



Universidad Mariano Gálvez



Programación

Laboratorio Stark



0905-24-9756

Luis Ángel Santiago Palma

Codigo

```
Program.cs*  X
proyecto stark  Program
using System;
using System.IO;

0 referencias
class Program
{
    0 referencias
    static void Main(string[] args)
    {
        // Definir las rutas de las carpetas y archivos
        string Laboratorio = "LaboratorioAvengers";
        string ArchivoInventos = Path.Combine(Laboratorio, "inventos.txt");
        string Backup = Path.Combine(Laboratorio, "Backup");
        string ArchivosClasificados = Path.Combine(Laboratorio, "ArchivosClasificados");
        string ProyectosSecretos = Path.Combine(Laboratorio, "ProyectosSecretos");

        // Crear la carpeta LaboratorioAvengers si no existe
        CrearCarpeta(Laboratorio);

        // Menú principal del programa
        while (true)
        {
            Console.WriteLine("\n--- Menú de Operaciones ---");
            Console.WriteLine("1. Crear archivo 'inventos.txt'");
            Console.WriteLine("2. Agregar un invento");
            Console.WriteLine("3. Leer archivo línea por línea");
            Console.WriteLine("4. Leer todo el contenido del archivo");
            Console.WriteLine("5. Copiar archivo a 'Backup'");
            Console.WriteLine("6. Mover archivo a 'ArchivosClasificados'");
            Console.WriteLine("7. Crear carpeta 'ProyectosSecretos'");
            Console.WriteLine("8. Listar archivos en 'LaboratorioAvengers'");
            Console.WriteLine("9. Eliminar archivo 'inventos.txt'");
            Console.WriteLine("10. Salir");
            Console.Write("Selecciona una opción: ");

            string opcion = Console.ReadLine();

            // Manejar la opción seleccionada por el usuario
            switch (opcion)
            {
                case "1":
                    CrearArchivo(ArchivoInventos);
                    break;
            }
        }
    }
}
```

90 % 0 6 Linea: 262 Carácter: 2

```
Program.cs*  X
proyecto stark  Program
Main(string[] args)
{
    case "1":
        CrearArchivo(ArchivoInventos);
        break;
    case "2":
        Console.Write("Ingresa el nombre del invento: ");
        string invento = Console.ReadLine();
        AgregarInvento(ArchivoInventos, invento);
        break;
    case "3":
        LeerLineaPorLinea(ArchivoInventos);
        break;
    case "4":
        LeerTodoElTexto(ArchivoInventos);
        break;
    case "5":
        CopiarArchivo(ArchivoInventos, Path.Combine(Backup, "inventos.txt"));
        break;
    case "6":
        MoverArchivo(ArchivoInventos, Path.Combine(ArchivosClasificados, "inventos.txt"));
        break;
    case "7":
        CrearCarpeta(ProyectosSecretos);
        break;
    case "8":
        ListarArchivos(Laboratorio);
        break;
    case "9":
        EliminarArchivo(ArchivoInventos);
        break;
    case "10":
        Console.WriteLine("Gracias por ayudar a Tony Stark");
        return;
    default:
        Console.WriteLine("Opción no válida. Intenta de nuevo.");
        break;
}

// Método para crear un archivo si no existe
1 referencia
static void CrearArchivo(string rutaArchivo)
{
}
```

90 % 0 6 Linea: 64 Carácter: 27

```
Program.cs*  X
proyecto stark  Program  Main(string[] args)

// Método para agregar un invento al archivo
1 referencia
static void AgregarInvento(string rutaArchivo, string invento)
{
    try
    {
        using (StreamWriter sw = File.AppendText(rutaArchivo))
        {
            sw.WriteLine(invento);
            Console.WriteLine($"Invento '{invento}' agregado exitosamente.");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para leer el archivo línea por línea
1 referencia
static void LeerLineaPorLinea(string rutaArchivo)
{
    try
    {
        if (File.Exists(rutaArchivo))
        {
            Console.WriteLine("\nInventos:");
            using (StreamReader sr = new StreamReader(rutaArchivo))
            {
                string linea;
                while ((linea = sr.ReadLine()) != null)
                {
                    Console.WriteLine(linea);
                }
            }
        }
        else
        {
            Console.WriteLine($"Error: El archivo '{rutaArchivo}' no existe. ¡Ultron debe haberlo borrado!");
        }
    }
}

90 %  0 6  Línea: 64  Carácter: 27
```

```
Program.cs*  X
proyecto stark  Program  LeerLineaPorLinea(string rutaArchivo)

    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para leer todo el contenido del archivo
1 referencia
static void LeerTodoElTexto(string rutaArchivo)
{
    try
    {
        if (File.Exists(rutaArchivo))
        {
            string contenido = File.ReadAllText(rutaArchivo);
            Console.WriteLine("\nContenido completo del archivo:");
            Console.WriteLine(contenido);
        }
        else
        {
            Console.WriteLine($"Error: El archivo '{rutaArchivo}' no existe. ¡Ultron debe haberlo borrado!");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para copiar un archivo a otra ubicación
1 referencia
static void CopiarArchivo(string origen, string destino)
{
    try
    {
        CrearCarpeta(Path.GetDirectoryName(destino));
        File.Copy(origen, destino, true);
        Console.WriteLine($"Archivo copiado a '{destino}' exitosamente.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

50 %  0 6  Línea: 129  Carácter: 50
```

```
Program.cs*  X
proyecto stark  Program  LeerLineaPorLinea(string rutaArchivo)

// Método para copiar un archivo a otra ubicación
1 referencia
static void CopiarArchivo(string origen, string destino)
{
    try
    {
        CrearCarpeta(Path.GetDirectoryName(destino));
        File.Copy(origen, destino, true);
        Console.WriteLine($"Archivo copiado a '{destino}' exitosamente.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para mover un archivo a otra ubicación
1 referencia
static void MoverArchivo(string origen, string destino)
{
    try
    {
        CrearCarpeta(Path.GetDirectoryName(destino));
        File.Move(origen, destino);
        Console.WriteLine($"Archivo movido a '{destino}' exitosamente.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para crear una carpeta si no existe
4 referencias
static void CrearCarpeta(string rutaCarpeta)
{
    try
    {
        if (!Directory.Exists(rutaCarpeta))
        {
            Directory.CreateDirectory(rutaCarpeta);
        }
    }
}
```

```
Program.cs*  X
proyecto stark  Program  LeerLineaPorLinea(string rutaArchivo)

        if (!Directory.Exists(rutaCarpeta))
        {
            Directory.CreateDirectory(rutaCarpeta);
            Console.WriteLine($"Carpeta '{rutaCarpeta}' creada exitosamente.");
        }
        else
        {
            Console.WriteLine($"La carpeta '{rutaCarpeta}' ya existe.");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para listar los archivos en una carpeta
1 referencia
static void ListarArchivos(string Carpeta)
{
    try
    {
        if (Directory.Exists(Carpeta))
        {
            string[] archivos = Directory.GetFiles(Carpeta);
            Console.WriteLine($"Archivos en la carpeta:");
            foreach (string archivo in archivos)
            {
                Console.WriteLine(Path.GetFileName(archivo));
            }
        }
        else
        {
            Console.WriteLine($"La carpeta '{Carpeta}' no existe.");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}
```

```
Program.cs*  X
proyecto stark  Program  LeerLineaPorLinea(string rutaArchivo)

// Método para copiar un archivo a otra ubicación
1 referencia
static void CopiarArchivo(string origen, string destino)
{
    try
    {
        CrearCarpeta(Path.GetDirectoryName(destino));
        File.Copy(origen, destino, true);
        Console.WriteLine($"Archivo copiado a '{destino}' exitosamente.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para mover un archivo a otra ubicación
1 referencia
static void MoverArchivo(string origen, string destino)
{
    try
    {
        CrearCarpeta(Path.GetDirectoryName(destino));
        File.Move(origen, destino);
        Console.WriteLine($"Archivo movido a '{destino}' exitosamente.");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para crear una carpeta si no existe
4 referencias
static void CrearCarpeta(string rutaCarpeta)
{
    try
    {
        if (!Directory.Exists(rutaCarpeta))
        {
            Directory.CreateDirectory(rutaCarpeta);
        }
    }
}
```

90 % 0 6 Linea: 129 Carácter: 53

```
Program.cs*  X
proyecto stark  Program  LeerLineaPorLinea(string rutaArchivo)

// Método para crear una carpeta si no existe
4 referencias
static void CrearCarpeta(string rutaCarpeta)
{
    try
    {
        if (!Directory.Exists(rutaCarpeta))
        {
            Directory.CreateDirectory(rutaCarpeta);
            Console.WriteLine($"Carpeta '{rutaCarpeta}' creada exitosamente.");
        }
        else
        {
            Console.WriteLine($"La carpeta '{rutaCarpeta}' ya existe.");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Método para listar los archivos en una carpeta
1 referencia
static void ListarArchivos(string Carpeta)
{
    try
    {
        if (Directory.Exists(Carpeta))
        {
            string[] archivos = Directory.GetFiles(Carpeta);
            Console.WriteLine($"Archivos en la carpeta:");
            foreach (string archivo in archivos)
            {
                Console.WriteLine(Path.GetFileName(archivo));
            }
        }
        else
        {
            Console.WriteLine($"La carpeta '{Carpeta}' no existe.");
        }
    }
}
```

90 % 0 6 Linea: 129 Carácter: 53

```
Program.cs* x
[+] proyecto stark Program LeerLineaPorLinea(string rutaArchivo)
    Console.WriteLine($"La carpeta '{Carpeta}' no existe.");
}
catch (Exception ex)
{
    Console.WriteLine($"Error: {ex.Message}");
}
}

// Método para eliminar un archivo
1 referencia
static void EliminarArchivo(string Archivo)
{
    try
    {
        if (File.Exists(Archivo))
        {
            File.Delete(Archivo);
            Console.WriteLine($"Archivo '{Archivo}' eliminado exitosamente.");
        }
        else
        {
            Console.WriteLine($"El archivo '{Archivo}' no existe.");
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}
```

Ejecución

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 1
Archivo 'LaboratorioAvengers\inventos.txt' creado exitosamente.
```

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 2
Ingresa el nombre del invento: Armadura k23
Invento 'Armadura k23' agregado exitosamente.
```

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 3

Inventos:
Armadura k23
Reactor ark
Proyecto Spider
```

```
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 4
```

```
Contenido completo del archivo:
Armadura k23
Reactor ark
Proyecto Spider
```

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 5
Carpeta 'LaboratorioAvengers\Backup' creada exitosamente.
```

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 6
Carpeta 'LaboratorioAvengers\ArchivosClasificados' creada exitosamente.
Archivo movido a 'LaboratorioAvengers\ArchivosClasificados\inventos.txt' exitosamente.
```

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 7
Carpeta 'LaboratorioAvengers\ProyectosSecretos' creada exitosamente.
```



```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 8
```

```
Archivos en la carpeta:
intetos.txt
```

```
--- Menú de Operaciones ---
1. Crear archivo 'inventos.txt'
2. Agregar un invento
3. Leer archivo línea por línea
4. Leer todo el contenido del archivo
5. Copiar archivo a 'Backup'
6. Mover archivo a 'ArchivosClasificados'
7. Crear carpeta 'ProyectosSecretos'
8. Listar archivos en 'LaboratorioAvengers'
9. Eliminar archivo 'inventos.txt'
10. Salir
Selecciona una opción: 9
Archivo 'LaboratorioAvengers\inventos.txt' eliminado exitosamente.
```

Explicación

El programa esta creado con el fin de poder organizar y tener guardados cada uno de los proyecto que vaya a tener tony stark y el programa ayuda a tener todo guardado que inicializa definiendo cada ruta que tendrán las carpetas y archivos, luego inicia el menú principal en el cual aparece cada una de las opciones para saber que función quieres realizar y el cual se configura con un switch para seleccionar la función a realizar con solo colocar un numero siendo mas simplificado con funciones las cuales en cada una esta creada un acción

En la función de crear archivo simplemente realiza la función de crear nuevo archivos para poder almacenar los inventos

En la función Agregar inventos al seleccionarla el programa pide el nombre del invento y lo escribe en el archivo de texto

En la función Leer un archivo El programa ofrece dos formas de leer el archivo:

Línea por línea usa StreamReader para leer el archivo línea por línea y mostrar cada invento en la consola.

Todo el contenido usa File.ReadAllText para leer todo el contenido del archivo de una vez y mostrarlo en la consola.

En la función crear carpetas solo verifica si la carpeta de proyectos secretos esta creada y si no la crea

En la función Listar archivos solo es para mostrarte una lista con todos los archivos

En la función la función eliminar archivos selecciona el archivo de la carpe laboratorio y los borra