PROGRAMA QUE GUARDA EN FICHEROS

Para este programa he utilizado tres clases : Alumno que tendrá el rol de modelo de datos para trabajar en la aplicación.

```
namespace Fichero
{
    10 referencias | Daniel Madrigal, Hace 2 días | 1 autor, 1 cambio
    public class Alumno
    {
        3 referencias | Daniel Madrigal, Hace 2 días | 1 autor, 1 cambio
        public int Id { get; set; }
        3 referencias | Daniel Madrigal, Hace 2 días | 1 autor, 1 cambio
        public string Nombre { get; set; }
        3 referencias | Daniel Madrigal, Hace 2 días | 1 autor, 1 cambio
        public string Apellidos { get; set; }
        3 referencias | Daniel Madrigal, Hace 2 días | 1 autor, 1 cambio
        public string Dni { get; set; }

        1 referencia | 0 cambios | 0 autores, 0 cambios
        public Alumno(int id , string nombre , string apellidos , string dni)
        {
            this.Id =id;
            this.Nombre = nombre;
            this.Apellidos = apellidos;
            this.Dni = dni;
        }
        0 referencias | Daniel Madrigal, Hace 2 días | 1 autor, 1 cambio
        public Alumno() { }
    }
}
```

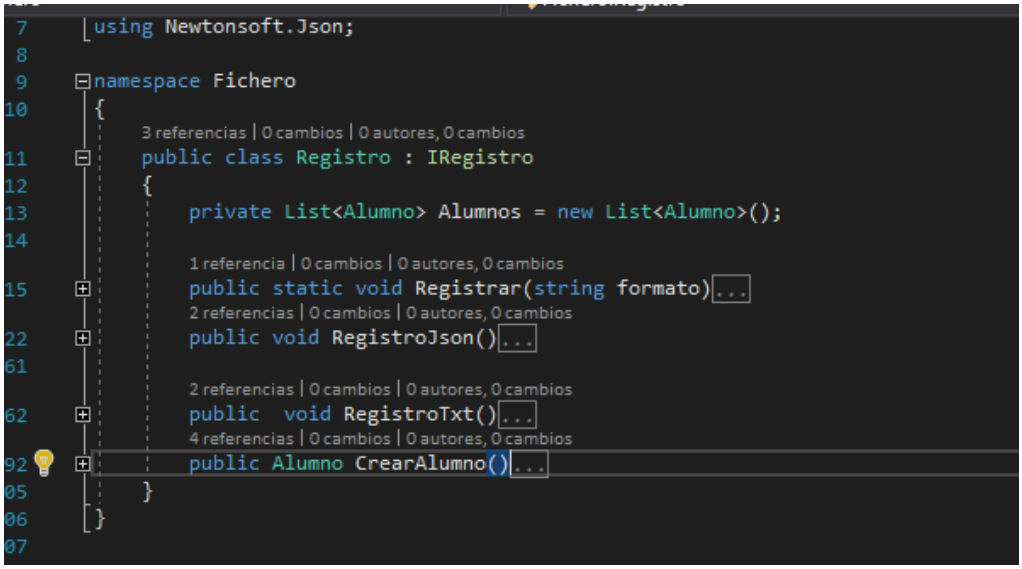
La clase Formato que define las variables de configuración ya que por defecto la primero vez que se ejecuta el programa es txt pero se puede cambiar guardando la config aun cerrando la aplicación.

```
2 referencias | 0 cambios | 0 autores, 0 cambios
public class Formato : IFormato
{
    3 referencias | 0 cambios | 0 autores, 0 cambios
    public enum Cambio { Si=1,No}
    2 referencias | 0 cambios | 0 autores, 0 cambios
    public string CambiarFormato()
    {
        string formato = ConfigurationManager.AppSettings["Formato"];
        Console.WriteLine($"El formato de registro es {formato} , desea cambiarlo? "+"\n"+"1.Si"+"\n"+"2.No");
        int cambio = Convert.ToInt32(Console.ReadLine());

        switch ((Cambio)cambio)
        {
            case Cambio.Si:
                if (formato.Equals("txt"))
                {
                    Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);
                    config.AppSettings.Settings["Formato"].Value = "json";
                    config.Save(ConfigurationSaveMode.Modified);
                    ConfigurationManager.RefreshSection("appSettings");
                    Console.WriteLine("Formato cambiado a json");
                    return formato = "json";
                }
                else
                {
                    Configuration config = ConfigurationManager.OpenExeConfiguration(ConfigurationUserLevel.None);
                    config.AppSettings.Settings["Formato"].Value = "txt";
                    config.Save(ConfigurationSaveMode.Modified);
                    ConfigurationManager.RefreshSection("appSettings");
                    Console.WriteLine("Formato cambiado a txt");
                    return formato = "txt";
                }

            case Cambio.No:
                return formato;
```

Y por ultimo la clase registro donde registramos los datos , si el fichero esta creado lo rellenamos y si no lo crea.



Como posibles mejoras faltaría sustituir las rutas de acceso por variables de entorno que el programa crea al compilar. Y también faltaría la implantación de test unitarios.