

PROGRAMACIÓN I

TP 9 ARCHIVOS

UNIDAD 4

ARCHIVOS

Autor de contenidos:
Nicolás Battaglia



OBJETIVOS

Realizar programas utilizando archivos

ENUNCIADO

Resuelva el TP en base a completar la Parte A y Parte B.

PARTE A

Realice los siguientes laboratorios

Laboratorio Archivos #2

La propuesta de este trabajo es guiarlo/a en la generación de un archivo de texto que posea como operación dar de Alta, de Baja, Modificar y Listar los registros del mismo.

A continuación, las consignas:

➔ Ingrese a **C#** y genere el siguiente formulario

archivo secuencial

legajo

apellido y nombre

categoria

lstlistado

Alta Baja Modificacion Salir

➔ Codifique en cada control lo que corresponda.

Atención! No codifique el número de línea. Son referencias para la explicación que incluimos a continuación del código.

Antes de la class, codificar **Imports system.IO**

(esto se hace para que reconozca todo lo que vamos a hacer con nuestros archivos).





Luego, en el **Form1_load** codificaremos **Listar()** para que llame a la función listar.

btnalta

```
FileStream archivo = new FileStream("d:\\archivo.txt", FileMode.Append);  
StreamWriter EscritorArchivo = new StreamWriter(archivo);  
  
string reg;  
  
reg = txtleg.Text + ";" + txtapynom.Text + ";" + txtcat.Text;  
  
reg = String.Format("{0};{1};{2}", txtleg.Text, txtapynom.Text, txtcat.Text);  
  
EscritorArchivo.WriteLine(reg);  
  
EscritorArchivo.Close();  
  
archivo.Close();  
  
txtleg.Text = "";  
  
txtapynom.Text = "";  
  
txtcat.Text = "";  
  
txtleg.Focus();  
  
listar();
```

¿Qué hace cada línea?

1. Abrimos un archivo de tipo **append** para agregarle registros.
2. Abrimos un **stream** donde escribir nuestros campos
3. Defino una variable de tipo cadena donde concatenaré los campos que ingresan en cada **textbox**.
6. Concateno los campos.
9. Genero el registro con esta variable de tipo cadena.



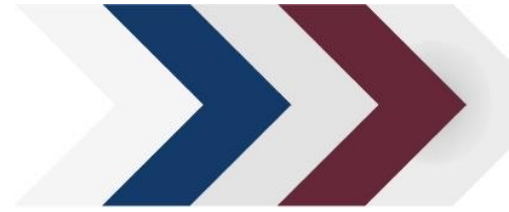


10. Cierro el **stream**.
11. Cierro el archivo.
- 12,13,14 . Limpio los **textbox**.
15. Reubico el cursor en el **textbox** que yo quiero.
16. Llamo a la función **listar**.

Btnbaja

```
FileStream archivo = new FileStream("d:\\archivo.txt", FileMode.Open);
FileStream archivoAux = new FileStream("d:\\archivoAux.txt", FileMode.Create);
StreamReader LectorArchivo = new StreamReader(archivo);
StreamWriter EscritorArchivoAux = new StreamWriter(archivoAux);
String[] VectorRegAux = new String[0];
string reg;
string legajo;
while (LectorArchivo.Peek() > -1)
{
    reg = LectorArchivo.ReadLine();
    VectorRegAux = reg.Split(';');
    legajo = VectorRegAux[0];
    if (legajo != txtleg.Text)
    {
        EscritorArchivoAux.WriteLine(reg);
    }
}
//CIERRO ARCHIVOS
```

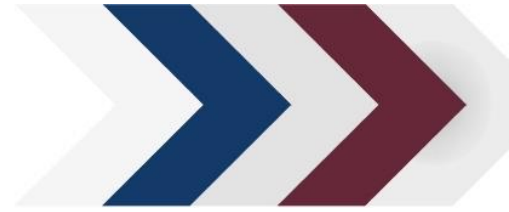




```
LectorArchivo.Close();  
archivo.Close();  
EscritorArchivoAux.Close();  
archivoAux.Close();  
//PISO EL ARCHIVO AORIGINAL CON EL AUX  
File.Delete("d:\archivo.txt");  
File.Move("d:\archivoAux.txt", "d:\archivo.txt");  
txtleg.Text = "";  
txtapynom.Text = "";  
txtcat.Text = "";  
txtleg.Focus();  
listar();
```

¿Qué hace cada línea?

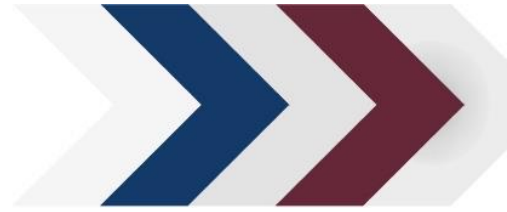
1. Abrimos un archivo para leerlo.
2. Abrimos un archivo para escribir (recordar que **create** borra lo que habia en el archivo).
3. Genero el **stream** de lectura.
4. Genero el **stream** de escritura.
5. Defino un vector donde le aplicare la funcion **split**.
6. Genero una variable de tipo cadena.
7. Genero una variable de tipo cadena.
8. Comienzo el ciclo hasta que la funcion peek me devuelva -1 o sea **EOF**.
10. Asigno la linea leida al **string**.



11. Le aplico al string la funcion **split**.
12. Asigno a la variable la primera posicion de la linea que me devuelve el **split**.
13. Si esa variable es distinta a la que ingrese en el **textbox** .
14. Genero la linea de escritura con los valores de la linea de lectura.
15. Cierro el **si**.
16. Cierro el ciclo.
18. Cierro el **stream** de lectura.
19. Cierro el archivo.
20. Cierro el **stream** de escritura.
21. Cierro el archivo auxiliar.
23. Borro el archivo original.
24. Renombro el archivo auxiliar como el original.
- 25,26,27 Limpio las cajas de texto.
28. Me posiciono en la caja de texto que yo deseo.
29. Llamo a la funcion **listar**.

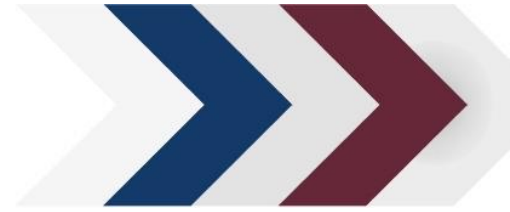
Btnmodificar

```
FileStream archivo = new FileStream("d:\archivo.txt", FileMode.Open);  
FileStream archivoAux = new FileStream("d:\archivoAux.txt", FileMode.Create);  
StreamReader LectorArchivo = new StreamReader(archivo);  
StreamWriter EscritorArchivoAux = new StreamWriter(archivoAux);  
String[] VectorRegAux = new String[0];
```



```
string reg = "";
string legajo;
while (LectorArchivo.Peek() > -1)
{
    reg = LectorArchivo.ReadLine();
    VectorRegAux = reg.Split(';');
    legajo = VectorRegAux[0];
    if (legajo == txtleg.Text)
    {
        reg = String.Format("{0};{1};{2}", txtleg.Text, txtapynom.Text, txtcat.Text);
    }
    EscritorArchivoAux.WriteLine(reg);
}
LectorArchivo.Close();
archivo.Close();
EscritorArchivoAux.Close();
archivoAux.Close();
//PISO EL ARCHIVO AORIGINAL CON EL AUX
File.Delete("d:\\archivo.txt");
File.Move("d:\\archivoAux.txt", "d:\\archivo.txt");
txtleg.Text = "";
txtapynom.Text = "";
txtcat.Text = "";
txtleg.Focus();
listar();
```





¿Qué hicimos?

Como habrá observado, esta rutina es muy similar a la de **baja**.

Le explicaremos la lógica empleada:

Los archivos secuenciales no se pueden sobre-escribir por lo que debemos crear un archivo auxiliar. Entonces, en la **baja** buscaremos dentro del archivo el registro a eliminar y mientras no sea el que estamos usando, lo escribiremos en el nuevo archivo, quedando el igual descartado. Luego renombraremos el archivo auxiliar como el original.

La **modificación** es similar. Lo que varía es que la escritura del nuevo registro está fuera del condicional. Además, cuando usted encuentra el registro para modificar dentro del condicional deberá cambiarle el contenido a los campos.

Btnsalir

Me.close ()

Por ultimo en el **listbox evento click** colocaremos el siguiente codigo que le permitirá seleccionar una linea del mismo ubicando su contenido en las cajas de texto correspondiente

```
if (lstlistado.SelectedItem != null)
{
    string seleccionado = lstlistado.SelectedItem.ToString();
    string[] vectorLista = new String[0];
    vectorLista = seleccionado.Split(';');
    txtleg.Text = vectorLista[0];
    txtapynom.Text = vectorLista[1];
    txtcat.Text = vectorLista[2];
}
```





Laboratorio Archivos #2

Este laboratorio lo guiará en el desarrollo de un logaritmo que le permitirá la visualización y control de sus directorios.

→ Genere un formulario con el siguiente diseño

→ En el **form load** codifique lo siguiente para que indique los distintos dispositivos logicos:

```
string[] Unidades = Directory.GetLogicalDrives();  
int i;  
for (i = 1; i < Unidades.Length - 1; i++)  
{  
    IstUnidades.Items.Add(Unidades[i]);  
}
```



→ En el **btncontenido**

```
IstContenido.Items.Clear();

DateTime creacion;

int i;

string Ruta = txtRuta.Text;

if (Directory.Exists(Ruta))
{
    carpetas = Directory.GetDirectories(Ruta);
    creacion = Directory.GetCreationTime(Ruta);
    archivos = Directory.GetFiles(Ruta);

    for (i = 1; i < carpetas.Length - 1)
    {
        IstContenido.Items.Add("Carpeta: " + Path.GetFileName(carpetas[i]));
    }

    //IMPRIMO LOS ARCHIVOS
    for (i = 1; i < archivos.Length - 1; i++)
    {
        IstContenido.Items.Add("Archivo: " + Path.GetFileName(archivos[i]));
    }
}
```

Habiendo realizado las soluciones anteriores le proponemos una complejización de la programación que le permitirá avanzar en el aprendizaje del lenguaje.





PARTE B

En base al prototipo de interfaz de usuario provisto, realizar:

- 1) ABM de alumnos
- 2) ABM de notas asociadas a un alumno y guardarlas en un archivo, con los siguientes datos:
dni del alumno, nota, fecha.

El prototipo de UI es el siguiente

Alumnos

▼ Head 1	▼ Head 2	▼ Head 3	
Cell 1	Cell 2	Cell 3	<input type="checkbox"/>
Cell 4	Cell 5	Cell 6	<input checked="" type="checkbox"/>
Cell 7	Cell 8	Cell 9	<input type="radio"/>

DNI

Nombre

Apellido

Notas

▼ Head 1	▼ Head 2	▼ Head 3	
Cell 1	Cell 2	Cell 3	<input type="checkbox"/>
Cell 4	Cell 5	Cell 6	<input checked="" type="checkbox"/>
Cell 7	Cell 8	Cell 9	<input type="radio"/>

DNI

Nota

Fecha