Archivos de Texto

19.1

Programación II y Laboratorio de Computación II

Edición 2018

Streams

- La clase **StreamWriter** escribe caracteres en archivos de texto.
- La clase **StreamReader** lee desde un archivo de texto.
- Ambas clases se encuentran en el espacio de nombres System.IO.

StreamWriter

- StreamWriter (string path)
  - Inicializa una nueva instancia de la clase StreamWriter, en un path específico. Si el archivo existe, se sobrescribirá, sino se creará.
- StreamWriter (string path, bool append)
  - Ídem anterior, si append es true, se agregarán datos al archivo existente. Caso contrario, se sobrescribirá el archivo.
- StreamWriter (string path, bool append, Encoding e)
  - Ídem anterior, dónde se le puede especificar el tipo de codificación que se utilizará al escribir en el archivo.

StreamWriter

- Write (string value)
  - Escribe una cadena en el archivo sin provocar salto de línea.
- WriteLine(string value)
  - Escribe una cadena en un archivo provocando salto de línea.
- Close()
  - Cierra el objeto StreamWriter.

StreamWriter

```
using System.IO;
// ...
// Abro el archivo ubicado en una dirección de la máquina
StreamWriter sw = new StreamWriter("C:\\prueba.txt");

// Agrego una línea de texto
sw.WriteLine("Hola mundo!!");

// Agrego otra línea de texto
sw.WriteLine("Chau mundo!!");

// Cierro el archivo
sw.Close();
```

StreamWriter

```
using System.IO;
// ...
// using maneja el archivo, encargándose de cerrarlo al finalizar
using (StreamWriter writer = new StreamWriter("bitacora.txt"))
{
    writer.Write("Palabra ");
    writer.WriteLine("Otras Palabras");
    writer.WriteLine("Línea");
}
```

StreamReader

- StreamReader (string path)
  - Inicializa una nueva instancia de la clase StreamReader. El path especifica de donde se leerán los datos.
- StreamReader (string path, Encoding e)
  - Ídem anterior, dónde se le especifica el tipo de codificación que se utilizará para leer el archivo.
- Close()
  - Cierra el objeto StreamReader.

StreamReader

- Read()
  - Lee un carácter del stream y avanza carácter a carácter. Retorna un entero.
- ReadLine()
  - Lee una línea de caracteres del stream y lo retorna como un string.
- ReadToEnd()
  - Lee todo el stream y lo retorna como una cadena de caracteres.

StreamReader

```
using System.IO;
// ...
// Abro el archivo ubicado en una dirección de la máquina
StreamReader sw = new StreamReader("C:\\prueba.txt");

// Leo una línea de texto
sr.ReadLine();

// Cierro el archivo
sw.Close();
```

Excepciones

- La ruta de acceso no es válida porque…
  - Es una cadena de longitud cero, contiene sólo espacios en blanco, o contiene caracteres no válidos. (ArgumentException).
  - La ruta de acceso es **Null** (ArgumentNullException).
- **File** señala a una ruta de acceso que no existe.
  - FileNotFoundException
  - DirectoryNotFoundException
- El archivo está en uso por otro proceso o hay un error de E/S
  - IOException

Excepciones

- La ruta supera la longitud máxima definida por el sistema
  - PathTooLongException
- Un nombre de archivo o de directorio de la ruta de acceso contiene un signo de dos puntos (:) o tiene un formato no válido
  - NotSupportedException
- El usuario no tiene los permisos necesarios para ver la ruta de acceso
  - SecurityException

Objetos útiles

File

- File.Exists(string path)
  - Es true si tiene los permisos necesarios y path contiene el nombre de un archivo existente; de lo contrario, es false.
  - También devuelve false si path es null, una ruta de acceso no válida o una cadena de longitud cero.
  - Si no tiene permisos suficientes para leer el archivo especificado, no se produce ninguna excepción y el método devuelve false, independientemente de la existencia del path.
- File.Copy(string, string):
  - Copia un archivo existente en un archivo nuevo. No se permite sobrescribir un archivo del mismo nombre.
- File.Delete(string path)
  - Elimina el archivo especificado.

Directory

- Directory.Delete(string path)
  - Elimina el directorio especificado, siempre y cuando esté vacio.
- Directory.Delete(string, boolean):
  - Elimina el directorio especificado y, si está indicado, los subdirectorios y archivos que contiene.
- Directory.Exists(string):
  - Determina si la ruta de acceso dada hace referencia a un directorio existente en el disco.
- GetFiles(String):
  - Devuelve los nombres de archivo (con sus rutas de acceso) del directorio especificado.

Special Folders

- Por medio del método de clase GetFolderPath de Environment podemos obtener la dirección de una carpeta:
  - Environment.GetFolderPath
- A través del enumerado Environment.SpecialFolder podemos acceder a las carpetas del sistema sin conocer su ruta completa:
  - Environment.GetFolderPath(Environment.SpecialFolder.Desktop) // retorna el path al escritorio
  - Environment.GetFolderPath(Environment.SpecialFolder. MyDocuments) // carpeta de Mis Documentos
  - Environment.GetFolderPath(Environment.SpecialFolder. ProgramFiles) // directorio de archivos de programa.