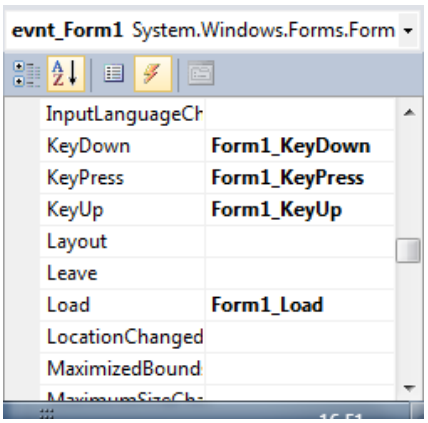
Programación Orientada a Eventos(PoE)

Me apoyaré con un ejercicio para desarrollar esta Unidad. En este ejercicio podemos aprender muchas cosas, puesto que el recurso **Formulario** es el "***contenedor***" sobre el cual "*diseñamos*" nuestros programas.

Es una buena idea estudiar muchos de los eventos que podemos capturar desde el formulario, ya que todos (*o la mayoría de ellos*) también existirán en los controles visuales que estudiaremos.

Primero para crear un evento sobre un control, sea un formulario o no hay que asegurarse que tenemos seleccionado el control que queremos, no todos los eventos están para todos los controles, y hacer click sobre el click sobre el rayo de la ventana de propiedades. Como se muestra en la imagen. Buscamos el evento en cuestión y hacemos doble click en la derecha del evento y automáticamente nos genera en el código del programa el método del evento, como se vio en los apuntes de formularioI, ahí se ceo haciendo doble_click sobre le botón, por que interesaba, fijaos en el evento del formulario KeyDown en el código nos ha generado




```
private void Form1_KeyDown(object sender, KeyEventArgs e)
    {
        // Código asociado a la tecla pulsada.
        // volvemos a utilizar <e>, pero en este caso, el parámetro
        // es de tipo: KeyEventArgs ... con lo cual, tendremos que
        // acceder al tipo a estudiar sus "posibilidades"
//que en este caso serán Alt, control,etc… la tecla pulsada

        }
```

Cuando creamos un evento por ejemplo el método asociado tiene unos argumentos, que paso a explicar con un ejemplo.

```
private void Form1_MouseMove(object sender, MouseEventArgs e)
```

MouseEventArgs es la clase base para las clases que contienen datos de eventos. por esta razón cada vez que tengamos una clase en la que queramos exponer eventos en sus argumentos, debemos heredarla.

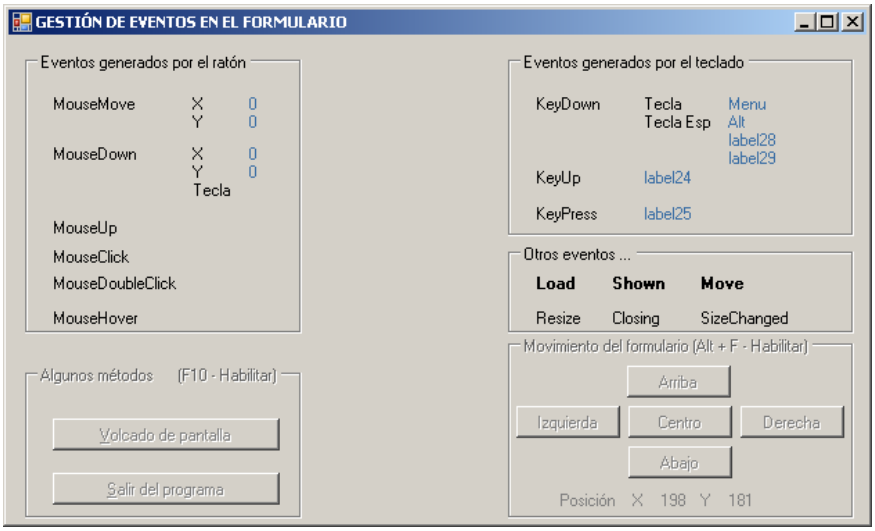
Hablando claro. Este evento se invoca cuando movemos el ratón, no? Si yo quiero saber la posición del ratón eso me lo dirá los argumentos e.X y e.Y.

¿Y como sabemos que propiedades lleva asociado el argumento e?

Primero por el msdn(ayuda de microsoft)

Segundo si en el código escribimos “e.” se despliega un menú contextual con todas las propiedades.
…y ¿Quién es sender? SENDER es el objeto que ha generado el evento. Por ejemplo si ponemos switch ((sender as Button).Text) { comprueba la propiedad “Text” del botón
case "Arriba":

Diseño del Formulario:



Para el diseño de este formulario se utilizan, solo, Button, Label y GroupBox, para insertar un control lo arrastraremos desde el cuadro de herramientas hasta el formulario y para poner el nombre iremos a las propiedades, buscaremos la propiedad **Text** y pondremos el nombre deseado.

Para mostrar el contenido de una etiqueta tendremos que poner:

```
label1.Text = "Ejecutado";
```

Donde label1 es el nombre genérico… nos va creando label1, label2, etc…

En las label’s solo se puede imprimir texto, por lo tanto, hay que hacer una conversión de tipo. Por ejemplo:

```
int PuntoY = e.Y;
label10.Text = System.Convert.ToString(PuntoY);
```

Para habilitar un control utilizarnos la propiedad Enabled=true/false; pero como es en tiempo de ejecución habrá que poner el nombre del control.Enabled=true; por ejemplo:

```
groupBox1.Enabled = true;
```

El funcionamiento del ejercicio parte de las siguientes situaciones:

1. **Eventos generados por el Ratón**.

    - Muestra las posiciones X e Y en las que se encuentra el ratón correspondientes a los eventos ***MouseMove*** y ***MouseDown***. En este último, además mostrará la tecla del ratón pulsada.

    - En la gestión de ***MouseUp*** se mostrará el mensaje "*Soltado*" una vez que se haya producido el evento.

    - La ejecución de ***MouseClick***, MouseDoubleClick o MouseHover, mostrarán el mensaje: "*Ejecutado*".

2. **Eventos generados por el Teclado**.

    - En la ejecución del evento ***KeyDown*** capturaremos y mostraremos la tecla pulsada y, si se da la situación oportuna, la secuencia especial de teclado pulsada (*ej. < Ctrl + Alt+ J >*).
    - Al soltar la secuencia de teclado pulsada, aparecerá el mensaje: "Soltado" junto al nombre del evento ***KeyUp***.
    - El carácter correspondiente a la tecla pulsada, aparecerá junto al nombre del evento: ***KeyPress***.
3. **Otros eventos**.
    - Los nombres de los eventos: ***Load***, ***Shown***, **Move**, ***Resize***, ***Closing*** y ***SizeChanged*** se irán presentando en **negrita** cuando hayan sido ejecutados.

4. **Movimiento del Formulario**.

    - Al inicio del programa, el contenido de este componente contenedor de tipo **groupBox** estará inhabilitado.

    - Al pulsar la secuencia de teclado <***Alt + F***> se habilitará el control y, por tanto, también su contenido.

    - La pulsación sobre los botones, permitirá al usuario final mover el Formulario en el sentido que indica el título de estos botones.

- Además, las etiquetas que hay bajo, mostrarán la posición inicial del formulario en la pantalla (*coordenadas <0,0> del formulario*).

5. ***Algunos métodos***.

   - Este groupBox también se encuentra inhabilitado al inicio del programa, pero se habilitará al pulsar la tecla de función <***F10***>.

   - A partir de estos momentos, la pulsación sobre los botones ejecutará la función que indique su título.