

DOCUMENTACIÓN TP FINAL POO2

Estudiantes:

Nombre: Lucio Jara, Luciojara8gmail.com

Nombre: Ivan Budzaj, budzajivan@gmail.com

Nombre: Agustin Campos,

Patrones utilizados:

- **Patrón Strategy:** Decidimos utilizar un patrón de Strategy en la Clase ModoAplicacion, ya que el usuario debe tener la flexibilidad de intercambiar el modo en que se encuentra la aplicación del celular, tanto sea como Manual o Automático.

Partes:

Context : AppEstacionamiento

Strategy: ModoAplicacion

ConcreteStrategy: ModoManual/ModoAutomatico

- **Patrón State:** Decidimos utilizar el patrón de State en la clase EstadoMovimiento, debido a que implementamos en el AppEstacionamiento un estado de movimiento, que son Walking y Driving.

Partes:

Context: AppEstacionamiento

State: EstadoMovimiento

ConcreteState: Walking/Driving

Decisiones de diseño:

- Se decidió no implementar un Observer para la clase Suscriptores, debido a que no existen clases concretas que reciban dicha notificación y , además, decidimos tener la interfaz ISuscriptor como colaborador interno, ya que de realizar una clase más que sea intermediaria de ambas, lo único que se realizaría sería una clase que reciba el mismo protocolo. Por lo que llegamos a la conclusión que tener una clase que sea “pasamanos” de protocolo no es viable.
- Se decidió no implementar un PATRÓN STATE para conocer el estado de vigencia de un estacionamiento, se decidió que sea un booleano y este se modifica su estado mediante el inicio o finalización de dicho estacionamiento en otras clases.

- Se decidió no implementar una clase Usuario debido a que se debatió y verificamos que solamente contaría con atributos y su protocolo se basaría en derivar las instrucciones. Por ende decidimos que no es necesario.
- Se decidió no implementar una clase Celular, sin embargo este está representado como AppEstacionamiento, que cumple la función de tener el número de celular.
- Se decidió no realizar una clase para el celular del Inspector, ya que todo el protocolo recae en el mismo Inspector y no es necesario.
- Se decidió realizar una clase Compras de la cual desprenden dos tipos de registros que se generan en el PuntoDeVenta, lo cual se guarda con dicha Clase una lista en el SEM para generalizar y poseer todas las compras en una sola lista.
- Se decidió realizar una clase abstracta Estacionamiento de la cual desprenda las subclases EstacionamientoCompraPuntual y EstacionamientoViaApp, lo cual ambos realizaron dichos estacionamientos de manera particular.
- La clase AppEstacionamiento, además de funcionar como celular, este es el encargado de seleccionar el modo de uso de la aplicación, así como si quiere recibir o no asistencia al usuario.
- AppEstacionamiento también hace uso de la interfaz provista de el GPS, donde toma los mensajes y los redefine a conveniencia para lograr el uso de la interfaz.