

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:
Agosto - Diciembre 2025

MATERIA:
Patrones de diseño

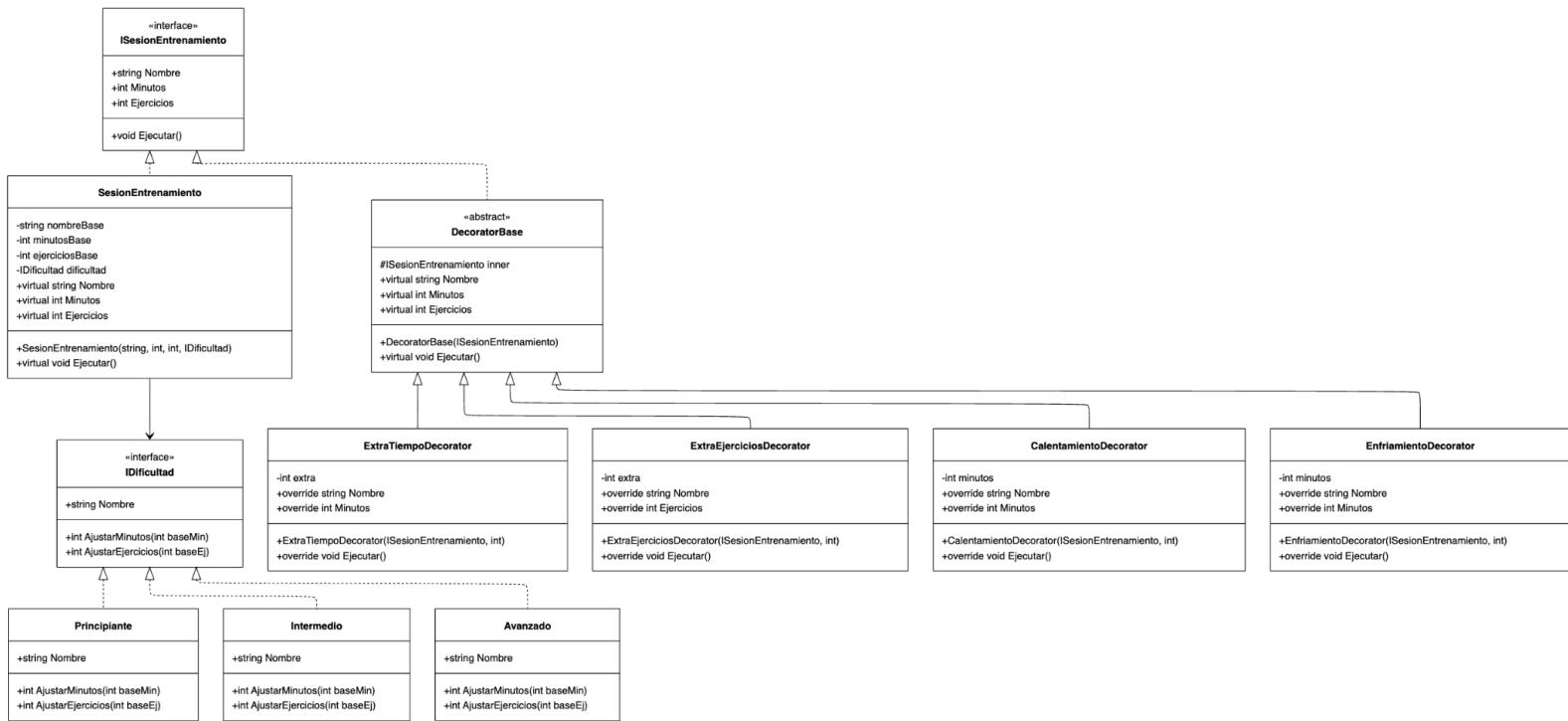
TÍTULO ACTIVIDAD:
Examen U3

UNIDAD A EVALUAR:
Unidad 3

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:
Vizuet Acosta Andre 21212372

NOMBRE DEL MAESTRO (A):
Maribel Guerrero Luis

UML



Program

```

1  using System;
2  using Entrenamiento.Bridge;
3  using Entrenamiento.Main;
4  using Entrenamiento.Decorators;
5
6  namespace Entrenamiento
7  {
8      0 references
9      class Program
10     {
11         0 references
12         static void Main(string[] args)
13         {
14             Console.WriteLine("== App de Entrenamiento ==");
15
16             IDificultad dificultad;
17             while (true)
18             {
19                 Console.WriteLine("\nElige dificultad:");
20                 Console.WriteLine("1) Principiante");
21                 Console.WriteLine("2) Intermedio");
22                 Console.WriteLine("3) Avanzado");
23                 Console.Write("Opción: ");
24                 string op = Console.ReadLine() ?? "";
25                 if (op == "1") { dificultad = new Principiante(); break; }
26                 if (op == "2") { dificultad = new Intermedio(); break; }
27                 if (op == "3") { dificultad = new Avanzado(); break; }
28
29             ISesionEntrenamiento sesion;
30             const int baseMinCardio = 30, baseEjCardio = 8, baseMinFuerza = 45, baseEjFuerza = 10, baseMinYoga = 60, baseEjYoga = 6;
  
```

```

30     while (true)
31     {
32         Console.WriteLine("\nElige tipo de sesión:");
33         Console.WriteLine($"1) Cardio {dificultad.AjustarMinutos(baseMinCardio)} min, {dificultad.AjustarEjercicios(baseEjCardio)} ejercicios");
34         Console.WriteLine($"2) Fuerza {dificultad.AjustarMinutos(baseMinFuerza)} min, {dificultad.AjustarEjercicios(baseEjFuerza)} ejercicios");
35         Console.WriteLine($"3) Yoga {dificultad.AjustarMinutos(baseMinYoga)} min, {dificultad.AjustarEjercicios(baseEjYoga)} ejercicios");
36         Console.Write("Opción: ");
37         string op = Console.ReadLine() ?? "";
38         if (op == "1") { sesion = new SesionEntrenamiento("Cardio", baseMinCardio, baseEjCardio, dificultad); break; }
39         if (op == "2") { sesion = new SesionEntrenamiento("Fuerza", baseMinFuerza, baseEjFuerza, dificultad); break; }
40         if (op == "3") { sesion = new SesionEntrenamiento("Yoga", baseMinYoga, baseEjYoga, dificultad); break; }
41     }
42
43     Console.Write("\nAgregar mas tiempo? (s/n): ");
44     if (Console.ReadLine()?.Trim().ToLower() == "s")
45     {
46         Console.WriteLine("Minutos extra: ");
47         if (int.TryParse(Console.ReadLine(), out int extraMins) && extraMins > 0)
48             sesion = new ExtraTiempoDecorator(sesion, extraMins);
49     }
50
51     Console.Write("\nAgregar mas ejercicios? (s/n): ");
52     if (Console.ReadLine()?.Trim().ToLower() == "s")
53     {
54         Console.WriteLine("Cantidad extra de ejercicios: ");
55         if (int.TryParse(Console.ReadLine(), out int extraEj) && extraEj > 0)
56             sesion = new ExtraEjerciciosDecorator(sesion, extraEj);
57     }
58
59     Console.Write("\nAgregar calentamiento? (s/n): ");
60     if (Console.ReadLine()?.Trim().ToLower() == "s")
61     {
62         Console.WriteLine("Duración del calentamiento (min): ");
63         if (int.TryParse(Console.ReadLine(), out int mins) && mins > 0)
64             sesion = new CalentamientoDecorator(sesion, mins);
65     }
66
67     Console.WriteLine("\nAgregar enfriamiento? (s/n): ");
68     if (Console.ReadLine()?.Trim().ToLower() == "s")
69     {
70         Console.WriteLine("Duración del enfriamiento (min): ");
71         if (int.TryParse(Console.ReadLine(), out int mins) && mins > 0)
72             sesion = new EnfriamientoDecorator(sesion, mins);
73     }
74     Console.WriteLine($"\nIniciando {sesion.Nombre}, nivel {dificultad.Nombre}, duracion de {sesion.Minutos} min, con {sesion.Ejercicios} ejercicios\n");
75     sesion.Ejecutar();
76 }
77 }
78 }
```

ISesionEntrenamiento

```
1  namespace Entrenamiento.Main
2  {
3      |
4      |     9 references
5      |     public interface ISesionEntrenamiento
6      |     {
7      |         |
8      |         |     12 references
9      |         |     string Nombre { get; }
10     |        |
11     |        |     10 references
12     |        |     int Minutos { get; }
13     |        |
14     |        |     6 references
15     |        |     int Ejercicios { get; }
16     |        |
17     |        |     12 references
18     |        |     void Ejecutar();
19     |
20 }
```

SesionEntrenamiento

```
1  using System;
2  using Entrenamiento.Bridge;
3
4  namespace Entrenamiento.Main
5  {
6      4 references
7      public class SesionEntrenamiento : ISesionEntrenamiento
8      {
9          2 references
10         private readonly string nombreBase;
11         2 references
12         private readonly int minutosBase;
13         2 references
14         private readonly int ejerciciosBase;
15         3 references
16         private readonly IDificultad dificultad;
17
18         3 references
19         public SesionEntrenamiento(string nombre, int minutosBase, int ejerciciosBase, IDificultad dificultad)
20         {
21             this.nombreBase = nombre;
22             this.minutosBase = minutosBase;
23             this.ejerciciosBase = ejerciciosBase;
24             this.dificultad = dificultad;
25         }
26
27         7 references
28         public virtual string Nombre => nombreBase;
29         6 references
30         public virtual int Minutos => dificultad.AjustarMinutos(minutosBase);
31         4 references
32         public virtual int Ejercicios => dificultad.AjustarEjercicios(ejerciciosBase);
33
34         3 references
35         public virtual void Ejecutar()
36         {
37             Console.WriteLine($"Terminaste la sesion.");
38         }
39     }
40 }
```

IDificultad

```
1  namespace Entrenamiento.Bridge
2  {
3      public interface IDificultad
4      {
5          string Nombre { get; }
6          int AjustarMinutos(int baseMin);
7          int AjustarEjercicios(int baseEj);
8      }
9 }
```

Principiante

```
1  namespace Entrenamiento.Bridge
2  {
3      public class Principiante : IDificultad
4      {
5          public string Nombre => "Principiante";
6          public int AjustarMinutos(int baseMin) => (int)Math.Floor(baseMin * 0.8);
7          public int AjustarEjercicios(int baseEj) => baseEj * 0.8 > 0 ? (int)Math.Floor(baseEj * 0.8) : 1;
8      }
9 }
```

Intermedio

```
1  namespace Entrenamiento.Bridge
2  {
3      1 reference
4      public class Intermedio : IDificultad
5      {
6          2 references
7          public string Nombre => "Intermedio";
8          5 references
9          public int AjustarMinutos(int baseMin) => baseMin;
10         5 references
11         public int AjustarEjercicios(int baseEj) => baseEj;
12     }
13 }
```

Avanzado

```
1  namespace Entrenamiento.Bridge
2  {
3      1 reference
4      public class Avanzado : IDificultad
5      {
6          2 references
7          public string Nombre => "Avanzado";
8          5 references
9          public int AjustarMinutos(int baseMin) => (int)Math.Floor(baseMin * 1.3);
10         5 references
11         public int AjustarEjercicios(int baseEj) => (int)Math.Floor(baseEj * 1.3);
12     }
13 }
```

DecoratorBase

```
1  using Entrenamiento.Main;
2
3  namespace Entrenamiento.Decorators
4  {
5      9 references
6      public abstract class DecoratorBase : ISesionEntrenamiento
7      {
8          13 references
9          protected readonly ISesionEntrenamiento inner;
10         4 references
11         protected DecoratorBase(ISesionEntrenamiento inner) => this.inner = inner;
12
13         11 references
14         public virtual string Nombre => inner.Nombre;
15         9 references
16         public virtual int Minutos => inner.Minutos;
17         5 references
18         public virtual int Ejercicios => inner.Ejercicios;
19
20         11 references
21         public virtual void Ejecutar() => inner.Ejecutar();
22     }
23 }
```

CalentamientoDecorator

```
1  using System;
2  using Entrenamiento.Main;
3
4  namespace Entrenamiento.Decorators
5  {
6      2 references
7      public class CalentamientoDecorator : DecoratorBase
8      {
9          3 references
10         private readonly int minutos;
11
12         1 reference
13         public CalentamientoDecorator(ISesionEntrenamiento inner, int minutos)
14             : base(inner) => this.minutos = minutos;
15
16         8 references
17         public override string Nombre => $"{inner.Nombre} + Calentamiento";
18
19         7 references
20         public override int Minutos => inner.Minutos + minutos;
21
22         8 references
23         public override void Ejecutar()
24         {
25             Console.WriteLine($"Calentamiento: {minutos} minutos para de comenzar.\n");
26             base.Ejecutar();
27         }
28     }
29 }
```

EnfriamientoDecorator

```
1  using System;
2  using Entrenamiento.Main;
3
4  namespace Entrenamiento.Decorators
5  {
6      2 references
7      public class EnfriamientoDecorator : DecoratorBase
8      {
9          3 references
10         private readonly int minutos;
11
12         1 reference
13         public EnfriamientoDecorator(ISesionEntrenamiento inner, int minutos)
14             : base(inner) => this.minutos = minutos;
15
16         8 references
17         public override string Nombre => $"{inner.Nombre} + Enfriamiento";
18         7 references
19         public override int Minutos => inner.Minutos + minutos;
20
21         8 references
22         public override void Ejecutar()
23         {
24             base.Ejecutar();
25             Console.WriteLine($"Enfriamiento: {minutos} minutos de estiramiento y respiración.\n");
26         }
27     }
28 }
```

ExtraEjerciciosDecorator

```
1  using System;
2  using Entrenamiento.Main;
3
4  namespace Entrenamiento.Decorators
5  {
6      2 references
7      public class ExtraEjerciciosDecorator : DecoratorBase
8      {
9          3 references
10         private readonly int extra;
11         1 reference
12         public ExtraEjerciciosDecorator(ISesionEntrenamiento inner, int extra) : base(inner)
13         {
14             this.extra = extra;
15         }
16
17         8 references
18         public override string Nombre => $"{inner.Nombre} + Ejercicios";
19         5 references
20         public override int Ejercicios => inner.Ejercicios + extra;
21
22         8 references
23         public override void Ejecutar()
24         {
25             Console.WriteLine($"+ {extra} ejercicios extra\n");
26             base.Ejecutar();
27         }
28     }
29 }
```

ExtraTiempoDecorator

```
1  using System;
2  using Entrenamiento.Main;
3
4  namespace Entrenamiento.Decorators
5  {
6      2 references
7      public class ExtraTiempoDecorator : DecoratorBase
8      {
9          3 references
10         private readonly int extra;
11
12         1 reference
13         public ExtraTiempoDecorator(ISesionEntrenamiento inner, int extra) : base(inner)
14         {
15             this.extra = extra;
16
17             8 references
18             public override string Nombre => $"{inner.Nombre} + Tiempo";
19             7 references
20             public override int Minutos => inner.Minutos + extra;
21
22             8 references
23             public override void Ejecutar()
24             {
25                 Console.WriteLine($"+ {extra} min extra\n");
26                 base.Ejecutar();
27             }
28         }
29     }
```

Ejecución

```
ExamenUnidad3_Patrones_VizuetAcostaAndre/src on ✘ main via .NET v9.0.102 🐄 net9.0
• dotnet run
== App de Entrenamiento ==

Elige dificultad:
1) Principiante
2) Intermedio
3) Avanzado
Opción: 1

Elige tipo de sesión:
1) Cardio 24 min, 6 ejercicios
2) Fuerza 36 min, 8 ejercicios
3) Yoga 48 min, 4 ejercicios
Opción: 1

Agregar mas tiempo? (s/n): s
Minutos extra: 1

Agregar mas ejercicios? (s/n): s
Cantidad extra de ejercicios: 1

Agregar calentamiento? (s/n): s
Duración del calentamiento (min): 5

Agregar enfriamiento? (s/n): s
Duracion del enfriamiento (min): 5

Iniciando Cardio + Tiempo + Ejercicios + Calentamiento + Enfriamiento, nivel Principiante, duracion de 35 min, con 7 ejercicios
Calentamiento: 5 minutos para de comenzar.

+ 1 ejercicios extra

+ 1 min extra

Terminaste la sesion.
Enfriamiento: 5 minutos de estiramiento y respiración.
```

Conclusión

El patrón bridge separó la abstracción de las sesiones de la implementación de los niveles de dificultad, y con el patrón decorador agrega funcionalidades extras a las sesiones de forma dinámica como el tiempo y ejercicios adicionales, sin modificar la estructura base.