



SUBDIRECCIÓN ACADÉMICA  
INSTITUTO TECNOLÓGICO DE TIJUANA TOMAS AQUINO  
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Agosto - Diciembre 2025

CARRERA:

Ingeniería en Sistemas Computacionales

MATERIA:

Patrones de Diseño de Software

Grupo:

02:00 - 03:00 pm

Unidad:

IV

Actividad:

Examen Unidad 4

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Gonzalez Vazquez Melanie Estefania -21211956-

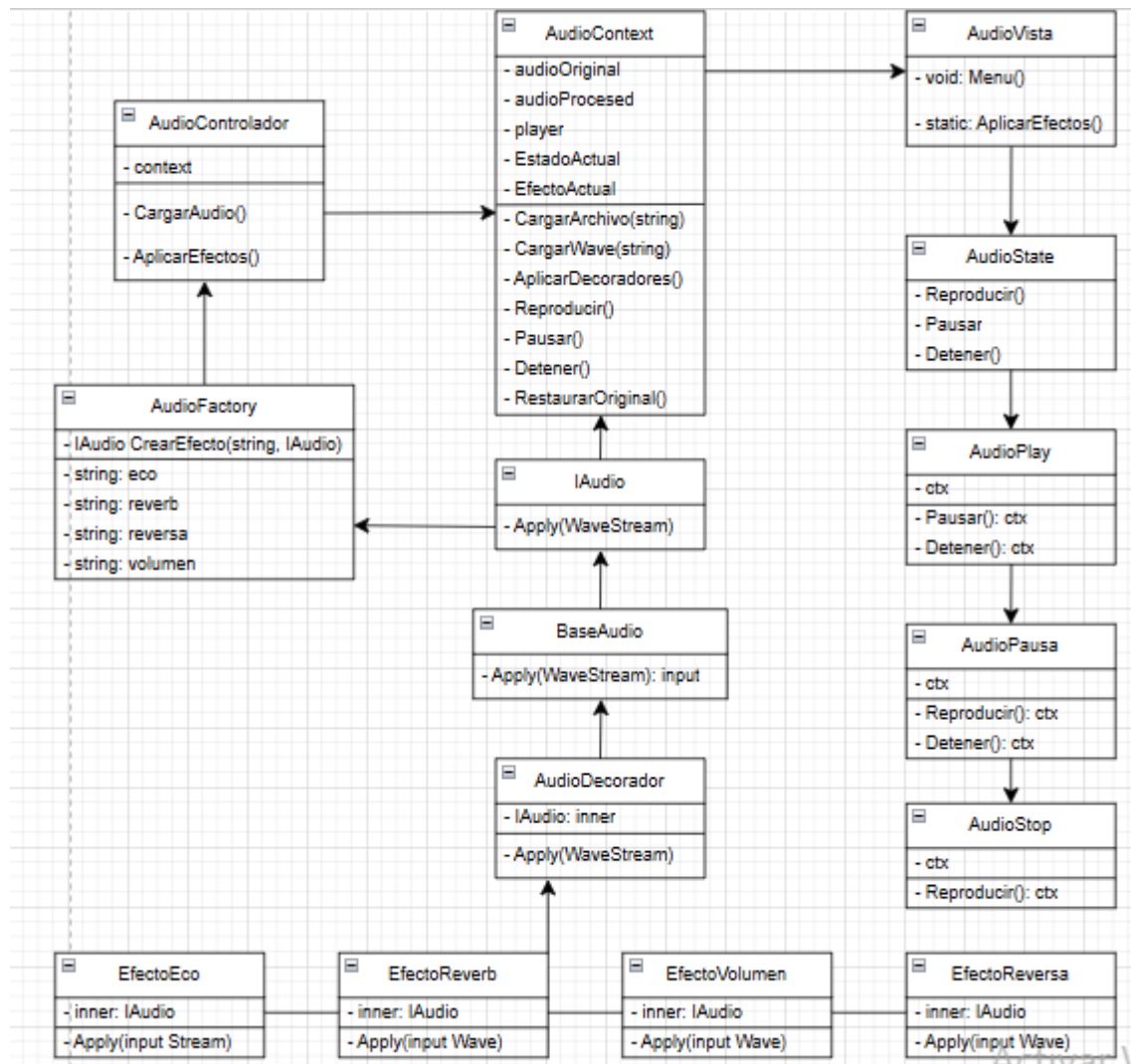
NOMBRE DEL DOCENTE:

Maribel Guerrero Luis

LUGAR Y FECHA:

Tijuana, Baja California, México a 11 de Diciembre del 2025

## Diagrama UML





## Código

### Program.cs

```
Examen_U4
Examen_U4_Patrones.Program
Main(string[] args)

1 using Examen_U4_Patrones.Controlador;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7 using Examen_U4_Patrones.Vista;
8
9 namespace Examen_U4_Patrones
10 {
11     0 referencias
12     public class Program
13     {
14         0 referencias
15         static void Main(string[] args)
16         {
17             AudioVista.Menu();
18         }
19     }
20 }
```

Buscar en Explorador de soluc...  
Solución "Examen\_U4" (1 de 1)  
Examen\_U4  
Properties  
Referencias  
Controlador  
Modelo  
Vista  
App.config  
packages.config  
C# Program.cs

### Controlador: AudioControlador.cs

```
Examen_U4
Examen_U4_Patrones.Controlador.Audio
CargarAudio(string ruta)

6 using Examen_U4_Patrones.Modelo;
7 using Examen_U4_Patrones.Modelo.Decorador;
8 using Examen_U4_Patrones.Modelo.Factory;
9 using Examen_U4_Patrones.Modelo.State;
10 using Examen_U4_Patrones.Vista;
11 using System.IO;
12
13 namespace Examen_U4_Patrones.Controlador
14 {
15     3 referencias
16     public class AudioControlador
17     {
18         private AudioContext context = new AudioContext();
19
20         1 referencia
21         public bool CargarAudio(string ruta)
22         {
23             return context.CargaArchivo(ruta);
24         }
25
26         1 referencia
27         public void Reproducir() => context.Reproducir();
28
29         1 referencia
30         public void Pausar() => context.Pausar();
31
32         1 referencia
33         public void Detener() => context.Detener();
34
35         1 referencia
36         public void RestaurarOriginal() => context.RestablecerOriginal();
37
38         1 referencia
39         public void AplicarEfectos(string[] efectos)
40         {
41             context.RestablecerOriginal();
42
43             foreach (string ef in efectos)
44             {
45                 string limpio = ef.Trim();
46                 var decorador = AudioFactory.CrearEfecto(limpio, context.EfectoActual);
47                 context.EfectoActual = decorador;
48             }
49
50             context.AplicarDecoradores();
51         }
52     }
53 }
```

Buscar en Explorador de soluc...  
Solución "Examen\_U4" (1 de 1)  
Examen\_U4  
Properties  
Referencias  
Controlador  
C# AudioControlador.cs  
Modelo  
Vista  
App.config  
packages.config  
C# Program.cs

### Modelo: AudioContext.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.AudioCon
CargarWave(string ruta)

7 using Examen_U4_Patrones.Modelo.Decorador;
8 using Examen_U4_Patrones.Modelo.State;
9 using NAudio.Wave;
10
11 namespace Examen_U4_Patrones.Modelo
12 {
13     9 referencias
14     public class AudioContext
15     {
16         9 referencias
17         public AudioEstado EstadoActual { get; set; }
18
19         4 referencias
20         public IAudio EfectoActual { get; set; } = new BaseAudio();
21
22         private WaveOutEvent player;
23         private WaveStream audioOriginal;
24         private WaveStream audioProcesado;
25         private string archivo;
26
27         1 referencia
28         public AudioContext()
29         {
30             EstadoActual = new AudioStop(this);
31         }
32
33         0 referencias
34         public WaveStream AudioStream => audioProcesado;
35
36         1 referencia
37         public bool CargaArchivo(string ruta)
38         {
39             try
40             {
41                 archivo = ruta;
42                 audioOriginal = CargarWave(ruta);
43                 audioProcesado = audioOriginal;
44
45                 player = new WaveOutEvent();
46                 player.Init(audioProcesado);
47
48                 return true;
49             }
50             catch { }
51         }
52     }
53 }
```

Buscar en Explorador de soluc...  
Solución "Examen\_U4" (1 de 1)  
Examen\_U4  
Properties  
Referencias  
Controlador  
Modelo  
Decorator  
Factory  
State  
C# AudioContext.cs  
Vista  
App.config  
packages.config  
C# Program.cs



```
Examen_U4
Examen_U4_Patrones.Modelo.AudioCon
CargarWave(string ruta)

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81

    catch
    {
        return false;
    }
}

1 referencia
public WaveStream CargarWave(string ruta)
{
    string ext = System.IO.Path.GetExtension(ruta).ToLower();

    if (ext == ".wav")
        return new WaveFileLoader(ruta);
    else if (ext == ".mp3")
        return new Mp3FileLoader(ruta);
    else
        throw new Exception("Formato invalido.");
}

1 referencia
public void AplicarDecoradores()
{
    audioProcesado = EfectoActual.Apply(audioOriginal);
    player.Stop();
    player.Init(audioProcesado);
}

1 referencia
public void Reproducir() => EstadoActual.Reproducir();
1 referencia
public void Pausar() => EstadoActual.Pausar();
1 referencia
public void Detener() => EstadoActual.Detener();

2 referencias
public void Play() => player.Play();
1 referencia
public void Pause() => player.Pause();
1 referencia
public void Stop() => player.Stop();

2 referencias
public void RestaurarOriginal()
{
    EfectoActual = new BaseAudio();
    audioProcesado = audioOriginal;
    player.Init(audioProcesado);
}
```

## Modelo: State: AudioEstado.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.State.Audi
Reproducir()

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Examen_U4_Patrones.Modelo.State
{
    4 referencias
    public interface AudioEstado
    {
        4 referencias
        void Reproducir();
        4 referencias
        void Pausar();
        4 referencias
        void Detener();
    }
}
```

## Modelo: State: AudioPausa.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.State.Audi
Detener()

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

using Examen_U4_Patrones.Controlador;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Examen_U4_Patrones.Modelo.State
{
    2 referencias
    public class AudioPausa : AudioEstado
    {
        13
        private AudioContext ctx;

        1 referencia
        public AudioPausa(AudioContext c) => ctx = c;

        2 referencias
        public void Reproducir()
        {
            17
            ctx.Play();
            ctx.EstadoActual = new AudioPlay(ctx);
        }

        2 referencias
        public void Pausar() { }

        2 referencias
        public void Detener()
        {
            25
            ctx.Stop();
            27
            ctx.EstadoActual = new AudioStop(ctx);
        }
    }
}
```



## Modelo: State: AudioPlay.cs

```
Examen_U4 Examen_U4_Patrones.Modelo.State.Audi Detener()
1 using Examen_U4_Patrones.Controlador;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Examen_U4_Patrones.Modelo.State
9 {
10     3 referencias
11     public class AudioPlay : AudioEstado
12     {
13         private AudioContext ctx;
14
15         2 referencias
16         public AudioPlay(AudioContext c) => ctx = c;
17
18         2 referencias
19         public void Reproducir() { }
20
21         2 referencias
22         public void Pausar()
23         {
24             ctx.Pause();
25             ctx.EstadoActual = new AudioPausa(ctx);
26         }
27
28         2 referencias
29         public void Detener()
30         {
31             ctx.Stop();
32             ctx.EstadoActual = new AudioStop(ctx);
33         }
34     }
35 }
```

## Modelo: State: AudioStop.cs

```
Examen_U4 Examen_U4_Patrones.Modelo.State.Audi Reproducir()
1 using Examen_U4_Patrones.Controlador;
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace Examen_U4_Patrones.Modelo.State
9 {
10     4 referencias
11     public class AudioStop : AudioEstado
12     {
13         private AudioContext ctx;
14
15         3 referencias
16         public AudioStop(AudioContext c) => ctx = c;
17
18         2 referencias
19         public void Reproducir()
20         {
21             ctx.Play();
22             ctx.EstadoActual = new AudioPlay(ctx);
23         }
24
25         2 referencias
26         public void Pausar() { }
27
28         2 referencias
29         public void Detener() { }
30     }
31 }
```

## Modelo: Factory: AudioFactory.cs

```
Examen_U4 Examen_U4_Patrones.Modelo.Factory.Al CrearEfecto(string tipo, IAudio baseEffect)
4 using System.Text;
5 using System.Threading.Tasks;
6 using Examen_U4_Patrones.Modelo.Decorador;
7 using NAudio.Wave;
8
9 namespace Examen_U4_Patrones.Modelo.Factory
10 {
11     1 referencia
12     public class AudioFactory
13     {
14         1 referencia
15         public static IAudio CrearEfecto(string tipo, IAudio baseEffect)
16         {
17             tipo = tipo.ToLower();
18
19             switch(tipo)
20             {
21                 case "eco":
22                     return new EfectoEco(baseEffect);
23                 case "reverb":
24                     return new EfectoReverb(baseEffect);
25                 case "volumen":
26                     return new EfectoVolumen(baseEffect);
27                 case "reversa":
28                     return new EfectoReversa(baseEffect);
29                 default:
30                     return baseEffect;
31             }
32         }
33     }
34 }
```



## Modelo: Decorador: IAudio.cs

```
Examen_U4 | Examen_U4_Patrones.Modelo.Decorador | Apply(WaveStream input)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using NAudio.Wave;
7
8 namespace Examen_U4_Patrones.Modelo.Decorador
9 {
10     11 referencias
11     public interface IAudio
12     {
13         10 referencias
14         WaveStream Apply(WaveStream input);
15     }
16 }
```

Buscar en Explorador de soluc...  
Solución "Examen\_U4" (1 de 1)  
Examen\_U4  
Properties  
Referencias  
Controlador  
Modelo  
Decorador  
AudioDecorator.cs  
BaseAudio.cs  
EfectoEco.cs  
EfectoReverb.cs  
EfectoReversa.cs  
EfectoVolumen.cs  
IAudio.cs  
Factory  
State  
AudioContext.cs  
Vista  
App.config  
packages.config  
Program.cs

## Modelo: Decorador: BaseAudio.cs

```
Examen_U4 | Examen_U4_Patrones.Modelo.Decorador | Apply(WaveStream input)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using NAudio.Wave;
7
8 namespace Examen_U4_Patrones.Modelo.Decorador
9 {
10     2 referencias
11     public class BaseAudio : IAudio
12     {
13         5 referencias
14         public WaveStream Apply(WaveStream input) => input;
15     }
16 }
```

Buscar en Explorador de soluc...  
Solución "Examen\_U4" (1 de 1)  
Examen\_U4  
Properties  
Referencias  
Controlador  
Modelo  
Decorador  
AudioDecorator.cs  
BaseAudio.cs  
EfectoEco.cs  
EfectoReverb.cs  
EfectoReversa.cs  
EfectoVolumen.cs  
IAudio.cs  
Factory  
State  
AudioContext.cs  
Vista  
App.config  
packages.config  
Program.cs

## Modelo: Decorador: AudioDecorator.cs

```
Examen_U4 | Examen_U4_Patrones.Modelo.Decorador | AudioDecorator(IAudio innerAudio)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6 using NAudio.Wave;
7
8 namespace Examen_U4_Patrones.Modelo.Decorador
9 {
10     9 referencias
11     public abstract class AudioDecorator : IAudio
12     {
13         protected IAudio inner;
14
15         4 referencias
16         protected AudioDecorator(IAudio innerAudio)
17         {
18             this.inner = innerAudio;
19         }
20
21         9 referencias
22         public abstract WaveStream Apply(WaveStream input);
23     }
24 }
```

Buscar en Explorador de soluc...  
Solución "Examen\_U4" (1 de 1)  
Examen\_U4  
Properties  
Referencias  
Controlador  
Modelo  
Decorador  
AudioDecorator.cs  
BaseAudio.cs  
EfectoEco.cs  
EfectoReverb.cs  
EfectoReversa.cs  
EfectoVolumen.cs  
IAudio.cs  
Factory  
State  
AudioContext.cs  
Vista  
App.config  
packages.config  
Program.cs



## Modelo: Decorador: EfectoEco.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.Decorador
ToWaveStream(IWaveProvider wave)

using System;
using NAudio.Wave;
using NAudio.Wave.SampleProviders;

namespace Examen_U4_Patrones.Modelo.Decorador
{
    public class EfectoEco : AudioDecorator
    {
        public EfectoEco(Audio inner) : base(inner) { }

        public override WaveStream Apply(WaveStream input)
        {
            var original = input.ToSampleProvider();
            var delayedSource = input.ToSampleProvider();
            var delay = new OffsetSampleProvider(delayedSource)
            {
                DelayBy = TimeSpan.FromMilliseconds(100)
            };
            var delayVol = new VolumeSampleProvider(delay)
            {
                Volume = 0.5f
            };
            var mixer = new MixingSampleProvider(WaveFormat.CreateFromFormat(input.WaveFormat,
                input.WaveFormat.SampleRate,
                input.WaveFormat.Channels));
            mixer.AddMixerInput(original);
            mixer.AddMixerInput(delayVol);
            return mixer.ToWaveProvider().ToWaveStream();
        }
    }

    public static class WaveExtensions
    {
        public static WaveStream ToWaveStream(this IWaveProvider wave)
        {
            return new RawSourceWaveStream(wave, wave.WaveFormat);
        }
    }
}
```

## Modelo: Decorador: EfectoReverb.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.Decorador
EfectoReverb(IAudio inner)

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NAudio.Wave;
using NAudio.Wave.SampleProviders;

namespace Examen_U4_Patrones.Modelo.Decorador
{
    public class EfectoReverb : AudioDecorator
    {
        public EfectoReverb(IAudio inner) : base(inner) { }

        public override WaveStream Apply(WaveStream input)
        {
            var processed = inner.Apply(input);
            processed.Position = 0;
            byte[] data = new byte[processed.Length];
            processed.Read(data, 0, data.Length);
            Array.Reverse(data);
            return new RawSourceWaveStream(data, 0, data.Length, processed.WaveFormat);
        }
    }
}
```

## Modelo: Decorador: EfectoReversa.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.Decorador
WaveFormat

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NAudio.Wave;
using NAudio.Wave.SampleProviders;

namespace Examen_U4_Patrones.Modelo.Decorador
{
    public class EfectoReversa : AudioDecorator
    {
        public EfectoReversa(IAudio inner) : base(inner) { }

        public override WaveStream Apply(WaveStream input)
        {
            var processed = inner.Apply(input);
            var sample = processed.ToSampleProvider();

            var delay1 = new OffsetSampleProvider(sample) { DelayBy = TimeSpan.FromMilliseconds(40) };
            var delay2 = new OffsetSampleProvider(sample) { DelayBy = TimeSpan.FromMilliseconds(80) };
            var delay3 = new OffsetSampleProvider(sample) { DelayBy = TimeSpan.FromMilliseconds(120) };

            var d1 = new VolumeSampleProvider(delay1) { Volume = 0.4f };
            var d2 = new VolumeSampleProvider(delay2) { Volume = 0.25f };
            var d3 = new VolumeSampleProvider(delay3) { Volume = 0.15f };

            var mixer = new MixingSampleProvider(sample.WaveFormat)
            {
                ReadFully = true
            };
            mixer.AddMixerInput(sample);
            mixer.AddMixerInput(d1);
            mixer.AddMixerInput(d2);
            mixer.AddMixerInput(d3);
        }
    }
}
```





```
Examen_U4
Examen_U4_Patrones.Modelo.Decorador
WaveFormat

37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

return new SampleProviderToWaveStream(mixer);

}

2 referencias
public class SampleProviderToWaveStream : WaveStream
{
    private readonly ISampleProvider source;
    private readonly WaveFormat waveFormat;
    private long position;

    1 referencia
    public SampleProviderToWaveStream(ISampleProvider source)
    {
        this.source = source;
        this.waveFormat = source.WaveFormat;
    }

    0 referencias
    public override WaveFormat WaveFormat => waveFormat;

    0 referencias
    public override long Length => long.MaxValue;

    0 referencias
    public override long Position
    {
        get => position;
        set => position = value;
    }

    0 referencias
    public override int Read(byte[] buffer, int offset, int count)
    {
        int samplesRequired = count / 4;
        float[] sampleBuffer = new float[samplesRequired];
        int samplesRead = source.Read(sampleBuffer, 0, samplesRequired);
        Buffer.BlockCopy(sampleBuffer, 0, buffer, offset, samplesRead * 4);
    }
}
```

## Modelo: Decorador: EfectoVolumen.cs

```
Examen_U4
Examen_U4_Patrones.Modelo.Decorador
Apply(WaveStream input)

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using NAudio.Wave;
using NAudio.Wave.SampleProviders;

namespace Examen_U4_Patrones.Modelo.Decorador
{
    2 referencias
    public class EfectoVolumen : AudioDecorator
    {
        1 referencia
        public EfectoVolumen(IAudio inner) : base(inner) { }

        6 referencias
        public override WaveStream Apply(WaveStream input)
        {
            var processed = inner.Apply(input);
            var vol = new WaveChannel32(processed);
            vol.Volume = 0.5f;
            return vol;
        }
    }
}
```

## Modelo: Vista: AudioVista.cs

```
Examen_U4
Examen_U4_Patrones.Vista.AudioVista
Menu()

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38

using Examen_U4_Patrones.Controlador;
using Examen_U4_Patrones.Modelo;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Examen_U4_Patrones.Vista
{
    1 referencia
    public static class AudioVista
    {
        1 referencia
        public static void Menu()
        {
            AudioControlador controller = new AudioControlador();
            //AudioContext context = new AudioContext();

            Console.WriteLine("=== GESTOR DE AUDIO ===");
            Console.WriteLine("Ingresa ruta del archivo (.wav o .mp3): ");
            string ruta = Console.ReadLine();

            if (!controller.CargarAudio(ruta))
            {
                Console.WriteLine("No se pudo cargar el archivo.");
                return;
            }

            while (true)
            {
                Console.WriteLine("\n=== MENU ===");
                Console.WriteLine("1) Reproducir");
                Console.WriteLine("2) Pausar");
                Console.WriteLine("3) Detener");
                Console.WriteLine("4) Aplicar efectos (mezclador)");
                Console.WriteLine("5) Restaurar audio original");
                Console.WriteLine("6) Salir");
                Console.WriteLine("Opción: ");
                string op = Console.ReadLine();
            }
        }
    }
}
```





```
Examen_U4
Examen_U4_Patrones.Vista.AudioVista
Menu()

39
40
41 switch (op)
42 {
43     case "1": controller.Reproducir(); break;
44     case "2": controller.Pausar(); break;
45     case "3": controller.Detener(); break;
46     case "4": AplicarEfectos(controller); break;
47     case "5": controller.RestaurarOriginal(); break;
48     case "6": return;
49     default: Console.WriteLine("Opción inválida"); break;
50 }
51
52
53 1 referencia
54 static void AplicarEfectos(AudioControlador controller)
55 {
56     Console.WriteLine("\nSelecciona efectos separados por coma:");
57     Console.WriteLine("eco, reverb, volumen, reversa");
58     Console.WriteLine("Ejemplo: eco, volumen: ");
59
60     string entrada = Console.ReadLine();
61     string[] lista = entrada.ToLower().Split(',');
62
63     controller.AplicarEfectos(lista);
64
65     Console.WriteLine("Efectos aplicados.");
66 }
67
68 }
```

## Ejecución

```
C:\Users\dell\source\repos\Examen_U4\Examen_U4\bin\Debug\Examen_U4.exe

=== GESTOR DE AUDIO ===
Ingresa ruta del archivo (.wav o .mp3): C:\Users\dell\Music\luna.mp3_
```

```
C:\Users\dell\source\repos\Examen_U4\Examen_U4\bin\Debug\Examen_U4.exe

=== GESTOR DE AUDIO ===
Ingresa ruta del archivo (.wav o .mp3): C:\Users\dell\Music\luna.mp3

=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 1

=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: _
```



```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 2
```

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 3
```

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 
```

```
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 4
```

Selecciona efectos separados por coma:  
eco, reverb, volumen, reversa  
Ejemplo: eco, volumen: reversa  
Efectos aplicados.

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 1
```

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 
```

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 5
```

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 1
```

```
=== MENÚ ===
1) Reproducir
2) Pausar
3) Detener
4) Aplicar efectos (mezclador)
5) Restaurar audio original
6) Salir
Opción: 
```