

Explicación de mi código

Lo primero que hice fue dividir mi código en funciones específicas, explicaré una a una:

1. Crear archivo 'inventos.txt'

```
3
4 // Crear el archivo
5 void CrearArchivo() {
6     string ruta = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers";
7     string archivo = Path.Combine(ruta, "inventos.txt");
8
9     // Evaluar si la ruta existe
10    if (!Directory.Exists(ruta)) {
11        Console.WriteLine($"Error: la ruta {ruta} no existe");
12        return;
13    }
14
15    // Evaluar si el archivo existe
16    if (!File.Exists(archivo)) {
17        File.WriteAllText(archivo, "LISTA DE INVENTOS:");
18        Console.WriteLine($"Archivo creado: {archivo}");
19    }
20 }
21
```

- Lo primero que hago es establecer la ruta donde quiero crear el archivo dentro de una variable y luego otra variable con la combinación de la ruta + el archivo a crear en ella, para unificarlo.
- Con `!Directory.Exists(ruta)` se evalúa que la ruta exista, sino existe se imprime un mensaje informándolo.
- Con `!File.Exists(archivo)` se evalúa que el archivo no exista, si ya existe no se vuelve a crear y si no existe se crea con el título "LISTA DE INVENTOS" y un mensaje en pantalla de confirmación.

2. Agrega un nuevo invento al archivo

```
21
22 // Agregar inventos al archivo
23 string AgregarInvento(string invento) {
24     string ruta = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
25     string contenido = File.ReadAllText(ruta);
26     string inventoNuevo = invento;
27
28     if (!File.Exists(ruta)) {
29         Console.WriteLine("El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
30     } else {
31         // Verificar si existe un salto de línea al final, sino agregarlo
32         if (!contenido.EndsWith("\n")) {
33             File.AppendAllText(ruta, Environment.NewLine);
34         }
35
36         // Agregar nuevo invento
37         using (StreamWriter writer = new StreamWriter(ruta, true)) {
38             writer.WriteLine(inventoNuevo);
39         }
40     }
41     return invento;
42 }
43
```

- Como en la función anterior, guardo en una variable la ruta, pero esta vez directamente hacia el archivo de texto.
- Guardo dentro de una variable, en ese caso nombrada como “contenido” la lectura previa del archivo, esto con `File.ReadAllText(ruta)`.
- Procedo a evaluar que la ruta exista, en caso de que no se muestra un mensaje en pantalla.
- Si la ruta del archivo existe hago otra evaluación, pero esta vez sobre la lectura del archivo, para identificar si en la última línea de mi archivo existe un salto de línea, si no existe se agrega el salto de línea.
- Luego de esto con `StreamWriter` agrego el invento nuevo en el archivo, sin que lo sobrescriba, este nuevo invento viene por parámetro en la función, el usuario nombre el invento y la función lo recibe hasta `StreamWriter`.
- (Aclaración): Al final utilizo un `return invento;` únicamente para que no haya errores en la sintaxis, ya que al ser la función de tipo `string` esto es obligatorio, pero no se le da uso real a ese `return`.

3. Leer el archivo 'inventos.txt' línea por línea

```
43
44 // Leer el archivo
45 void LeerArchivo() {
46     string ruta = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
47
48     if (!File.Exists(ruta)) {
49         Console.WriteLine("El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
50     } else {
51         using (StreamReader reader = new StreamReader(ruta)) {
52             string contenido = reader.ReadToEnd();
53             Console.WriteLine(contenido);
54         }
55     }
56 }
57
```

- Guardo la ruta al archivo en una variable
- Evalúo que el archivo exista, en caso contrario se muestra un mensaje en pantalla.
- Si el archivo existe con StreamReader se lee el archivo línea por línea para mostrarlo.
- Guardo la lectura en una variable y muestro su contenido en pantalla.

4. Leer todo el texto del archivo 'inventos.txt'

```
57
58 // Leer todo el archivo (con otro metodo)
59 void LeerArchivoCompleto() {
60     string ruta = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
61     string contenido = File.ReadAllText(ruta);
62
63     if (!File.Exists(ruta)) {
64         Console.WriteLine("El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
65     } else {
66         Console.WriteLine(contenido);
67     }
68 }
69
```

- Repito el mismo proceso, pero esta vez utilizando File.ReadAllText para almacenar toda la lectura del archivo en una variable.
- Procedo a evaluar que el archivo exista, si no existe se muestra un mensaje y si sí existe se muestra su contenido en pantalla.

5. Copiar 'inventos.txt' a la carpeta 'Backup'

```
70
71 // OPERACIONES CON LA CLASE FILE
72 // Copiar inventos.txt a Backup
73 void CopiarArchivo() {
74     string archivo = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
75     string carpetaBackup = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\Backup";
76     string destinoArchivo = Path.Combine(carpetaBackup, Path.GetFileName(archivo));
77
78     if (!Directory.Exists(carpetaBackup)) {
79         Directory.CreateDirectory(carpetaBackup);
80     }
81
82     if (!File.Exists(archivo)) {
83         Console.WriteLine("El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
84     } else {
85         File.Copy(archivo, destinoArchivo, true);
86         Console.WriteLine("Archivo 'inventos.txt' copiado correctamente a 'Backup'");
87     }
88 }
89
```

- Guardo la ruta al archivo en una variable.
- Guardo la ruta de la carpeta a crear en una variable.
- Guardo la creación o acción de copiar el archivo dentro de la variable destinoArchivo
- Para evaluar si una carpeta existe utilizo Directory.Exists
- Si la carpeta no existe procedo a crearla, utilizando Directory.CreateDirectory
- Si el archivo no existe se manda un mensaje en pantalla informándolo
- Si el archivo sí existe con File.Copy se copia el archivo dentro de la carpeta

6. Mover 'inventos.txt' a la carpeta 'ArchivosClasificados'

```
89
90 // Mover inventos.txt a ArchivosClasificados
91 void MoverArchivo() {
92     string archivo = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
93     string carpetaArchivosClasificados = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\ArchivosClasificados";
94     string destinoArchivo = Path.Combine(carpetaArchivosClasificados, Path.GetFileName(archivo));
95
96     if (!Directory.Exists(carpetaArchivosClasificados)) {
97         Directory.CreateDirectory(carpetaArchivosClasificados);
98     }
99
100     if (!File.Exists(archivo)) {
101         Console.WriteLine("El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
102     } else {
103         File.Move(archivo, destinoArchivo);
104         Console.WriteLine("Archivo 'inventos.txt' movido correctamente a 'ArchivosClasificados'");
105     }
106 }
107
```

- Bastante similar al anterior, pero utilizando File.Move para que al momento de mover el archivo hacia la carpeta destino se elimine de donde originalmente estaba.

7. Crear carpeta 'ProyectosSecretos'

```
17
18 // Crear carpeta ProyectosSecretos
19 void CrearCarpeta() {
20     string proyectosSecretos = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\ProyectosSecretos";
21
22     if (!Directory.Exists(proyectosSecretos)) {
23         Directory.CreateDirectory(proyectosSecretos);
24         Console.WriteLine("Carpeta 'ProyectosSecretos' creada exitosamente");
25     } else {
26         Console.WriteLine("La carpeta 'ProyectosSecretos' ya existe");
27     }
28 }
29
```

- Guardo en una variable la ruta que debería tener la carpeta si existiera
- Evalúo si ya existe y si aún no existe procedo a crearla con `Directory.CreateDirectory` junto con un mensaje informando que se creó
- Si la carpeta ya existe se manda un mensaje en pantalla informando que ya existe y no se crea nada

8. Eliminar el archivo 'inventos.txt'

```
19
20 // Eliminar archivo inventos.txt
21 void EliminarArchivo() {
22     string archivo = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
23     string copiaSeguridad = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\Backup\inventos.txt";
24     string carpetaBackup = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\Backup";
25     string destinoArchivo = Path.Combine(carpetaBackup, Path.GetFileName(archivo));
26
27     if (!Directory.Exists(carpetaBackup)) {
28         Directory.CreateDirectory(carpetaBackup);
29     }
30
31     if (!File.Exists(archivo)) {
32         Console.WriteLine("El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
33     } else if (!File.Exists(copiaSeguridad)) {
34         // Si la copia de seguridad no existe, se crea
35         File.Copy(archivo, destinoArchivo, true);
36         File.Delete(archivo);
37
38         Console.WriteLine("Copia de seguridad de 'inventos.txt' creada con éxito en 'Backup'");
39         Console.WriteLine("Archivo 'inventos.txt' eliminado con éxito");
40     } else {
41         File.Delete(archivo);
42         Console.WriteLine("Archivo 'inventos.txt' eliminado con éxito");
43     }
44 }
45
```

- Guardo las rutas del archivo, de la copia de seguridad, de la carpeta donde se guardará la copia de seguridad y de la creación del archivo en una variable cada uno
- Primero evalúo que exista la carpeta Backup, si no existe se crea

- Evalúo que exista el archivo a eliminar, si no existe se manda un mensaje en pantalla
- Si el archivo sí existe primero se valida que exista su copia de seguridad, si esta no existe se crea antes y luego se elimina con File.Delete
- Si la copia de seguridad ya existía, se elimina directamente el archivo

9. Listar todos los archivos dentro de 'LaboratorioAvengers'

```
// Listar archivos de LaboratorioAvengers
void ListarArchivos() {
    string carpeta = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers";
    string[] elementos = Directory.GetFileSystemEntries(carpeta);
    int contador = 0;

    if (Directory.Exists(carpeta)) {
        Console.WriteLine("Archivos dentro de la carpeta:\n");
        foreach (string elemento in elementos) {
            string element = Path.GetFileName(elemento);
            contador++;

            Console.WriteLine($"- {element}");
        }
    } else {
        Console.WriteLine("La carpeta 'LaboratorioAvengers' no existe");
    }
}
```

- Guado la ruta de la carpeta en una variable
- Creo un arreglo de elementos que almacena el nombre de cada archivo dentro de la carpeta utilizando Directory.GetFileSystemEntries
- Si la carpeta existe se listan sus elementos utilizando un foreach que recorre el arreglo elementos y los va listando en pantalla uno a uno
- Si la carpeta no existe, se informa con un mensaje en pantalla

Menú:

```
164
165 // Menú para el usuario
166 void Menu() {
167     Console.WriteLine("\nElige la opción que deseas realizar:\n");
168     Console.WriteLine("1. Crear archivo 'inventos.txt'");
169     Console.WriteLine("2. Agrega un nuevo invento al archivo");
170     Console.WriteLine("3. Leer el archivo 'inventos.txt' línea por línea");
171     Console.WriteLine("4. Leer todo el texto del archivo 'inventos.txt'");
172     Console.WriteLine("5. Copiar 'inventos.txt' a la carpeta 'Backup'");
173     Console.WriteLine("6. Mover 'inventos.txt' a la carpeta 'ArchivosClasificados'");
174     Console.WriteLine("7. Crear carpeta 'ProyectosSecretos'");
175     Console.WriteLine("8. Eliminar el archivo 'inventos.txt'");
176     Console.WriteLine("9. Listar todos los archivos dentro de 'LaboratorioAvengers'");
177 }
178
```

- Cree una función llamada Menu para guardar el listado de opciones a realizar mostrado al usuario

Ejecución:

```
178
179 int opcion = -1;
180 while (opcion != 0) {
181     Menu();
182     opcion = int.Parse(Console.ReadLine());
183
184     switch(opcion) {
185     case 1:
186         string archivo = @"C:\Users\Jeiner\Desktop\LaboratorioAvengers\inventos.txt";
187
188         if (File.Exists(archivo)) {
189             Console.WriteLine("El archivo 'inventos.txt' ya existe");
190         } else {
191             CrearArchivo();
192         }
193         break;
194     case 2:
195         Console.WriteLine("Agrega un nuevo invento: ");
196         string nuevoInvento = Console.ReadLine();
197         AgregarInvento(nuevoInvento);
198         Console.WriteLine("Nuevo invento agregado correctamente");
199         break;
200     case 3:
201         LeerArchivo();
202         break;
203     case 4:
204         LeerArchivoCompleto();
205         break;
206     case 5:
207         CopiarArchivo();
208         break;
209     case 6:
210         MoverArchivo();
211         break;
212     case 7:
213         CrearCarpeta();
214         break;
215     case 8:
216         EliminarArchivo();
217         break;
218     case 9:
219         ListarArchivos();
220         break;
221     default:
222         Console.WriteLine("\nOpcion no valida\n");
223         break;
224     }
225 }
```

- Se llama al Menu y se guarda la respuesta del usuario en una variable para posteriormente evaluar el valor de esa opción en un switch, dependiendo de la opción elegida se ejecutará un código u otro.

- Todo esto dentro de un while que evalúa que la opción sea diferente de 0, si es diferente se repetirá el bucle para seguir mostrando el Menu una vez ejecuta la opción que el usuario eligió
- Si el usuario selecciona un número fuera de listado brindado se mandará un mensaje en pantalla diciendo que esa opción no es válida