

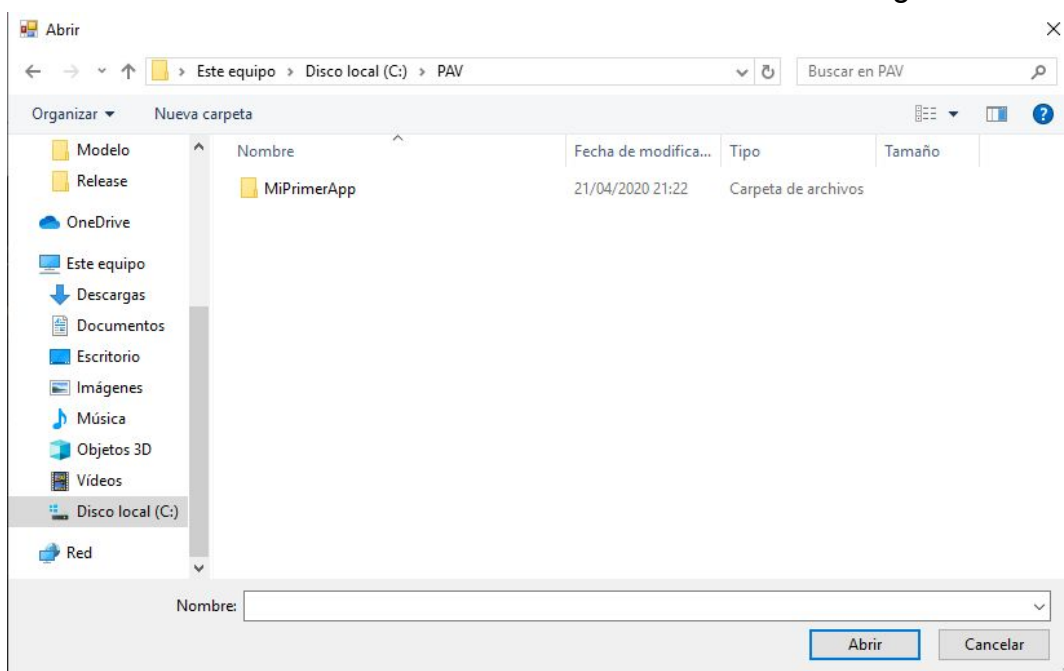


## UNIDAD N° 2

### EJERCICIO N° 7

#### ESCRIBIR UNA SENTENCIA WHILE

1. Descargar el ejemplo **WhileStatement** de:  
<https://frtutneduar.sharepoint.com/:u:/r/sites/PAV2020/Documentos%20comp%20artidos/Unidad%202/Ejercicios/Codigo/WhileStatement.zip?csf=1&web=1&e=jI5WXL>
2. Iniciar Visual Studio 2017
3. Ir al menú Archivo → Abrir → Proyecto / Solución
4. Ubicar la carpeta donde se descargó el ejemplo
5. Seleccionar la solución WhileStatement y elegir Abrir.
6. Ir al menú Depurar → Iniciar Depuración  
La aplicación es un simple visualizador del contenido de un archivo.
7. Presionar Abrir Archivo. Se mostrará una ventana como la siguiente:



8. Navegar hasta la carpeta WhileStatement de la solución.
9. Seleccionar el archivo Principal.cs y presionar Abrir.

El nombre del archivo aparecerá en el cuadro de texto a la par del botón, pero el contenido del archivo no aparecerá en el cuadro de texto inferior. En los próximos pasos se agrega la funcionalidad para esto.

10. Cerrar la venta y regresar a Visual Studio.
11. Visualizar el código del formulario *Principal.cs* (hacer click con botón derecho y elegir Visualizar Código o utilizar el ícono en la barra de herramientas del Explorador de Soluciones). Ubicar el método *openFileClick*.

Este método se ejecuta cuando el usuario hace click en el botón Abrir Archivo para seleccionar un archivo en el cuadro de diálogo Abrir. No es necesario comprender los detalles exactos de cómo funciona este método en este momento: simplemente aceptar el hecho de que este método solicita al usuario un archivo y abre el archivo seleccionado para leerlo (el método simplemente muestra la ventana *OpenFileDialog*, y cuando el usuario selecciona un archivo, se ejecuta el método *openFileDialogFileOk*, por lo que en realidad es este método el que abre el archivo para leerlo).

Sin embargo, las dos últimas sentencias en el método *openFileDialogFileOk* son importantes:

```
StreamReader reader = src.OpenText();  
displayData(reader);
```

La primera sentencia declara una variable de *StreamReader* llamada *reader*. *StreamReader* es una clase proporcionada por el Framework de Microsoft.NET que se puede usar para leer secuencias de caracteres de fuentes como archivos. Se encuentra en el espacio de nombres (namespace) *System.IO*. Esta declaración hace que los datos en el archivo especificado por el usuario en el cuadro de diálogo *OpenFileDialog* estén disponibles para el objeto *StreamReader*, que luego puede usarse para leer los datos del archivo. La declaración final llama al método *displayData*, pasando *reader* como parámetro. El método *displayData* lee los datos utilizando el objeto *reader* y los muestra en la pantalla (o lo hará una vez que se haya escrito el código para lograr esto).

La variable *src* es del tipo *FileInfo*, otra clase del Framework de Microsoft.NET, ésta se carga con la información del archivo seleccionado durante la apertura. La clase *FileInfo* también provee el método *OpenText* que realiza la apertura del archivo para lectura. La variable *reader* se encargará de proveer el contenido del archivo.

12. Examinar el contenido del método *displayData*:

```
private void displayData(StreamReader reader)
{
    // TODO: add while loop here
}
```

Se puede ver que el método solo tiene un comentario.

13. Reemplazar el comentario //TODO...por:

```
txtOrigen.Text = "";
```

La variable *txtOrigen* se refiere al cuadro de texto grande en el formulario. Al establecer la propiedad *Text* en una cadena vacía ("") se borra el contenido de éste.

14. Agregar la siguiente sentencia:

```
string line = reader.ReadLine();
```

Esta sentencia declara una variable de cadena llamada *line* y llama al método *ReadLine* de *reader* para leer la primera línea del archivo en esta variable. Este método devuelve la siguiente línea de texto del archivo o un valor especial *null* (nulo) cuando no hay más líneas para leer.

15. A continuación agregar las siguientes líneas:

```
while (line != null)
{
    txtOrigen.Text += line + '\n';
    line = reader.ReadLine();
}
```

El *while* itera a través del archivo línea por línea hasta que no haya más líneas disponibles. La expresión booleana al comienzo del ciclo *while* examina el valor en la variable *line*, si no es nulo, el cuerpo del bucle muestra la línea de texto actual y la agrega a la propiedad *Text* del cuadro de texto *txtOrigen*, junto con un carácter de nueva línea ('\n': el método *ReadLine* del objeto *TextReader* elimina los caracteres de nueva línea a medida que lee cada línea, por lo que el código debe volver a agregarla). El ciclo *while* lee la siguiente línea de texto antes de realizar la siguiente iteración. El ciclo finaliza cuando no hay más texto para leer en el archivo y el método *ReadLine* devuelve *null*.

16. Al final agregar la sentencia:

```
reader.Close();
```

Esta sentencia libera los recursos asociados a la apertura del archivo y lo cierra. Esta es una buena práctica porque permite que otras aplicaciones utilicen el archivo y también libera memoria y otros recursos utilizados para acceder al archivo.

17. Ir al menú Depurar → Iniciar Depuración
18. Hacer click en Abrir Archivo.
19. Navegar hasta la carpeta WhileStatement de la solución.
20. Seleccionar el archivo Principal.cs y presionar Abrir.
21. Verificar que se agregó el contenido del archivo en el cuadro de texto inferior.
22. Retornar a Visual Studio y detener la ejecución o cerrar la ventana.