

Explicación

```
static void Main()
{
    CrearCarpetasIniciales(); // Crear las carpetas iniciales necesarias para el programa
    Console.WriteLine("\n\t*****");
    Console.WriteLine("\t;Bienvenido a la base de datos de Tony Stark!");
    Console.WriteLine("\t*****\n");
}
```

en este punto es donde se inicia el programa, esto sería lo primero que se va a ejecutar al iniciar el programa, por ejemplo en esta parte puse la creación de carpetas necesarias, las carpetas backup, y la carpeta, archivos clasificados, y también que me haga la creación de la carpeta laboratorioAverngers y de ahí se guie de las rutas, no es necesario crearla, sino que la crea automáticamente.

```
// Menú interactivo
while (true)
{
    Console.Clear(); // Limpiar la consola antes de mostrar el menú
    Console.WriteLine("\t----- ¿COMO PUEDO AYUDARTE EL DIA HOY? -----\n");
    // Mostrar opciones del menú sin colores
    Console.WriteLine("1. Crear Archivo");
    Console.WriteLine("2. Agregar Inventos");
    Console.WriteLine("3. Leer Archivo Línea por Línea");
    Console.WriteLine("4. Leer Todo el texto");
    Console.WriteLine("5. Copiar y Eliminar Archivo");
    Console.WriteLine("6. Mover Archivo");
    Console.WriteLine("7. Crear Carpeta 'ProyectosSecretos'");
    Console.WriteLine("8. Listar Archivos y Carpetas");
    Console.WriteLine("9. Salir");
    Console.Write("Seleccione una opción: ");

    string opcion = Console.ReadLine(); // Leer la opción ingresada por el usuario
    Console.Clear(); // Limpiar la consola después de seleccionar una opción
}
```

En esta parte use un while (true) este bucle se va a ejecutar indefinidamente hasta que el programa se detenga o que el usuario lo finalice, esta es la estructura principal es lo primero que me muestra para dejarme elegir que quiero hacer, y dependiendo lo que elija, llamaré la función, esto lo hice con un switch

```

string opcion = Console.ReadLine(); // Leer la opción ingresada por el usuario
Console.Clear(); // Limpiar la consola después de seleccionar una opción

switch (opcion)
{
    case "1":
        Console.WriteLine("1. Crear Archivo\n");
        CrearArchivo(); // Llamar a la función para crear un archivo
        break;
    case "2":
        Console.WriteLine("2. Agregar Inventos\n");
        if (VerificarExistenciaArchivo()) // Verificar si el archivo existe
            AgregarInventos(); // Llamar a la función para agregar inventos
        break;
    case "3":
        Console.WriteLine("3. Leer Archivo Línea por Línea\n");
        if (VerificarExistenciaArchivo()) // Verificar si el archivo existe
            LeerLíneaPorLínea(); // Llamar a la función para leer el archivo línea por
        break;
    case "4":
        Console.WriteLine("4. Leer Todo el texto\n");
        if (VerificarExistenciaArchivo()) // Verificar si el archivo existe
            LeerTodoElTexto(); // Llamar a la función para leer todo el texto del archi
        break;
    case "5":
        Console.WriteLine("5. Copiar y Eliminar Archivo\n");
        if (VerificarExistenciaArchivo()) // Verificar si el archivo existe
            CopiarYEliminarArchivo(); // Llamar a la función para copiar y eliminar el
        break;
    case "6":
        Console.WriteLine("6. Mover Archivo\n");
        if (VerificarExistenciaArchivo()) // Verificar si el archivo existe
            MoverArchivo(); // Llamar a la función para mover el archivo
        break;
}

```

Acá fue donde use el switch para lo del menú y acá fue donde use la comprobación de la existencia del archivo inventos.txt ahí use un if y luego hice un void donde está la líneas de código y luego las metí acá en los case para que me analice el inventos.txt por ejemplo si deseo copiar el archivo me tiene que tirar error por la comprobación que metí, que me va comprobar si ese archivo existe

```

1 referencia
static void CrearArchivo()
{
    string path = "c:/LaboratorioAvengers/inventos.txt";

    try
    {
        // Crear un archivo si no existe y escribir un contenido inicial
        if (!File.Exists(path)) // Verificar si el archivo no existe
        {
            File.WriteAllText(path, "Este archivo fue creado por Tony Stark.\n");
            Console.WriteLine("\nArchivo creado con éxito. - Tony Stark");
        }
        else
        {
            Console.WriteLine("\nEl archivo ya existe.");
        }
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al crear el archivo y mostrar un mensaje de error
        Console.WriteLine("\nError al crear el archivo: " + ex.Message);
    }
}

```

en esta parte sería el void y declaro creararchivo que sería la función que se va realizar, en este caso definir la variable path pero la hice como local mejor, por que si la hacia global me

iba a confundir y se me iba dificultar, en este caso path contiene la ruta completa

del archivo que se va a crear , que en este caso , el archivo tendrá el nombre de inventos.txt y se va ubicar en el directorio de la carpeta laboratorioAvengers seria "c:/LaboratorioAvengers/inventos.txt"; y también use try y catch , que el try contiene el código que intenta crear el archivo, que si ocurre alguna excepción durante la ejecución de este bloque, el control se va transferir al bloque catch y también agregue un if que la función es verificar si el archivo especifico por la variable path ya existe , se crea el archivo y se escribe un contenido inicial en el utilizao , que seria (File.WriteAllText(path, "Este archivo fue creado por Tony Stark.\n");) y luego muestra que el archivo fue creado correctamente , y si el archivo ya existe también me lo da a conocer y me muestra el mensaje el archivo ya existe, y utilice el manejo de excepciones ue si ocurre una excepción durante la creacion del archivo , esto me captura el bloque cath y se imprime un menaje de error en el programa que me va indicar que hubo un problema al crear el archivo la variable excaptura la excepción que se produjo y permite acceder almensaje de error detallado mediante el ex.message.

```
static void AgregarInventos()
{
    string path = "c:/LaboratorioAvengers/inventos.txt";
    Console.Write("\nIngrese el número de inventos que desea agregar: ");
    int numeroDeInventos = int.Parse(Console.ReadLine()); // Leer el número de inventos a agregar

    try
    {
        // Usar un bucle for para agregar múltiples inventos al archivo
        for (int i = 0; i < numeroDeInventos; i++)
        {
            Console.WriteLine("*****");
            Console.Write($"Ingrese el nombre del invento {i + 1}: ");
            string invento = Console.ReadLine(); // Leer el nombre del invento
            File.AppendAllText(path, invento + "\n"); // Agregar el invento al archivo
        }
        Console.WriteLine("\nInventos agregados.");
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al agregar los inventos y mostrar un mensaje de error
        Console.WriteLine("\nError al agregar los inventos: " + ex.Message);
    }
}
```

primero defino la ruta del archivo inventos.txt donde se guardaran los inventos y luego lee el numero de inventos que deseo agregar y convierte la entrada a un entero , y use un bucle par iterar sobre el numero de inventos especificados ingresados , en cada iteración , se va solicitar lo principal que seria el nombre del invento y lo incorpora al archivo inventos.txt ,y uso igual el manejo de excepciones que captura cualquier excepción como su nombre lo dice , y hace que durante la operación de agregar inventos y muestra un menaje de error.

```

static void LeerLineaPorLinea()
{
    string path = "c:/LaboratorioAvengers/inventos.txt";

    try
    {
        Console.WriteLine("\n*****");
        Console.WriteLine("\nEl archivo ha sido leído línea por línea:\n");
        string[] lineas = File.ReadAllLines(path); // Leer todas las líneas del archivo
        // Usar un bucle for para mostrar cada línea del archivo con su número
        for (int i = 1; i < lineas.Length; i++) // Empezar desde 1 para omitir el título
        {
            Console.WriteLine((i) + ". " + lineas[i]); // Mostrar cada línea con su número
        }
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al leer el archivo y mostrar un mensaje de error
        Console.WriteLine("\nError al leer el archivo: " + ex.Message);
    }
}

```

Acá empecé definiendo la ruta del archivo, que lo hice igual con la variable path. Podría haber usado path como global, pero como dije anteriormente, preferí hacerlo local para llevar un mejor orden. Hice que me muestre y lea las líneas, así una por una. Acá utilicé File.ReadAllLines(path) para poder leer todas las líneas del archivo y almacenarlas en un array de cadenas lineas. Utilicé un bucle for para iterar sobre el array líneas. Definí que se comenzara desde el índice 1 para poder omitir el título del documento .txt. En cada iteración, muestra la línea correspondiente.

```

static void LeerLineaPorLinea()
{
    string path = "c:/LaboratorioAvengers/inventos.txt";

    try
    {
        Console.WriteLine("\n*****");
        Console.WriteLine("\nEl archivo ha sido leído línea por línea:\n");
        string[] lineas = File.ReadAllLines(path); // Leer todas las líneas del archivo
        // Usar un bucle for para mostrar cada línea del archivo con su número
        for (int i = 1; i < lineas.Length; i++) // Empezar desde 1 para omitir el título
        {
            Console.WriteLine((i) + ". " + lineas[i]); // Mostrar cada línea con su número
        }
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al leer el archivo y mostrar un mensaje de error
        Console.WriteLine("\nError al leer el archivo: " + ex.Message);
    }
}

```

En este caso es lo mismo, solo con la diferencia de que me muestra solo el contenido del texto sin enumerar y acá sí me aparece el título, me lo imprime en la consola a comparación del anterior. El título solo lo puse para que me aparezca entrando al documento inventos.txt y en todo usé la excepción para que me muestre error si captura cualquier excepción que ocurra al leer el archivo y me muestre el mensaje de error al leer el archivo.

```
static void CopiarYEliminarArchivo()
{
    string origen = "c:/LaboratorioAvengers/inventos.txt";
    string destino = "c:/LaboratorioAvengers/Backup/inventos(copia).txt";

    try
    {
        // Copiar el archivo a la carpeta de respaldo y luego eliminar el archivo original
        File.Copy(origen, destino, true);
        File.Delete(origen);
        Console.WriteLine("\nAlerta: Parece que alguien ha robado el archivo principal 'inventos.txt', ¡pero no te preocupes! Tenemos una copia de seguridad en 'Backup'. - Tony Stark");
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al copiar y eliminar el archivo y mostrar un mensaje de error
        Console.WriteLine("\nError al copiar y eliminar el archivo: " + ex.Message);
    }
}
```

En esta parte, definí el origen que sería donde se creó el archivo inventos.txt inicialmente, y también definí la carpeta donde se va a mover el archivo, en este caso sería la copia, y sería en la carpeta Backup. Usé File.Copy(origen, destino, true) y File.Delete(origen) que sería eliminar el archivo original (el primer archivo inventos.txt creado), y al hacerlo me muestra un mensaje que trate de hacerlo algo intuitivo y amigable.

```
static void MoverArchivo()
{
    string origen = "c:/LaboratorioAvengers/inventos.txt";
    string destinoDirectorio = "c:/LaboratorioAvengers/ArchivosClasificados"; // Carpeta de destino
    string destinoArchivo = Path.Combine(destinoDirectorio, "inventos.txt"); // Ruta completa del archivo de destino

    try
    {
        // Mover el archivo a la carpeta de archivos clasificados
        File.Move(origen, destinoArchivo);
        Console.WriteLine("\nArchivo movido a " + destinoArchivo + " - Tony Stark");
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al mover el archivo y mostrar un mensaje de error
        Console.WriteLine("\nError al mover el archivo: " + ex.Message);
    }
}
```

Se define la ruta del archivo de origen inventos.txt y la carpeta de destino ArchivosClasificados. Luego, se construye la ruta completa del archivo de destino utilizando Path.Combine. Utilicé el File.Move(origen, destinoArchivo) para poder mover el archivo de origen a la ruta de destino, y luego se muestra el mensaje en el programa en el cual me informa que el archivo se ha movido. También usé las excepciones con los bloques try y catch.

```

static void CrearCarpetaProyectosSecretos()
{
    try
    {
        // Crear la carpeta 'ProyectosSecretos'
        Directory.CreateDirectory("c:/LaboratorioAvengers/ProyectosSecretos");
        Console.WriteLine("\n*****");
        Console.WriteLine("Carpeta 'ProyectosSecretos' creada exitosamente.");
        Console.WriteLine("*****\n");
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al crear la carpeta y mostrar un mensaje de error
        Console.WriteLine("\nError al crear la carpeta 'ProyectosSecretos': " + ex.Message);
    }
}

```

Pongo la dirección del directorio donde quiero que me cree la carpeta , y si la carpeta se creo con éxito me va mostrar el mensaje que fue creada exitosamente , y luego manejo lo de las excepciones como en las demás opciones del menú

```

static void ListadoDeArchivosYCarpetas()
{
    string path = "c:/LaboratorioAvengers";

    try
    {
        if (Directory.Exists(path)) // Verificar si el directorio existe
        {
            Console.WriteLine("\nArchivos y carpetas en " + path + ":\n");

            // Listar todos los archivos en el directorio
            string[] archivos = Directory.GetFiles(path);
            Console.WriteLine("Archivos:");
            // Usar un bucle for para mostrar cada archivo con su número
            for (int i = 0; i < archivos.Length; i++)
            {
                Console.WriteLine((i + 1) + ". " + Path.GetFileName(archivos[i]));
            }

            // Listar todas las carpetas en el directorio
            string[] carpetas = Directory.GetDirectories(path);
            Console.WriteLine("\nCarpetas:");
            // Usar un bucle for para mostrar cada carpeta con su número
            for (int i = 0; i < carpetas.Length; i++)
            {
                Console.WriteLine((i + 1) + ". " + Path.GetFileName(carpetas[i]));
            }
        }
        else
        {
            // Mensaje de error si el directorio no existe
            Console.WriteLine("\nError: La base de datos fue analizada y no se pudo encontrar el directorio. - Tony Stark*");
        }
    }
    catch (Exception ex)
    {
        // Capturar cualquier excepción que ocurra al listar archivos y carpetas y mostrar un mensaje de error
        Console.WriteLine("\nError al listar archivos y carpetas: " + ex.Message);
    }
}

```

Primero verifico si el directorio especificado en la variable path existe. Si el directorio existe, se obtienen todos los archivos en el directorio usando Directory.GetFiles(path) y luego se muestra cada archivo junto con su número en la lista. También puse un mensaje de error si el directorio no existe y utilicé el manejo de excepciones. En este caso, yo hice que me muestre los archivos como txt y las carpetas creadas en la carpeta LaboratorioAvengers. En la parte de archivos y carpetas casi usé las mismas líneas, solo con la diferencia de que definí que quería

las carpetas, y luego al darle en esa opción me muestra por separado los resultados para mejor entendimiento.

```
5 referencias
static bool VerificarExistenciaArchivo()
{
    string path = "c:/LaboratorioAvengers/inventos.txt";

    if (File.Exists(path)) // Verificar si el archivo existe
    {
        return true; // El archivo existe
    }
    else
    {
        // Mensaje interactivo cuando el archivo no existe
        Console.WriteLine("\nUps, parece que el archivo 'inventos.txt' no existe. ¡Tony Stark debe haber olvidado crearlo!");
        return false; // El archivo no existe
    }
}
```

Acá está la parte de la verificación si el archivo existe. Pongo la dirección del archivo que quiero que me analice. Si el archivo existe, lo hice con un if que, si existe, que lo verifiqué, pero si no lo encuentra en la dirección que puse, me va a mostrar en el programa el mensaje de error que el archivo no pudo haberse encontrado.