



Trabajo Práctico de Análisis y Diseño Orientado a Objetos

Grupo

Conformamos el grupo 3

Profesor

Stricagnoli, Matias Damian

Justificación de patrones de diseño utilizados

A lo largo del Trabajo Práctico Obligatorio hemos adquirido las prácticas necesarias para cumplir con el mismo, teniendo en cuenta los Patrones GRASP y los Principios SOLID. Como también, hemos utilizado los siguientes patrones: Adapter, Strategy, Decorator y Observer, donde se explicará el uso de cada uno a continuación.

Patrón Adapter

Hemos hecho uso de este patrón estructural en tres ocasiones debido a que encapsula el uso de la clase que se quiere adaptar en otra clase que concuerda con la interfaz requerida:

En el “login”, ya que la consigna afirma que es un servicio externo a la plataforma, por ende, no sabemos cómo funciona.

Lo mismo sucede con las mediciones, en donde también lo utilizamos ya que no sabemos cómo es que se toman esas mediciones, si es a través de algún sistema especial, una balanza, etc.

Usamos también un Adapter en el Firebase, ya que consideramos que en un futuro puede llegar a cambiar este servicio, y de esta forma se puede agregar uno nuevo sin necesidad de tocar el código.

Patrón Strategy

Se ha utilizado este patrón de comportamiento debido a que permite disponer de varios métodos para resolver un problema puntual y elegir cual usar en el momento de ejecución.

En esta ocasión, usamos en objetivo un Strategy, dado que tenemos tres estrategias que podemos seleccionar al momento de buscar una rutina, y el socio puede cambiar de estrategia en cualquier momento.

Patrón Decorator

Utilizamos este patrón ya que permite que cualquier objeto pueda complementarse con un comportamiento sin influir en las funcionalidades de otros objetos de la misma clase.

En este caso, lo utilizamos en EjercicioReforzado porque contamos con condicionales (los valores reforzados) que amplían las acciones que realiza el objeto. Entonces si Ejercicio recibe EjercicioReforzado, cambia la forma en la que se muestra dicho ejercicio.

Patrón Observer

Se ha utilizado este patron debido a que permite observar cambios producidos por un objeto, donde cada cambio que afecte el estado del objeto observado lanzará una notificación a los observadores. En este caso, se les notificará a los socios que ganaron un trofeo. No interesa que hará el socio con la información de que recibió un trofeo.