Trabajo Práctico Final

Sistema de Estacionamiento Medido

Integrantes:

- Emiliano Borzi emilianoborzi@gmail.com
- Maribel Claros claross.maribel@gmail.com
- Ernesto Domato edomat@gmail.com

Universidad Nacional De Quilmes
Primer Cuatrimestre 2024

Decisiones de diseño:

Lo primero que hicimos fue juntarnos virtualmente para releer todos juntos el enunciado del TP mientras empezamos a hacer el UML. Para eso utilizamos la herramienta www.plantuml.com ya que nos permite escribir el diagrama en código y olvidarnos de dibujarlo nosotros mismos.

Una vez que estuvimos satisfechos con la base del diseño, nos dividimos los objetos entre todos y así avanzar con la siguiente etapa que era el desarrollo.

Al terminal los desarrollos con sus respectivos test, completamos entre todos el UML para que quede consistente con el código.

Patrones de diseño:

En este proyecto utilizamos 3 patrones de diseño:

- **Strategy:** Para implementar el modo de estacionamiento (Manual, Automático). Tiene tres métodos polimórficos. El primero se usa para notificar el inicio y fin de estacionamiento según el tipo de modo; y los otros dos implementan la lógica de inicio o fin de estacionamiento según el modo.
  El rol que cumple al patrón es desligar a la aplicación de la estrategia para generar un estacionamiento dependiendo el modo que se seleccione, y permite que en el futuro se puedan crear nuevos modos sin necesidad de modificación de los existentes.
- **State:** Para el estado del GPS (Apagado, Encendido), tiene dos métodos polimórficos. El primero que según su estado, cambia al otro; y el segundo devuelve si está encendido o apagado.
  Para el estado de movimiento (Walking, Driving), tiene tres métodos. Uno solo devuelve si está en estado driving o no; los otros dos son los que se llaman según la implementación del “movement sensor” para cambiar si corresponde al otro estado.
  El rol que cumple el patrón en la solución es que los cambios de estados del sistema puedan ser agnosticos al modo de

cambiarlos, sí tienen que conocerse entre sí para poder cambiar a otro estado según su lógica interna.

- **Observer:** Para que cualquier Entidad pueda suscribirse y/o desuscribirse de las notificaciones que les manda el Sistema de Estacionamiento Medido (SEM).
  El rol que cumple el patrón es poder implementar una manera de suscripción a eventos, que el sistema notifica a cada uno de sus suscriptores en el momento que ocurre el mismo sin importar quienes son los mismos, o que hacen con el evento.

Publisher: interfaz que permite registrar y des-registrar observadores, así como también notifica los eventos que suceden en el SEM (notificar inicio/fin de estacionamientos). Basándonos en el libro de Gamma et. al., Publisher cumple el rol de *Subject*.

Entidad: interfaz de actualización de los objetos que serán notificados de los cambios del SEM. Basándonos en el libro de Gamma et. al, Entidad cumple el rol de *Observer*.

SEM: es el encargado de almacenar las actualizaciones que les interesa a los objetos observadores, además de notificar cuando sucede algún evento (inicio/fin de estacionamiento). Basándonos en el libro de Gamma et. al., SEM es el *ConcreteSubject*.