



Centro Integrado de Formación Profesional
AVILÉS
Principado de Asturias

ACTIVIDAD INDIVIDUAL EJEMPLO

DISEÑO WEB EN ENTORNO CLIENTE

2º CURSO

C.F.G.S. DESARROLLO DE APLICACIONES WEB

1. UNIDADES QUE SE TRABAJAN EN ESTA ACTIVIDAD

[UT2] El lenguaje JavaScript. Sintaxis básica

[UT3] El lenguaje JavaScript. Arrays, funciones y objetos

[UT4] Modelo de objetos de documento (DOM)

2. INTRODUCCIÓN

La empresa de autobuses AVILESA necesita una aplicación que gestione sus líneas de autobuses entre municipios. Para ello, se necesita almacenar tanto origen como destino y el itinerario recorrido junto con el intervalo de tiempo en horas y minutos que se tarda desde la salida.

3. TIPOS DE DATOS

De un modo general, la aplicación manejará los siguientes datos:

- Línea. En este objeto se recogen los datos de la línea. Requerirá la siguiente información:
 - Número de línea. Un número entero a partir de 1 para identificar de forma única la línea.
 - Origen. Localidad de la que sale el autobús.
 - Destino. Localidad al que llega el autobús.
 - Hora inicial de salida. La primera hora del día a la que sale el bus.
 - Intervalo entre buses. Intervalo expresado en HH:MM desde que sale un bus hasta el siguiente.
- Paradas. Recogerá los datos de las paradas que de las que consta el itinerario a realizar. Debe incluir la siguiente información:
 - Número de parada. Un número entero a partir de 1 para identificar de forma única la parada.
 - Número de línea. Línea a la que se refiere el itinerario.
 - Localidad. Localidad por la que para el bus en el itinerario.
 - Intervalo desde la hora de salida. Intervalo expresado en HH:MM desde que sale el bus para llegar a ese destino.

El itinerario debe recoger todas las paradas excepto la salida, incluyendo el destino para saber a qué hora se llega. Por supuesto, el destino es el que debe tener el mayor intervalo desde la salida.

4. FUNCIONALIDAD JAVASCRIPT

Como ya se ha explicado antes, la aplicación será capaz de gestionar una empresa de autobuses. Existirá una página principal desde la que se podrán abrir cada una de las páginas correspondientes a cada apartado: líneas, paradas y detalle de línea.

4.1. GESTIÓN DE LÍNEAS

Se permitirá dar de alta, baja, modificar y consultar líneas de autobús. Se deberá comprobar lo siguiente:

- a) Que no se intenta dar de alta un número de línea ya existente y que este es mayor o igual que 1 no admitiendo decimales.
- b) Que los tipos son adecuados a cada campo.
- c) Que el origen no es igual al destino.
- d) Que la hora de salida tiene un formato correcto.
- e) Que el intervalo tiene un formato correcto.

4.2. GESTIÓN DE PARADAS

La aplicación permitirá dar de alta, baja, modificar y consultar paradas de una línea. Se deberá comprobar lo siguiente:

- a) Que no se intenta dar de alta un número de parada ya existente y que este es mayor o igual que 1 no admitiendo decimales.
- b) Que el número de línea introducido ya existe en la lista de líneas.
- c) Que el intervalo tiene un formato correcto.
- d) Que si se introduce una parada que equivale al destino de una línea, es la que tiene el intervalo mayor (hay que tener en cuenta que es la parada destino).

4.3. DETALLE DE LÍNEA

El sistema permitirá consultar el detalle de una línea. Este contendrá el listado de paradas correspondientes a una línea dado su número y ordenadas por intervalo. Un ejemplo de tabla sería el siguiente:

Num. Parada	Localidad	Intervalo

4.4. PERSISTENCIA DE DATOS

El sistema permitirá almacenar los datos usando LocalStorage. Al arrancar el sistema se pueden cargar una serie de registros de ejemplo si así se desea.

5. RECOMENDACIONES DE DISEÑO

Se valorará positivamente la separación lógica del código JavaScript en ficheros .js dejando que el código HTML contenga solo la parte imprescindible de marcado junto con los scripts mínimos necesarios. También aspectos de diseño (desplegables en datos como el número de líneas, localidades, etc.).