

PROGRAMA EN EJECUCIÓN

```
--- Menú de Laboratorio Avengers ---  
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  
El archivo 'inventos.txt' ya existe.
```

```
--- Menú de Laboratorio Avengers ---  
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: Iron Leg 2.0  
  
Invento 'Iron Leg 2.0 ' agregado exitosamente.
```

```
--- Menú de Laboratorio Avengers ---  
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: Iron Fist 3.0  
  
Invento 'Iron Fist 3.0' agregado exitosamente.
```

```
--- Menú de Laboratorio Avengers ---  
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: Megatronic 2.1  
  
Invento 'Megatronic 2.1' agregado exitosamente.
```

```
--- Menú de Laboratorio Avengers ---  
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  
1. Inteligencia Artificial JARVIS  
2. Iron Leg 2.0  
3. Iron Fist 3.0  
4. Megatronic 2.1
```

```
--- Menú de Laboratorio Avengers ---  
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 del archivo 'inventos.txt':  
-----  
Inteligencia Artificial JARVIS  
Iron Leg 2.0  
Iron Fist 3.0  
Megatronic 2.1  
-----
```

--- Menú de Laboratorio Avengers ---

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

Archivo copiado a Backup exitosamente.

--- Menú de Laboratorio Avengers ---

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

Archivo movido a ArchivosClasificados exitosamente.

--- Menú de Laboratorio Avengers ---

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 Laboratorio Avengers ---

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 LaboratorioAvengers:
inventos.txt

MENSAJE DE ERRORES

--- Menú de Laboratorio Avengers ---

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 'inventos.txt' eliminado exitosamente.

--- Menú de Lab4_misiónT ---

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 Lab4_misiónT
9. Eliminar archivo 'inventos.txt'
10. Salir

Selecciona una opción: 1

El archivo 'inventos.txt' ya existe.

--- Menú de Laboratorio Avengers ---

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: Iron Fist 3.0

Error: El invento 'Iron Fist 3.0' ya existe.

--- Menú de Laboratorio Avengers ---

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

Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!

--- Menú de Laboratorio Avengers ---

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

Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!

--- Menú de Laboratorio Avengers ---

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

Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!

--- Menú de Laboratorio Avengers ---

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

Error: No se puede crear un archivo que ya existe.

--- Menú de Laboratorio Avengers ---

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

La carpeta 'LaboratorioAvengers/ProyectosSecretos' ya existe.

--- Menú de Lab4_misiónT ---

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 Lab4_misiónT
9. Eliminar archivo 'inventos.txt'
10. Salir

Selecciona una opción: 8

Archivos en Lab4_misiónT:

--- Menú de Lab4_misiónT ---

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 Lab4_misiónT
9. Eliminar archivo 'inventos.txt'
10. Salir

Selecciona una opción: 9

Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!

Archivo Editar Ver

Inteligencia Artificial JARVIS
Iron Fist 3.0
Iron Leg 2.0
IA Friday 1.0

--- Menú de Laboratorio Avengers ---

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

La carpeta 'LaboratorioAvengers' está vacía.

--- Menú de Laboratorio Avengers ---

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: 11

Error: La opción debe ser un número entre 1 y 10.
Intenta de nuevo.

--- Menú de Laboratorio Avengers ---

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: 10

C:\Users\USUARIO\Documents\LABORATORIO AVENGERS - P1\LaboratorioAvengers\LaboratorioAvengers\bin\Debug\net8.0\LaboratorioAvengers.exe (process 25128) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .

Iniciar respaldo > ... LaboratorioAvengers > bin > Debug > net8.0 > LaboratorioAvengers >				
Ordenar Ver ...				
Nombre	Fecha de modificación	Tipo	Tamaño	
ArchivosClasificados	26/02/2025 20:02	Carpeta de archivos		
Backup	26/02/2025 20:02	Carpeta de archivos		
ProyectosSecretos	26/02/2025 18:52	Carpeta de archivos		

CÓDIGO LABORATORIO AVENGERS

```
using System; // Importar la biblioteca System para usar funcionalidades básicas como Console.
```

```
using System.IO; // Importar la biblioteca System.IO para manejar archivos y directorios.
```

```
using System.Linq; // Importar la biblioteca System.Linq para usar métodos como Any().
```

```
class LaboratorioAvengers // Clase principal del programa.
```

```
{
```

```
    // Definir rutas de archivos y carpetas.
```

```
    static string rutaArchivo = "LaboratorioAvengers/inventos.txt"; // Ruta del archivo principal.
```

```
    static string rutaBackup = "LaboratorioAvengers/Backup/inventos.txt"; // Ruta del archivo de backup.
```

```
    static string rutaArchivosClasificados = "LaboratorioAvengers/ArchivosClasificados/inventos.txt"; // Ruta del archivo clasificado.
```

```
    static string rutaProyectosSecretos = "LaboratorioAvengers/ProyectosSecretos"; // Ruta de la carpeta de proyectos secretos.
```

```
    static void Main(string[] args) // Método principal del programa.
```

```
{
```

```
    // Crear la carpeta LaboratorioAvengers si no existe.
```

```
    if (!Directory.Exists("LaboratorioAvengers")) // Verificar si la carpeta no existe.
```

```
{
```

```
        Directory.CreateDirectory("LaboratorioAvengers"); // Crear la carpeta.
```

```
}
```

```

// Menú interactivo.
while (true) // Bucle infinito para mantener el menú activo.
{
    Console.WriteLine("\n--- Menú de Laboratorio Avengers ---\n"); //
Mostrar el menú.
    Console.WriteLine("1. Crear archivo 'inventos.txt'"); // Opción 1:
Crear archivo.
    Console.WriteLine("2. Agregar un invento"); // Opción 2: Agregar
invento.
    Console.WriteLine("3. Leer archivo línea por línea"); // Opción 3:
Leer línea por línea.
    Console.WriteLine("4. Leer todo el contenido del archivo"); //
Opción 4: Leer todo el contenido.
    Console.WriteLine("5. Copiar archivo a Backup"); // Opción 5:
Copiar archivo.
    Console.WriteLine("6. Mover archivo a ArchivosClasificados"); //
Opción 6: Mover archivo.
    Console.WriteLine("7. Crear carpeta ProyectosSecretos"); // Opción
7: Crear carpeta.
    Console.WriteLine("8. Listar archivos en LaboratorioAvengers"); //
Opción 8: Listar archivos.
    Console.WriteLine("9. Eliminar archivo 'inventos.txt'"); // Opción
9: Eliminar archivo.
    Console.WriteLine("10. Salir"); // Opción 10: Salir del programa.
    Console.Write("Selecciona una opción: "); // Solicitar al usuario
que seleccione una opción.

    string input = Console.ReadLine(); // Leer la entrada del usuario.
    int opcion; // Variable para almacenar la opción seleccionada.

    Console.Clear(); // Limpiar la consola para mejorar la legibilidad.

    // Validar si la entrada es un número.
    if (!int.TryParse(input, out opcion)) // Intentar convertir la
entrada a un número.

```

```

    {
        Console.WriteLine("\nError: Solo se permiten números. Intenta
de nuevo."); // Mostrar mensaje de error.
        continue; // Volver al inicio del bucle.
    }

    // Validar si la opción está dentro del rango válido.
    if (opcion < 1 || opcion > 10) // Verificar si la opción está entre
1 y 10.
    {
        Console.WriteLine("\nError: La opción debe ser un número entre
1 y 10. Intenta de nuevo."); // Mostrar mensaje de error.
        continue; // Volver al inicio del bucle.
    }

    switch (opcion) // Evaluar la opción seleccionada.
    {
        case 1:
            CrearArchivo(); // Llamar a la función para crear el
archivo.

            break;
        case 2:
            Console.Write("\nIngresa el nombre del invento: "); //
Solicitar el nombre del invento.

            string invento = Console.ReadLine(); // Leer el nombre del
invento.

            AgregarInvento(invento); // Llamar a la función para
agregar el invento.

            break;
        case 3:
            LeerLineaPorLinea(); // Llamar a la función para leer línea
por línea.

            break;
        case 4:

```

```

        LeerTodoElTexto(); // Llamar a la función para leer todo
el contenido.

        break;
    case 5:
        CopiarArchivo(rutaArchivo, rutaBackup); // Llamar a la
función para copiar el archivo.
        break;
    case 6:
        MoverArchivo(rutaArchivo, rutaArchivosClasificados); //
Llamar a la función para mover el archivo.
        break;
    case 7:
        CrearCarpeta(rutaProyectosSecretos); // Llamar a la
función para crear la carpeta.
        break;
    case 8:
        ListarArchivos("LaboratorioAvengers"); // Llamar a la
función para listar archivos.
        break;
    case 9:
        EliminarArchivo(rutaArchivo); // Llamar a la función para
eliminar el archivo.
        break;
    case 10:
        return; // Salir del programa.
    }
}

}

// Función para crear el archivo "inventos.txt".
static void CrearArchivo()
{
    try // Manejo de excepciones.

```



```

        {
            if (!File.Exists(rutaArchivo)) // Verificar si el archivo no
existe.
            {
                File.Create(rutaArchivo).Close(); // Crear el archivo y
cerrarlo.
                Console.WriteLine("\nArchivo 'inventos.txt' creado
exitosamente."); // Mensaje de éxito.
            }
            else
            {
                Console.WriteLine("\nEl archivo 'inventos.txt' ya existe.");
// Mensaje si el archivo ya existe.
            }
        }
    catch (Exception ex) // Capturar excepciones.
    {
        Console.WriteLine($"Error: {ex.Message}"); // Mostrar mensaje de
error.
    }
}

```

// Función para agregar un invento al archivo y verificar que no sea un invento duplicado.

```
static void AgregarInvento(string invento)
```

```
{
```

```
    try // Manejo de excepciones.
```

```
    {
```

```
        // Verificar si el archivo existe.
```

```
        if (File.Exists(rutaArchivo))
```

```
        {
```

```
            // Leer todas las líneas del archivo.
```

```
            string[] inventosExistentes = File.ReadAllLines(rutaArchivo);
```

```

        // Verificar si el invento ya existe (ignorando
mayúsculas/minúsculas y espacios en blanco).
        if (inventosExistentes.Any(linea =>
linea.Trim().Equals(invento.Trim(), StringComparison.OrdinalIgnoreCase)))
        {
            Console.WriteLine($"Error: El invento '{invento}' ya
existe."); // Mensaje si el invento ya existe.
            return; // Salir de la función sin agregar el invento.
        }
    }

    // Si el invento no existe, agregarlo al archivo.
    using (StreamWriter sw = File.AppendText(rutaArchivo)) // Abrir el
archivo en modo append.
    {
        sw.WriteLine(invento); // Escribir el invento en una nueva
línea.
    }

    Console.WriteLine($"Invento '{invento}' agregado
exitosamente.\n"); // Mensaje de éxito.
}
catch (Exception ex) // Capturar excepciones.
{
    Console.WriteLine($"Error: {ex.Message}"); // Mostrar mensaje de
error.
}
}

// Función para leer el archivo línea por línea.
static void LeerLineaPorLinea()
{
    try // Manejo de excepciones.

```

```

{
    // Verificar si el archivo existe.
    if (File.Exists(rutaArchivo))
    {
        // Abrir el archivo para lectura.
        using (StreamReader sr = new StreamReader(rutaArchivo))
        {
            string linea; // Variable para almacenar cada línea leída.
            int numeroLinea = 1; // Contador para numerar las líneas.

            // Leer cada línea del archivo.
            while ((linea = sr.ReadLine()) != null) // Leer hasta el
final del archivo.
            {
                // Mostrar la línea en la consola con un número de
línea.

                Console.WriteLine($"{numeroLinea}. {linea}");
                numeroLinea++; // Incrementar el contador de líneas.
            }
        }
    }
    else
    {
        // Mensaje si el archivo no existe.
        Console.WriteLine("Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!");
    }
}
catch (Exception ex) // Capturar excepciones.
{
    // Mostrar mensaje de error.
    Console.WriteLine($"Error: {ex.Message}");
}

```

```

    }
}

// Función para leer todo el contenido del archivo.
static void LeerTodoElTexto()
{
    try // Manejo de excepciones.
    {
        // Verificar si el archivo existe.
        if (File.Exists(rutaArchivo))
        {
            // Leer todo el contenido del archivo.
            string contenido = File.ReadAllText(rutaArchivo);

            // Mostrar el contenido en la consola.
            Console.WriteLine("\nContenido          del          archivo
'inventos.txt':\n");
            Console.WriteLine("-----");
            Console.WriteLine(contenido);
            Console.WriteLine("-----");
        }
        else
        {
            // Mensaje si el archivo no existe.
            Console.WriteLine("Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!");
        }
    }
    catch (Exception ex) // Capturar excepciones.
    {
        // Mostrar mensaje de error.
        Console.WriteLine($"Error: {ex.Message}");
    }
}

```

```

    }
}

// Función para copiar el archivo a la carpeta Backup.
static void CopiarArchivo(string origen, string destino)
{
    try // Manejo de excepciones.
    {
        if (File.Exists(origen)) // Verificar si el archivo de origen
existe.
        {
            string carpetaBackup = Path.GetDirectoryName(destino); //
Obtener la ruta de la carpeta Backup.
            if (!Directory.Exists(carpetaBackup)) // Verificar si la
carpeta no existe.
            {
                Directory.CreateDirectory(carpetaBackup); // Crear la
carpeta Backup.
            }

            File.Copy(origen, destino, true); // Copiar el archivo.
            Console.WriteLine("\nArchivo copiado a Backup exitosamente.");
// Mensaje de éxito.
        }
        else
        {
            // Mensaje si el archivo no existe.
            Console.WriteLine("Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!");
        }
    }
    catch (Exception ex) // Capturar excepciones.
    {

```

```

        // Mostrar mensaje de error.
        Console.WriteLine($"Error: {ex.Message}");
    }
}

// Función para mover el archivo a la carpeta ArchivosClasificados.
static void MoverArchivo(string origen, string destino)
{
    try // Manejo de excepciones.
    {
        if (File.Exists(origen)) // Verificar si el archivo de origen
existe.
        {
            string carpetaDestino = Path.GetDirectoryName(destino); //
Obtener la ruta de la carpeta ArchivosClasificados.
            if (!Directory.Exists(carpetaDestino)) // Verificar si la
carpeta no existe.
            {
                Directory.CreateDirectory(carpetaDestino); // Crear la
carpeta ArchivosClasificados.
            }

            File.Move(origen, destino); // Mover el archivo.
            Console.WriteLine("\nArchivo movido a ArchivosClasificados
exitosamente."); // Mensaje de éxito.
        }
        else
        {
            // Mensaje si el archivo no existe.
            Console.WriteLine("Error: El archivo 'inventos.txt' no existe.
¡Ultron debe haberlo borrado!");
        }
    }
}

```

```

        catch (Exception ex) // Capturar excepciones.
        {
            // Mostrar mensaje de error.
            Console.WriteLine($"Error: {ex.Message}");
        }
    }

    // Función para crear la carpeta ProyectosSecretos.
    static void CrearCarpeta(string nombreCarpeta)
    {
        try // Manejo de excepciones.
        {
            if (!Directory.Exists(nombreCarpeta)) // Verificar si la carpeta
no existe.
            {
                Directory.CreateDirectory(nombreCarpeta); // Crear la carpeta.
                Console.WriteLine($"\\nCarpeta    '{nombreCarpeta}'    creada
exitosamente.\\n"); // Mensaje de éxito.
            }
            else
            {
                // Mensaje si la carpeta ya existe.
                Console.WriteLine($"\\nLa    carpeta    '{nombreCarpeta}'    ya
existe.");
            }
        }
        catch (Exception ex) // Capturar excepciones.
        {
            // Mostrar mensaje de error.
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}

```

```

// Función para eliminar el archivo "inventos.txt".
static void EliminarArchivo(string rutaArchivo)
{
    try // Manejo de excepciones.
    {
        if (File.Exists(rutaArchivo)) // Verificar si el archivo existe.
        {
            File.Delete(rutaArchivo); // Eliminar el archivo.
            Console.WriteLine("\nArchivo 'inventos.txt' eliminado exitosamente."); // Mensaje de éxito.
        }
        else
        {
            // Mensaje si el archivo no existe.
            Console.WriteLine("Error: El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
        }
    }
    catch (Exception ex) // Capturar excepciones.
    {
        // Mostrar mensaje de error.
        Console.WriteLine($"Error: {ex.Message}");
    }
}

```

```

// Función para listar archivos en una carpeta.
static void ListarArchivos(string ruta)
{
    try // Manejo de excepciones.
    {
        // Verificar si la carpeta existe.
    }
}

```



```

        if (Directory.Exists(ruta))
        {
            // Obtener la lista de archivos en la carpeta.
            string[] archivos = Directory.GetFiles(ruta);

            // Verificar si la carpeta está vacía.
            if (archivos.Length == 0)
            {
                Console.WriteLine($"\\nLa carpeta '{ruta}' está vacía.");
// Mensaje si la carpeta está vacía.
            }
            else
            {
                // Mostrar los archivos en la carpeta.
                Console.WriteLine($"\\nArchivos en '{ruta}':");
                foreach (string archivo in archivos) // Recorrer cada
archivo.
                {
                    Console.WriteLine(Path.GetFileName(archivo)); //
Mostrar solo el nombre del archivo.
                }
            }
        }
        else
        {
            // Mensaje si la carpeta no existe.
            Console.WriteLine($"\\nLa carpeta '{ruta}' no existe o ha sido
removida.");
        }
    }
    catch (Exception ex) // Capturar excepciones.
    {

```

```
        // Mostrar mensaje de error.  
        Console.WriteLine($"\\nError: {ex.Message}");  
    }  
}  
}
```