

1. Diseño de una clase que simula un bucle despachador de eventos.

Ejemplo1:

NO SE ENCUENTRAN BUCLES DE DESPACHADOR DE EVENTOS, SOLO DESPACHADORES

Ejemplo2:

NO SE ENCUENTRAN BUCLES DE DESPACHADOR DE EVENTOS, SOLO DESPACHADORES

2. Definición de eventos personalizados dentro de la clase.

Ejemplo1:

```
public class Publicador
{
    // Declaración de un evento
    public event EventHandler EventoOcurrido;

    // Método que dispara el evento
    public void DispararEvento()
    {
        // Cuando ocurre el evento, se llama a todos los manejadores registrados
        EventoOcurrido?.Invoke(this, EventArgs.Empty);
    }
}
```

Ejemplo2:

```
public class EventDispatcher
{
    private readonly Dictionary<string, EventHandler> eventHandlers = new Dictionary<string, EventHandler>();

    public event EventHandler EventDispatched;
```

3. Manipulación de eventos mediante suscripciones y desencadenamientos.

Ejemplo1:

```
0 referencias
class Program
{
    0 referencias
    static void Main()
    {
        Publicador publicador = new Publicador();
        Suscriptor suscriptor = new Suscriptor();

        // Suscripción del manejador de eventos
        publicador.EventoOcurrido += suscriptor.ManejadorEvento;

        // Disparo del evento
        publicador.DispararEvento();
    }
}
```

Ejemplo2:

```
2 referencias
public void Subscribe(string eventName, EventHandler eventHandler)
{
    if (!eventHandlers.ContainsKey(eventName))
    {
        eventHandlers[eventName] = eventHandler;
    }
    else
    {
        eventHandlers[eventName] += eventHandler;
    }
}

1 referencia
public void Unsubscribe(string eventName, EventHandler eventHandler)
{
    if (eventHandlers.ContainsKey(eventName))
    {
        eventHandlers[eventName] -= eventHandler;
    }
}
```

4. Simulación de la recepción y procesamiento de mensajes utilizando eventos.

Ejemplo1:

```
public void DispararEvento()
{
    // Cuando ocurre el evento, se llama a todos los manejadores registrados
    EventoOcurrido?.Invoke(this, EventArgs.Empty);
}

// Referencias
public class Suscriptor

// Manejador de eventos que responde al evento
1 referencia
public void ManejadorEvento(object sender, EventArgs e)
{
    Console.WriteLine("El evento ha ocurrido.");
}
```

Recibido y procesado después por la clase “suscriptor”.

Ejemplo2:

```
4 referencias
public void DispatchEvent(string eventName)
{
    Console.WriteLine($"Evento '{eventName}' despachado.");
    EventDispatched?.Invoke(this, EventArgs.Empty);

    // Simulación de procesamiento
    Thread.Sleep(1000);
}
```

Luego es procesado por la función “evethandler”.