



# Universidad de Alcalá

Escuela Politécnica Superior

Universidad de Alcalá

**PECL6**

## **Arquitectura y Diseño de Sistemas Web y C/S**

### ***Implementación del Patrón MVC***

Laboratorio Jueves 10:00 – 12:00

Grado en Ingeniería Informática & Ingeniería en Sistemas de  
Información – Curso 2019/2020

Marcos Barranquero Fernández – 51129104N

Daniel Manzano Estébanez – 03220212M

## ÍNDICE

Análisis del problema .....	3
Implementación.....	3
Modelo de datos.....	3
Implementación de la web .....	4
Crear circuito y crear coche .....	4
Calculador de ahorro .....	6
Diagrama de clases.....	7
Manual de usuario .....	7
Inicio .....	7
Dar de alta un circuito.....	8
Dar de alta un coche .....	10
Calcular ahorro.....	10

## ANÁLISIS DEL PROBLEMA

Se quiere desarrollar una aplicación donde se puedan registrar circuitos y coches, para posteriormente calcular la ganancia de potencia que tendría un determinado coche en un determinado circuito en función del número de curvas.

Para ello se parte de la base del modelo de datos: se crearán dos tablas en una base de datos que representen a los circuitos y a los coches.

## IMPLEMENTACIÓN

### MODELO DE DATOS

Este es el modelo de datos creado. Las tablas no están relacionadas entre sí, ya que son totalmente independientes. Los coches pueden correr en cualquier circuito, y en cada circuito puede correr cualquier coche. Solo se utilizarán unos determinados atributos para realizar el cálculo.



Con este diagrama hecho, se ha generado el código SQL correspondiente y se ha adaptado a la sintaxis de Apache Derby.

```
-- DROP TABLE IF EXISTS "CIRCUITO" CASCADE;  
CREATE TABLE "CIRCUITO"(  
    nombre varchar(20) NOT NULL,  
    ciudad varchar(20) NOT NULL,  
    pais varchar(20) NOT NULL,  
    vueltas integer NOT NULL,  
    longitud integer NOT NULL,  
    curvas integer NOT NULL,  
    CONSTRAINT "CIRCUITO_pk" PRIMARY KEY (nombre)  
);  
  
-- DROP TABLE IF EXISTS "COCHE" CASCADE;  
CREATE TABLE "COCHE"(  
    nombre varchar(20) NOT NULL,  
    kwcurva integer NOT NULL,  
    CONSTRAINT "COCHE_pk" PRIMARY KEY (nombre)  
);
```

Este código ha sido ejecutado sobre una base de datos llamada "Practica6" que no tiene ni usuario ni contraseña, y que se encuentra adjuntada con los datos de esta práctica. Se puede copiar en la siguiente ruta para hacer que funcione sin tener que configurar nada.

`%AppData%\Roaming\NetBeans\Derby`

## IMPLEMENTACIÓN DE LA WEB

La estructura de la web se ha llevado a cabo de la siguiente manera. Comienza en index.html, desde donde se puede acceder a las páginas de dar de alta circuitos y coches y a la calculadora de ahorro. Todas las páginas hacen uso de la librería Bootstrap para mejorar la presentación.

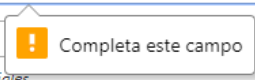
### CREAR CIRCUITO Y CREAR COCHE

Son dos páginas diferentes, pero con un funcionamiento muy parecido. Cuentan con un formulario donde se pueden especificar los datos deseados. Todos los campos tienen verificación, es decir, no se pueden dejar vacíos. De lo contrario, aparecerá lo siguiente.

Nombre del coche:

Ganancia de potencia (entre 4 y 10 kW):

Longitud máxima del texto: 20 caracteres. Por compatibilidad, el nombre no puede contener caracteres especiales.



Los números, pese a ser inputs de tipo texto, tienen expresiones regulares que comprueban que el valor introducido sea un número y que esté en el rango.

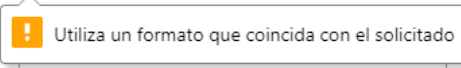
```
<input type="text" pattern="[4-9]|10" size="40" name="txtPotencia">
```

País del circuito:

Numero de vueltas del circuito (entre 40 y 80):

Longitud de la vuelta del circuito en metros (entre 3 y 9 km):

Numero de curvas del circuito (entre 6 y 20):



Por compatibilidad con la base de datos, el nombre no puede tener caracteres especiales (tildes o ñ). La longitud máxima de los textos es de 20 caracteres.

Una vez rellenado todo y enviado, los datos se redirigen a un Servlet, CargarCircuito.java o CargarCoche.java, según sea el caso. Estos Servlets insertan los datos, verificándolos antes. Los procesos de verificación también son similares. En el caso de CargarCoche.java, el proceso es el siguiente.

1. Se obtiene el nombre y se comprueba si existe en la base de datos, ya que es la clave primaria.

```
// Se comprueba primero si no existe el coche
String nombre = req.getParameter("txtNombre");

if (bd.existeCoche(nombre)) {
    res.sendRedirect(res.encodeRedirectURL("errorCoche.html"));
}
```

Mediante un método contenido en ModeloDatos.java, la clase que se encarga de operar sobre la base de datos se comprueba si existe. De ser así, se redirige a la siguiente página de error.

## No se ha podido registrar ese coche

Ya figura en la base de datos uno con el mismo nombre.

Si no existe, se pasa a las siguientes comprobaciones, que se hacen por la especificación del enunciado de que deben realizarse en el servidor.

2. Se obtienen los parámetros restantes y se comprueba que no estén vacíos.

```
} else {
    // Si no existe, se pasa a comprobar el resto de datos
    String kwcurvaTxt = req.getParameter("txtPotencia");

    // Se comprueba que no estén vacíos
    if (nombre.isEmpty() || kwcurvaTxt.isEmpty()) {
        System.out.println("Algún campo está vacío.");
        res.sendRedirect(res.encodeRedirectURL("datosMalCoche.html"));
    }
}
```

Si falla, se muestra este error, que será el mismo para los próximos fallos.

## No se ha podido registrar ese coche

Algún dato ha sido introducido mal.

En la consola de salida de Glassfish se puede ver detalladamente a qué se debe el error.

3. Tras esto, se comprueba que la longitud del texto es apta para la base de datos.

```
} else if (nombre.length() > 20) {
    // Si el nombre ocupa más de lo permitido
    System.out.println("La longitud de algún texto excede el
                                                                máximo permitido.");
    res.sendRedirect(res.encodeRedirectURL("datosMalCoche.html"));
}
```

4. Se intenta convertir los números de String a Integer. Si se pasara algo diferente, lanzará una excepción.

```

    } else {
        try {
            // Se comprueba que kwcurva sea un número
            Integer kwcurva = Integer.parseInt(kwcurvaTxt);

            // [...] Aquí están las últimas comprobaciones

        } catch (Exception e) {
            System.out.println("No se ha introducido un número en
                               algún campo numérico.");
            res.sendRedirect(res.encodeRedirectURL(
                           "datosMalCoche.html "));
        }
    }

```

5. Por último, se comprueba el valor de los números para asegurarse que no están fuera de rango.

```

        if (kwcurva < 4 || kwcurva > 10){
            // Y que esté en el rango correcto
            System.out.println("Algún número está fuera de rango."
);
            res.sendRedirect(res.encodeRedirectURL("datosMalCoche.
html"));
        }

```

6. Si ha pasado todos los filtros, se añade a la base de datos.

```

    } else {
        // Si ha pasado todos los filtros, se inserta
        bd.insertarCoche(nombre, kwcurva);
        res.sendRedirect(res.encodeRedirectURL(
                           "exitoInsertado.html"));
    }

```

Este es el mensaje de confirmación.

**Se ha insertado con éxito**

[Volver a inicio](#)

---

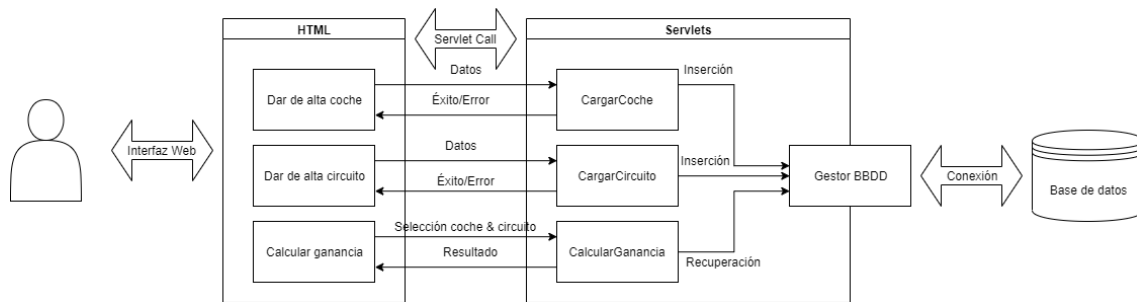
## CALCULADOR DE AHORRO

Se trata de una página que cuenta con dos combos boxes, donde aparecerán los nombres de los coches y circuitos que se encuentren en la base de datos. Al seleccionar una combinación y pulsar "Calcular", ejecutaremos el Servlet `CalcularGanancia.java`. Este obtiene los elementos seleccionados en el formulario y ejecuta dos consultas sobre la base de datos.

Hecho esto, guarda en variables los datos necesarios: ahorro del coche y el número de curvas y de vueltas del circuito. Con ellos realiza el cálculo y genera un código para mostrar el resultado.

## DIAGRAMA DE CLASES

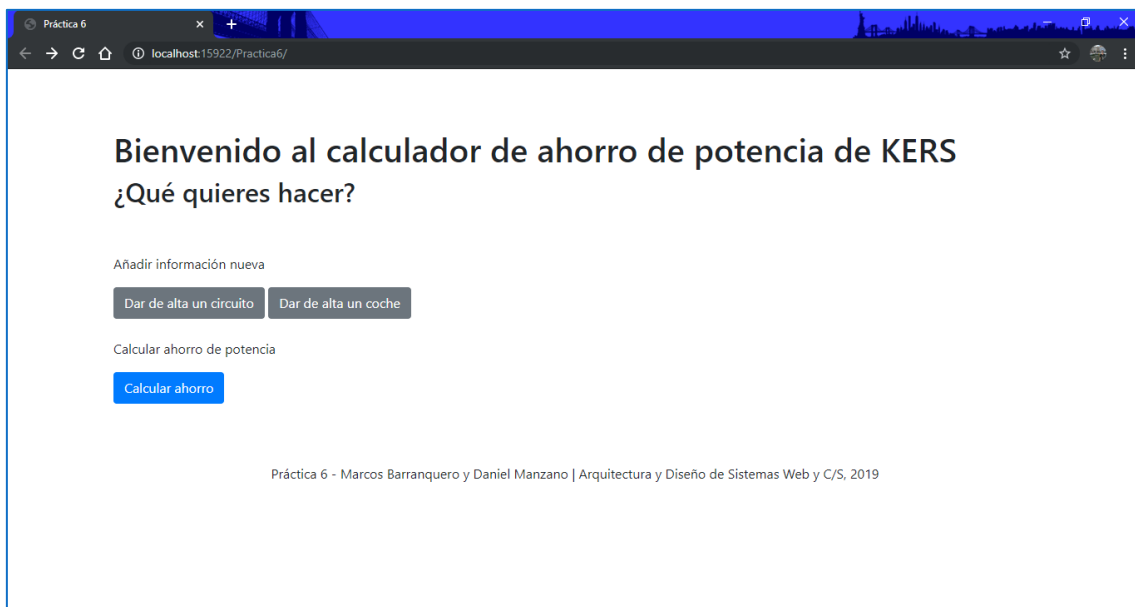
El siguiente diagrama representa la estructura de la aplicación:



Podemos ver que el cliente utiliza la interfaz web de archivos HTML. Para interactuar con la base de datos, se llama a distintos servlets que validan los datos y realizan los cálculos para la ganancia. Estos servlets se comunican con la base de datos con la clase gestora de la base de datos, que es la encargada de conectarse y llevar a cabo las consultas sobre la BBDD.

## MANUAL DE USUARIO

### INICIO

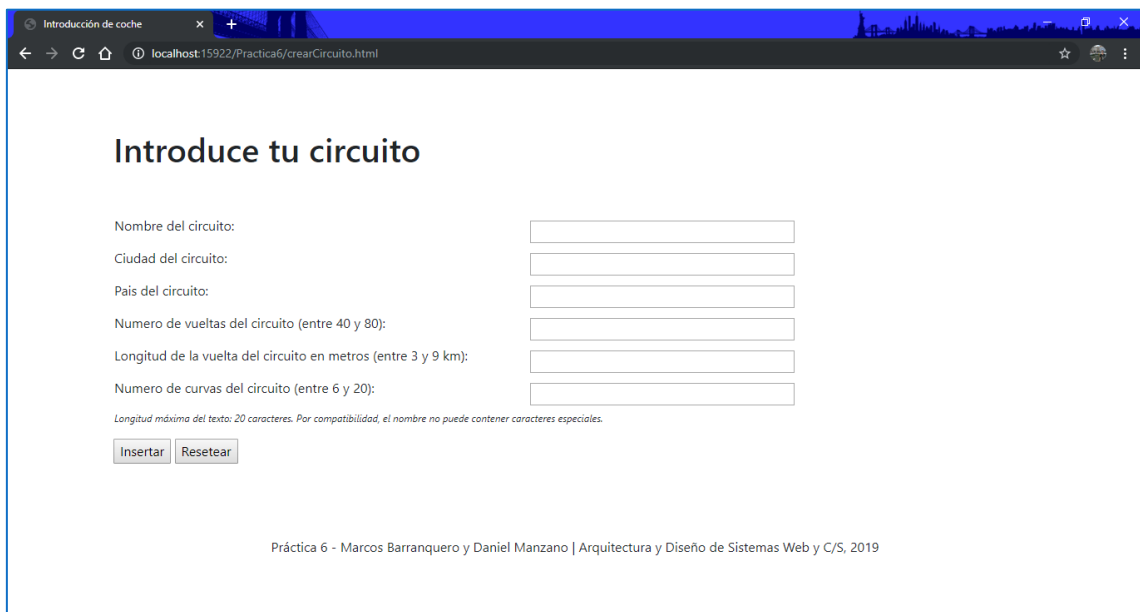


Bienvenido al calculador de ahorro de potencia de KERS. Con esta herramienta podrás registrar circuitos y coches, para calcular el ahorro de potencia que tendrán los coches al correr en un circuito.

En la página de bienvenida encontrarás tres funciones:

[Dar de alta un circuito](#) - [Dar de alta un coche](#) - [Calcular ahorro](#)

## DAR DE ALTA UN CIRCUITO



The screenshot shows a web browser window with the address bar displaying 'localhost:15922/Practica6/crearCircuito.html'. The page title is 'Introduce tu circuito'. The form contains the following fields and labels:

- Nombre del circuito: [text input]
- Ciudad del circuito: [text input]
- País del circuito: [text input]
- Numero de vueltas del circuito (entre 40 y 80): [text input]
- Longitud de la vuelta del circuito en metros (entre 3 y 9 km): [text input]
- Numero de curvas del circuito (entre 6 y 20): [text input]

Below the inputs, there is a small note: 'Longitud máxima del texto: 20 caracteres. Por compatibilidad, el nombre no puede contener caracteres especiales.' At the bottom of the form are two buttons: 'Insertar' and 'Resetear'.

At the very bottom of the page, there is a footer: 'Práctica 6 - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2019'.

En esta página puedes registrar tus circuitos. Solo tienes que introducir estos datos:

- **Nombre de circuito:**  
Aquí puedes escribir letras y números, aunque los caracteres especiales como las tildes o la ñ no están soportados, por compatibilidad con la base de datos. Longitud máxima del texto: 20 caracteres.
- **Ciudad del circuito y País del circuito:**  
Aquí sí puedes poner tildes o la ñ. Máximo 20 caracteres.
- **Número de vueltas del circuito:**  
Debe introducirse un número entero entre 40 y 80.
- **Longitud de la vuelta del circuito:**  
Entre 3 y 9 kilómetros, y se especifica en metros. Es decir, hay que introducir un número entero entre 3000 y 9000.
- **Número de curvas:**  
Admite números enteros comprendidos entre 6 y 20.

Una vez introducidos los datos, puedes insertarlos en la base de datos con el botón "Insertar" (o vaciar los campos con "Resetear"). Si están bien, obtendrás la siguiente ventana, donde podrás volver al inicio.





Si introduces algún dato mal, tu navegador te avisará. Si no, nuestro servidor te avisará del error.

Numero de vueltas del circuito (entre 40 y 80):

Longitud de la vuelta del circuito en metros (entre 3 y 9 km):

Numero de curvas del circuito (entre 6 y 20):

Longitud máxima del texto: 20 caracteres. Por compatibilidad, el nombre no puede contener caracteres especiales.

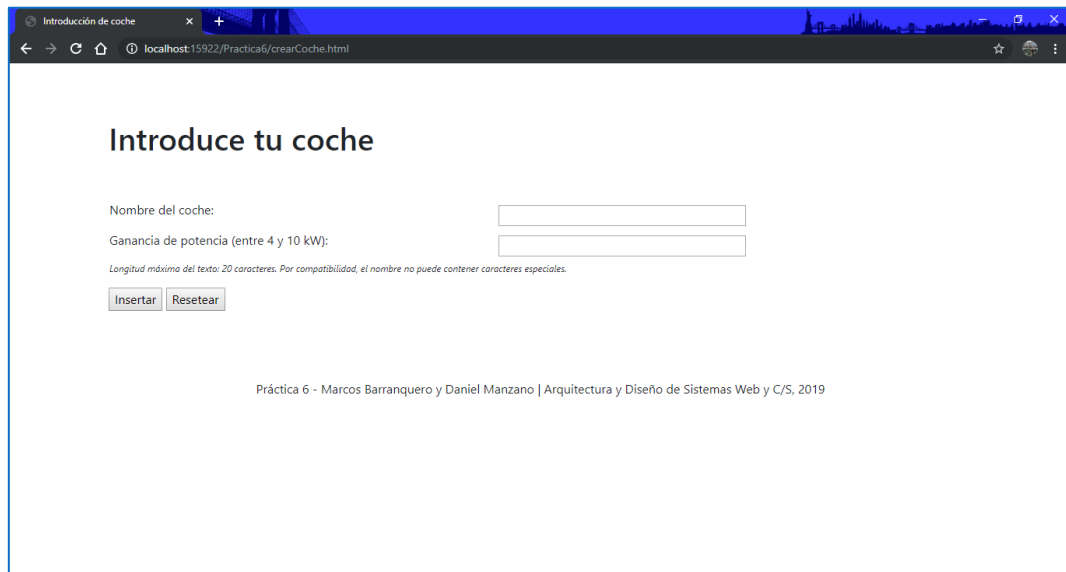
Práctica 6 - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2019



Recuerda: no puedes registrar un circuito cuyo nombre ya exista.



## DAR DE ALTA UN COCHE



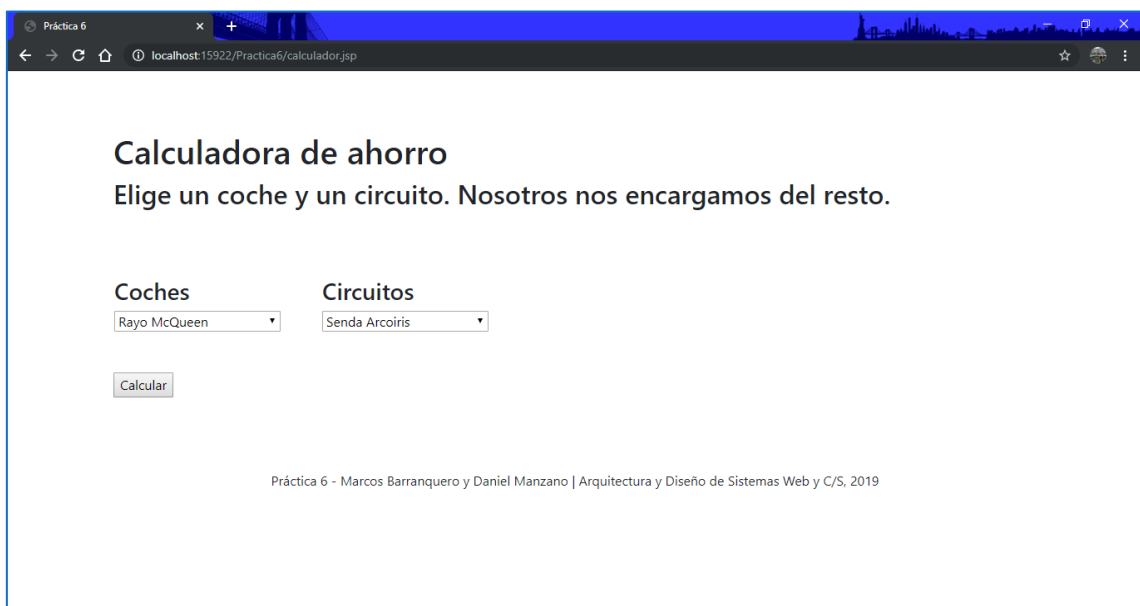
The screenshot shows a web browser window with the address bar displaying 'localhost:15922/Practica6/crearCoche.html'. The page has a title 'Introducción de coche' in the browser tab. The main heading is 'Introduce tu coche'. Below it, there are two input fields: 'Nombre del coche:' and 'Ganancia de potencia (entre 4 y 10 kW):'. A small note below the second field states: 'Longitud máxima del texto: 20 caracteres. Por compatibilidad, el nombre no puede contener caracteres especiales.' At the bottom of the form are two buttons: 'Insertar' and 'Resetear'. At the very bottom of the page, there is a footer: 'Práctica 6 - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2019'.

Aquí puedes dar de alta tus coches. Es muy parecido a la página de registrar circuitos, pero sólo te pedimos dos cosas:

- **Nombre del coche:**  
Aquí puedes escribir letras y números, aunque los caracteres especiales como las tildes o la ñ no están soportados, por compatibilidad con la base de datos. Longitud máxima del texto: 20 caracteres.
- **Ganancia de potencia:**  
¿Cuántos kW ahorra el coche en cada curva? Puedes poner un número entero entre 4 y 10.

Los mensajes de error y las confirmaciones son similares a los del circuito.

## CALCULAR AHORRO



The screenshot shows a web browser window with the address bar displaying 'localhost:15922/Practica6/calculador.jsp'. The page has a title 'Práctica 6' in the browser tab. The main heading is 'Calculadora de ahorro'. Below it, there is a sub-heading: 'Elige un coche y un circuito. Nosotros nos encargamos del resto.' There are two dropdown menus: 'Coches' with 'Rayo McQueen' selected, and 'Circuitos' with 'Senda Arcoiris' selected. Below these is a 'Calcular' button. At the very bottom of the page, there is a footer: 'Práctica 6 - Marcos Barranquero y Daniel Manzano | Arquitectura y Diseño de Sistemas Web y C/S, 2019'.

Finalmente tenemos la funcionalidad estrella. Solo hay que elegir un coche y un circuito en cada lista desplegable, y pulsar "Calcular". Esto te mostrará el resultado, junto a los datos empleados en la operación.

