

## Teoría

### a. Qué son los archivos en programación y cuales son los tipos de archivos que se manejan.

Los archivos, o ficheros (files), son colecciones de información, en este caso datos (llamados registros, compuestos por campos y estos a la vez por subcampos), que están relacionados entre sí, los datos pueden ser usados en un mismo proceso o tratamiento de información, estos se guardan como una unidad en algún medio persistente (donde tienen nombre o identificador y ubicación), estando disponibles para ser usados desde un programa, estos sirven para la entrada y salida de información.

#### tipos de archivos:

1. **Archivos permanentes o maestros:** Son archivos que tienen pocas o ninguna modificación o variación a través del tiempo, en algunos casos se deben actualizar periódicamente, estos pueden ser:
  - **Constantes:** Son archivos con registros con campos fijos y muy baja variación a través del tiempo.
  - **De situación:** son los que en cada instante tienen información actualizada
  - **Históricos:** Acumulan información a lo largo del tiempo de los archivos que han sido modificados o también acumulan datos de variación periódica, es decir, en estos archivos se lleva la información de los cambios que sufra algún otro archivo específico.
2. **Archivos de movimiento:** Estos se encargan de actualizar los archivos permanentes, estos suelen tener algún campo en común con el archivo permanente que se va a mover, estos archivos contienen registros de altas, bajas y modificaciones de archivos permanentes.
3. **Archivos de maniobra o transitorios:** Tienen un tiempo de existencia limitado, son archivos auxiliares creados durante la ejecución de un programa (suelen tener una vida menor que la duración de ejecución del programa).

### b. Qué son los archivos de acceso aleatorio y para qué sirven en programación.

Estos archivos nos permiten tomar directamente el registro que necesitamos, es decir, no hay necesidad de leer los registros anteriores, para esto necesitamos saber exactamente la posición del registro dentro del archivo,

para esto hay 2 reglas, tener una longitud de registro definida (todos los registros tienen la misma longitud), además teniendo el número de registros podemos posicionarnos en el registro deseado, estos pueden abrirse en un modo que permita lectura/escritura o en un modo binario (algunos campos del registro pueden no ser caracteres) .

En programación son útiles porque se pueden acceder a cualquier parte del fichero en cualquier momento, dando un mayor dinamismo a la lectura/escritura en cualquier punto del archivo, esto reduce el tiempo necesario para las operaciones sobre el mismo, ya que no es necesario procesar los otros registros y solo nos centramos en el que necesitamos.

**c. Qué son archivos de acceso directo, para qué sirven en programación, cuáles son las operaciones básicas sobre archivos y explique cómo se implementan en C#**

Es el mismo concepto que anteriormente se trató, son archivos a los que podemos acceder en cualquier momento, a cualquiera de sus registros de forma directa, sin tener que pasar por los anteriores registros y en programación son útiles porque se pueden acceder a cualquier parte del fichero en cualquier momento, dando un mayor dinamismo a la lectura/escritura en cualquier punto del archivo, esto reduce el tiempo necesario para las operaciones sobre el mismo, ya que no es necesario procesar los otros registros y solo nos centramos en el que necesitamos.

las operaciones básicas sobre archivos son:

- Crear
- Abrir
- Leer
- Escribir
- Cerrar
- Buscar
- Borrar

**Implementación en C#**

```
using System.IO;  
using System;
```

```
// para crear el archivo  
void GenerarTXT()  
{  
    string rutaCompleta = @"D:\mi archivo.txt";  
    string texto = "HOLA MUNDO ";  
    using (StreamWriter mylogs = File.AppendText(rutaCompleta))    //se crea el archivo  
    {  

```

//se adiciona alguna información y la fecha

```
DateTime dateTime = new DateTime();
dateTime = DateTime.Now;
string strDate = Convert.ToDateTime(dateTime).ToString("yyMMdd");
mylogs.WriteLine(texto+ strDate);
mylogs.Close();

}
}
// para escribir en el archivo
void AdicionarInfoAltTxt()
{
    string rutaCompleta = @"D:\mi archivo.txt";
    string texto = "HOLA DE NUEVO";
    using (StreamWriter file = new StreamWriter(rutaCompleta, true))
    {
        file.WriteLine(texto); //se agrega información al documento
        file.Close();
    }
}
// para leer la información el archivo
void LeerInfoTxt()
{
    string rutaCompleta = @"D:\mi archivo.txt";
    string line = "";
    using (StreamReader file = new StreamReader(rutaCompleta))
    {
        while ((line = file.ReadLine()) != null) //Leer linea por linea
        {
            Console.WriteLine(line);
        }
        // OTRA FORMA DE LEER TODO EL ARCHIVO
        line = file.ReadToEnd();
        Console.WriteLine(line);
        file.close();
    }
}
```