TrabajoPractico Final:Estacionamiento

Integrantes:

*Herrera Juan Ignacio (mail: JuanHerreraUnq@outlook.com.ar)

*Luis Favatier (mail: favatier@gmail.com)

Patrones de diseño utilizados: Template Method,Observer,Strategy y State.

Se ha usado el patron Strategy en la clase AppCelularEstacionamiento para poder cambiar entre los modos manual y automatico de forma manual. Para que la clase cambie de estado de “a pie” o “en auto” de forma automatica haciendo uso de los mensajes de la interface MovementSensor usamos el patron State.

Para las siguientes jerarquias se ha usado el patron Template:

- Ventas (VentaApp, VentaPuntual)

- Estacionamiento (EstatacionamientoApp, EstacionamientoPorHoras)

Para las alertas a suscriptores se ha utilizado el patron Observer implementado con la clase SEM_Alertas y suscriptores.

Se ha creado las clases SEM_Alertas, SEM_ZonasDeEstacionamiento, SEM_Usuarios, SEM_Estacionamientos y SEM_Multa. Se ha hecho esto para juntar todo el comportamiento que deberia tener el sistema en lugares focalizados y para que no quede una clase con muchas responsabilidades de distinto tipo. Ademas de esta forma cumplir con el principio solid de single resposability. Si estuvieran unidas en una sola clase a largo plazo podria traer problema ya que si se modifica aluna regla de negocio de un sector de la SEM afectaria al resto de la clase.

Los principios solid se han tratado de mantener en todo el sistema.

Cada SEM implementa una o dos interfaces para comunicarse con el resto del sistema para que de esa forma las clases esten desacopladas y que dependan de una abstraccion.