Clase 14
Archivos

¿Qué es un SISTEMA DE ARCHIVOS?

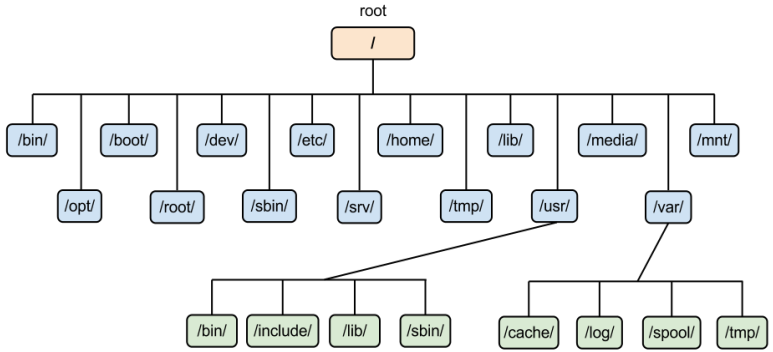
Un **sistema de archivos (*file system*)** es un elemento del sistema operativo que organiza y controla cómo se almacenan y recuperan los datos guardados en un medio de almacenamiento.

Proveen métodos para crear, mover, renombrar y eliminar tanto archivos como directorios. También asignan propiedades como sólo lectura y permisos de acceso.

La mayoría de los sistemas operativos manejan su propio sistema de archivos.

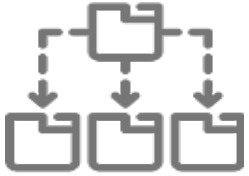
Funciones de un sistema de archivos

- Asignación de espacio a los archivos.
- Administración del espacio libre y del acceso a los datos resguardados.
- Estructurar la información guardada en un dispositivo de almacenamiento de datos.
- Permite representar la estructura de forma textual o gráficamente (gestor de archivos).



Información del sistema operativo

- **Environment**
  - **Is64BitOperatingSystem** - true si la arquitectura es de 64 bits.
  - **ProcessorCount** - retorna la cantidad de procesadores lógicos de la máquina.
  - **UserName** - retorna el nombre del usuario conectado al sistema operativo.
  - **NewLine** - salto de línea agnóstico al sistema operativo.
  - **OSVersion** - OperatingSystem
    - **ToString** - Muestra la información sobre sistema operativo y versión.



¿Qué es una RUTA?

Una **ruta (*path*)** es la forma de referenciar un archivo informático o directorio en un sistema de archivos de un sistema operativo determinado.

Señala la localización exacta de un archivo o directorio mediante una cadena de caracteres concreta.

Esta puede ser de diversas formas dependiendo del sistema operativo y del sistema de archivos en cuestión.

¿Qué es un DIRECTORIO?

Dentro de un sistema de archivos, un **directorio** es una colección de archivos y otros directorios (sub-directorios) creado con fines organizacionales.

Rutas relativas vs absolutas

Rutas absolutas
Señalan la ubicación de un archivo o directorio desde el directorio raíz del sistema de archivos.

C:\usuarios\juan\archivo.txt

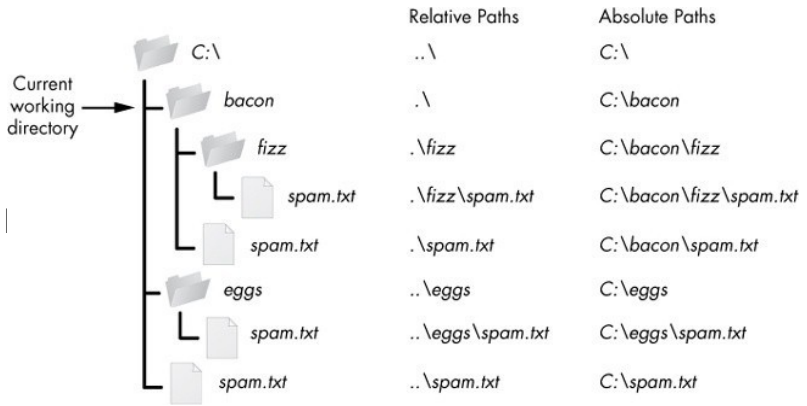
Rutas relativas
Señalan la ubicación de un archivo o directorio en relación a l posición actual.

. → representa la posición actual en la que estamos ubicados sistema de archivos.

.\archivo.txt

.. → nos mueve al directorio padre de la ubicación actual.

..\..\otra_carpeta\archivo.txt



Trabajando con rutas

Path

- **DirectorySeparatorChar** - Retorna el caracter separador de directorios para el sistema operativo sobre el que se ejecuta la aplicación.
- **PathSeparator** - Retorna el caracter que se usa para separar una lista de rutas en el sistema operativo sobre el que se ejecuta la aplicación.
- **Combine()** - Path.PathSeparator and it checks whether the first path already has a separator at the end so it will not duplicate the separators. Additionally, it checks whether the path elements to combine have invalid chars.
- **Join()** – Contatena paths.

Trabajando con directorios

Directory

- **GetCurrentDirectory()** - Retorna la ruta correspondiente al directorio sobre el que se está ejecutando la aplicación. *Alternativa a* ***Environment.CurrentDirectory***.
- **Directory.Delete(string path)** - Elimina el directorio especificado, siempre y cuando esté vacio.
- **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
- **Directory.Delete(string, boolean) -** Elimina el directorio especificado y, si está indicado, los subdirectorios y archivos que contiene

Archivos

- 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 (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.

StremWriter

- ✔ 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.

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.

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.