Ejercicio1:

```
using System.IO;

namespace Examen
{
	class Program
	{
	static void Main(string[] args)
	{
	StreamReader Original = new StreamReader("C:/Users/aleja/OneDrive/Escritorio/original.txt");
	StreamWriter Final = new StreamWriter("C:/Users/aleja/OneDrive/Escritorio/final.txt");
	string linea;
	do {
	linea = Original.ReadLine();
	if (linea != null) {

		linea = linea.Replace("Console.WriteLine", "printf");
		linea = linea.Replace("Main", "main");
		linea = linea.Replace("public", "");
		linea = linea.Replace("string", "char[80]");
		linea = linea.Replace("Console.ReadLine", "scanf");
		linea = linea.Replace("static", "");
		linea = linea.Replace("public", "");
		if (!linea.StartsWith("Using"))
		{
			Final.WriteLine(linea);
		}
	}
	} while (linea != null);

	Original.Close();
	Final.Close();


	}
	}
}
```

Ejercicio2:

```
using System;

namespace ConsoleApp1
{
	class Program
	{
		static void Main(string[] args)
		{
		EstadoJuego estadojuego = new EstadoJuego();
		ManejoEstado manejoestado = new ManejoEstado();

		estadojuego.Vidas = 3;
		estadojuego.Nivel = 5;
		estadojuego.Nombre = "alex";

		manejoestado.Guardarpartida(estadojuego);


		EstadoJuego estadojuegorecuperado = new EstadoJuego();

		estadojuegorecuperado = manejoestado.Recuperar();

		Console.WriteLine(estadojuegorecuperado.Nivel);
		Console.WriteLine(estadojuegorecuperado.Vidas);
		Console.WriteLine(estadojuegorecuperado.Nombre);

		Console.ReadLine();
	}
}
}

class ManejoEstado
	{
		public void Guardarpartida(EstadoJuego estadojuego)
		{
		StreamWriter fichero = new
	StreamWriter("C:\\Users\\aleja\\OneDrive\\Escritorio\\EstadoPartida.json");
		fichero.Write(JsonConvert.SerializeObject(estadojuego));
		fichero.Close();
		}
		public EstadoJuego Recuperar()
		{
		StreamReader fichero = new
	StreamReader("C:\\Users\\aleja\\OneDrive\\Escritorio\\EstadoPartida.json");
		string estadojuego = fichero.ReadToEnd();
		return JsonConvert.DeserializeObject<EstadoJuego>(estadojuego);
		fichero.Close();
		}
	}

	class EstadoJuego
	{
```

```
        private string nombre;
        private int vidas;
        private int nivel;

        public string Nombre { get => nombre; set => nombre = value; }
        public int Vidas { get => vidas; set => vidas = value; }
        public int Nivel { get => nivel; set => nivel = value; }
    }
```

Consulta 1:

```
SELECT Jugadores.nombre, TiposArma.Nombre FROM Jugadores LEFT OUTER JOIN TiposArma ON
Jugadores.CodigoArma = TiposArma.codigo;
```

Consulta 2

```
ALTER TABLE TiposArma ADD nombreplater varchar(10), FOREIGN KEY(nombreplater) REFERENCES
Jugadores(nombre) on delete cascade;
```

Consulta 3

```
select * from Jugadores where Jugadores.Nombre like '%u%'
 in
 (
   SELECT min(ataque) FROM TiposArma
)
```

Consulta 4

```
create trigger guardarfecha after insert on Jugadores for each row begin
   INSERT INTO Registro VALUES (datetime, new.nombre);
end;
```

Consulta 5

```
SELECT Jugadores.nombre, TiposArma.Nombre, Jugadores.CodigoArmadura FROM Jugadores LEFT OUTER
JOIN TiposArma ON Jugadores.CodigoArma = TiposArma.codigo
ORDER BY Jugadores.nombre DESC;
```