**Módulo: 02 - Características del Lenguaje C#**

**Tutorial: Eventos**

**Ejemplo 1**

1. Crear la siguiente clase:

class Empleados

{

private string \_nombre = string.Empty;

public string Nombre

{

get

{

return \_nombre;

}

set

{

\_nombre = value;

}

}

private string \_apellidos = string.Empty;

public string Apellidos

{

get

{

return \_apellidos;

}

set

{

\_apellidos = value;

}

}

private int \_edad = 0;

public int Edad

{

get

{

return \_edad;

}

set

{

\_edad = value;

}

}

public void Empleados(string nombre, string apellidos, int edad)

{

this.\_nombre = nombre;

this.\_apellidos = apellidos;

this.\_edad = edad;

}

}

1. La propuesta es reconocer de alguna manera cuando se cambia alguno de los campos de la clase instanciada. Por lo tanto, cuando se setea una propiedad se dispara un evento.
2. Para que el evento pueda ser tratado se debe definir un evento a través de un delegado genérico para tal fin. Agregar el delegado para manejar el evento en la clase Empleado para :

class Empleado

{

public event EventHandler DatosCambiados;

1. Para definir cuando se debe lanzar el evento:

public string Nombre

{

get

{

return \_nombre;

}

set

{

\_nombre = value;

DatosCambiados(this, new EventArgs());

}

}

1. Crear un ejemplo para modificar el nombre y disparar el evento

class Program

{

static void Main(string[] args)

{

Empleados emp = new Empleados("Jose", "Perez", 34);

emp.DatosCambiados += new EventHandler(emp\_DatosCambiados);

}

}

1. Crear el método para que el delegado pueda tratar el evento.

private void emp\_DatosCambiados(object sender, EventArgs e)

{

Console.WriteLine("Los datos fueron cambiados");

}

1. Compilar para ver si hay errores.

**Ejemplo 2**

1. Continuando ver como se comporta el tratamiento del evento. Agregar el siguiente código.

static void Main(string[] args)

{

Program p = new Program();

Console.ReadLine();

}

1. Agregar el nuevo seteo de la propiedad

public Program()

{

Empleados emp = new Empleados ("Jose", "Perez", 34);

emp.DatosCambiados += new EventHandler(emp\_DatosCambiados);

emp.Nombre = "Nuevo Nombre";

}

1. Compilar y ejecutar y ver que la salida muestra que el evento se disparado.
2. Poner un breakpoint en Empleados emp = new Empleados ("Jose", "Perez", 34); y ver como se van asignando las variables.
3. Que ocurre si el evento no tiene asignado un manejador de eventos. Comentar la línea:

//emp.DatosCambiados += new EventHandler(emp\_DatosCambiados);

1. Compilar y ejecutar y observar que aparece una excepción de referencia nula.
2. Para resolver se evalua si el evento es tiene asociado una referencia nula:

public string Nombre

{

get

{

return \_nombre;

}

set

{

\_nombre = value;

if (DatosCambiados != null)

{

DatosCambiados(this, new EventArgs());

}

}

1. Compilar y Ejecutar.
2. Para mostrar los resultados agregar la siguiente sobrecarga en la clase Empleados:

public override string ToString()

{

return \_apellidos + "," + \_nombre;

}

1. Agregar la salida en la clase Program:

Empleados emp = new Empleados ("Jose", "Perez", 34);

//emp.DatosCambiados += new EventHandler(emp\_DatosCambiados);

emp.Nombre = "Nuevo Nombre";

Console.WriteLine(emp.ToString());

1. Compilar y ejecutar. Observar el resultado.
2. Ahora manejemos el evento, descomentando la asignación del evento al manejador
3. Compilar y ejecutar. Observar el resultado.

**Ejemplo 3**

1. Vamos a crear un delegado propio para ello se crea el siguiente código

class Empleados

{

public delegate void DatosCambiadosEventHandler();

public event DatosCambiadosEventHandler DatosCambiados;

1. Modificar el método del evento de acuerdo al firma del delegado definido

set

{

\_nombre = value;

if (DatosCambiados != null)

{

DatosCambiados();

}

}

1. Modificar la asignación del evento con el método de manejará el evento.

Empleados emp = new Empleados ("Jose", "Perez", 34);

emp.DatosCambiados += new empleados.DatosCambiadosEventHandler(emp\_DatosCambiados);

emp.Nombre = "Nuevo Nombre";

1. Modificar el método definido según la firma del delegado asociado

private void emp\_DatosCambiados()

{

Console.WriteLine("Los datos fueron cambiados");

}

1. Compilar y ejecutar.