



## PRÁCTICA 10-0: HOGARES INTELIGENTES QUE NO LO SON TANTO

### 1. Introducción

La mayoría de las aplicaciones software que se usan a diario están desarrolladas basadas en el uso de Tipos Abstractos de Datos, implementados por los propios equipos de desarrollo o simplemente reutilizados de terceras partes. A la hora de desarrollar software con Tipos Abstractos de Datos, todo Ingeniero Software debe tomar dos decisiones: (1) seleccionar el Tipo Abstracto de Datos más adecuado para la tarea que desea realizar; y (2) elegir la implementación más adecuada para dicho Tipo Abstracto de Datos de acuerdo con las características propias de la aplicación a desarrollar.

La primera actividad, la selección del Tipo Abstracto de Datos, no suele conllevar mayores dificultades, dado que se deduce normalmente de forma muy natural de las propias características del problema que TAD debemos usar.

La segunda actividad es la que suele conllevar mayores problemas, pues no suele existir una respuesta única al problema, sino que es tarea del desarrollador analizar las ventajas e inconvenientes de cada técnica de implementación para hacer uso de aquella que proporciona unas mayores ventajas y unos menores inconvenientes, de acuerdo con la naturaleza del problema. En muchas ocasiones no hay una clara elección en este sentido, por lo que el desarrollador software debe tomar ciertas decisiones basándose en su propia experiencia y en ciertos supuestos. Dos desarrolladores software podrían tomar decisiones diferentes y ser las dos correctas, al estar ambas perfectamente justificadas. Sólo el tiempo y en ocasiones el azar harán que la decisión tomada por uno de ellos se torne mejor que la otra.

Dado la debilidad de la corrección de dichas decisiones, se torna vital que las mismas se puedan deshacer fácilmente, con objeto de modificarlas en cuanto se detecte que los supuestos sobre los que se habían tomado eran en realidad falsos.

En esta práctica se desarrollará un software, adaptado al nivel del alumno, para el control de un hogar inteligente, donde varios dispositivos se gestionan de forma coordinada para mejorar el confort y la seguridad de sus habitantes, así como para hacer un uso más inteligente de los recursos. Dicho sistema está basada en descripción del software para un hogar inteligente proporcionada por Siemens AG.

Esta práctica se desarrollará en varias sesiones de prácticas, en cada una de las cuales se entregará un nuevo enunciado donde se detallen los aspectos específicos de cada práctica.

### 2. Objetivos

Los objetivos generales de estas prácticas son:

- (1) Aprender a usar TADs para la construcción de software.
- (2) Aprender a diferenciar el concepto de interfaz e implementación.
- (3) Aprender a escoger la implementación más adecuada para un TAD.
- (4) Aprender a usar implementaciones de TADs realizadas por terceros.
- (5) Aprender a crear estructuras complejas basadas en TADs.



### 3. Descripción General del Software para Hogares Inteligentes

Esta sección describe de forma global el sistema software sobre el que versarán las prácticas de las siguientes secciones. En ningún momento se deberá implementar el sistema software completo. De hecho, se desarrollará una parte mínima del mismo. Esta descripción simplemente se proporciona para que el alumno pueda obtener una visión general del sistema como un todo, en caso de que dicha visión le ayude a la hora de diseñar y desarrollar software; y para que asimile la complejidad natural de los sistemas software.

El objetivo de un hogar inteligente es aumentar la comodidad y seguridad de sus habitantes, así como hacer un uso más eficiente de la energía consumida. Los ejemplos más comunes de tareas automatizadas dentro de un hogar inteligente son el control de las luces, ventanas, puertas, persianas, aparatos de frío/calor, así como otros dispositivos, que forman parte de un hogar. Un hogar inteligente también busca incrementar la seguridad de sus habitantes mediante sistemas automatizados de vigilancia y alerta de potenciales situaciones de riesgo. Por ejemplo, el sistema debería encargarse de detección de humos o de la detección de ventanas abiertas cuando se abandona el hogar.

El funcionamiento de un hogar inteligente se basa en el siguiente esquema: (1) el sistema lee o recibe datos de una serie de sensores; (2) se procesan dichos datos; y (3) se activan los actuadores para realizar las acciones que correspondan en función de los datos recibidos de los sensores. El sistema también lleva a cabo una serie de funciones a petición expresa del usuario.

Todos los sensores y actuadores se comunican a través de un dispositivo especial denominado puerta de enlace (*Gateway*, en inglés). Dicho dispositivo se encarga de coordinar de forma adecuada los diferentes dispositivos existentes en el hogar, de acuerdo a los parámetros y preferencias especificados por los habitantes del mismo. Los habitantes del hogar se comunicarán con la puerta de enlace a través de una interfaz.

Cada hogar puede tener un número variable de plantas y habitaciones. El número de habitaciones por planta es también variable. Un hogar inteligente ofrecer varios servicios, Dichos servicios se clasifican en funciones básicas y complejas, las cuales describimos a continuación.

#### **Funciones básicas**

1. **Control automático de luces:** Los habitantes del hogar deben ser capaces de encender, apagar y ajustar la intensidad de las diferentes luces de la casa. El número de luces por habitación es variable. El ajuste debe realizarse especificando un valor de intensidad.
2. **Control automático de ventanas:** Los residentes tienen que ser capaces de controlar las ventanas automáticamente. De tal modo que puedan indicar la apertura de una ventana desde las interfaces de usuario disponibles.
3. **Control automático de persianas:** Los habitantes podrán subir y bajar las persianas de las ventanas de manera automática desde las interfaces de usuario.



4. **Control automático de temperatura:** El usuario será capaz de ajustar la temperatura de los aparatos de frío/calor desde las interfaces de usuario.

#### ***Funciones complejas***

1. **Control inteligente de energía:** Esta funcionalidad trata de coordinar el uso de ventanas y aparatos de frío/calor para regular la temperatura interna de la casa de manera que se haga un uso más eficiente de la energía. Por ejemplo, si se recibe la orden de calentar la casa, a la vez que se activan los radiadores se cerrarán las ventanas para evitar las pérdidas de calor.
2. **Presencia simulada:** Para evitar posibles robos, cuando los habitantes abandonen la casa por un periodo largo de tiempo, se deberá poder simular la presencia de personas en las casas. Hay dos opciones de simulación (no exclusivas):
  - a. **Simulación de las luces:** Las luces se deberán apagar y encender para simular la presencia de habitantes en la casa.
  - b. **Simulación de persianas:** Las persianas se deberán subir y bajar automática para simular la presencia de individuos dentro de la casa.

#### **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.

En general, para una correcta resolución de la práctica, se deberán satisfacer los siguientes criterios:

- (1) Se cumplen las normas expuestas en la *Guía de Buenas Prácticas en Programación*. Cada incumplimiento de una norma se penalizará en la prueba evaluable correspondientes a estas prácticas con una merma de entre 0.1 y 0.5 en la calificación final del alumno.
- (2) Las variables nunca poseen como tipo implementaciones concretas de un TAD, sino su interfaz. Por ejemplo, si una variable es una secuencia, su tipo declarado es *List* y no *ArrayList* o *LinkedList*.
- (3) Para cualquier método, si se satisface su precondition, dicho método no genera ninguna excepción y termina correctamente.
- (4) Ninguna llamada a método viola su precondition.
- (5) No se acceden directamente a los atributos de una clase desde fuera de dicha clase. El incumplimiento de esta norma supondrá el suspenso automático.
- (6) No se usan, salvo imponderables, métodos o atributos estáticos.
- (7) La visibilidad de cada atributo es la adecuada.
- (8) No existen redundancias dentro del código desarrollado.

*Pablo Sánchez Barreiro.*