



**Instituto Tecnológico de Tijuana, Departamento de Sistemas y Computación
Tijuana, Baja California, México**

Materia:

Patrones de Diseño

Semestre:

8

Unidad:

III

Docente:

Maribel Guerrero Luis

Alumno:

Leonel Martinez Huitron

Número de control:

22210777

Fecha:

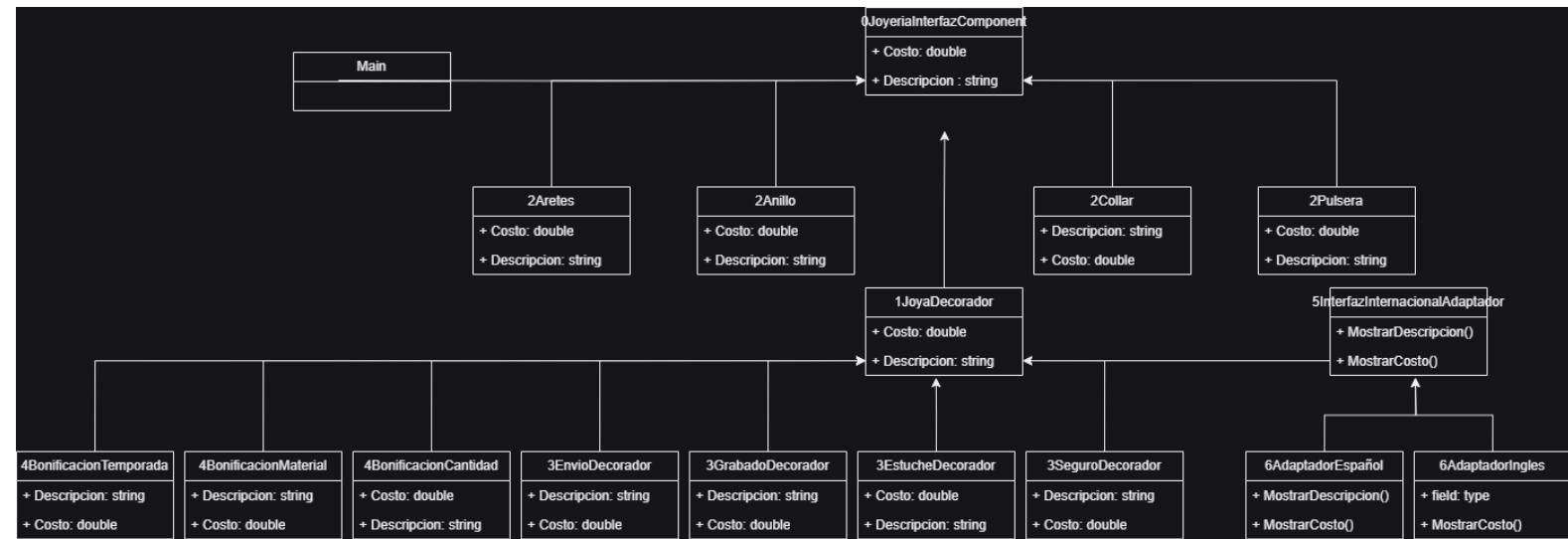
Viernes, 14 de noviembre del 2025





EXAMEN UNIDAD 3

DIAGRAMA UML





CÓDIGO

TECNOLÓGICO
NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE TIJUANA





```
9      internal class Program
10     {
11         static void Main(string[] args)
12         {
13             bool joyaValido = false, personalizado = true;
14             string opcion = "";
15             _JoyeriaInterfazComponent joya = null;
16             while (!joyaValido)
17             {
18                 Console.WriteLine("-----");
19                 Console.WriteLine("\t\tJOYERIA CHULA VISTA");
20                 Console.WriteLine("-----\n");
21
22                 Console.WriteLine("-----");
23                 Console.WriteLine("Bienvenido a Joyeria Chula Vista");
24                 Console.WriteLine("-----\n");
25
26                 Console.WriteLine("1. Anillo\t\t\t$250.00");
27                 Console.WriteLine("2. Aretes\t\t\t$300.00");
28                 Console.WriteLine("3. Collar\t\t\t$350.00");
29                 Console.WriteLine("4. Pulsera\t\t\t$400.00");
30                 Console.WriteLine("Elige una opción:");
31                 opcion = Console.ReadLine();
32                 Console.Clear();
33                 switch (opcion)
34                 {
35                     case "1":
36                         joya = new _Anillo();
37                         joyaValido = true;
38                         break;
39                     case "2":
40                         joya = new _Aretes();
41                         joyaValido = true;
42                         break;
43                     case "3":
44                         joya = new _Collar();
45                         joyaValido = true;
46                         break;
47                     case "4":
48                         joya = new _Pulsera();
49                         joyaValido = true;
50                         break;
51                     default:
52                         Console.WriteLine("Opción inválida. Intente de nuevo.");
53                         Console.ReadKey();
54                         Console.Clear();
55                         break;
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
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111

        Console.WriteLine("-----");
        Console.WriteLine($"\\tJOYA SELECCIONADO\\n{joya.Descripcion}");

        Console.WriteLine("\n¿Deseas personalizar tu JOYA?");
        Console.WriteLine("1. Sí/\n2. No");
        Console.WriteLine("Elige una opción:");
        opcion = Console.ReadLine();
        Console.Clear();
        bool deseaEnvio = false;
        if (opcion == "1")
        {
            while (personalizado)
            {
                Console.WriteLine("\nOpciones de personalización:");
                Console.WriteLine("1. Agregar Estuche Lujoso");
                Console.WriteLine("2. Agregar Grabado");
                Console.WriteLine("3. Agregar Seguro");
                Console.WriteLine("4. Finalizar personalización");

                Console.WriteLine("\nElige una opción:");
                opcion = Console.ReadLine();

                switch (opcion)
                {
                    case "1":
                        joya = new _3EstucheDecorador(joya);
                        Console.WriteLine("Estuche lujoso añadido.");
                        break;
                    case "2":
                        joya = new _3GrabadoDecorador(joya);
                        Console.WriteLine("Grabado personalizado añadido.");
                        break;
                    case "3":
                        joya = new _3SeguroDecorador(joya);
                        Console.WriteLine("Seguro de protección añadido.");
                        break;
                    case "4":
                        personalizado = false;
                        break;
                    default:
                        Console.WriteLine("Opción inválida. Intente de nuevo.");
                        break;
                }
                Console.ReadKey();
                Console.Clear();
            }
            Console.WriteLine("¿Te gustaría un envío a domicilio?");
            Console.WriteLine("1. Sí/\n2. No");
            Console.WriteLine("Elige una opción:");
            opcion = Console.ReadLine();
        }
```



```
111     opcion = Console.ReadLine();
112     switch (opcion)
113     {
114         case "1":
115             deseaEnvio = true;
116             if (deseaEnvio)
117             {
118                 joya = new _3EnvioDecorador(joya);
119             }
120             break;
121         case "2":
122             deseaEnvio = false;
123             Console.WriteLine("¡Disfruta tu pedido!");
124             break;
125         default:
126             Console.WriteLine("Opción inválida. Intente de nuevo.");
127             break;
128     }
129     Console.ReadKey();
130     Console.Clear();
131 }
132 Console.WriteLine("¿Deseas aplicar bonificación por material?");
133 Console.WriteLine("1. Sí\n2. No");
134 opcion = Console.ReadLine();
135 if (opcion == "1")
136 {
137     Console.WriteLine("Ingresa el tipo de material (Oro, Plata, Platino):");
138     string material = Console.ReadLine();
139     joya = new _4BonificacionMaterial(joya, material);
140     Console.Clear();
141 }
142
143 Console.WriteLine("¿Deseas aplicar bonificación por temporada?");
144 Console.WriteLine("1. Sí\n2. No");
145 opcion = Console.ReadLine();
146 if (opcion == "1")
147 {
148     Console.WriteLine("Ingresa la temporada (Navidad, San Valentín, Verano):");
149     string temporada = Console.ReadLine();
150     joya = new _4BonificacionTemporada(joya, temporada);
151     Console.Clear();
152 }
153
154 Console.WriteLine("¿Cuántas piezas deseas comprar?");
155 int cantidad = int.Parse(Console.ReadLine());
156 joya = new _4BonificacionCantidad(joya, cantidad);
157 Console.Clear();
158
159 Console.WriteLine("¿Deseas ver el resumen en otro idioma?");
160 Console.WriteLine("1. Español\n2. Ingles");
161 Console.WriteLine("Elige una opción:");
162 opcion = Console.ReadLine();
163 string etiquetaIdioma = "";
```



```
163     string etiquetaIdioma = "";
164
165     _5InterfazInternacionalAdaptador joyaTraducida;
166
167     switch (opcion)
168     {
169         case "1":
170             joyaTraducida = new _6AdaptadorEspañol(joya);
171             etiquetaIdioma = "EN ESPAÑOL";
172             break;
173         case "2":
174             joyaTraducida = new _6AdaptadorIngles(joya);
175             etiquetaIdioma = "IN ENGLISH";
176             break;
177         default:
178             Console.WriteLine("Idioma no reconocido. Se mostrará en español.");
179             joyaTraducida = new _6AdaptadorEspañol(joya);
180             etiquetaIdioma = "EN ESPAÑOL";
181             break;
182     }
183
184     Console.WriteLine("-----");
185     Console.WriteLine($"---\t\t\tORDER SUMMARY ({etiquetaIdioma})\t\t---");
186     Console.WriteLine("-----");
187     joyaTraducida.MostrarDescripcion();
188     joyaTraducida.MostrarCosto();
189
190     Console.WriteLine("\nThank you for your purchase. Come back soon to Joyeria Chula Vista!");
191     Console.WriteLine();
192
193     double total = joya.Costo;
194     Console.WriteLine("-----");
195     Console.WriteLine($"---\t\t\tRESUMEN DE TU PEDIDO\t\t---");
196     Console.WriteLine("-----");
197
198     Console.WriteLine(joya.Descripcion);
199     Console.WriteLine($" \nTOTAL A PAGAR\t\t\t${total}");
200     Console.WriteLine("\nGracias por tu compra. ¡Vuelve pronto a Joyeria Chula Vista!");
201
202     Console.ReadKey();
203
204 }
205
206 }
207 }
```

```
namespace ExamenU3P
{
    19 referencias
    public abstract class _0JoyeriaInterfazComponent
    {
        22 referencias
        public abstract double Costo { get; }
        23 referencias
        public abstract string Descripcion { get; }
    }
}
```

```
namespace ExamenU3P
{
    15 referencias
    public abstract class _1JoyaDecorador : _0JoyeriaInterfazComponent
    {
        protected _0JoyeriaInterfazComponent _joyas;
        7 referencias
        public _1JoyaDecorador(_0JoyeriaInterfazComponent joyas)
        {
            _joyas = joyas;
        }
    }
}
```





```
namespace ExamenU3P
{
    2 referencias
    public class _3EnvioDecorador : _1JoyaDecorador
    {
        1 referencia
        public _3EnvioDecorador(_0JoyeriaInterfazComponent joyas) : base(joyas)
        {
        }
        12 referencias
        public override double Costo => _joyas.Costo + 50.0;
        13 referencias
        public override string Descripcion => _joyas.Descripcion + "\n\n (+) Envio a domicilio\t\t$50.0";
    }
}
```

```
namespace ExamenU3P
{
    2 referencias
    public class _3EstucheDecorador : _1JoyaDecorador
    {
        1 referencia
        public _3EstucheDecorador(_0JoyeriaInterfazComponent joyas) : base(joyas)
        {
        }
        12 referencias
        public override double Costo => _joyas.Costo + 100.0;
        13 referencias
        public override string Descripcion => _joyas.Descripcion + "\n\n (+) Estuche Lujoso\t\t$100.0";
    }
}
```

```
namespace ExamenU3P
{
    2 referencias
    public class _3GrabadoDecorador : _1JoyaDecorador
    {
        1 referencia
        public _3GrabadoDecorador(_0JoyeriaInterfazComponent joyas) : base(joyas)
        {
        }
        12 referencias
        public override double Costo => _joyas.Costo + 150.0;
        13 referencias
        public override string Descripcion => _joyas.Descripcion + "\n\n (+) Grabado\t\t$150.0";
    }
}
```

```
namespace ExamenU3P
{
    2 referencias
    public class _3SeguroDecorador : _1JoyaDecorador
    {
        1 referencia
        public _3SeguroDecorador(_0JoyeriaInterfazComponent joyas) : base(joyas)
        {
        }
        12 referencias
        public override double Costo => _joyas.Costo + 500.0;
        13 referencias
        public override string Descripcion => _joyas.Descripcion + "\n\n (+) Seguro\t\t$500.0";
    }
}
```



```
✓ namespace ExamenU3P
{
    ✓ 2 referencias
    public class _4BonificacionCantidad : _1JoyaDecorador
    {
        ✓ private int _cantidad;
        ✓ 1 referencia
        public _4BonificacionCantidad(_0JoyeriaInterfazComponent joyas, int cantidad) : base(joyas)
        {
            _cantidad = cantidad;
        }
        ✓ 13 referencias
        public override string Descripcion
        {
            get
            {
                if (_cantidad >= 2)
                {
                    return _joyas.Descripcion + $"\n\n + Bonificación por cantidad {_cantidad} piezas) 10%";
                }
                else
                {
                    return _joyas.Descripcion + $"\n\n - No hay bonificación para {_cantidad} pieza";
                }
            }
        }
        ✓ 12 referencias
        public override double Costo
        {
            get
            {
                double precioBase = _joyas.Costo;
                if (_cantidad >= 2)
                {
                    precioBase = _joyas.Costo * _cantidad;
                    precioBase = precioBase * 0.9;
                    return precioBase;
                }
                return precioBase;
            }
        }
    }
}
```



```
namespace ExamenU3P
{
    2 referencias
    public class _4BonificacionMaterial : _1JoyaDecorador
    {
        private string _material;
        1 referencia
        public _4BonificacionMaterial(_0JoyeriaInterfazComponent joyas, string material) : base(joyas)
        {
            _material = material;
        }
        13 referencias
        public override string Descripcion
        {
            get
            {
                string detalle = "";
                if (_material == "Plata") detalle = "5%";
                else if (_material == "Platino") detalle = "8%";
                else if (_material == "Oro") detalle = "0%";
                return _joyas.Descripcion + $"\\n\\n + Bonificación por material {_material} {detalle}";
            }
        }
        12 referencias
        public override double Costo
        {
            get
            {
                double precioBase = _joyas.Costo;
                double descuento = 0;
                if (_material == "Plata") descuento = 0.05;
                else if (_material == "Platino") descuento = 0.08;
                else if (_material == "Oro") descuento = 0.0;
                if (descuento > 0)
                {
                    return precioBase * (1 - descuento);
                }
                return precioBase;
            }
        }
    }
}
```



```
namespace ExamenU3P
{
    2 referencias
    public class _4BonificacionTemporada : _1JoyaDecorador
    {
        private string _temporada;
        1 referencia
        public _4BonificacionTemporada(_0JoyeriaInterfazComponent joyas, string temporada) : base(joyas)
        {
            _temporada = temporada;
        }
        13 referencias
        public override string Descripcion
        {
            get
            {
                string detalle = "";

                if (_temporada == "Navidad") detalle = "15%";
                else if (_temporada == "San Valentín") detalle = "10%";
                else if (_temporada == "Verano") detalle = "5%";

                return _joyas.Descripcion + $"{\n\n} + Bonificación por temporada {_temporada} {detalle}";
            }
        }
        12 referencias
        public override double Costo
        {
            get
            {
                double precioBase = _joyas.Costo;
                double descuento = 0;

                if (_temporada == "Navidad") descuento = 0.15;
                else if (_temporada == "San Valentín") descuento = 0.10;
                else if (_temporada == "Verano") descuento = 0.05;

                if (descuento > 0)
                {
                    return precioBase * (1 - descuento);
                }

                return precioBase;
            }
        }
    }
}
```

```
namespace ExamenU3P
{
    3 referencias
    public abstract class _5InterfazInternacionalAdaptador
    {
        3 referencias
        public abstract void MostrarDescripcion();
        3 referencias
        public abstract void MostrarCosto();
    }
}
```



```
namespace ExamenU3P
{
    3 referencias
    public class _6AdaptadorEspañol : _5InterfazInternacionalAdaptador
    {
        private _0JoyeriaInterfazComponent _joya;

        2 referencias
        public _6AdaptadorEspañol(_0JoyeriaInterfazComponent joya)
        {
            _joya = joya;
        }

        2 referencias
        public override void MostrarDescripcion()
        {
            Console.WriteLine("Descripción:\n" + _joya.Descripcion);
        }

        2 referencias
        public override void MostrarCosto()
        {
            Console.WriteLine($"Costo total: ${_joya.Costo}");
        }
    }
}
```

```
namespace ExamenU3P
{
    2 referencias
    public class _6AdaptadorIngles : _5InterfazInternacionalAdaptador
    {
        private _0JoyeriaInterfazComponent _joya;
        1 referencia
        public _6AdaptadorIngles(_0JoyeriaInterfazComponent joya)
        {
            _joya = joya;
        }

        2 referencias
        public override void MostrarDescripcion()
        {
            Console.WriteLine("Description:\n" + _joya.Descripcion);
        }

        2 referencias
        public override void MostrarCosto()
        {
            Console.WriteLine($"Total cost: ${_joya.Costo}");
        }
    }
}
```



Explorador de soluciones

Buscar en Explorador de soluciones (Ctrl+)

Solución "ExamenU3P" (1 de 1 proyecto)

ExamenU3P

- Properties
- Referencias
- 0JoyeriaInterfazComponent.cs
- 1JoyaDecorador.cs
- 2Anillo.cs
- 2Aretes.cs
- 2Collar.cs
- 2Pulsera.cs
- 3EnvioDecorador.cs
- 3EstucheDecorador.cs
- 3GrabadoDecorador.cs
- 3SeguroDecorador.cs
- 4BonificacionCantidad.cs
- 4BonificacionMaterial.cs
- 4BonificacionTemporada.cs
- 5InterfazInternacionalAdaptador.cs
- 6AdaptadorEspanol.cs
- 6AdaptadorIngles.cs
- App.config
- Program.cs



EJECUCIÓN

```
C:\Users\Layone1\source\repo X + ▾  
  
JOYERIA CHULA VISTA  
  
-----  
  
Bienvenido a Joyeria Chula Vista  
  
-----  
  
1. Anillo $250.00  
2. Aretes $300.00  
3. Collar $350.00  
4. Pulsera $400.00  
Elige una opción:
```

```
C:\Users\Layone1\source\repo X + ▾  
  
JOYERIA CHULA VISTA  
  
-----  
  
Bienvenido a Joyeria Chula Vista  
  
-----  
  
1. Anillo $250.00  
2. Aretes $300.00  
3. Collar $350.00  
4. Pulsera $400.00  
Elige una opción:  
1|
```



```
C:\Users\Layonel\source\repo X + ▾  
-----  
JOYA SELECCIONADO  
Anillo $250.0  
(-) Estuche $0.00  
(-) Grabado $0.00  
(-) Seguro $0.00  
  
¿Deseas personalizar tu JOYA?  
1. Sí/  
2. No  
Elige una opción:  
1
```

```
C:\Users\Layonel\source\repo X + ▾  
-----  
Opciones de personalización:  
1. Agregar Estuche Lujoso  
2. Agregar Grabado  
3. Agregar Seguro  
4. Finalizar personalización  
  
Elige una opción:  
1  
Estuche lujoso añadido.
```

```
C:\Users\Layonel\source\repo X + ▾  
-----  
Opciones de personalización:  
1. Agregar Estuche Lujoso  
2. Agregar Grabado  
3. Agregar Seguro  
4. Finalizar personalización  
  
Elige una opción:  
2  
Grabado personalizado añadido.
```



```
C:\Users\Layonel\source\repo X +  
  
Opciones de personalización:  
1. Agregar Estuche Lujoso  
2. Agregar Grabado  
3. Agregar Seguro  
4. Finalizar personalización  
  
Elige una opción:  
4  
|
```

```
C:\Users\Layonel\source\repo X + | v  
  
¿Te gustaría un envío a domicilio?  
1. Sí/  
2. No  
Elige una opción:  
1
```

```
C:\Users\Layonel\source\repo X + | v  
  
¿Deseas aplicar bonificación por material?  
1. Sí  
2. No  
1
```

```
C:\Users\Layonel\source\repo X + | v  
  
¿Deseas aplicar bonificación por material?  
1. Sí  
2. No  
1  
Ingresa el tipo de material (Oro, Plata, Platino):  
Platino
```

```
C:\Users\Layonel\source\repo X + | v  
  
¿Deseas aplicar bonificación por temporada?  
1. Sí  
2. No  
1
```



```
C:\Users\Layonel\source\repo X + ▾  
¿Deseas aplicar bonificación por temporada?  
1. Sí  
2. No  
1  
Ingresa la temporada (Navidad, San Valentín, Verano):  
Navidad|
```

```
C:\Users\Layonel\source\repo X + ▾  
¿Cuántas piezas deseas comprar?  
2
```

```
C:\Users\Layonel\source\repo X + ▾  
¿Deseas ver el resumen en otro idioma?  
1. Español  
2. Ingles  
Elige una opción:  
2
```



RESUMEN DE TU PEDIDO	
Anillo	\$250.0
(-) Estuche	\$0.00
(-) Grabado	\$0.00
(-) Seguro	\$0.00
(+) Estuche Lujoso	\$100.0
(+) Grabado	\$150.0
(+) Seguro	\$500.0
(+) Envio a domicilio	\$50.0
+ Bonificación por material (Platino) 8%	
+ Bonificación por temporada (Navidad) 15%	
+ Bonificación por cantidad (2 piezas) 10%	
TOTAL A PAGAR	\$1477.98
Gracias por tu compra. ¡Vuelve pronto a Joyeria Chula Vista!	



C:\Users\Layonel\source\repo

X + ▾

----- ORDER SUMMARY (IN ENGLISH) -----

Description:

Anillo	\$250.0
(-) Estuche	\$0.00
(-) Grabado	\$0.00
(-) Seguro	\$0.00

(+) Estuche Lujoso	\$100.0
--------------------	---------

(+) Grabado	\$150.0
-------------	---------

(+) Seguro	\$500.0
------------	---------

(+) Envio a domicilio	\$50.0
-----------------------	--------

+ Bonificación por material (Platino) 8%

+ Bonificación por temporada (Navidad) 15%

+ Bonificación por cantidad (2 piezas) 10%

Total cost: \$1477.98

Thank you for your purchase. Come back soon to Joyeria Chula Vista!



CONCLUSIÓN

En conclusión, aplicar el patrón Decorador nos permite agregar funcionalidades dinámicamente a objetos sin modificar su estructura base, es decir, podemos asignar comportamientos adicionales de forma flexible, ya sean de cualquier tipo siempre y cuando no se modifique la clase base. Además de aplicar el segundo patrón el cuál nos permite adaptar una interfaz existente a una nueva, facilita mucho la integración de clases incompatibles, en este caso, la aplicación de este segundo patrón me permitió adaptar la presentación del Decorador a distintos idiomas, sin alterar la estructura principal del Decorador.

Considero difícil este examen, ya que no tenía muy bien claro la idea de cómo implementar el Adaptador en este sistema (a nivel práctico), el cuál fue un control de inventarios con reglas de bonificación, pese a tener días para desarrollarlo, no lograba compactar el sistema lo más refactorizado posible (Decorador)...



**TECNOLÓGICO
NACIONAL DE MÉXICO**
INSTITUTO TECNOLÓGICO DE TIJUANA

