

Este programa es un Gestor de archivos que permite a los usuarios agregar inventos a un archivo llamado Inventos.txt y permite copiar, mover, eliminar, agregar inventos y listar archivos.

## Funcionalidades del programa

### Parte inicial

Desde el inicio cree un Static void Main() para crear funciones y poder llamarla para realizar las acciones que desea hacer el usuario seguido de eso añadí dos clases llamada Directory que crea n dos archivos llamados Backup y ArchivosClasificados con excepción por si ocurre un error y así el programa no se cierre de inesperadamente.

```
static void Main()
{
    // Creacion de carpetas
    try
    {
        Directory.CreateDirectory(@"C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Backup");
        Directory.CreateDirectory(@"C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\ArchivosClasificados");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al crear directorios: " + ex.Message);
    }
}
```

### Menú de opciones para el usuario

Imprime en la consola las opciones que el usuario puede realizar y están enumeradas para que el ingrese un numero que lo lleve a la acción que desea.

```
{
    Console.WriteLine("\n ***Bienvenido***");
    Console.WriteLine("1. Crear archivo");
    Console.WriteLine("2. Agregar invento");
    Console.WriteLine("3. Leer linea por linea");
    Console.WriteLine("4. Leer todo el texto");
    Console.WriteLine("5. Copiar archivo a backup");
    Console.WriteLine("6. Mover archivo a Archivos Clasificados");
    Console.WriteLine("7. Crear carpeta ProyecrtosSecretos");
    Console.WriteLine("8. Listar archivos");
}
```

## Llamado a las funciones

Cuando el usuario selecciona un numero se guarda en una variable y se selecciona dentro del switch la cual conduce a las funciones que realizaran la tarea deseada, si el usuario escribe un numero que no aparece en el menú o ingresa texto el programa imprime un mensaje en la consola indicando que vuelva a intentarlo.

```
if (int.TryParse(Console.ReadLine(), out opcion))
{
    switch (opcion)
    // Llamado a las funciones
    {
        case 1: CrearArchivo(); break;
        case 2: if (VerificarExistenciaArchivo()) AgregarInvento(Console.ReadLine()); break;
        case 3: if (VerificarExistenciaArchivo()) LeerLineaPorLinea(); break;
        case 4: if (VerificarExistenciaArchivo()) LeerTodoElTexto(); break;
        case 5: if (VerificarExistenciaArchivo()) CopiarArchivo(); break;
        case 6: MoverArchivo(); break;
        case 7: CrearCarpeta(); break;
        case 8: ListadoDeArchivos(); break;
    }
}
else
{
    Console.WriteLine("Al parecer no ingreso un numero. Intentelo de nuevo");
}
```

## Verifica si el archivo Inventos.txt existe

Creé una función que antes de realizar la acción que desea el usuario verifica si el archivo existe, sino existe el programa no se cerrara inesperadamente, sino que imprime un error en la consola y se devuelve al menú aquí utilice la clase File para verificar la existencia del archivo.

```
static bool VerificarExistenciaArchivo()
{
    //Verifica si la carpeta invnetos.txt existe, si el archivo no existe aparece un mensaje (Archivo no existe)
    string path = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Inventos.txt";
    if (File.Exists(path))
    {
        return true;
    }
    else
    {
        Console.WriteLine("Error: El archivo 'inventos.txt' no existe. ¡Ultron debe haberlo borrado!");
        return false;
    }
}
```

## Crear archivo Inventos.txt

Esta función crea el archivo Inventos.txt y agrega un mensaje al archivo indicando que se creó esto pasa porque utilice el File.WriteAllText, si la ruta para crear el archivo no se encuentra el programa no se cierra inesperadamente, sino que mostrara un mensaje indicando que el archivo no se pudo crear.

```
static void CrearArchivo()
{
    try
    {
        string path = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Inventos.txt";
        string contenido = "Archivo plano creado";
        File.WriteAllText(path, contenido); //Crea el archivo en la ruta especificada y agrega un mensaje dentro del archivo
        Console.WriteLine("Archivo creado");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al crear el archivo: " + ex.Message);
    }
}
```

## Agregar inventos

Esta función agrega inventos que el usuario ingreso a la dirección del archivo Inventos.txt esto se realiza mediante el método File.AppendAllText que busca la dirección del archivo y agrega el invento.

```
static void AgregarInvento(string invento)
{
    try
    {
        string path = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Inventos.txt"; //Ubicacion del archivo que deseo agregar contenido
        File.AppendAllText(path, "\n" + invento); //Agrega el invento que coloque en la variable invento
        Console.WriteLine("Se agrega contenido");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al agregar invento: " + ex.Message);
    }
}
```

## Leer línea por línea

Esta función permite leer las líneas del archivo Inventos.txt, inicié una variable con valor 0 que me sirve de contador y cree un array para que en la estructura de control foreach recorriera todas las líneas del archivo.

```
static void LeerLineaPorLinea()
{
    try
    {
        int cont = 0; //Inicio esta variable en 0 para utilizarla como contador de líneas
        string path = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Inventos.txt";
        string[] lineas = File.ReadAllLines(path);
        foreach (string linea in lineas) //Recorre las líneas dentro del array string[] lineas
        {
            cont++;
            Console.WriteLine("Linea Numero: " + cont + "\\t" + linea);
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al leer el archivo: " + ex.Message);
    }
}
```

## Leer todo el texto

Esta función permite leer todos los inventos agregados por el usuario.

```
static void LeerTodoElTexto()
{
    try
    {
        string path = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Inventos.txt"; //Ubicacion del texto que se va a leer
        string contenido = File.ReadAllText(path); //Lee todo el texto
        Console.WriteLine("\\n--- Contenido del Archivo ---");
        Console.WriteLine(contenido);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al leer el archivo: " + ex.Message);
    }
}
```

## Copiar Archivo

Este bloque sirve para copiar un archivo a una carpeta llamada Backup, le doy una ruta a cada variable, una para el origen del archivo y el otro al destino donde quiero que se copie utilizando el método File.Copy y dentro de la función CopiarArchivo llama a otra función llamada EliminarArchivo para que este sea eliminado de su origen.

```
static void CopiarArchivo()
{
    try
    {
        string pathOrigen = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Inventos.txt";
        string pathDestino = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Backup\\Inventos.txt";
        File.Copy(pathOrigen, pathDestino, true); // Copia el archivo
        Console.WriteLine("Archivo copiado");
        EliminarArchivo(pathOrigen); //Llama a la función EliminarArchivo para eliminar el archivo
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al copiar el archivo: " + ex.Message);
    }
}
```

## Eliminar archivo

Elimina el archivo de su origen luego de haberse copiado en la carpeta backup utilizando el método File.Delete.

```
static void EliminarArchivo(string ruta)
{
    //Elimina el archivo Invnetos.txt de su origen
    if (File.Exists(ruta))
    {
        File.Delete(ruta);
        Console.WriteLine("Archivo eliminado después de la copia.");
    }
}
```

## Mover archivo

Mueve el archivo de la carpeta Backup a la carpeta ArchivosClasificados utilizando el método File.Move.

```
static void MoverArchivo()
{
    try
    {
        string pathOrigen = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\Backup\\Inventos.txt"; //Ubicacion del archivo
        string pathDestino = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\ArchivosClasificados\\Inventos.txt"; //Ubicacion
        File.Move(pathOrigen, pathDestino); //Mueve el archivo a la carpeta deseada
        Console.WriteLine("Archivo movido correctamente");
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al mover el archivo: " + ex.Message);
    }
}
```

## Crear carpeta ProyectosSecretos

Crea la carpeta con el método Directory.CreateDirectory con la variable path donde esta almacenada la dirección donde se crea la carpeta.

```
static void CrearCarpeta()
{
    try
    {
        string nomcarpeta = "ProyectosSecretos"; //Nombre de la carpeta que se va a crear

        string path = Path.Combine(@"C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers\\", nomcarpeta); //Crea la carpeta en esta ubicacion
        Directory.CreateDirectory(path);
        Console.WriteLine("Carpeta creada exitosamente en: " + path);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al crear la carpeta: " + ex.Message);
    }
}
```

## Listado de archivos

Esta función imprime los archivos existentes colocando la dirección del archivo LaboratorioAvengers en un array y los enumera con el metodo Get.Files.

```
static void ListadoDeArchivos()
{
    string path = "C:\\Users\\joseg\\OneDrive\\Escritorio\\LaboratorioAvengers";
    try
    {
        if (Directory.Exists(path))
        {
            //Muestra y enumera los archivos existentes
            Console.WriteLine("\nArchivos y carpetas en " + path + ":\n");
            string[] archivos = Directory.GetFiles(path);

            Console.WriteLine("Archivos:");
            foreach (string archivo in archivos)
            {
                Console.WriteLine(Path.GetFileName(archivo));
            }
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error al listar archivos: " + ex.Message);
    }
}
```