Una manera de plantearse el ejercicio de eventos y cualquier otro que tenga una interfaz gráfica, seria:

En primer lugar definiremos la interfaz, situaremos todas las etiquetas, y en general todos los controles donde se nos indique dándole un aspecto “amigable” para el usuario, intentando optimizar al máximo posible el espacio de la ventana(formulario), ni muchos espacios vacios ni muy condensado, sino todo lo contrario.

En segundo lugar, ya tenemos la ventana bien estructura ahora vamos a darle la funcionalidad que se nos pide, en este caso capturar los eventos del ratón y del teclado, para ello iremos a la ventana de propiedades y buscaremos los eventos y una vez localizados haremos click para que me genere la función o el código o el método para poder actuar…

Mouse	
MouseDown	**Form1_MouseDown**
MouseEnter	
MouseHover	**Form1_MouseHover**
MouseLeave	
MouseMove	**Form1_MouseMove**
MouseUp	**Form1_MouseUp**
Tecla	
KeyDown	**Form1_KeyDown**
KeyPress	**Form1_KeyPress**
KeyUp	**Form1_KeyUp**
PreviewKeyDown	

Mención especial evento Click y DoubleClick del ratón

Click	**Form1_Click**
DoubleClick	**Form1_DoubleClick**

Una vez hacemos click en la pestaña de Form1.cs nos aparecerá de manera inmediata el siguiente fragmento de código que nosotros tendremos que ampliar…

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
        {

            label11.Text = "Pulsado";

        }
private void Form1_MouseHover(object sender, EventArgs e)
        {
            label42.Text = "Ejecutado";

        }
private void Form1_MouseMove(object sender, MouseEventArgs e)
        {

            label24.Text = "ejecutado";
        }
```

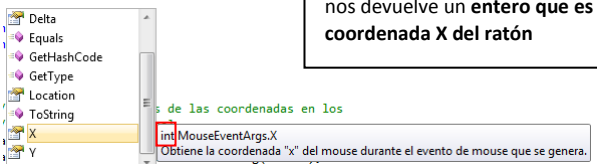
```
private void Form1_KeyDown(object sender, KeyEventArgs e)
        {
…

}
```

…y así con todos los evento, de tal manera que podemos comprobar que realmente cuando pulso una tecla o muevo el ratón, etc… lo tengo localizado, ahora bien quiero saber más, quiero saber que tecla o en qué posición se encuentra el puntero del ratón, ¿Cómo? Que hace la "e" según leí tiene las propiedades del evento, no es lo mismo un ratón que un teclado.

Por ejemplo, vayamos al evento MouseMove

```
private void Form1_MouseMove(object sender, MouseEventArgs e)
        {
            // Leemos los valores de las
propiedades X, Y de
            // <e> que es de tipo MouseEventArgs
            // ---       (ver prototipo de evento)
            // tecleamos la e. y no sale un menú como
el siguiente ---
```



De donde podemos deducir que la X nos devuelve un **entero que es la coordenada X del ratón**

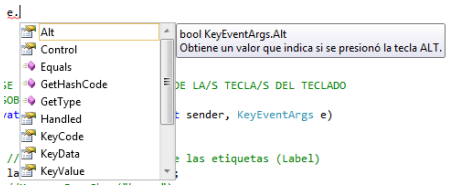
Por lo tanto solo nos queda imprimirla o mostrarla por pantalla, como las labels solo pueden sacar strings o cadena de caracteres hacemos la conversión siguiente:

```
            // Mostramos los valores de las coordenadas en los
            // controles de tipo Label
            label9.Text = e.X.ToString();
            label10.Text = e.Y.ToString();
            label24.Text = "";
        }
```

Ahora veamos un ejemplo de Teclado, cada vez que pulsemos una tecla que me diga que tecla es

```
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"
…y lo mismo e. y nos sale el menú contextual de "e" para el
KeyDown
```



Y nos queda buscar la opción que más se aproxime a lo que buscamos y mostrarla

Ahora la conversión de tipo de otra manera

```
            label26.Text = System.Convert.ToString(e.KeyCode);
```

podemos crear unas etiquetas para ver si se ha pulsado el Alt, Shift o Control que, como podemos ver en el ejemplo son de tipo booleano.

```
              // Devolverá True o False dependiendo si se pulsa o no.
              labelXX.Text=e.Alt.ToString();

        }
```

…y así sucesivamente.