



PRÁCTICA 10-2: ¡¡EXTRA, EXTRA!! UN DISPOSITIVO HA CAMBIADO SU ESTADO

1. Introducción

A la hora de desarrollar software contamos con una serie de soluciones plantilla, o técnicas de diseño, que suelen resultar de utilidad en un amplio caso de situaciones recurrentes. Una de dichas técnicas son los esquemas de *publicación-suscripción*. Estos esquemas permiten que un objeto pueda suscribirse de forma dinámica a una serie de eventos que son de su interés, así cómo retirar, también de forma dinámica, dicha suscripción. Cuando se produce un evento, se notifica a todos sus suscriptores, de forma que éstos puedan actualizar su estado.

Para implementar este esquema, es necesario crear una estructura que asocie a cada tipo de evento una lista de de suscriptores a dicho evento. Dicha estructura debe proporcionar métodos para:

- (1) Permitir a los objetos que generan eventos registrarse como *autores*.
- (2) Permitir un objeto suscribirse a un determinado tipo de evento;
- (3) Permitir a un objeto retirar su suscripción a un determinado evento.
- (4) Permitir a los objetos autores publicar eventos. En este caso, se notificará a la lista de suscriptores acerca de la publicación de dicho evento. Adviértase que para recibir tales notificaciones los objetos que se suscriben a los eventos deben poseer algún tipo de método que se invoque con tal propósito.

Por ejemplo, supongamos que en el hogar inteligente tenemos como dispositivos, entre otros, aparatos de frío/calor y termómetros. Los termómetros actuarían como objetos *autores* y publicarían eventos para indicar que la temperatura ha cambiado. Los aparatos de frío/calor, cuando se encuentren activos, se suscribirán al evento "*cambio de temperatura*" de aquellos termómetros que sean de su interés, como por ejemplo, los que estén en su misma habitación. De esta forma, si un termómetro cambie de temperatura, los aparatos de frío/calor que estén activos e interesados en dicho termómetro recibirán la correspondiente notificación y podrán actuar en consecuencia, entrando en modo enfriar o calentar con objeto de mantener la temperatura deseada.

Las ventajas de este esquema son dos:

- (1) Reduce el acoplamiento entre dispositivos. Si añadimos un nuevo dispositivo que esté interesado en los cambios de temperatura de dicho termómetro, sólo habrá que añadirlo a la lista de suscriptores.
- (2) Permite que los dispositivos puedan suscribirse y eliminar suscripciones a eventos dinámicamente, es decir, en tiempo de ejecución, sin necesidad de parar la aplicación modificarla, recompilarla y ejecutarla de nuevo.

En esta práctica añadiremos al código creado en la práctica anterior termómetros como nuevos dispositivos. Además implementaremos un esquema de publicación suscripción para coordinar el funcionamiento de termómetros y aparatos de frío/calor.



2. Objetivos

Los objetivos generales de estas prácticas son:

- (1) Aprender a implementar esquemas de cierta complejidad a partir de su descripción utilizando TADs.
- (2) Aprender a crear estructuras de cierta complejidad que involucren TADs de diferentes tipos.
- (3) Aprender a usar implementaciones de TADs realizadas por terceros.
- (4) Aprender a usar TADs para implementar un esquema de publicación-suscripción.
- (5) Aprender a diferenciar el concepto de interfaz e implementación.
- (6) Aprender a escoger la implementación más adecuada para un TAD concreto en función del contexto donde se va a usar dicho TAD.

3. Actividades

El alumno deberá completar las siguientes actividades para una correcta realización de la práctica.

- (1) El alumno deberá obtener de *moodle* un fichero *zip* con una serie de ficheros de código fuente que le serán de utilidad para la realización de la práctica.
- (2) El alumno deberá hacer que la clase *Gateway*, creada en la práctica 10-1, implemente la interfaz *IThermometerMng* (proporcionada al alumno), con objeto de añadir la gestión de termómetros al hogar.
- (3) El alumno deberá implementar un esquema de publicación-suscripción en una clase *PublishSubscribe* con los siguientes objetivos:
 - a. Cada dispositivo en la casa deberá ser capaz de publicar eventos. Cada dispositivo se identifica por su *id*, que es único para cada tipo de dispositivo.
 - b. Cada dispositivo podrá publicar diversos eventos, que están recogidos en la clase *HouseEvent* (proporcionada al alumno)¹.
 - c. Un dispositivo deberá poder darse de alta en el esquema como *autor*, indicando que tipo de evento publica. Igualmente, deberá poder darse de baja.
 - d. Un dispositivo deberá poder darse de alta como *suscriptor*, indicando a qué tipo dispositivo y evento publicado por dicho dispositivo se suscribe. Igualmente, todo dispositivo deberá poder darse de baja de las listas a las cuales está suscrito.
 - e. Un dispositivo autor deberá ser capaz de publicar eventos. La publicación de un evento conllevará la correspondiente notificación a la lista de suscriptores.

NOTA: Todos los dispositivos suscriptores deberán ser del tipo *ISuscriptor*, es decir, deberán implementar dicha interfaz.

¹ Actualmente dicha clase solo contiene un evento *ThermometerChanged* el cual debería poder ser solo publicado por termómetros.



- (4) A continuación se usará la estructura creada en el punto (3) para coordinar el funcionamiento de los aparatos de frío/calor y los termómetros de la siguiente forma:
- Cada vez que añadamos un termómetro nuevo a la casa, deberemos darlo de alta en el esquema de publicación-suscripción como *autor* para el evento *ThermometerChanged*.
 - Cada vez que un aparato de frío-calor se active (la clase *Heater* se proporciona al alumno), deberemos darlo de alta como suscriptor en la lista de suscriptores correspondientes a todos los termómetros situados en su misma habitación.
 - Cada vez que un aparato de frío-calor se desactive, deberemos darlo de baja como suscriptor de todas las listas a las que se encuentre suscrito.
 - Siempre que se activa un aparato de frío-calor o se ajusta su temperatura, éste no comienza a funcionar hasta que recibe un evento de cambio de temperatura. Por tanto, para forzar a los aparatos de frío-calor a trabajar cuando se activan o se les modifica la temperatura deseada, siempre que el *Gateway* recibe un método relacionado con alguna de dichas acciones, obtendrá la temperatura de todos los termómetros en la misma habitación que el aparato correspondiente, y por cada termómetro, el propio *Gateway* publicará el correspondiente evento.
- (5) El alumno deberá probar que su aplicación funciona correctamente. Para tal propósito se le proporciona a través de la plataforma *moodle* una clase *Runner* con un amplio abanico de casos de prueba.

4. Criterios Generales de Autoevaluación y Aclaraciones

La práctica se deberá entregar el día de la segunda prueba evaluable parcial y será evaluada y calificada en dicho momento. Los criterios expuestos en esta sección están destinados a permitir que el alumno pueda autoanalizar cuando la solución dada a la práctica cumple con los criterios de calidad necesarios para superar la asignatura.

Se aplican todos los criterios expuestos en la Practica 10-0. Además, la práctica se considerará resuelta correctamente cuando se satisfagan los casos de prueba proporcionados en la clase *Runner*. Se valorará la eficiencia de los métodos, la elección adecuada de las implementaciones,

Pablo Sánchez Barreiro.