

Eventos

23

Programación II y Laboratorio de Computación II

Edición 2018

Eventos

- Un evento es el modo que tiene una clase dada de proporcionar notificaciones a sus clientes cuando ocurre algo en particular dentro del objeto.
- El uso más habitual para los eventos lo vemos en las interfaces gráficas (evento Click de un botón, evento Load de un Form, etc.).
- Los eventos proporcionan un medio apropiado para que los objetos puedan señalizar cambios de estado que pueden resultar útiles para los clientes de ese objeto.

Eventos

- Un evento es un mensaje enviado por un objeto para indicar que se ha producido una acción invocada programáticamente o por un usuario.
- Cada evento tiene un emisor que produce el evento y un receptor que lo captura.
- Utilizando eventos, los componentes de la interfaz avisan a la lógica de negocios que el usuario ha ejecutado alguna acción sobre los componentes de la misma (por ejemplo: presionar el botón del Mouse o presionar una tecla).

Delegados y eventos

- El objeto que produce (desencadena) el evento se denomina *emisor* del evento.
- El procedimiento que captura el evento se denomina *receptor* o *manejador* del evento.
- En cualquier caso, el emisor no sabe qué objeto o método responderá a los eventos que produzca.
- Por ello, es necesario tener un componente que enlace el emisor del evento con el receptor del evento.

Delegados y eventos

- El Framework .NET utiliza un tipo de delegado para trabajar como un puntero a función entre el emisor y el receptor del evento.
- En la mayoría de casos, el Framework .NET crea el delegado y se ocupa de gestionar los detalles por nosotros.
- Sin embargo, es posible crear delegados para los casos en que se desee que un evento utilice diferentes controladores de eventos en diferentes circunstancias.

Delegados

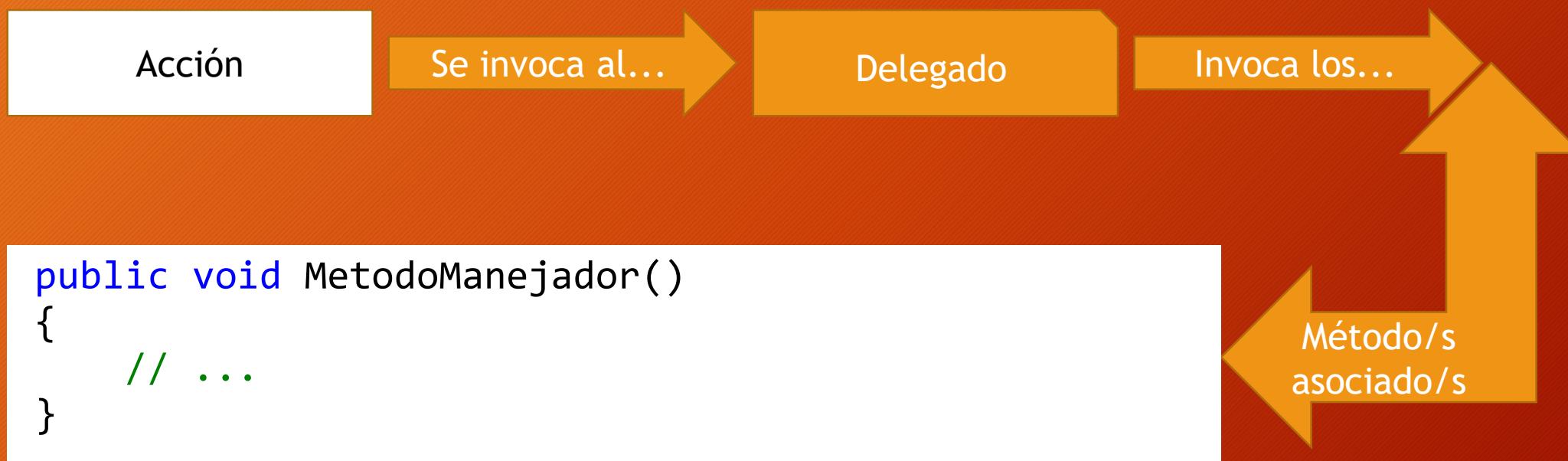
- Los eventos se declaran mediante delegados.
- Un delegado es un tipo que representa referencias a métodos con una lista de parámetros determinada y un tipo de valor devuelto.
- Un objeto delegado encapsula un método de modo que se pueda llamar de forma anónima.
- Cuando ocurre el evento, se llama a los delegados que proporcionan los clientes para el evento.

Delegados

- Los delegados son como los punteros de función de C++, pero tienen seguridad de tipos.
- Los delegados permiten pasar los métodos como parámetros.
- Los delegados pueden encadenarse entre sí; por ejemplo, se puede llamar a varios métodos en un solo evento.
- Un Evento puede tener múltiples manejadores y viceversa.

```
public delegate void MiDelegado(object sender, EventArgs e);  
  
public event MiDelegado ElEvento;
```

Esquema de Eventos



Manejadores

- Para asociar un evento a un manejador de eventos en tiempo de ejecución, hay que 'agregarlo' al evento del emisor.

```
objEmisor.evento += MetodoManejador;
```

- La instrucción **+ =** agrega a la lista de invocación del evento del 'emisor', el nuevo manejador.

Ejemplo

```
button1.Click += MetodoManejador;  
// ...  
private void MetodoManejador(Object sender, EventArgs e)  
{  
    MessageBox.Show("Uso el manejador de eventos");  
}
```

Manejadores

- Para quitar un evento de un manejador de eventos en tiempo de ejecución, hay que utilizar la instrucción -=.

```
objEmisor.evento -= MetodoManejador;
```

- La instrucción -= quita de la lista de invocación del evento del 'emisor', el manejador.