

**Universidad Mariano Gálvez Guatemala, Campus
Jutiapa**

Facultad de Ingeniería en Sistemas

Curso: Programación 1



Catedrático: Ing. Ruldin Ayala

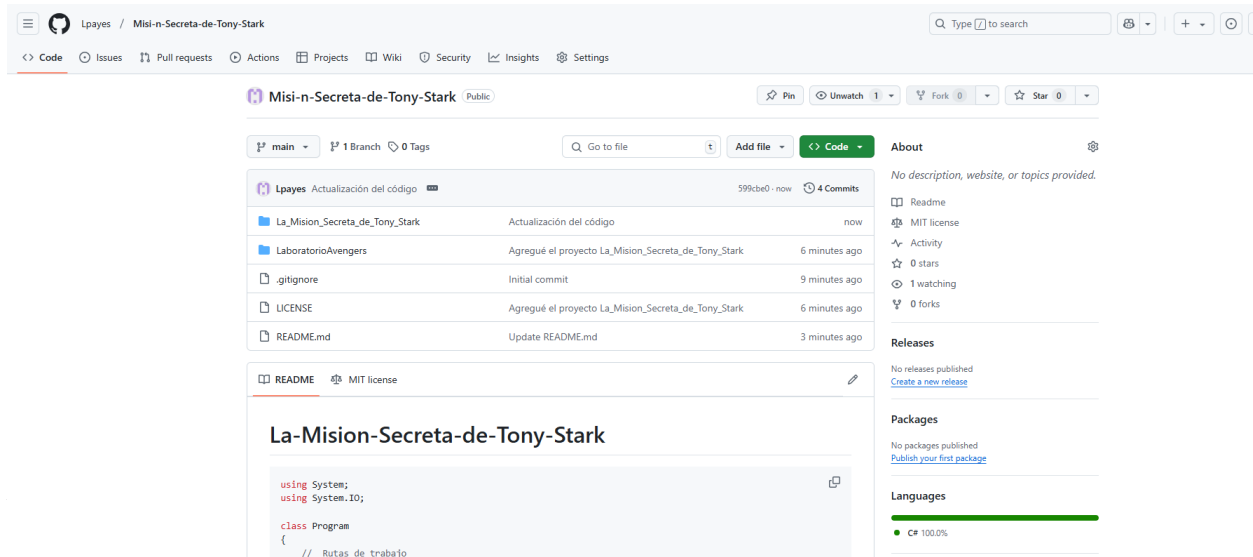
Nombre del estudiante: Lester David Payes Méndez

Carné: 0905-24.22750

Tarea: La Misión Secreta de Tony Stark

Proyecto: La Misión Secreta de Tony Stark

Repositorio remoto



```
C:\Users\LESTER\Downloads\PROGRAMACIÓN 1\La-Misi-n-Secreta-de-Tony-Stark>git commit -m"Agregué
Secreta-de-Tony-Stark"
[main 1964b6d] Agregué el proyecto de La-Misi-n-Secreta-de-Tony-Stark
3 files changed, 323 insertions(+)
create mode 100644 La_Mision_Secreta_de_Tony_Stark/La_Mision_Secreta_de_Tony_Stark.csproj
create mode 100644 La_Mision_Secreta_de_Tony_Stark/La_Mision_Secreta_de_Tony_Stark.sln
create mode 100644 La_Mision_Secreta_de_Tony_Stark/Program.cs

C:\Users\LESTER\Downloads\PROGRAMACIÓN 1\La-Misi-n-Secreta-de-Tony-Stark>git push origin main
Enumerating objects: 7, done.
```

Objetivo del Proyecto:

Este proyecto simula un laboratorio de inventos en el que se gestionan archivos relacionados con inventos secretos, respaldos, clasificaciones y proyectos. El objetivo principal es ayudar a Tony Stark a recuperar y gestionar sus inventos tras el ataque de Ultron. A través de este sistema, se implementan diversas funcionalidades para manipular archivos y carpetas de manera eficiente, como crear, leer, agregar, mover, respaldar y

eliminar archivos, asegurando que los inventos y datos valiosos estén protegidos y sean accesibles para poder enfrentar la amenaza de Ultron y futuras emergencias tecnológicas.

- **Lenguaje de Programación utilizado C#**
- **Herramientas: .NET Framework para desarrollo de aplicaciones de consola.**

Descripción de las Funciones Principales

1. CrearCarpeta(string rutaCarpeta)

Función encargada de crear una carpeta en la ruta especificada si no existe. En caso de que la carpeta ya exista, no realiza ninguna acción. La creación de la carpeta se maneja mediante un bloque try-catch para capturar excepciones comunes como `DirectoryNotFoundException` o `IOException`.

2. CrearArchivoinventos()

Esta función crea un archivo `inventos.txt` en el directorio principal del laboratorio (`C:\LaboratorioAvengers`). Si el archivo ya existe, muestra un mensaje indicando que el archivo ya está creado. Si el archivo no existe, se crea con un contenido inicial que representa algunos de los inventos del laboratorio.

3.

4. AgregarInvento(string nuevoInvento)

Permite agregar un nuevo invento al archivo `inventos.txt`. Si el archivo no existe, lo crea previamente con el contenido inicial. Los inventos se agregan al final del archivo sin sobrescribir el contenido anterior.

5. LeerInventosLineaPorLinea()

Lee el contenido del archivo `inventos.txt` línea por línea y lo muestra en consola. Si el archivo no existe, muestra un mensaje de error.

6. **LeerTodoElTexto()**

Esta función lee todo el contenido del archivo inventos.txt de una sola vez y lo muestra en consola.

7. **CopiarArchivo()**

Permite hacer una copia de seguridad del archivo inventos.txt y guardarlo en la carpeta Backup del laboratorio. Si el archivo original no existe, muestra un mensaje de error.

8. **MoverArchivo()**

Mueve el archivo inventos.txt desde su ubicación original a la carpeta ArchivosClasificados. Si el archivo no existe, muestra un mensaje de error.

9. **EliminarArchivo()**

Elimina el archivo inventos.txt si existe un respaldo del mismo en la carpeta Backup. Si no existe el archivo de respaldo, muestra un mensaje de error.

10. **ListarArchivosEnCarpeta(string rutaCarpeta)**

Lista todos los archivos presentes en la carpeta especificada. Si la carpeta no existe, muestra un mensaje de error.

Funcionamiento General del Programa

El programa utiliza un menú interactivo para que el usuario seleccione qué acción desea realizar. Este menú incluye las opciones para:

- 1. Crear el archivo inventos.txt.**
- 2. Agregar un nuevo invento al archivo.**
- 3. Leer los inventos línea por línea.**
- 4. Leer todo el contenido del archivo.**
- 5. Hacer una copia de seguridad del archivo.**

- 6. Mover el archivo a una carpeta de clasificación.**
- 7. Eliminar el archivo después de respaldarlo.**
- 8. Listar los archivos en la carpeta del laboratorio.**
- 9. Salir del programa.**

Para cada operación, el programa realiza comprobaciones para asegurarse de que los archivos y las carpetas necesarias existen antes de ejecutar las acciones correspondientes.

Estructura de Carpetas y Archivos

El proyecto maneja una estructura de directorios y archivos como sigue:

- **C:\LaboratorioAvengers**
 - **Carpeta principal que contiene las siguientes subcarpetas y archivos:**
 - **Backup**
 - **Carpeta para almacenar las copias de seguridad de los archivos.**
 - **ArchivosClasificados**
 - **Carpeta destinada a almacenar archivos clasificados.**
 - **ProyectosSecretos**
 - **Carpeta para proyectos secretos del laboratorio.**
 - **inventos.txt**
 - **Archivo de texto que almacena la lista de inventos registrados.**

Manejo de Errores

Se implementan bloques try-catch para capturar y manejar excepciones comunes que pueden ocurrir durante la ejecución del programa, tales como:

- **DirectoryNotFoundException:** Se lanza si se intenta acceder a una carpeta que no existe.
- **FileNotFoundException:** Se lanza si se intenta leer un archivo que no existe.
- **UnauthorizedAccessException:** Se lanza si no se tienen los permisos necesarios para realizar una operación en un archivo o carpeta.
- **IOException:** Se lanza si ocurre algún problema al interactuar con el sistema de archivos, como un problema de lectura o escritura.