**Enunciado**

Un amigo nuestro tiene un taller y quiere hacer una aplicación para llevar el control de los coches que han pasado por el. Cada coche se caracteriza por un número de bastidor, por una marca, por el año en el que fue fabricado, por el número de kilometros que tenía la última vez que entró en el taller y por el número de veces que ha entrado en el taller en cada uno de los últimos 9 años (que son los que lleva abierto el negocio). El número de bastidor es una serie de 17 caracteres. Cada caracter puede ser un número del 0 al 9 o una letra (sin contar la letra "ñ"). La marca de un coche puede ser una de entre las siguientes: AUDI, BMW, CHEVROLET, CITROEN, FIAT, FORD, HONDA, HYUNDAI, JAGUAR, JEEP, KIA, MAZDA, MERCEDES, MITSUBISHI, NISSAN, OPEL, PEUGEOT, RENAULT, SAAB, SEAT, SKODA, SUZUKI, TOYOTA, VOLKSWAGEN, VOLVO. El número de kilometros es un número entero a partir de 1. Para realizar el programa vamos a emplear dos clases: la clase `Taller` y la clase `Coche`. Se pueden crear más clases en caso de considerarlo oportuno.

[](https://github.com/miguelbayon/miguelbayon.github.io/blob/master/images/taller.png)

**Pasos a completar**

1. Crea un repositorio local de git en tu equipo.
2. Asocia dicho repositorio con el repositorio remoto adecuado conforme las instrucciones del anexo II.
3. 20 puntos Codifica el código necesario para que se pueda cºrear un objeto de tipo Taller desde la interfaz de BlueJ. Ese objeto taller debe construir y almacenar un número de coches determinado por un parámetro del constructor. Los coches deben tener sus datos generados de forma aleatoria.

Haz un commit en la rama master y realiza un push.

1. 20 puntos Codifica el código necesario para que, interactuando gráficamente con el objeto de la clase Tallercreado, podamos ver un listado de todos los coches que han pasado por el taller en el que se visualicen en una única línea por coche todos los datos de un vehículo.

Haz un commit en la rama master y realiza un push.

1. 15 puntos Codifica el código necesario para que, interactuando gráficamente con el objeto de la clase Tallercreado, podamos introducir una marca y obtener el número de coches de esa marca que han pasado por el taller.

Haz un commit en la rama master y realiza un push.

1. 15 puntos Codifica el código necesario para que, interactuando gráficamente con el objeto de la clase Tallercreado, podamos introducir un número de bastidor y ver por pantalla si el coche asociado ha pasado o no por el taller. En caso de que haya pasado, se deben visualizar sus datos completos.

Haz un commit en la rama master y realiza un push.

1. 15 puntos Codifica el código necesario para que, interactuando gráficamente con el objeto de la clase Tallercreado, podamos saber si todos los coches de una marca determinada (indicada por el usuario) que han pasado por el taller fueron fabricados antes de un año determinado (indicado también por el usuario como parámetro).

Haz un commit en la rama master y realiza un push.

1. 15 puntos Codifica el código necesario para que, interactuando gráficamente con el objeto de la clase Tallercreado, podamos visualizar los coches ordenados alfabéticamente por marca y dentro de los coches de una marca ordenados por el número de kiloémtros (antes los que menos kilómetros tienen).

Haz un commit en la rama master y realiza un push.

**Observaciones:**

* Es obligatorio para que se corrija el examen que:
  + El código compile sin errores sobre Java 7.
  + Se utilice git de forma correcta para el control de versiones
* No se permite:
  + El uso de la sentencia break excepto en claúsulas switch.
  + Utilizar más de un return en los métodos.
* Se valora negativamente:
  + La redundancia de código.
  + La incorrección del código.
  + La utilización de código ineficiente.
  + La indentación incorrecta.
  + La utilización de números literales dentro del código en vez de constantes.
  + La incorrección en el diseño.

**Importante**

* No se permite el uso de ninguna forma de comunicación (verbal, escrita, por Internet) entre alumnos o con cualquier otra persona.
* Tampoco se permite ningún mecanismo que permita transmitir el código fuente al resto de alumnos, incluido el simple acceso a repositorios de otros alumnos.

En caso de detectar alguna de estas prácticas, el alumno suspende directamente la convocatoria actual.