# INFORME LIBRERÍA “YURCAR” – MARKEL SETIEN Y UNAI DURAN

El objetivo de esta tarea consiste en crear una librería funcional de “*Pypi*” de acuerdo a todos los conocimientos obtenidos durante estos cuatro años académicos de programación, especialmente aquellos aprendidos durante este último curso, como, por ejemplo, programación orientada a objetos y tratamiento de errores y excepciones.

En este caso, se propone una librería dirigida a la búsqueda y recomendación de coches satisfaciendo unos requisitos establecidos por el usuario. Además, se permite la opción de realizar una reserva del vehículo elegido.

Como ya se ha mencionado, la programación orientada a objetos era un requisito fundamental a la hora de desarrollar la librería por lo que sobre ésta se cimenta la propuesta ofrecida.

De forma adicional, y para poder realizar la recomendación de coches, se ha hecho uso de una API utilizada para realizar búsqueda de vehículos de acuerdo a unos parámetros de entrada (<https://api-ninjas.com/>).

A continuación, se explica paso a paso y punto por punto el funcionamiento de la aplicación, así como su configuración y estructura primitiva. Con el fin de facilitar la comprensión, se informa que los paréntesis indicarán el tipo de dato en caso de los atributos y el tipo de *input* y *output* (*input* -> *output*) en caso de los métodos.

### Clase “Cliente”

En primer lugar, se genera una clase llamada “Cliente”, que representa un cliente con información básica. Esta clase está formada por los siguientes atributos (no precisan explicación):

* *edad (int)*
* *mayoria*\_*edad (bool)*
* *nombre (str)*
* *residencia (str)*
* *permiso (list)*

Y la constituyen los siguientes tres métodos:

* *comprobacion\_edad (-> bool):* Comprueba si el cliente es mayor de edad, devuelve e inserta *True* en el atributo *mayoria\_edad* en caso de que lo sea, *False* en caso contrario.
* *añadir*\_*permiso (str ->)*: Añade un permiso a la lista (atributo) de permisos de conducción del cliente.
* *show*\_*ciente*: Muestra los datos del cliente en consola.

### Clase “ClienteExt”

Esta segunda clase representa una versión extendida de cliente con versión adicional.

La forman los siguientes atributos:

* *cliente (Cliente)*: Clase *Cliente* heredada.
* *vigencia*\_*permiso (str)*: La vigencia del permiso de conducción del cliente.
* *necesidades*\_*especiales (bool)*: Indica si el cliente tiene necesidades especiales a considerar a la hora de conducir y realizar la reserva de un vehículo.
* *compromiso*\_*medioambiental (str)*: El nivel del compromiso mediambiental del cliente.
* *nivel*\_*cliente (str)*: Nivel del cliente.

Y los siguientes métodos:

* *es*\_*vigente (-> bool)*: Comprueba si el permiso del cliente está vigente (haciendo uso de la fecha actual y el atributo *vigencia\_permiso*).
* *añadir\_necesidad\_especial*: Añade necesidades especiales al cliente. Establece atributo *necesidades\_especiales* a *True*.
* *show\_clientext*: Muestra los datos del cliente extendido en consola.

### Clase “Coche”

Esta clase representa un coche deseado por el usuario. Es decir, aquí se guardan las características deseadas por el cliente para considerar en la recomendación de su coche.

Está formada por los siguientes atributos:

* *make (str)*: La marca del coche.
* *Model (str)*: El modelo del coche.
* *fuel*\_*type (str)*: El tipo de combustible del coche (gasolina, diesel).
* *Year (str)*: El año del coche.
* *Transmission (str)*: El tipo de transmisión del coche.
* *Cylinders (str)*: El número de cilindros del motor.

Y los siguientes métodos:

* *buscar\_coche (list ->)*: Realiza una búsqueda de coches en la API de acuerdo a la información especificada. En este método se incorpora un *try…except* que comprueba la respuesta devuelta por la API.
* *mostrar\_inform\_busquda\_coche*: Muestra los resultados de la búsqueda en forma de tabla.

### Clase “Reserva”

Para finalizar, se define la clase *Reserva* que representa una reserva de un coche realizada por un cliente, de ahí el hecho de que herede información de las clases *ClienteExt* y *Coche*.

Está formada por los siguientes atributos:

* *num\_resultado\_coche\_reservado (int)*: El número de resultado de la búsqueda de coche a reservar.

Y los siguientes métodos:

* *realizar\_reserva (int ->)*: Realiza una reserva de un coche en función del número de resultado de búsqueda. En este método se realizan dos *try…except*: El primero comprueba que el dato introducido sea un entero válido; el segundo, en cambio, comprueba que el número de resultado insertado realmente exista entre los resultados de búsqueda ofrecidos. Por otro lado, y para poder realizar la reserva, se comprueba que el cliente disponga de permiso de conducción tipo *B*, que sea mayor de edad y que su permiso de conducción esté vigente.

### Funcionamiento cronológico esperado de la librería

En primer lugar, se espera la creación de un objeto *Cliente* con la información de la clase requerida. Después, se espera la creación del objeto *ClienteExt* en el que ultimar detalles necesarios sobre el objeto previamente creado.

Para continuar, el cliente deberá crear un objeto *Coche* en el que especificará los atributos deseados para su búsqueda. Después, éste llevará a cabo la búsqueda y la selección del coche deseado haciendo uso de los métodos relativos a esta clase.

Para finalizar, y en caso de que el cliente lo decida, realizará la reserva de uno de los vehículos previamente mostrados mediante la creación del objeto *Reserva* y su método correspondiente.