

MODULO PROFESIONAL:	PROGRAMACIÓN
CICLO FORMATIVO:	DESARROLLO DE APLICACIONES WEB

P02R - Examen Python 2do trimestre / Fecha: 05-04-2019 / Duración: 120 minutos

Nombre y Apellidos: _____

1. Nos solicitan un programa para simular la apertura de cajas del juego Brawl Star. Características:
 - Se suministrará un archivo denominado `brawlers` con los distintos personajes del juego organizados en 2 líneas cada uno. La primera es el tipo de brawler y la segunda el nombre. Ejemplo:
`especial`
`barley`
`superespecial`
`rico`
 - Se deben cargar los brawlers en memoria (preferentemente como una lista de diccionarios).
 - Se solicitará el nombre del jugador. Cada jugador tendrá un archivo con su nombre y la misma estructura que `brawlers` con los personajes de los que ya disponga. Si ese archivo no existe se indicará que es un jugador nuevo y comenzará sin personajes.
 - Se mostrarán los personajes que tenga el jugador y se pedirá introducir un texto.
 - Si el usuario escribe **fin** se generará el archivo del jugador con los personajes de los que disponga.
 - Si el usuario escribe **simular** seguido de un número se debe simular la apertura de cajas durante los días que indique el número.
 - Cada día se abrirán dos *cajas brawler*. Cada *caja brawler* puede traer un brawler según las siguientes probabilidades: legendario : 0.0624 %, mitico : 0.2496 %, epico : 0.5472 %, superespecial : 1.2096 % y especial : 2.7312 %
 - Cada 2,5 días se abrirá una *caja grande*. Cada *caja grande* equivale a 3 *cajas brawl*.
 - Cuando el jugador reciba un brawler de un tipo determinado, debe ser uno de los que no posea. Si ya posee todos los brawlers de un tipo no recibirá ninguno más de ese tipo.

Calificación: 5 puntos (Cada apartado será calificado en múltiplos de 0,25)

- Carga archivos `brawlers` y jugador: **1 punto**
- Estructura de datos para almacenar brawlers: **1 punto**
- Desarrollo del juego: **1,5 puntos**
- Escritura de resultados en archivo: **1 punto**
- Programa sin errores: **0,5 puntos**

NOTAS:

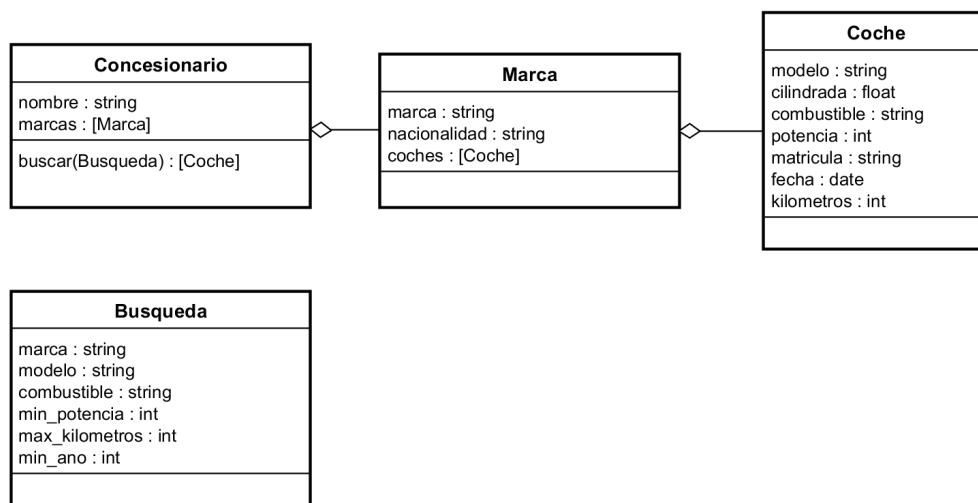
- El archivo `brawlers` será suministrado por el profesor.
- Se usará la librería `random`.
- La sentencia `random.uniform(0.00001, 100.0)` proporciona un número float entre 0,00001 y 100.
- Es interesante crear una lista desordenada con los brawlers que no tiene el jugador para decidir el nuevo brawler.

MODULO PROFESIONAL:	PROGRAMACIÓN
CICLO FORMATIVO:	DESARROLLO DE APLICACIONES WEB

P02R - Examen Python 2do trimestre / Fecha: 05-04-2019 / Duración: 120 minutos

Nombre y Apellidos: _____

2. Nuestra empresa ha recibido el encargo de desarrollar un programa para controlar las búsquedas de coches en un concesionario multimarca. Nuestras analistas han decidido implementarlo usando programación orientada a objetos y han diseñado el siguiente diagrama de clases:



- Crear las clases del diagrama en un solo archivo denominado concesionario.py.
- Aunque no se indiquen en el diagrama, cada clase debe tener implementados los métodos `__init__` y `__str__`.
- Una vez implementadas las clases se deben comprobar con el código de prueba que se suministrará.
- Tras esta comprobación se deberá crear un programa denominado consulta.py que haga uso de las clases para permitir la consulta.
- Este programa debe solicitar los parámetros definidos en la clase Busqueda permitiendo dejar cualquiera de ellos en blanco.
- Si al solicitar la marca se escribe fin terminará el programa.

```
Concesionario: Rafael Multimarcas
¿Marca? Volvo
¿Modelo?
¿Combustible?
¿Potencia mínima?
¿Máximo de kilómetros?
¿Año de matriculación mínimo?
Resultados:
Volvo V40 Diesel 2.0 120CV 2019 (0 km)
Volvo S90 Diesel 2.0 150CV 2017 (23435 km)
```

Calificación: 5 puntos (Cada apartado será calificado en múltiplos de 0,25)

- Implementación de las clases: **2 puntos**
- Ejecución correcta del código de prueba: **1 punto**
- Sistema de consulta: **1,5 puntos**
- Programa sin errores: **0,5 puntos**