



Centro Integrado de Formación Profesional
AVILÉS
Principado de Asturias

ACTIVIDAD INDIVIDUAL SEGUNDA EVALUACIÓN

DISEÑO WEB EN ENTORNO CLIENTE

2º CURSO

C.F.G.S. DESARROLLO DE APLICACIONES WEB (DISTANCIA)

1. UNIDADES QUE SE TRABAJAN EN ESTA ACTIVIDAD

- [UT5] Bibliotecas y Frameworks: React
- [UT6] Componentes y objetos predefinidos
- [UT7] Interacción con el usuario: eventos y formularios
- [UT8] Comunicación asíncrona

2. INTRODUCCIÓN

La siguiente actividad se basa en una aplicación web de realizar de un concesionario de venta de vehículos. Se trata de crear una aplicación en **React** que cumpla con los apartados a continuación.

3. GESTIÓN DE VEHÍCULOS

La aplicación debe ser capaz de gestionar los vehículos del sistema. Se da por hecho que es una aplicación interna que no será publicada al exterior, por lo que no requerirá una autenticación previa. La gestión de vehículos debe cumplir con los siguientes requisitos:

- a) Poder consultar todos los vehículos del concesionario. Se pueden visualizar de forma tabular. Cada vehículo debe contener al menos los siguientes datos:
 - a. Id de vehículo. Es un dato interno que no se muestra al usuario, solo se usa en operaciones de servidor y de obtención de datos.
 - b. Número de chasis.
 - c. Marca del vehículo.
 - d. Modelo del vehículo.
 - e. Color
 - f. Potencia (en CV)
 - g. Fecha de fabricación
- b) Poder dar de alta un vehículo de forma manual.
- c) Poder actualizar un vehículo cualquiera.
- d) Poder eliminar un vehículo.

Para poder hacer estas operaciones, se proporciona un script en PHP (vehículos.php) que implementa las siguientes operaciones CRUD en el servidor:

1. getAllVehiculos: Obtiene todos los vehículos del concesionario (GET)
2. getVehiculoById: Obtiene un vehículo por ID (GET)
3. createVehiculo: Crea un vehículo dado un JSON con sus datos. No debe contener el ID (POST)
4. updateVehiculoById: Actualiza un vehículo dado un JSON con sus datos. Debe contener el ID. (PUT)
5. deleteVehiculoById: Borra un vehículo dado un JSON con su ID. (DELETE)

Por otra parte, debe incluir las siguientes validaciones de los datos introducidos en el formulario:

1. Todos los campos son requeridos.
2. Al dar de alta un coche, el número de chasis no puede existir previamente en el servidor.
3. Al actualizar un coche, si se modifica el número de chasis de un vehículo, hay que comprobar que no se introduce uno ya existente en otro registro.
4. El número de chasis debe contener 8 dígitos. Realizar la validación mediante expresiones regulares.
5. Tanto marca como color solo pueden contener texto, no números.
6. La potencia debe ser mayor que 50CV.
7. La fecha de fabricación no puede ser posterior al día de hoy.
8. Las expresiones regulares de validación deben almacenarse en variables de contexto (React Context).

4. ESTADÍSTICAS DE VEHÍCULOS

Además de consultar los datos de cada vehículo, se podrán visualizar las siguientes estadísticas en relación con todos los vehículos:

1. Valores medios de la potencia en CV.
2. Valor mínimo de la potencia en CV.
3. Valor máximo de la potencia en CV.

Se repetirán las mismas estadísticas filtrando por marca. Por ejemplo, potencia media, mínima y máxima de los vehículos Audi almacenados en el servidor.

Aunque en condiciones normales, estos valores deberían calcularse en el servidor, en este caso se hará exclusivamente en cliente.

5. ENTREGA DE LA PRÁCTICA

La práctica estará completamente alojada en un repositorio de GitHub, aunque también se podrá entregar como un archivo comprimido que contenga la solución. Formará parte de la evaluación el trabajo realizado sobre esta plataforma. La actividad puede estar subida progresivamente al repositorio para poder ver el trabajo realizado durante toda la evaluación.

Importante: Entrega el directorio completo en un archivo comprimido **exceptuando node_modules**.