

```
using System;

using System.IO;

public class InventosTony
{
    static string rutaLaboratorio = "LaboratorioAvengers";

    public static void Main(string[] args)
    {
        if (!Directory.Exists(rutaLaboratorio))
        {
            Directory.CreateDirectory(rutaLaboratorio);
            Console.WriteLine("Carpeta 'Laboratorio Avengers' creada.");
        }

        while (true)
        {
            MostrarMenu();
            string opcion = Console.ReadLine();

            if (int.TryParse(opcion, out int opcionNumerica))
            {
                switch (opcionNumerica)
                {
                    {
                        case 1:
                            CrearArchivo();
                            break;

                        case 2:
                            AgregarInvento();
                            break;

                        case 3:
                            LeerLineaPorLinea();
                            break;

                        case 4:
```

```
        LeerTodoElTexto();

        break;

    case 5:

        CopiarArchivo();

        break;

    case 6:

        MoverArchivo();

        break;

    case 7:

        CrearCarpeta();

        break;

    case 8:

        VerificarExistencia();

        break;

    case 9:

        ListarArchivos();

        break;

    case 10:

        EliminarArchivo();

        break;

    case 11:

        Console.WriteLine("Saliendo del programa...");

        return;

    default:

        Console.WriteLine("Opción no válida. Intenta de nuevo.");

        break;

    }

}

else

{

    Console.WriteLine("Opción no válida. Debes ingresar un número.");

}

Console.WriteLine("Presiona cualquier tecla para continuar...");
```

```

        Console.ReadKey();

        Console.Clear();
    }
}

public static void CrearArchivo()
{
    string path = Path.Combine(rutaLaboratorio, "inventos.txt");
    string[] inventos = { "1. Traje Mark 1", "2. Reactor ARC", "3. JARVIS" };
    string contenido = string.Join(Environment.NewLine, inventos);

    try
    {
        File.WriteAllText(path, contenido);
        Console.WriteLine("Archivo con inventos creado.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al crear el archivo: {ex.Message}");
    }
}

public static void AgregarInvento()
{
    string path = Path.Combine(rutaLaboratorio, "inventos.txt");

    try
    {
        string contenido = "Drones de combate" + Environment.NewLine;
        File.AppendAllText(path, contenido);
        Console.WriteLine("Invento agregado.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al agregar invento: {ex.Message}");
    }
}

```

```
}  
}
```

```
public static void LeerLineaPorLinea()  
{  
    string path = Path.Combine(rutaLaboratorio, "inventos.txt");  
    try  
    {  
        string[] lineas = File.ReadAllLines(path);  
        foreach (string linea in lineas)  
        {  
            Console.WriteLine(linea);  
        }  
    }  
    catch (Exception ex)  
    {  
        Console.WriteLine($"Error al leer el archivo: {ex.Message}");  
    }  
}
```

```
public static void LeerTodoElTexto()  
{  
    string path = Path.Combine(rutaLaboratorio, "inventos.txt");  
    try  
    {  
        string contenido = File.ReadAllText(path);  
        Console.WriteLine("Inventos: ");  
        Console.WriteLine(contenido);  
    }  
    catch (Exception ex)  
    {  
        Console.WriteLine($"Error al leer el archivo: {ex.Message}");  
    }  
}
```

```

public static void CopiarArchivo()
{
    string carpetaBackup = "Backup";
    if (!Directory.Exists(carpetaBackup))
    {
        Directory.CreateDirectory(carpetaBackup);
        Console.WriteLine("Carpeta 'Backup' creada.");
    }

    string pathOrigen = Path.Combine(rutaLaboratorio, "inventos.txt");
    string pathDestino = Path.Combine(carpetaBackup, "inventos.txt");
    try
    {
        File.Copy(pathOrigen, pathDestino, true);
        Console.WriteLine("Archivo copiado.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al copiar el archivo: {ex.Message}");
    }
}

public static void MoverArchivo()
{
    string carpetaClasificados = "ArchivosClasificados";
    if (!Directory.Exists(carpetaClasificados))
    {
        Directory.CreateDirectory(carpetaClasificados);
        Console.WriteLine("Carpeta 'Archivos Clasificados' creada.");
    }

    string pathOrigen = Path.Combine(rutaLaboratorio, "inventos.txt");
    string pathDestino = Path.Combine(carpetaClasificados, "inventos.txt");

```

```
try
{
    File.Move(pathOrigen, pathDestino);
    Console.WriteLine("Archivo movido.");
}
catch (Exception ex)
{
    Console.WriteLine($"Error al mover el archivo: {ex.Message}");
}
}
```

```
public static void CrearCarpeta()
{
    string path = "ProyectosSecretos";
    try
    {
        Directory.CreateDirectory(path);
        Console.WriteLine("Directorio creado.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al crear el directorio: {ex.Message}");
    }
}
```

```
public static void VerificarExistencia()
{
    string path = Path.Combine(rutaLaboratorio, "inventos.txt");
    if (File.Exists(path))
    {
        Console.WriteLine("El archivo existe.");
    }
    else
    {

```

```
        Console.WriteLine("El archivo no existe.");
    }
}

public static void EliminarArchivo()
{
    string path = Path.Combine(rutaLaboratorio, "inventos.txt");
    try
    {
        File.Delete(path);
        Console.WriteLine("Archivo eliminado.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al eliminar el archivo: {ex.Message}");
    }
}

public static void ListarArchivos()
{
    string path = rutaLaboratorio;
    try
    {
        string[] archivos = Directory.GetFiles(path);
        foreach (string archivo in archivos)
        {
            Console.WriteLine(archivo);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error al listar archivos: {ex.Message}");
    }
}
```

```
public static void MostrarMenu()
{
    Console.WriteLine("Gestión de Archivos");
    Console.WriteLine("--- Menú ---");
    Console.WriteLine("1. Crear archivo");
    Console.WriteLine("2. Agregar invento");
    Console.WriteLine("3. Leer archivo (línea por línea)");
    Console.WriteLine("4. Leer archivo (todo el texto)");
    Console.WriteLine("5. Copiar archivo");
    Console.WriteLine("6. Mover archivo");
    Console.WriteLine("7. Crear carpeta");
    Console.WriteLine("8. Verificar Existencia");
    Console.WriteLine("9. Listar archivos");
    Console.WriteLine("10. Eliminar archivo");
    Console.WriteLine("11. Salir");
    Console.Write("Tony selecciona una opción: ");
}
}
```

**Este es el código en el cual trabaje sin apoyarme demasiado en la inteligencia artificial, lo utilice para descartar dudas.**

**Pero lo que realice me tiraba algunos errores los cuales no les pude dar solución, aunque lo intenté.**

**Si cuenta con tiempo podría darme algunas observaciones acerca del código se lo agradecería mucho.**

El Program.cs que está en el repositorio también lo trabajé, pero en el hice muchos más usos de la IA.







