Objetos II – TP Integrador

**Integrantes:**

- **Contardo Juan Pablo – contardo.juanpablo@gmail.com**
- **Roberto Leonardo Medici - leomedici06@gmail.com**
- **Matias Gonzales Aduco - matias.aduco@gmail.com**

## Implementacion del principio SOLID

### S - Single Responsibility Principle (SRP)

Gran parte del trabajo se genero a la par de este principio. Desde las clases mas básicas como pueden ser los estacionamientos o las mas complejas como el SEM, siempre se intento que tengan una única responsabilidad y cumplan su fin.

### O - Open/Closed Principle (OCP)

Desdes un comienzo se implementaron las clases que serian la base del trabajo practico, siempre pensando en realizar la menor cantidad de modificaciones a futuro. El código fue creciendo con el correr de los días y mediante los test se fue comprobando su funcionamiento.

### L - Liskov Substitution Principle (LSP)

Una gran porción del código perteneciente al trabajo practico son subclases. Por lo tanto una manera de llevar control sobre la utilidad que tenían estas mismas fue mediante este principio. Es decir, un caso que encontramos fue el de la app, cuando pusimos en la balanza si la app debería ser una superclase que se extienda a la app del inspector o la app del usuario descubrimos que no tenían puntos en común, por lo tanto se decidió ponerlas en clases separadas.

### I - Interface Segregation Principle (ISP)

Solo se implantó la interfaz perteneciente al sensor de movimiento. Pero cumple con las bases pertenecientes a este principio ya la interfaz cuenta con muy pocos métodos.

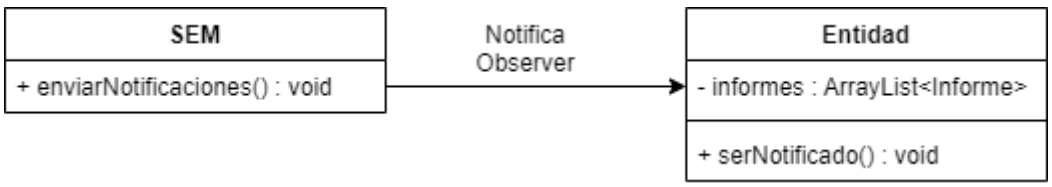
### D - Dependency Inversion Principle (DIP)

Este fue el principio que mas nos costo poder mantener, la clases pertenecientes al trabajo integrador dependen en muchos de los casos que exista el SEM. Pero esas clases que tienen de alguna forma una dependencia, seria raro implementarlas fuera del contexto dado. Un caso para poner de ejemplo es el inspector, que en sus metodos recibe como parametro al SEM, no tendría mucho sentido la figura de un inspector sin el ente al que pertenece.

Pratones de diseño:

Observer:

Para reolver el SEM pueda notificar a la entidad, se penso en patron de diseño observer, la Entidad espera que se produsca un cambio interno en el SEM para resivir una notificacion.



Considereciones:
A la hora de que un usuario registre mediante su app un estacionamiento, solo tendra en cuenta si esta dentro de la franja horaria especificada en el TP (7 A 20 HS).

Una zona siempre tendra asigando un inspector que se encargue de realizar su labor en esa area.

La app del usuario y la app del inspector no tienen ninguna relacion que las vincule para que sean una subclase de App, por lo tanto se decidio que sean dos apps por separado sin extension.

JRE Version: 1.8.0_271