

Laboratorio 4

“La misión secreta de Tony Stark”

Acerca del laboratorio 4

Se efectuó un programa en C# que su función es organizar archivos. Se creó un programa modularizado que permite gestionar archivos de manera eficiente, donde se hizo uso de funciones y manejo de archivos; la funcionalidad del programa o lo que ofrece es ¹ Crear un archivo plano, ² Agregar líneas al archivo (contenido), ³ Leer el archivo línea por línea, ⁴ Leer todo el texto del archivo, ⁵ Copiar el archivo, ⁶ Mover el archivo, ⁷ Crear carpeta y ⁸ Listar los archivos. Se uso manejo de excepciones y errores, además de una interfaz amigable al usuario.

Programa y su Funcionalidad

De primero creé 3 rutas específicas a usar denominadas path1, path2 y path3, path1 me lleva a lo que es la carpeta de Archivos en LaboratorioAvengers, path2 me lleva a lo que es la carpeta de Backup en LaboratorioAvengers y path3 directamente me lleva a LaboratorioAvengers.

1. Crear archivos

Lo que hice fue realizar una función la cual lo que hace es crear un archivo plano llamado “archivo.txt”, esta función la llame CrearArchivo, usando un console.writelo que hace es que me da opción a ingresarle contenido de una vez al archivo plano; uso el File.WriteAllText para que esto suceda que me permite la creación de este archivo.

Parte del código:

```
void CrearArchivo()
{ //Función para crear un archivo
    Console.Write("Ingrese el contenido del archivo: ");
    string contenido = Console.ReadLine() + "\n"; //Aca hago que el archivo se le pueda agregar contenido
    File.WriteAllText(Path.Combine(path1, "archivo.txt"), contenido); //Creo el archivo ya con el contenido agreagado
    Console.WriteLine("JARVIS- Archivo creado con éxito en " + path1);
} //CrearArchivo();
```

2. Agregar líneas

Lo que hago para agregar líneas al código es usar `path.combine` que es un método que me permite combinar dos o más rutas de archivo en una sola cadena, definiendo con un `if` para ver si el archivo existe usando un `file.exists`.

Parte del código:

```
void AgregarLineas(string invento)
{ //Función para agregar líneas a un archivo
    string filePath = Path.Combine(path1, "archivo.txt");
    if (File.Exists(filePath))
    { //Verifico si el archivo existe
        File.AppendAllText(filePath, invento + Environment.NewLine); //Agrego el contenido al archivo y salto de línea
        Console.WriteLine("Invento agregado con éxito");
    }
    else
    {
        Console.WriteLine("JARVIS- Ultron ha borrado el archivo. Primero cree el archivo Sr.Stark .");
    }
} //AgregarLineas();
```

3. Leer línea por línea

De igual manera para hacer que me lea línea por línea también use el `combine` para tener una sola cadena y con un `if` verifico si existe o no el archivo, leo todas las líneas en un array de cadenas y recorro cada línea, aumentando un contador que me imprime el número de línea y el contenido.

Parte del código:

```
void LeerLineaPorLinea()
{ //Función para leer el archivo línea por línea
    string filePath = Path.Combine(path1, "archivo.txt");
    if (File.Exists(filePath))
    {
        int cont = 0;
        string[] lineas = File.ReadAllLines(filePath); //Leo todas las líneas del archivo
        foreach (string linea in lineas)
        {
            cont++;
            Console.WriteLine("Linea Núm: " + cont + " " + linea); //Imprimo cada línea del archivo con su número
        }
    }
    else
    {
        Console.WriteLine("JARVIS- No se ha encontrado nada en el sistema, probablemente Ultron lo borró.");
    }
}
```

4. Leer todo el archivo

Combino para formar la ruta y verifico siempre si el archivo existe o no con un if, leo todo el contenido en una cadena e imprimo el contenido del archivo. Use un file.exists siempre para la verificación de existencia de archivo.

Parte del código:

```
void LeerTodoElArchivo()
{ //Función para leer todo el archivo
    string filePath = Path.Combine(path1, "archivo.txt");
    if (File.Exists(filePath))
    {
        string contenido = File.ReadAllText(filePath); //Leo todo el contenido del archivo
        Console.WriteLine("Contenido del archivo: ");
        Console.WriteLine(contenido);
    }
    else
    {
        Console.WriteLine("JARVIS- No se ha encontrado nada en el sistema, probablemente Ultron lo borró.");
    }
} //LeerTodoElArchivo();
```

5. Copiar el archivo

Para esto establezco dos rutas para copiar el archivo, origen y destino, origen es la ruta combinada de path1 junto al archivo y destino es la ruta que me lleva a la carpeta Backup donde creara la copia del archivo, usando un If junto a file.exists veo la existencia del archivo y con file.copy establezco que lo que esta en origen se copiara en destino, además al realizar esto elimino el archivo con un file.delete.

Parte del código:

```
void CopiarArchivo(string origen, string destino)
{ //Función para copiar un archivo
    origen = Path.Combine(path1, "archivo.txt");
    destino = "C:\\LaboratorioAvengers\\Backup\\Archivo(Copia).txt";
    if (File.Exists(origen))
    {
        File.Copy(origen, destino, true); //Copio el archivo a la carpeta de Backup
        Console.WriteLine("JARVIS- Su información ahora está a salvo de Ultron");
        Console.WriteLine("Archivo copiado con éxito");
        File.Delete(origen);
    }
    else
    {
        Console.WriteLine("JARVIS- No se encontró el archivo en la ruta especificada.");
    }
} //CopiarArchivo();
```

6. Mover el archivo

Volví a establecer el origen y el destino, origen es mi ruta combinada del path1 y el archivo y el destino es una carpeta llamada ArchivosClasificados donde se moverá el archivo, verifico la existencia del archivo con un If y usando file.exists, uso el file.move que me permite establecer que el archivo en origen pasar al destino.

Parte del código:

```
void MoverArchivo(string origen, string destino)
{ //Función para mover un archivo
    origen = Path.Combine(path1, "archivo.txt");
    destino = "C:\\LaboratorioAvengers\\ArchivosClasificados\\Archivo(Movido).txt";
    if (File.Exists(origen))
    { //Verifico si el archivo existe
        File.Move(origen, destino); //Muevo el archivo a la carpeta de ArchivosClasificados
        Console.WriteLine("JARVIS- Su información ahora está a salvo de Ultron");
        Console.WriteLine("Archivo movido con éxito");
    }
    else
    {
        Console.WriteLine("JARVIS- No se encontró el archivo en la ruta especificada.");
    }
} //MoverArchivo();
```

7. Crear carpeta

Creé una función simple donde se crea una carpeta en una ruta especificada como PSecreto, asigno esta variable o mi path como la ruta hacia la carpeta de LaboratorioAvengers donde ahí me crea la carpeta llamada "ProyectoSecreto", use el Directory.CreateDirectory para esto que como dije me permite crear la carpeta, además me imprime un mensaje donde me confirma la creación de la ruta de acceso.

Parte del código:

```
void CrearCarpeta(string PSecreto)
{ //Función para crear una carpeta
    PSecreto = "C:\\LaboratorioAvengers\\ProyectoSecreto";
    Directory.CreateDirectory(PSecreto); //Creo la carpeta de ProyectoSecreto
    Console.WriteLine("JARVIS- Ruta de acceso secreta creada\n");
} //CrearCarpeta();
```

8. Listar archivos

La función lo que me hace es listar los archivos de una carpeta proporcionada, obtengo los archivos usando `Directory.GetFiles(ruta)` y me imprime en mensaje indicando que se ha encontrado la ruta, recorre cada array en archivos e imprime el nombre.

Parte del código:

```
void ListarArchivos(string ruta)
{ //Función para listar los archivos de una carpeta
    string[] archivos = Directory.GetFiles(ruta); //Obtengo los archivos de la carpeta
    Console.WriteLine("JARVIS- Archivos encontrados en la carpeta: ");
    foreach (string archivo in archivos) //Imprimo los archivos encontrados
    {
        Console.WriteLine(archivo);
    }
    Console.WriteLine();
} //ListarArchivos();
```

Adicional

Como adicional hice que al iniciar el programa por si la persona no tiene la carpeta o las carpetas este las creé, iniciando crea las rutas donde trabajara, siendo estas LaboratorioAvengers y dentro de esta se crean Archivos, Backup y ArchivosClasificados, aparte ya esté definido como ruta que dentro de LaboratorioAvengers se crea ProyectosSecretos.

Parte del código:

```
//Creacion de carpetas para inicio de programa
void CarpetasDeInicio()
{ //Función para crear una carpeta
    string LabAvenger, Backup, Archivos, ArchivosClasificados;
    LabAvenger = "C:\\LaboratorioAvengers";
    Backup = "C:\\LaboratorioAvengers\\Backup";
    Archivos = "C:\\LaboratorioAvengers\\Archivos";
    ArchivosClasificados = "C:\\LaboratorioAvengers\\ArchivosClasificados";
    Directory.CreateDirectory(LabAvenger); //Creo la carpeta de LaboratorioAvengers
    Directory.CreateDirectory(Backup); //Creo la carpeta de Backup
    Directory.CreateDirectory(Archivos); //Creo la carpeta de Archivos
    Directory.CreateDirectory(ArchivosClasificados); //Creo la carpeta de ArchivosClasificados
}
CarpetasDeInicio();
```