



## UNIDAD 1

### Confección de interfaces de usuario 1.8 Eventos en Windows Forms

## EJERCICIO 1: CREAR UN MONITOR DE EVENTOS

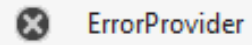
- En Visual Studio, crear una nueva aplicación de Windows Forms.
- Desde el Cuadro de herramientas, arrastrar un control Label y un control ListBox al formulario.
- En la ventana Propiedades, establecer la propiedad Text de la etiqueta con el texto: *El ratón aquí!!!* Ajusta el tamaño y tipo de fuente también en las propiedades.
- Seleccionar la etiqueta. En la ventana Propiedades, generar los controladores de eventos predeterminados de ratón siguientes: *Click, DoubleClick, MouseClick, MouseDoubleClick, MouseDown, MouseEnter, MouseHover, MouseLeave, MouseMove* y *MouseUp*.
- En el Editor de código, agregar código para cada uno de los controladores de eventos predeterminados, de tal manera que se añada al ListBox un texto indicando el evento que se ha producido, por ejemplo, si se hace clic sobre la etiqueta, se añadirá un elemento al ListBox con el texto "Clic" y así sucesivamente.
- Guarda y prueba el resultado de tu proyecto.

## EJERCICIO 2: EVENTOS DE TECLADO

- En Visual Studio, crear una nueva aplicación de Windows Forms.
- Añadir un control de *etiqueta*, un *cuadro de texto* y un *botón de comando*.
- El cuadro de texto permitirá controlar mediante los eventos *KeyPress*, *KeyDown* y *KeyUp* que los ocho primeros caracteres sean números y el noveno únicamente letra. (*llega con controlar el primero: KeyPress*)
- El botón de comando permanecerá desactivado mientras no se introduce esa cantidad de números y letra.
- Al pulsar el botón verificar NIF se comprobará si la letra se corresponde con el NIF introducido. En caso afirmativo mostrará un cuadro de mensaje indicando que es correcto y en caso negativo mostrará esa imagen de admiración en rojo al lado del cuadro de texto y el *tooltip* con el texto informando del error.

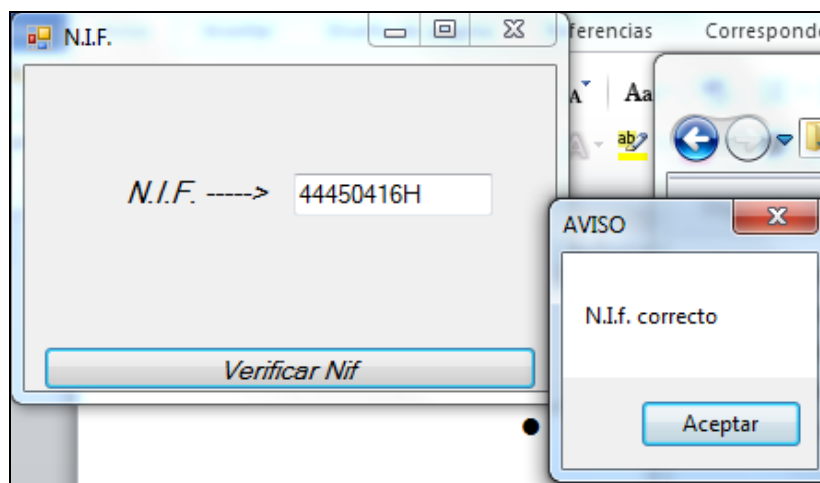
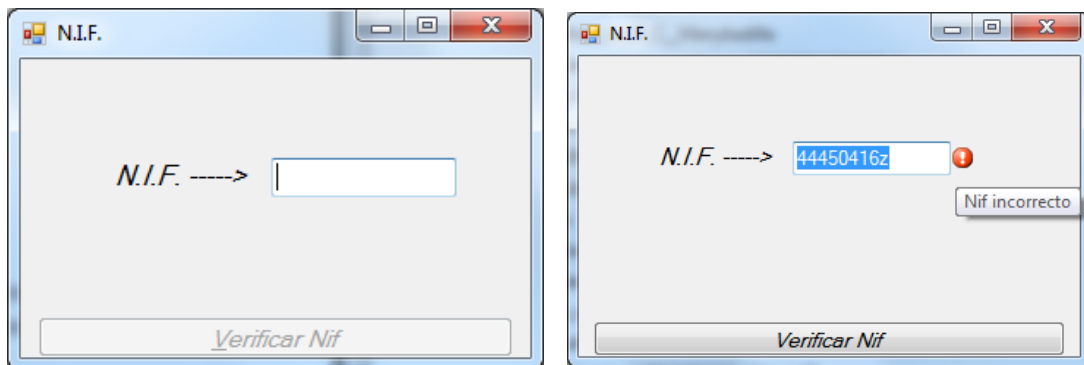


Esto puede hacerse mediante el objeto **ErrorProvider**, que podemos añadir arrastrándolo del cuadro de herramientas de Visual Studio.



Este componente se utiliza para informar al usuario de manera gráfica que ha ocurrido un error. [Más información](#)

El método más importante de `ErrorProvider` es `setError`



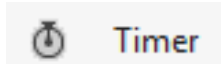
Para ello se pueden emplear también los eventos **Validating** y **Validated** aplicados, en este caso al cuadro de texto.

La principal diferencia es que en el primer caso (*Validating*) tiene lugar cuando el control se está validando y en el segundo caso (*Validated*) se produce cuando finaliza la validación del control.

## EJERCICIO 3: CREAR UNA ALARMA CON EL COMPONENTE TIMER

Realiza un proyecto que cree la alarma de un reloj para el día o para todos los días de la semana (a la misma hora).

Para ello se utilizará el objeto **timer**, que se puede añadir desde el cuadro de herramientas.



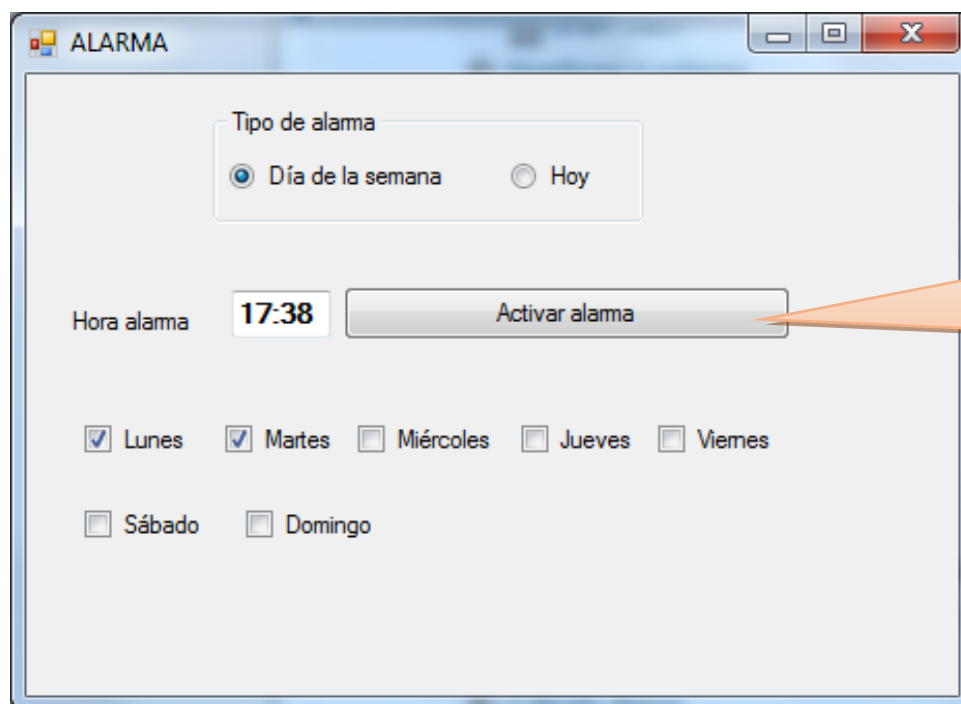
*Este componente produce un evento a intervalos regulares.*

*El evento asociado al componente timer es `Tick`*

Los métodos para iniciar o parar el temporizador son `Start` y `Stop`.

[Más información del componente timer](#)

*La apariencia podrá ser similar a la siguiente:*



El botón cambiará a "Desactivar alarma" cuando esté activada. Y se deshabilitarán los botones de opción...

- Cuando se pulse **Activar alarma** se activará la alarma del sistema, que sonará a la hora y minutos indicados en el cuadro de texto de la hora de la alarma.



*La alarma del sistema puede activarse con las llamadas a los métodos: `Console.Beep()` o bien `SystemSounds.Beep.Play()`, añadiendo las librerías `System.Media` en este último caso.*

- Habrá que comprobar si se ha seleccionado el botón de opción *hoy*, con lo cual sólo se comprobará la hora del temporizador. Si se ha seleccionado la opción *día de la semana*, habrá que ver qué días de la semana están marcados en los checkbox para que la alarma suene los días indicados.
- En la barra de tareas del sistema operativo, situado a la **derecha**, deberá aparecer un icono de la alarma (poner la imagen que se quiera).  
Para esto haremos uso del componente **NotifyIcon**, asignándole la imagen que consideremos necesario.

[Más información sobre NotifyIcon](#)

