



## ENTREGABLE 2ª EVALUACIÓN

Desarrollo Web entorno cliente  
CFGS DAW

Sergio García Barea  
[sergio.garcia@ceedcv.es](mailto:sergio.garcia@ceedcv.es)  
2018/2019  
Versión:190113.2013


## Licencia




**Reconocimiento - NoComercial - CompartirIgual (by-nc-sa):** No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

## Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:

 Importante

 Atención

 Interesante


## ÍNDICE DE CONTENIDO


1. Aclaraciones previas a los ejercicios.....	3
2. Boletín de ejercicios.....	4


## ENTREGABLE 2ª EVALUACIÓN

### 1. ACLARACIONES PREVIAS A LOS EJERCICIOS

Este entregable se realiza de forma individual.

 Intentar que el programa sea lo más claro y estructurado posible, así como incluir comentarios en sitios clave para facilitar la comprensión del mismo.

 La entrega de estos ejercicios se realizará mediante vuestra cuenta BitBucket.

 Para considerar un ejercicio realizado, no basta únicamente con entregar el código. El alumno ha de SER CAPAZ de defender su ejercicio a petición del profesor y SER CAPAZ de realizar pequeñas modificaciones relacionadas con el mismo, con el fin de demostrar la adquisición de conocimiento y evitar cualquier sospecha de copia.

La copia se castiga con el suspenso del módulo completo.

**Plazo de entrega:** hasta el 24 de febrero de 2019.

**Peso de la nota:** 15% de la nota (Total entregables 35% de la nota).

## 2. BOLETÍN DE EJERCICIOS

### 1) Tienda online (5 puntos)

**Nota importante:** Para ello haremos una página que usará Vue y para AJAX jQuery/Axios. Opcionalmente puede realizarse usando Quasar.

El objetivo es hacer una especie de tienda online y un carrito de compra (aunque sin su funcionalidad completa). Solo nos encargamos de la parte cliente, ya que la parte servidor está ya hecha.

#### 1.1) Carga de la web

Al cargarse la página deberemos realizar las siguientes acciones

(Para esto se recomienda usar JSON + WebStorage [https://www.w3schools.com/html/html5\\_webstorage.asp](https://www.w3schools.com/html/html5_webstorage.asp)):

- Si ya se cargo una tienda antes, obtenga de los datos almacenados localmente, tanto de la tienda como del carrito de compra.
- Si no había una tienda antes, al cargar pida mediante AJAX una tienda online.

La tienda online nos la proporciona un servidor ya escrito y estará formada por un array de objetos con los siguientes atributos:

- Identificador
- Categoría
- Nombre articulo
- Unidades disponibles
- Precio

#### El array podrá estar desordenado.

La tienda online al recibirse se guardará en datos locales (Otra vez JSON + WebStorage), para hacer que sea persistente.

#### 1.2) Mostrando la web

Deberá mostrarse la tienda online en una tabla siguiendo las siguientes especificaciones:

- Deberán mostrarse los artículos ordenados por categorías. Dentro de una misma categoría el orden dará igual.
- Cada articulo tendrá un input de tipo "text" y un botón "Añadir al carrito".
  - Ese botón añadirá al carrito de la compra los elementos que indique el input de tipo "text". Antes de añadir se comprobará que el campo es numérico. No se realizará ninguna comprobación adicional (ni siquiera del número de existencias)
  - Cuando se añada algo al carrito, además de mostrarse como se indica más abajo, se

almacenará en datos locales (JSON + WebStorage).

### 1.3) Mostrando el carrito

Habrà otra tabla para representar el array de compras del carrito.

El carrito de la compra estarà formado por array de objetos como el de la tienda, solo que “unidades” aquì indicarà cuantas unidades del elemento estàn en el carrito.

Al final de la página existiràn los siguientes botones:

- Un botón “Vaciar carrito” que vaciarà el carrito tanto a nivel visual como en datos locales.
- Un botón “Recargar tienda” que volverà a cargar la tienda de nuevo mediante una petición AJAX, vaciando el carrito tanto visualmente como en datos locales. La antigua tienda se perderà y se guardará la nueva recibida en datos locales.

⚡ Se incluye código de ejemplo que genera una posible tienda “entregable2-2.php” y subido a <http://hispabyte.net/DWEC/entregable2-2.php>

**NOTA IMPORTANTE:** este código incluido es para pruebas, debemos diseñar la tienda de forma que funcione para cualquier tienda en el formato proporcionado.

## 2) Buscando noticias en New York Times (5 puntos)

**Nota importante:** Este ejercicio deberá realizarse usando Quasar (con Vue y para el Ajax jQuery/Axios), de forma que pueda generarse una aplicación web, una de escritorio y otra para Android.

Para este primer ejercicio utilizaremos la API del periódico New York Times. Para usarla necesitaremos una API KEY que obtendremos registrándonos de manera gratuita:

<https://developer.nytimes.com/>

Tras registrarnos, deberemos LEER ATENTAMENTE EL FAQ de uso de la API.

📖 Si tenéis problemas en el registro o dudas en general del funcionamiento de la API podéis plantear vuestros problemas en el foro.

En concreto, para el ejercicio utilizaremos la API de búsqueda de artículos

[https://developer.nytimes.com/article\\_search\\_v2.json](https://developer.nytimes.com/article_search_v2.json)

Nuestra página deberá tener los siguientes elementos Quasar:

- Un input de tipo “text” donde podamos escribir una búsqueda.
  - <https://quasar-framework.org/components/input-textfield.html>
- Dos input de tipo “text”, uno para indicar “Fecha mínima de la noticia” y otro para “Fecha máxima”. El formato será YYYYMMDD

- <https://quasar-framework.org/components/input-textfield.html>
- Un combo box que tenga dos opciones “primero más modernas” y “primero más antiguas”.
- <https://quasar-framework.org/components/select.html>
- Un botón “Realizar búsqueda”
- <https://quasar-framework.org/components/button.html>

**NOTA:** esto es una recomendación. e permiten usar otros elementos “Quasar” siempre que se justifique su usabilidad.

Al pulsar “realizar búsqueda”, se realizará una búsqueda mediante AJAX en New York Time con noticias que incluyan la búsqueda, entre las dos fechas indicadas y ordenadas según lo que diga el combo box (más nuevas primero, más viejas primero).

Al realizar la búsqueda será obligatorio comprobar:

- Que todos los campos están rellenos.
- Que ambas fechas tengan correctamente el formato YYYYMMDD. Comprobar que sean 8 números (no es necesario comprobar si el día es válido, por ejemplo si alguno pone 20001340 es problema del usuario y no debemos comprobarlo).
- Que la fecha mínima sea menor que la máxima.

La respuesta de la petición se cargará dinámicamente en una lista de items <https://quasar-framework.org/components/lists-and-list-items.html>

Al realizar una nueva petición AJAX, el resultado de la anterior se eliminará visualizándose únicamente la nueva petición.