

MARIA CAMILA PRECIADO ROJAS

ROBINSON FERNANDO RODRIGUEZ APONTE



UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA

FACULTAD DE INGENIERÍA

ESCUELA DE SISTEMAS Y COMPUTACION

TUNJA, BOYACÁ

2017

**FACADE**

Es un patrón de diseño que nos permite simplificar el interface de comunicación entre varios objetos de tal forma que para un objeto A sea más sencillo interactuar con los demás permitiendo acceder a un subsistema de forma más sencilla y proporciona una interfaz unificada a otro conjunto de interfaces que componen el sistema.

VENTAJAS:

La principal ventaja del patrón fachada consiste en que para modificar las clases de los subsistemas, sólo hay que realizar cambios en la interfaz/fachada, y los clientes pueden permanecer ajenos a ello. Además, y como se mencionó anteriormente, los clientes no necesitan conocer las clases que hay tras dicha interfaz.

DESVENTAJAS:

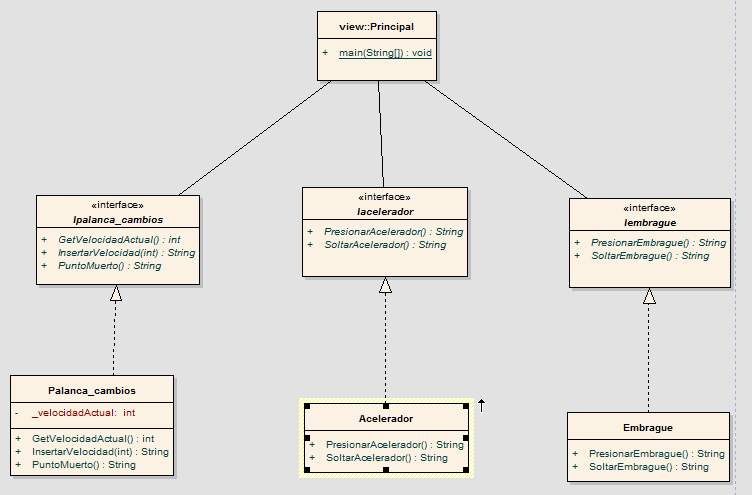
Si se considera el caso de que varios clientes necesiten acceder a subconjuntos diferentes de la funcionalidad que provee el sistema, podrían acabar usando sólo una pequeña parte de la fachada, por lo que sería conveniente utilizar varias fachadas más específicas en lugar de una única global.

**Ejemplo:** se plantea un caso en el que se necesita la comunicación de interfaces con clases concretas las cuales usan algunas características del tipo interfaz, para el caso de arrancar un vehículo declaramos tres interface como Ipalanca\_cambios, Iacelerador y Iembrague; de las cuales se heredan los métodos para las clases concretas: palanca de cambios, embrague y acelerador; por lo cual cuando el conductor vaya a hacer la simulación se comunicara con la interfaz de cada tipo, por medio de un objeto de la clase y hace más fácil la comunicación del sistema además que el usuario no se conecta directamente con las clases concretas sino con las interfaces, en el caso que se necesite más clase concretas la interfaz permite unificar en un solo punto la comunicación y hace más eficiente el acceso.

El proceso de comunicación seria el siguiente:

1. Llama a la Interfaz Iacelerador y al método soltarAcelerador() en la clase acelerador.
2. Llama a la inerfaz Iembrague y al método presionarEmbrague() en la clase embrague.
3. Llama a la interfaz Ipalanca\_cambios y al método putoMuerto() en la clase palanca\_cambios.
4. Llama a la interfaz Ipalanca\_cambios y al método insertsarVelocidad() en la clase palanca\_cambios.
5. Llama a la inerfaz Iembrague y al método soltarEmbrague() en la clase embrague;
6. Llama a la Interfaz Iacelerador y al método presionarAcelerador() en la clase acelerador.

**Diagrama de clases:**



El código del ejemplo se encuentra en el archivo **código ejemplo facade.rar.**