TrabajoPractico Final:SEMEstacionamiento

Integrantes:

\*Herrera Juan Ignacio (mail: [JuanHerreraUnq@outlook.com.ar](mailto: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 automático de forma manual. Para que la clase cambie de estado de “a pie” o “en auto” de forma automática haciendo uso de los mensajes de la interface MovementSensor usamos el patron State.

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 debería tener el sistema en lugares focalizados y para que no quede una clase con muchas responsabilidades de distinto tipo. Además de esta forma cumplir con el principio solid de single responsability. Si estuvieran unidas en una sola clase a largo plazo podría traer problema ya que si se modifica alguna regla de negocio de un sector de la SEM afectaría 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 estén desacopladas y que dependan de una abstracción.

Cambios Re entrega:

* Se arregló el modelo al agregar una sem principal que conoce a todas las otras sem individuales, y su funcionalidad es la de proveer las sem a otras cuando se le solicite.
* Se implementaron todas las interfaces que faltaban implementar.
* Se corrigió el uml.
* Parte de state agregada al uml y realizados test.
* Coverage al 96.6% de la carpeta src.