



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Agosto- Diciembre 2025

CARRERA:

Ingeniería Informática

MATERIA:

Patrones de Diseño

TÍTULO ACTIVIDAD:

Examen Unidad 3

UNIDAD A EVALUAR:

Unidad 3

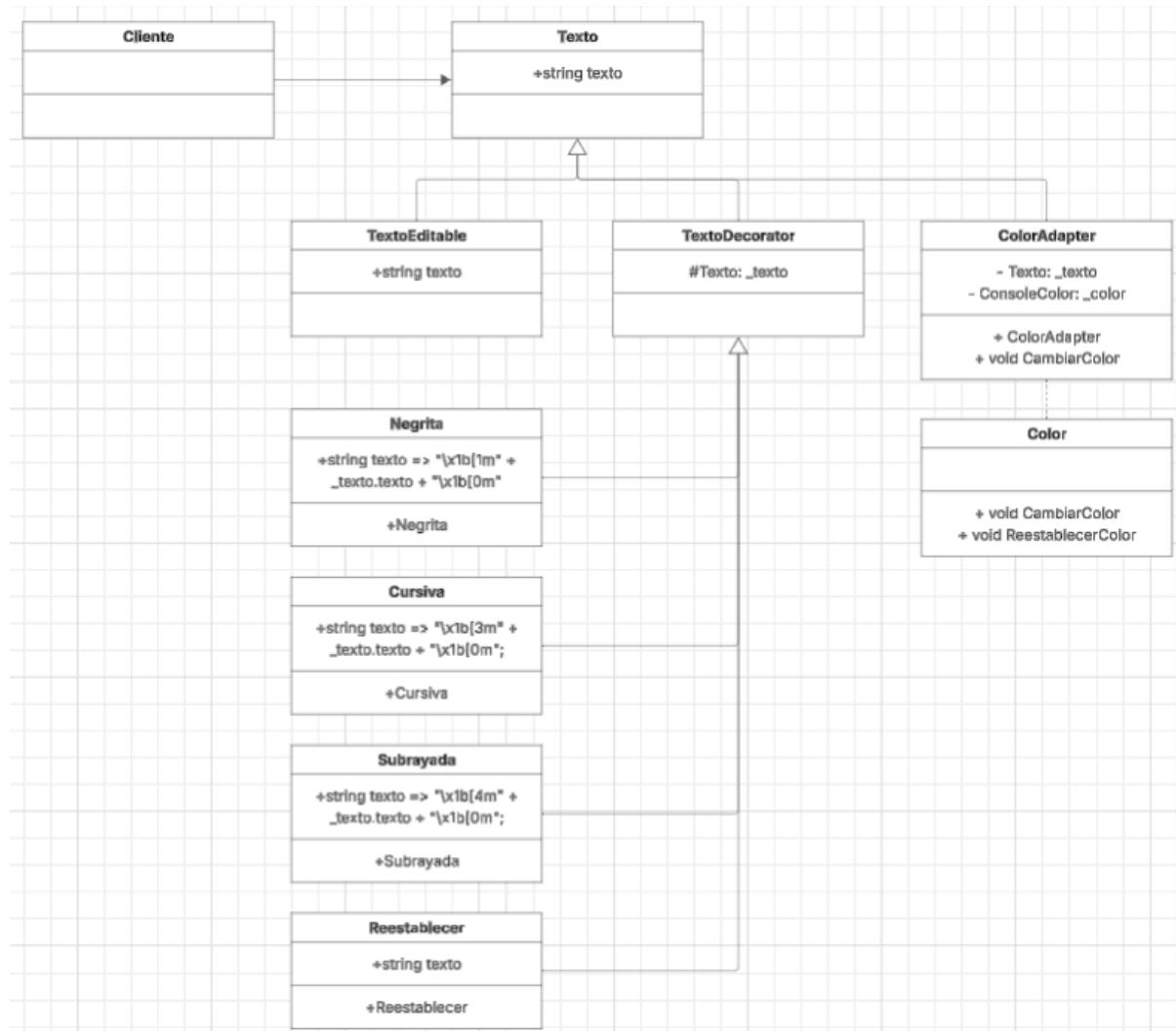
NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Miguel Angel Briones Hernandez 21212325

NOMBRE DEL MAESTRO (A):

Maribel Guerrero Luis

DIAGRAMA UML



CÓDIGO

Program:

```
namespace Examen_U3

{
    internal class Program

    {
        static void Main(string[] args)

        {

            Texto textoEditable = new TextoEditable();

            ColorAdapter adaptadorColor = null;

            string cadena;

            Console.WriteLine("Bienvenido al editor de Texto - Examen Unidad 3");

            Console.Write("Ingrese su texto para comenzar: ");

            cadena = Console.ReadLine();

            Console.Clear();

            ((TextoEditable)textoEditable).text = cadena;

            string continuar = "si";

            while (continuar == "si")

            {

                Console.WriteLine("Seleccione el estilo que desea aplicar:");

                Console.WriteLine("1. Negrita");

                Console.WriteLine("2. Cursiva");
```

```
Console.WriteLine("3. Subrayada");

Console.WriteLine("4. Reestablecer estilos");

Console.WriteLine("5. Cambiar color");

Console.Write("Opción: ");

int opc;

while (!int.TryParse(Console.ReadLine(), out opc) || opc < 1 || opc > 5)

{

    Console.Write("Opción inválida. Intente de nuevo: ");

}

Console.Clear();

switch (opc)

{

    case 1:

        textoEditable = new Negrita(textoEditable);

        break;

    case 2:

        textoEditable = new Cursiva(textoEditable);

        break;

    case 3:

        textoEditable = new Subrayada(textoEditable);

        break;

}
```

```
case 4:  
    textoEditable = new Reestablecer(textoEditable);  
    break;  
  
case 5:  
    Console.WriteLine("Seleccione un color:");  
    Console.WriteLine("1. Rojo");  
    Console.WriteLine("2. Verde");  
    Console.WriteLine("3. Azul");  
    Console.WriteLine("4. Amarillo");  
    Console.WriteLine("5. Morado");  
    Console.WriteLine("6. Reestablecer color");  
    Console.Write("Opción: ");  
  
    int colorOpc;  
    while (!int.TryParse(Console.ReadLine(), out colorOpc) || colorOpc < 1 ||  
colorOpc > 6)  
    {  
        Console.Write("Opción inválida. Intente de nuevo: ");  
    }  
  
    ConsoleColor colorSeleccionado = ConsoleColor.White;  
  
    switch (colorOpc)  
    {  
        case 1: colorSeleccionado = ConsoleColor.Red; break;  
        case 2: colorSeleccionado = ConsoleColor.Green; break;  
        case 3: colorSeleccionado = ConsoleColor.Blue; break;  
        case 4: colorSeleccionado = ConsoleColor.Yellow; break;  
        case 5: colorSeleccionado = ConsoleColor.Magenta; break;  
        case 6: colorSeleccionado = ConsoleColor.White; break;  
    }  
    texto.setEditable = colorSeleccionado;
```

```
        case 2: colorSeleccionado = ConsoleColor.Green; break;
        case 3: colorSeleccionado = ConsoleColor.Blue; break;
        case 4: colorSeleccionado = ConsoleColor.Yellow; break;
        case 5: colorSeleccionado = ConsoleColor.Magenta; break;
        case 6: colorSeleccionado = ConsoleColor.White; break;
    }

    Console.Clear();

    if (adaptadorColor == null)
    {
        adaptadorColor = new ColorAdapter(textoEditable, colorSeleccionado);
        textoEditable = adaptadorColor;
    }
    else
    {
        adaptadorColor.CambiarColor(colorSeleccionado);
    }
    break;
}

Console.Clear();
Console.WriteLine("Texto con estilos aplicados:");
Console.WriteLine(textoEditable.texto);

Console.WriteLine();
```

```
        Console.WriteLine("¿Desea agregar otro estilo? (si/no): ");

        continuar = Console.ReadLine().Trim().ToLower();

        Console.Clear();

    }

    Console.WriteLine("Resultado final:");

    Console.WriteLine(textoEditable.texto);

    Console.ResetColor();

    Console.ReadKey();

}

}

}


```

Texto:

```
namespace Examen_U3

{

    public abstract class Texto

    {

        public abstract string texto { get; }

    }

}
```

TextoEditable:

```
namespace Examen_U3

{

    public class TextoEditable : Texto

    {

}
```

```
public string text { get; set; }

public override string texto => text;

}

}
```

TextoDecorator:

```
namespace Examen_U3

{

    public abstract class TextoDecorator:Texto

    {

        protected Texto _texto;

        public TextoDecorator(Texto texto)

        {

            _texto = texto;

        }

    }

}
```

Negrita:

```
namespace Examen_U3

{

    public class Negrita : TextoDecorator

    {

        public Negrita(Texto texto) : base(texto) {}

        public override string texto => "\x1b[1m" + _texto.texto + "\x1b[0m";

    }

}
```

```
    }  
}  
  
}
```

Cursiva:

```
namespace Examen_U3  
{  
    internal class Cursiva:TextoDecorator  
    {  
        public Cursiva(Texto texto) : base(texto) {}  
        public override string texto => "\x1b[3m" + _texto.texto + "\x1b[0m";  
    }  
}
```

Subrayada:

```
namespace Examen_U3  
{  
    internal class Subrayada:TextoDecorator  
    {  
        public Subrayada(Texto texto) : base(texto) {}  
        public override string texto => "\x1b[4m" + _texto.texto + "\x1b[0m";  
    }  
}
```

Reestablecer:

```
namespace Examen_U3
```

```

{
internal class Reestablecer:TextoDecorator
{
    public Reestablecer(Texto texto) : base(texto) { }

    public override string texto
    {
        get
        {
            string limpio = Regex.Replace(_texto.texto, @"\x1b\[0-9;]*m", "");
            return limpio;
        }
    }
}

```

Color:

```

namespace Examen_U3
{
    public class Color
    {
        public void CambiarColor (ConsoleColor color)
        {
            Console.ForegroundColor = color;
        }

        public void ReestablecerColor()
    }
}

```

```
{  
    Console.ResetColor();  
}  
}  
}
```

ColorAdapter:

```
namespace Examen_U3
```

```
{  
    public class ColorAdapter : Texto  
    {  
        private readonly Texto _texto;  
        private ConsoleColor _color;
```

```
        public ColorAdapter(Texto texto, ConsoleColor color)
```

```
        {  
            _texto = texto;  
            _color = color;  
        }
```

```
        public void CambiarColor(ConsoleColor nuevoColor)
```

```
        {  
            _color = nuevoColor;  
        }
```

```
        public override string texto
```

```
{  
    get  
    {  
        Console.ForegroundColor = _color;  
        return _texto.texto;  
    }  
}  
}  
}
```

CAPTURAS

```
Bienvenido al editor de Texto - Examen Unidad 3  
Ingrese su texto para comenzar: |
```

```
Seleccione el estilo que desea aplicar:  
1. Negrita  
2. Cursiva  
3. Subrayada  
4. Reestablecer estilos  
5. Cambiar color  
Opción:
```

```
Seleccione un color:  
1. Rojo  
2. Verde  
3. Azul  
4. Amarillo  
5. Morado  
6. Reestablecer color  
Opción: |
```

```
Texto con estilos aplicados:  
Hola Mundo!
```

```
¿Desea agregar otro estilo? (si/no):
```

```
Resultado final:  
Hola Mundo!
```