar todos los valores de las variables de cadena iniciales en la NameAge.
4. Agrega una actividad de **Write Text File**.
5. En el panel **Propiedades**, en el campo **FileName**, escribe la ruta del archivo en el que quieres escribir entre comillas, como "%HOMEPATH%\Desktop\array\_variable.txt".

### 📘

**Nota:**

Si el archivo no existe en la ruta proporcionada, se crea.

Les recomendamos crear una variable para almacenar la ruta del archivo.

1. En el campo de **Texto**, escribe NameAge(0) + " " + NameAge(1) + " " +NameAge(2) + " ".

### 📘

**Nota:**

Agregando el número de índice de los elementos del array es posible acceder a sus valores y escribirlos, en este ejemplo, en un archivo de texto.

1. Presiona Ctrl + F5 para ejecutar el proyecto.
2. Navega hasta el archivo proporcionado en el paso 17 y haz doble clic en él. Se muestra una ventana del **Bloc de notas** con la información que has agregado en el paso 20.

**Ejercicio 3**

Vamos a crear una automatización que pida al usuario dos números, compruebe si uno es divisible por el otro y, en función del resultado, muestre un mensaje diferente en el panel de **Salida**.

1. Crea una nueva secuencia.
2. Crea dos variables enteras, FirstNumber y SecondNumber, por ejemplo.
3. Agrega dos actividades de **Input Dialog** al panel **Diseñador**.
4. En el panel de **Propiedades**, escribe las etiquetas y los títulos de ambas actividades y, en los campos de **Resultados**, agrega las variables FirstNumber y SecondNumber.
5. Agrega una actividad **IF** al panel **Diseñador**, en los **Diálogos de entrada** previamente agregados.
6. En la sección **Condición**, escribe FirstNumber mod SecondNumber = 0. En esta expresión se comprueba si el primer número es divisible por el segundo, utilizando el operador [mod](https://msdn.microsoft.com/en-us/library/se0w9esz.aspx).
7. En la sección **Then**, agrega una actividad [**Write**](https://activities.uipath.com/docs/write-line) **Line**.
8. En el campo de **Texto**, escribe SecondNumber.ToString + " es divisible por "+ FirstNumber.ToString + ".". Este es el mensaje que se muestra si el primer número es divisible por el segundo.
9. Haz clic en **Mostrar Else** para mostrar la sección Else y entonces añadir otra actividad **Escribir línea** en la sección **Else**.
10. En el campo de **Texto**, escribe FirstNumber.ToString + " NO es divisible por "+ SecondNumber.ToString + ".". Este es el mensaje que se muestra si el primer número no es divisible por el segundo.

**Ejercicio 4**

Vamos a crear una automatización que incremente una variable entera de 0 a 10, y muestre un mensaje cada vez que se incremente.

1. Crea una nueva secuencia.
2. Crea una variable entera, Counter, con un valor por defecto de 0.
3. Agrega una actividad **Do While** al panel **Diseñador**.
4. En la sección **Body**, agrega una actividad de **Assign**.
5. Selecciona la actividad **Assign**, y en el panel **Propiedades**, en el campo **To**, agrega la variable Counter.
6. En el campo **Value**, escribe Counter + 1. Esto te ayuda a incrementar el valor de la variable con uno.
7. Agrega una actividad de [**Write**](https://activities.uipath.com/docs/write-line) **line** debajo de la **Assign**.
8. En el campo **Texto**, escribe "El contador es " + Counter.ToString. Se escribe el valor del contador en el panel de **Salida** cada vez que se incrementa.
9. En la sección **Condición** de la actividad **Do while**, escribe Counter < 10. El cuerpo de la actividad **Do while** se repite hasta que el valor de la variable Counter sea mayor que 10.

**Ejercicio 5**

1. Crear un array (ArrayNumeros)
2. Poblar la lista con un assign
3. Agregar for each y en el body un message box que me mueste cada item de la lista

**Ejercicio 6**

Vamos a crear una automatización que pide al usuario un número, comprueba si es par o impar, y en función de ello, se escribe un mensaje diferente en el panel de **Salida**.

En vista de que todos los números impares divididos por dos tienen un resto igual a 1, este proyecto solo necesita dos casos (0 y 1) , pero ten en cuenta que esta actividad admite múltiples casos.

1. Crea una nueva secuencia.
2. Crea una variable numérica, Number.
3. Agrega una actividad de **Input Dialog** al panel **Diseñador**.
4. Agrega un **Título** y una **Etiqueta** para solicitar al usuario un número.
5. En el campo **Resultado**, agrega la variable Number.
6. Se agrega una actividad de **Switch**, en el **Diálogo de entrada**.
7. En el campo **Expresión**, escribe Number mod 2. Esto verifica si el número del usuario es divisible por 2.
8. En la sección de **Predeterminados**, agrega una actividad de **WriteLine**.
9. En el campo **Texto**, escribe Number.ToString + " es un número par".
10. Haz clic en la línea **Agrega nuevo caso**, y en el campo **Valor del caso**, escribe 1.
11. Agrega una actividad **Escribir línea** a este caso.
12. En la actividad **Texto**, escribe Number.ToString + " es un número impar".

**Ejercicio 7**

1. Empezar un nuevo proyecto y crear una secuencia inicial.
2. Crear una variable de tipo Array, la cual contendrá un arreglo inicial, y asignarle valores numéricos (estos valores van separados por coma, y todos dentro de un solo par de corchetes).
3. Crear 3 variables de tipo Int32, las cuales contendrán el valor mínimo, el valor máximo y un valor "index", el cual corresponderá a la posición de cada elemento dentro del arreglo, y se usará para realizar las comparaciones entre los elementos.
4. A las variables que contendrán el valor mínimo y máximo se le asigna el primer elemento del arreglo (ya que se usará para comparar el resto de los valores inicialmente).
5. Con la actividad "For Each" ("Para cada") se recorren los elementos del arreglo, la cual por default llama "item" a cada elemento del arreglo. El tipo de argumento que debe aceptar dicha actividad debe ser "Int32" y su salida será la variable "index".
6. Dentro del For Each se debe ir evaluando cada elemento. Si el elemento que se está evaluando es menor al valor de la variable que contiene el mínimo, se asignará a la variable creada para el mínimo, actualizando así el valor de dicha variable. Se hace lo mismo para el valor máximo.
7. Se imprime el valor mínimo y máximo del arreglo, ya sea a través de un "Log message" o "Message box". Lo ideal es probar ambos para entender cómo trabaja cada uno.

Recordar:

* Los nombres de las variables deben ser descriptivos, y deben comenzar con el tipo de dato que acepta dicha variable. Ejemplo, si se crea una variable que acepte valores del tipo string, el nombre debería comenzar con "str\_".
* El primer elemento de un arreglo siempre se identifica con 0, y se define: arr\_NombreVariable(0).