



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

Materia:

Patrones de Diseño

Semestre:

8

Unidad:

IV y V

Docente:

Maribel Guerrero Luis

Alumno:

Leonel Martinez Huitron

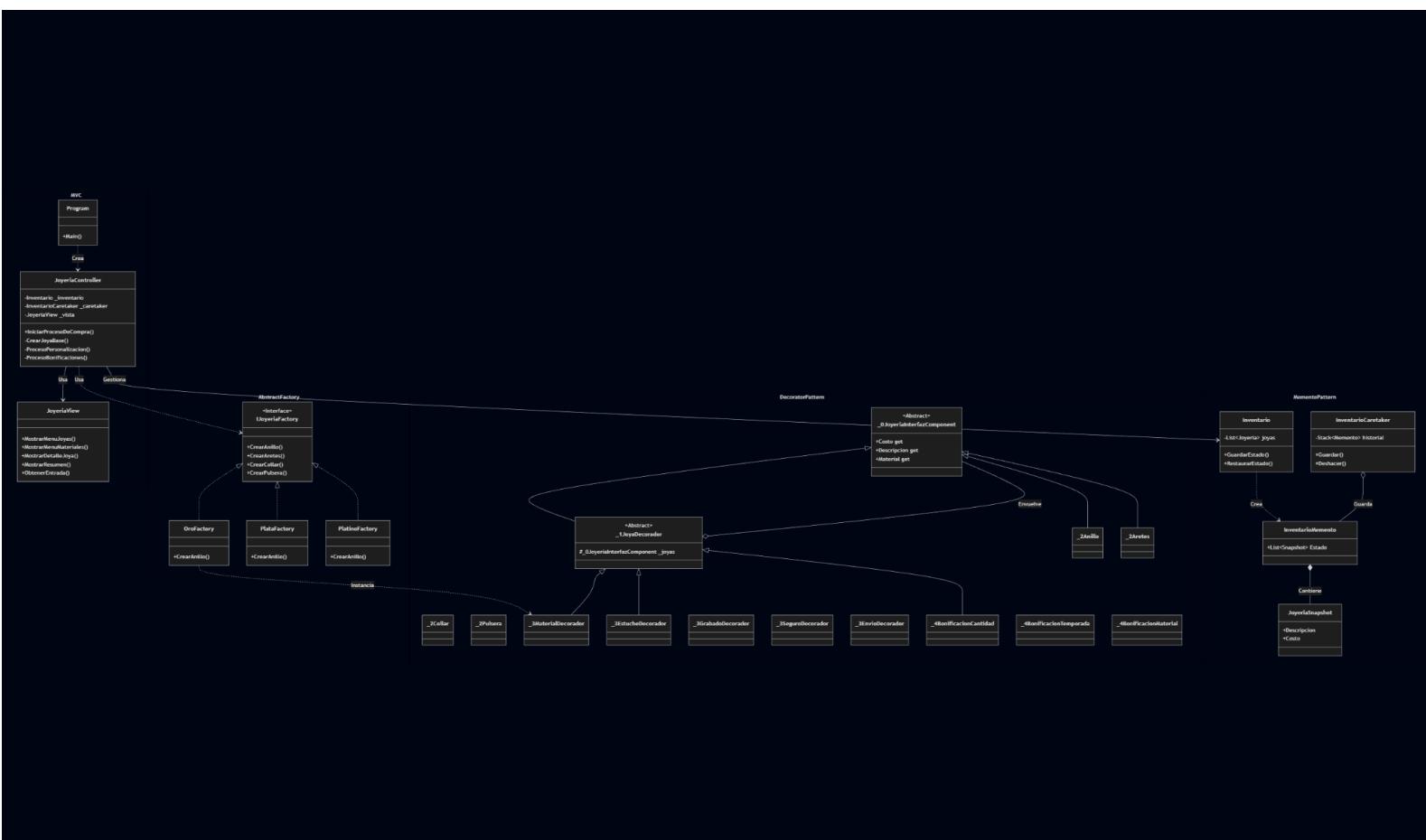
Número de control:

22210777





DIAGRAMA UML





CÓDIGO

```
namespace ExamenFinal
{
    0 referencias
    internal class Program
    {
        0 referencias
        static void Main(string[] args)
        {

            JoyeriaController controlador = new JoyeriaController();
            controlador.IniciarProcesoDeCompra();
        }
    }
}
```

```
namespace ExamenFinal
{
    17 referencias
    public abstract class _1JoyaDecorador : _0JoyeriaInterfazComponent
    {
        protected _0JoyeriaInterfazComponent _joyas;
        8 referencias
        public _1JoyaDecorador(_0JoyeriaInterfazComponent joyas)
        {
            _joyas = joyas;
        }
    }
}
```

```
namespace ExamenFinal
{
    2 referencias
    public class _3EnvioDecorador : _1JoyaDecorador
    {
        1 referencia
        public _3EnvioDecorador(_0JoyeriaInterfazComponent joyas) : base(joyas)
        {
        }
        16 referencias
        public override double Costo => _joyas.Costo + 50.0;
        15 referencias
        public override string Descripcion => _joyas.Descripcion + "\n\n (+) Envío a domicilio\t\t\t$50.0";
        8 referencias
        public override string Material => _joyas.Material;
    }
}
```

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



```
namespace ExamenFinal
{
    13 referencias
    public class _3MaterialDecorador : _1JoyaDecorador
    {
        private string material;

        12 referencias
        public _3MaterialDecorador(_0JoyeriaInterfazComponent joyas, string material) : base(joyas)
        {
            this.material = material;
        }

        8 referencias
        public override string Material => material;

        15 referencias
        public override string Descripcion
        {
            get
            {
                double costoFinal = Costo;
                double costoExtra = costoFinal - _joyas.Costo;

                return _joyas.Descripcion +
                    $"\\n (-) Material {material}\\t\\t+${costoExtra}";
            }
        }

        17 referencias
        public override double Costo
        {
            get
            {
                if (material == "Oro") return _joyas.Costo * 2;
                if (material == "Plata") return _joyas.Costo * 1.2;
                if (material == "Platino") return _joyas.Costo * 1.5;
                return _joyas.Costo;
            }
        }
}
```



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

```
namespace ExamenFinal
{
    2 referencias
    public class _4BonificacionCantidad : _1JoyaDecorador
    {
        private int _cantidad;
        1 referencia
        public _4BonificacionCantidad(_0JoyeriaInterfazComponent joyas, int cantidad) : base(joyas)
        {
            _cantidad = cantidad;
        }
        15 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)";
                }
            }
        }
        16 referencias
        public override double Costo
        {
            get
            {
                double precioBase = _joyas.Costo * _cantidad;
                if (_cantidad >= 2)
                    return precioBase * 0.9;
                return precioBase;
            }
        }
        8 referencias
        public override string Material => _joyas.Material;
    }
}
```



```
namespace ExamenFinal
{
    2 referencias
    public class _4BonificacionMaterial : _1JoyaDecorador
    {
        private string material;

        1 referencia
        public _4BonificacionMaterial(_0JoyeriaInterfazComponent joyas, string material) : base(joyas)
        {
            this.material = material;
        }

        8 referencias
        public override string Material => material;

        15 referencias
        public override string Descripcion
        {
            get
            {
                string detalle = material == "Plata" ? "5%" :
                                material == "Platino" ? "8%" : "0%";
                return _joyas.Descripcion + $"\n (-) Bonificación por material ({material}) {detalle}";
            }
        }

        16 referencias
        public override double Costo
        {
            get
            {
                double descuento = material == "Plata" ? 0.05 :
                                    material == "Platino" ? 0.08 : 0.0;
                return _joyas.Costo * (1 - descuento);
            }
        }
}
```



```
namespace ExamenFinal
{
    2 referencias
    public class _4BonificacionTemporada : _1JoyaDecorador
    {
        private string _temporada;
        1 referencia
        public _4BonificacionTemporada(_0JoyeriaInterfazComponent joyas, string temporada) : base(joyas)
        {
            _temporada = temporada;
        }
        15 referencias
        public override string Descripcion
        {
            get
            {
                string detalle = _temporada == "Navidad" ? "15%" :
                    _temporada == "San Valentín" ? "10%" :
                    _temporada == "Verano" ? "5%" :
                    "0%";
                return _joyas.Descripcion + $"\\n\\n (+) Bonificación por temporada {_temporada} {detalle}";
            }
        }
        16 referencias
        public override double Costo
        {
            get
            {
                double descuento = _temporada == "Navidad" ? 0.15 :
                    _temporada == "San Valentín" ? 0.10 :
                    _temporada == "Verano" ? 0.05 :
                    0.0;
                return _joyas.Costo * (1 - descuento);
            }
        }
        8 referencias
        public override string Material => _joyas.Material;
    }
}
```



```
namespace ExamenFinal
{
    2 referencias
    public class OroFactory : IJoyeriaFactory
    {
        3 referencias
        public _0JoyeriaInterfazComponent CrearAnillo()
        {
            var joya = new _2Anillo();
            return new _3MaterialDecorador(joya, "Oro");
        }
        2 referencias
        public _0JoyeriaInterfazComponent CrearAretes()
        {
            var joya = new _2Aretes();
            return new _3MaterialDecorador(joya, "Oro");
        }
        2 referencias
        public _0JoyeriaInterfazComponent CrearCollar()
        {
            var joya = new _2Collar();
            return new _3MaterialDecorador(joya, "Oro");
        }

        2 referencias
        public _0JoyeriaInterfazComponent CrearPulsera()
        {
            var joya = new _2Pulsera();
            return new _3MaterialDecorador(joya, "Oro");
        }
    }
}
```



```
namespace ExamenFinal
{
    1 referencia
    public class PlataFactory : IJoyeriaFactory
    {
        3 referencias
        public _0JoyeriaInterfazComponent CrearAnillo()
        {
            var joya = new _2Anillo();
            return new _3MaterialDecorador(joya, "Plata");
        }
        2 referencias
        public _0JoyeriaInterfazComponent CrearAretes()
        {
            var joya = new _2Aretes();
            return new _3MaterialDecorador(joya, "Plata");
        }
        2 referencias
        public _0JoyeriaInterfazComponent CrearCollar()
        {
            var joya = new _2Collar();
            return new _3MaterialDecorador(joya, "Plata");
        }

        2 referencias
        public _0JoyeriaInterfazComponent CrearPulsera()
        {
            var joya = new _2Pulsera();
            return new _3MaterialDecorador(joya, "Plata");
        }
    }
}
```



```
namespace ExamenFinal
{
    1 referencia
    public class PlatinoFactory : IJoyeriaFactory
    {
        3 referencias
        public _0JoyeriaInterfazComponent CrearAnillo()
        {
            var joya = new _2Anillo();
            return new _3MaterialDecorador(joya, "Platino");
        }
        2 referencias
        public _0JoyeriaInterfazComponent CrearAretes()
        {
            var joya = new _2Aretes();
            return new _3MaterialDecorador(joya, "Platino");
        }
        2 referencias
        public _0JoyeriaInterfazComponent CrearCollar()
        {
            var joya = new _2Collar();
            return new _3MaterialDecorador(joya, "Platino");
        }

        2 referencias
        public _0JoyeriaInterfazComponent CrearPulsera()
        {
            var joya = new _2Pulsera();
            return new _3MaterialDecorador(joya, "Platino");
        }
    }
}
```

```
namespace ExamenFinal
{
    43 referencias
    public abstract class _0JoyeriaInterfazComponent
    {
        30 referencias
        public abstract double Costo { get; }
        28 referencias
        public abstract string Descripcion { get; }
        21 referencias
        public abstract string Material { get; }
    }
}
```



```
namespace ExamenFinal
{
    4 referencias
    public interface IJoyeriaFactory
    {
        5 referencias
        _0JoyeriaInterfazComponent CrearAnillo();
        4 referencias
        _0JoyeriaInterfazComponent CrearAretes();
        4 referencias
        _0JoyeriaInterfazComponent CrearCollar();
        4 referencias
        _0JoyeriaInterfazComponent CrearPulsera();
    }
}

namespace ExamenFinal
{
    4 referencias
    public class Inventario
    {
        private List<_0JoyeriaInterfazComponent> joyas = new List<_0JoyeriaInterfazComponent>();

        1 referencia
        public void AgregarJoyeria(_0JoyeriaInterfazComponent joya)
        {
            joyas.Add(joya);
        }

        1 referencia
        public void MostrarInventario()
        {
            Console.WriteLine("Inventario actual:");
            foreach (var joya in joyas)
            {
                Console.WriteLine($"{joya.Descripcion}");
            }
            Console.WriteLine($"\\nTOTAL INICIAL: ${CalcularTotal()}");
        }
        1 referencia
        public double CalcularTotal()
        {
            return joyas.Sum(j => j.Costo);
        }
        1 referencia
        public InventarioMemento GuardarEstado()
        {
            return new InventarioMemento(joyas);
        }
        1 referencia
        public void RestaurarEstado(InventarioMemento memento)
        {
            joyas = new List<_0JoyeriaInterfazComponent>(memento.EstadoInventario);
        }
    }
}
```



```
namespace ExamenFinal
{
    2 referencias
    public class InventarioCaretaker
    {
        private Stack<InventarioMemento> historial = new Stack<InventarioMemento>();

        1 referencia
        public void Guardar(Inventario inventario)
        {
            historial.Push(inventario.GuardarEstado());
        }

        1 referencia
        public void Deshacer(Inventario inventario)
        {
            if (historial.Count > 0)
            {
                var memento = historial.Pop();
                inventario.RestaurarEstado(memento);
            }
        }
    }
}
```

```
namespace ExamenFinal
{
    6 referencias
    public class InventarioMemento
    {
        2 referencias
        public List<JoyeriaSnapshot> EstadoInventario { get; private set; }

        1 referencia
        public InventarioMemento(List<_0JoyeriaInterfazComponent> joyas)
        {
            EstadoInventario = joyas.Select(j =>
                new JoyeriaSnapshot(j.Descripcion, j.Costo, j.Material)).ToList();
        }
    }
}
```



```
namespace ExamenFinal
{
    3 referencias
    public class JoyeriaSnapshot : _0JoyeriaInterfazComponent
    {
        16 referencias
        public override string Descripcion { get; }
        17 referencias
        public override double Costo { get; }
        9 referencias
        public override string Material { get; }
        1 referencia
        public JoyeriaSnapshot(string descripcion, double costo, string material)
        {
            Descripcion = descripcion;
            Costo = costo;
            Material = material;
        }
    }
}
```



```
namespace ExamenFinal
{
    3 referencias
    public class JoyeriaController
    {
        private Inventario _inventario;
        private InventarioCaretaker _caretaker;

        private JoyeriaView _vista;

        1 referencia
        public JoyeriaController()
        {
            _inventario = new Inventario();
            _caretaker = new InventarioCaretaker();
            _vista = new JoyeriaView();
        }
    }
}
```



```
1 referencia
public void IniciarProcesoDeCompra()
{
    _vista.MostrarBienvenida();

    _JoyeriaInterfazComponent joya = CrearJoyaBase();

    _vista.LimpiarPantalla();
    _vista.MostrarDetalleJoya(joya.Descripcion);

    _inventario.AgregarJoyeria(joya);
    _caretaker.Guardar(_inventario);

    joya = ProcesoPersonalizacion(joya);

    joya = ProcesoBonificaciones(joya);

    _vista.LimpiarPantalla();
    _vista.MostrarResumen(joya.Descripcion, joya.Costo);

    _caretaker.Deshacer(_inventario);
    Console.WriteLine("\n(Sistema: Inventario restaurado por Memento)");
    _inventario.MostrarInventario();
    _vista.Pausa();
}
```



```
1 referencia
private _@JoyeriaInterfazComponent CrearJoyaBase()
{
    _vista.MostrarMenuJoyas();
    string tipo = _vista.ObtenerEntrada();
    _vista.LimpiarPantalla();

    _vista.MostrarMenuMateriales();
    string materialOp = _vista.ObtenerEntrada();

    IJoyeriaFactory factory = null;
    switch (materialOp)
    {
        case "1":
            factory = new OroFactory();
            break;
        case "2":
            factory = new PlataFactory();
            break;
        case "3":
            factory = new PlatinoFactory();
            break;
        default: factory = new OroFactory(); break;
    }

    switch (tipo)
    {
        case "1":
            return factory.CrearAnillo();
        case "2":
            return factory.CrearAretes();
        case "3":
            return factory.CrearCollar();
        case "4":
            return factory.CrearPulsera();
        default:
            return factory.CrearAnillo();
    }
}
```



```
1 referencia
private _0JoyeriaInterfazComponent ProcesoPersonalizacion(_0JoyeriaInterfazComponent joya)
{
    string op = _vista.ObtenerEntrada("\n¿Deseas personalizar tu JOYA? (1. Sí / 2. No)");
    if (op != "1")
        return joya;

    bool personalizando = true;
    while (personalizando)
    {
        _vista.LimpiarPantalla();
        _vista.MostrarDetalleJoya(joya.Descripcion);
        _vista.MostrarMenuPersonalizacion();

        op = _vista.ObtenerEntrada();
        switch (op)
        {
            case "1":
                joya = new _3EstucheDecorador(joya);
                break;
            case "2":
                joya = new _3GrabadoDecorador(joya);
                break;
            case "3":
                joya = new _3SeguroDecorador(joya);
                break;
            case "4":
                personalizando = false;
                break;
        }
    }

    op = _vista.ObtenerEntrada("¿Envío a domicilio? (1. Sí / 2. No)");
    if (op == "1")
        joya = new _3EnvioDecorador(joya);

    return joya;
}
```



```
1 referencia
private _0JoyeriaInterfazComponent ProcesoBonificaciones(_0JoyeriaInterfazComponent joya)
{
    _vista.LimpiarPantalla();
    Console.WriteLine("---- SECCIÓN DE BONIFICACIONES ----\n");

    string op = _vista.ObtenerEntrada("¿Deseas aplicar bonificación extra por tipo de material? (1. Sí / 2. No)");
    if (op == "1")
    {
        string material = _vista.ObtenerEntrada("Ingresa el material para validar bonificación (Plata, Platino, Oro):");
        joya = new _4BonificacionMaterial(joya, material);
        Console.WriteLine("> Bonificación de material aplicada.");
    }

    op = _vista.ObtenerEntrada("\n¿Estamos en temporada de ofertas? (1. Sí / 2. No)");
    if (op == "1")
    {
        string temporada = _vista.ObtenerEntrada("Ingresa la temporada (Navidad, San Valentín, Verano):");
        joya = new _4BonificacionTemporada(joya, temporada);
        Console.WriteLine("> Bonificación de temporada aplicada.");
    }

    string cantidadStr = _vista.ObtenerEntrada("\n¿Cuántas piezas iguales deseas llevar?");
    int cantidad = 1;
    if (!int.TryParse(cantidadStr, out cantidad))
    {
        cantidad = 1;
        Console.WriteLine("(Entrada inválida, se calculará 1 pieza)");
    }

    joya = new _4BonificacionCantidad(joya, cantidad);

    return joya;
}
```



Explorador de soluciones

Buscar en Explorador de soluciones (Ctrl+)

Solución "ExamenFinal" (1 de 1 proyecto)

- ExamenFinal
 - Properties
 - Referencias
 - Controllers
 - JoyeriaController.cs
 - Models
 - Decoradores
 - 1JoyaDecorador.cs
 - 3EnvioDecorador.cs
 - 3EstucheDecorador.cs
 - 3GrabadoDecorador.cs
 - 3MaterialDecorador.cs
 - 3SeguroDecorador.cs
 - 4BonificacionCantidad.cs
 - 4BonificacionMaterial.cs
 - 4BonificacionTemporada.cs
 - Factories
 - OroFactory.cs
 - PlataFactory.cs
 - PlatinoFactory.cs
 - Interfaces
 - 0JoyeriaInterfazComponent.cs
 - IJoyeriaFactory.cs
 - Memento
 - Inventario.cs
 - InventarioCaretaker.cs
 - InventarioMemento.cs
 - JoyeriaSnapshot.cs
 - Productos
 - 2Anillo.cs
 - 2Aretes.cs
 - 2Collar.cs
 - 2Pulsera.cs
 - Views
 - JoyeriaView.cs
 - App.config
 - Program.cs



FLUJO DEL CÓDIGO CON CAPTURAS

```
C:\Users\Admin\Source\Repo X + ▾  
  
-----  
JOYERIA CHULA VISTA  
  
-----  
Selecciona el tipo de joya:  
1. Anillo $250.00  
2. Aretes $300.00  
3. Collar $350.00  
4. Pulsera $400.00  
1|
```

```
C:\Users\Admin\Source\Repo X + ▾  
  
-----  
Selecciona el material:  
1. Oro  
2. Plata  
3. Platino  
3
```

```
C:\Users\Admin\Source\Repo X + ▾  
  
-----  
JOYA SELECCIONADA  
Anillo $250.0  
(-) Estuche $0.00  
(-) Grabado $0.00  
(-) Seguro $0.00  
(-) Material Platino +$125  
  
¿Deseas personalizar tu JOYA? (1. Sí / 2. No)  
1|
```



```
C:\Users\Admin\Source\Repo + ▾  
  
-----  
JOYA SELECCIONADA  
Anillo $250.0  
(-) Estuche $0.00  
(-) Grabado $0.00  
(-) Seguro $0.00  
(-) Material Platino+$125  
  
Opciones de personalización:  
1. Agregar Estuche Lujoso  
2. Agregar Grabado  
3. Agregar Seguro  
4. Finalizar personalización  
1
```

```
C:\Users\Admin\Source\Repo + ▾  
  
-----  
JOYA SELECCIONADA  
Anillo $250.0  
(-) Estuche $0.00  
(-) Grabado $0.00  
(-) Seguro $0.00  
(-) Material Platino+$125  
  
(+ ) Estuche Lujoso $100.0  
  
Opciones de personalización:  
1. Agregar Estuche Lujoso  
2. Agregar Grabado  
3. Agregar Seguro  
4. Finalizar personalización  
2
```



```
C:\Users\Admin\Source\Repo X + | ▾
```

JOYA SELECCIONADA

Anillo	\$250.0
(-) Estuche	\$0.00
(-) Grabado	\$0.00
(-) Seguro	\$0.00
(-) Material Platino	+\$125
 (+) Estuche Lujoso	\$100.0
 (+) Grabado	\$150.0

Opciones de personalización:

1. Agregar Estuche Lujoso
2. Agregar Grabado
3. Agregar Seguro
4. Finalizar personalización

3

```
C:\Users\Admin\Source\Repo X + | ▾
```

JOYA SELECCIONADA

Anillo	\$250.0
(-) Estuche	\$0.00
(-) Grabado	\$0.00
(-) Seguro	\$0.00
(-) Material Platino	+\$125
 (+) Estuche Lujoso	\$100.0
 (+) Grabado	\$150.0
 (+) Seguro	\$500.0

Opciones de personalización:

1. Agregar Estuche Lujoso
2. Agregar Grabado
3. Agregar Seguro
4. Finalizar personalización

4



```
C:\Users\Admin\Source\Repo X + | -  
-----  
JOYA SELECCIONADA  
Anillo $250.0  
(-) Estuche $0.00  
(-) Grabado $0.00  
(-) Seguro $0.00  
(-) Material Platino +$125  
  
(+ ) Estuche Lujoso $100.0  
  
(+ ) Grabado $150.0  
  
(+ ) Seguro $500.0  
  
Opciones de personalización:  
1. Agregar Estuche Lujoso  
2. Agregar Grabado  
3. Agregar Seguro  
4. Finalizar personalización  
4  
¿Envío a domicilio? (1. Sí / 2. No)  
1
```

```
C:\Users\Admin\Source\Repo X + | -  
--- SECCIÓN DE BONIFICACIONES ---  
  
¿Deseas aplicar bonificación extra por tipo de material? (1. Sí / 2. No)  
1
```

```
C:\Users\Admin\Source\Repo X + | -  
--- SECCIÓN DE BONIFICACIONES ---  
  
¿Deseas aplicar bonificación extra por tipo de material? (1. Sí / 2. No)  
1  
Ingresa el material para validar bonificación (Plata, Platino, Oro):  
Platino
```



```
C:\Users\Admin\Source\Repo X + ▾  
--- SECCIÓN DE BONIFICACIONES ---  
  
¿Deseas aplicar bonificación extra por tipo de material? (1. Sí / 2. No)  
1  
Ingresa el material para validar bonificación (Plata, Platino, Oro):  
Platino  
-> Bonificación de material aplicada.  
  
¿Estamos en temporada de ofertas? (1. Sí / 2. No)  
1  
  
C:\Users\Admin\Source\Repo X + ▾  
--- SECCIÓN DE BONIFICACIONES ---  
  
¿Deseas aplicar bonificación extra por tipo de material? (1. Sí / 2. No)  
1  
Ingresa el material para validar bonificación (Plata, Platino, Oro):  
Platino  
-> Bonificación de material aplicada.  
  
¿Estamos en temporada de ofertas? (1. Sí / 2. No)  
1  
Ingresa la temporada (Navidad, San Valentín, Verano):  
Navidad
```



C:\Users\Admin\Source\Repo X + ▾

--- SECCIÓN DE BONIFICACIONES ---

¿Deseas aplicar bonificación extra por tipo de material? (1. Sí / 2. No)

1

Ingresa el material para validar bonificación (Plata, Platino, Oro):

Platino

-> Bonificación de material aplicada.

¿Estamos en temporada de ofertas? (1. Sí / 2. No)

1

Ingresa la temporada (Navidad, San Valentín, Verano):

Navidad

-> Bonificación de temporada aplicada.

¿Cuántas piezas iguales deseas llevar?

2



C:\Users\Admin\Source\Repo

X + ▾

RESUMEN DE TU PEDIDO

Anillo	\$250.0
(-) Estuche	\$0.00
(-) Grabado	\$0.00
(-) Seguro	\$0.00
(-) Material Platino	+\$125
 (+) 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	 \$1653.93

Gracias por tu compra. ¡Vuelve pronto a Joyeria Chula Vista!

(Sistema: Inventario restaurado por Memento)

Inventario actual:

Anillo	\$250.0
(-) Estuche	\$0.00
(-) Grabado	\$0.00
(-) Seguro	\$0.00
(-) Material Platino	+\$125

TOTAL INICIAL: \$375