



## Program.cs (clase principal con el Main)

csharp

 Copiar

 Editar

```
using System;

namespace EjerP004
{
    internal class Program
    {
        static void Main(string[] args)
        {
            // Creamos un objeto de tipo Carpeta, usando el constructor con 3 parámetros
            Carpeta micar = new Carpeta("Matemática", false, color.negro);

            // Usamos la propiedad 'etiqueta' de la carpeta para asignar datos
            micar.etiqueta.apelNom = "Cecchi, David";
            micar.etiqueta.salon = "Labo 2";

            // Mostramos los valores por consola
            Console.WriteLine($"Materia: {micar.materia}");
            Console.WriteLine($"Apellido y Nombre: {micar.etiqueta.apelNom}");
            Console.WriteLine($"Aula: {micar.etiqueta.salon}");

            Console.ReadKey(); // Esperamos que el usuario presione una tecla
        }
    }
}
```





## Carpeta.cs (clase con propiedades y constructores)

csharp

📄 Copiar

🔧 Editar

```
namespace EjerP004
{
    // Enum para los colores posibles de la tapa de la carpeta
    public enum color
    {
        blanco,
        negro,
        rojo,
        azul
    }

    internal class Carpeta
    {
        // Campos privados
        private string _materia;
        private bool _esAnillada;
        private color _colorTapa;
        private Etiqueta _etiqueta;

        // Constructor por defecto
        public Carpeta()
        {
            this.materia = string.Empty;
            this.esAnillada = false;
            this.colorTapa = color.blanco;
            this.etiqueta = new Etiqueta(); // ⬇ inicializa con etiqueta vacía
        }
    }
}
```

```

// Constructor con 3 parámetros (crea una etiqueta vacía)
public Carpeta(string materia, bool esAnillada, color colorTapa)
{
    this.materia = materia;
    this.esAnillada = esAnillada;
    this.colorTapa = colorTapa;
    this.etiqueta = new Etiqueta(); // Se inicializa con etiqueta vacía
}

// Constructor con 4 parámetros (permite pasar una etiqueta creada)
public Carpeta(string materia, bool esAnillada, color colorTapa, Etiqueta etiqueta)
{
    this.materia = materia;
    this.esAnillada = esAnillada;
    this.colorTapa = colorTapa;
    this.etiqueta = etiqueta;
}

// Propiedades públicas
public string materia
{
    get { return this._materia; }
    set { this._materia = value; }
}

public bool esAnillada
{
    get { return this._esAnillada; }
    set { this._esAnillada = value; }
}

```

```

public color colorTapa
{
    get { return this._colorTapa; }
    set { this._colorTapa = value; }
}

public Etiqueta etiqueta
{
    get { return this._etiqueta; }
    set { this._etiqueta = value; }
}
}
}

```



## Etiqueta.cs (clase simple con propiedades automáticas)

csharp

📄 Copiar

📄 Editar

```
namespace EjerP004
{
    internal class Etiqueta
    {
        // Propiedades automáticas
        public string apelNom { get; set; }
        public string salon { get; set; }

        // Constructor por defecto
        public Etiqueta()
        {
            this.apelNom = string.Empty;
            this.salon = string.Empty;
        }

        // Constructor con parámetros
        public Etiqueta(string apelNom, string salon)
        {
            this.apelNom = apelNom;
            this.salon = salon;
        }
    }
}
```



### ¿Qué estás aprendiendo con este ejercicio?

- Cómo usar **clases compuestas** (Carpeta tiene una Etiqueta adentro).
- Qué son los **constructores sobrecargados** (varias versiones para crear objetos).
- Cómo usar un **enum** para representar un conjunto fijo de valores (`color`).
- Cómo organizar un proyecto en archivos separados pero trabajando en conjunto.

¿Querés que te lo resuma en una infografía visual o diagrama con flechas para entender cómo se relacionan?