

Catálogo de tienda online

Examen desarrollo Convocatoria Ordinaria

DWEC 2DAW 10Mar25

Necesidad de negocio

Esta actividad consiste en realizar un desarrollo para el control del catálogo de artículos de una tienda online.

Los artículos del catálogo podrán ser de tipo: *Asterix*, *CajaSorpresa* y *StarWars*.

Consideraciones técnicas

- El desarrollo se realizará sobre Javascript, con aportaciones de HTML y CSS
- El alumno utilizará el editor que considere más adecuado, sin conexión a Internet y sin ninguna extensión o plugin instalados relativos a Inteligencia Artificial
- Como base del desarrollo, se proporciona el fichero **catálogo.html**. Este fichero se proporciona al alumno dentro del material del ejercicio
- Todo el tratamiento del UI se realizará mediante Javascript. Dentro de **catálogo.html** se enlazará con el fichero o ficheros **.js** que el alumno genere. No se podrá poner código de Javascript dentro de **catálogo.html**.
- El fichero **catálogo.html** **NO** podrá modificarse, **SALVO** la parte entre **<script>** y **</script>**
- El alumno decidirá sobre detalles no especificados de forma lógica y profesional
- El alumno documentará convenientemente el código para facilitar el entendimiento del mismo
- Se utilizará código lo más declarativo posible (y lo menos imperativo posible), utilizando funcionalidades específicas y avanzadas de Javascript
- Adicionalmente a las funciones que se especificarán posteriormente, el alumno podrá crear otras funciones a su conveniencia. Se penalizará desarrollar varias veces un código similar para realizar una misma tarea o muy parecida
- Se proporciona al alumno:
 - **catálogo.html**: es el esqueleto sobre el que construir el desarrollo de este ejercicio
 - **eventoTeclado.html**: Ejemplo para la gestión de un evento de teclado
 - **mostrarImagen.html**: Ejemplo para añadir un elemento que contiene una imagen
 - Ficheros que contienen artículos y que se importarán en el punto 1 de las tareas especificadas más adelante. Estos ficheros son:
 - **Artículos Asterix.csv**: Lista de artículos a importar como objeto **articuloAsterix**
 - **Artículos CajaSorpresa.csv**: Lista de artículos a importar como objeto **articuloCajaSorpresa**

- **Artículos StarWars.csv**: Lista de artículos a importar como objeto **articuloStarWars**
- Imágenes de los artículos que se importarán de los ficheros anteriores (el nombre de cada imagen será una de las propiedades del objeto de cada artículo del catálogo), además de algunas imágenes adicionales para poder dar de alta nuevos artículos. El alumno creará un directorio al mismo nivel que el fichero **catálogo.html** llamado **images**; dentro de ese directorio se copiarán las imágenes proporcionadas (sin crear otros subdirectorios)
- Un **SKU** es código asignado a un artículo (en inglés, Stock-keeping unit) para poder identificarlo en un catálogo. Debe de ser único dentro de ese catálogo

Estructuras de datos

Se crearán 3 estructuras de datos, una por cada tipo de artículo (**Asterix**, **CajaSorpresa** y **StarWars**).

Cada una de estas estructuras será una array de objetos. Dentro de cada uno de los arrays de objetos, cada elemento del array vendrá definido por los siguientes tipos:

- **articuloAsterix** para los artículos de Asterix
- **articuloCajaSorpresa** para los artículos de CajaSorpresa
- **articuloStarWars** para los artículos de StarWars

Tipos de objetos

- Tipo de objeto **articulo**. Tendrá las siguientes propiedades:

Nombre de propiedad	Tipo	Formato
sku	String	3 letras en mayúscula, seguidas de 3 ó 4 dígitos <ul style="list-style-type: none"> • Letras: [A-Z], sin acentos ni Ñ (como en el alfabeto inglés) • Dígitos: [0-9] <i>Ejemplos: EAA122, MBO5050, FGE0035</i>
nombreArticulo	String	Varias palabras compuestas de caracteres (letras o dígitos) separadas por espacio o por el carácter “-” <ul style="list-style-type: none"> • Letras: [a-z] y [A-Z], sin acentos ni Ñ (como en el alfabeto inglés) • Dígitos: [0-9] <i>Ejemplos: Felpudo Asterix Y Obelix, Mini Busto Luke Skywalker</i>
precio	Integer	Mayor que 0 . Se representará en pantalla con 2 decimales
numArticulos	Integer	Número entero mayor que 0
imagen	String	Cualquier String que admita el navegador como nombre de fichero en Windows 10/11

- Tipo de objeto **articuloAsterix**. Tendrá TODAS las propiedades de **articulo** más:

Nombre de propiedad	Tipo	Formato
tipoObjeto	String	Una palabra (caracteres sin espacios) compuesta de caracteres (letras o dígitos) <ul style="list-style-type: none"> • Letras: [a-z] y [A-Z], sin acentos ni Ñ (como en el alfabeto inglés) <i>Ejemplos: Taza, Felpudo, Poster</i>

La adquisición de las propiedades de **articulo** se realizará a través de prototipos.

- Tipo de objeto **articuloCajaSorpresa**. Tendrá TODAS las propiedades de **articulo** más:

Nombre de propiedad	Tipo	Formato
tematica	String	Una palabra (caracteres sin espacios) compuesta de caracteres (letras o dígitos) <ul style="list-style-type: none"> Letras: [a-z] y [A-Z], sin acentos ni “Ñ” (como en el alfabeto inglés) Ejemplos: <i>Beatles, Stitch, Snoopy</i>

La adquisición de las propiedades de **articulo** se realizará a través de prototipos.

- Tipo de objeto **articuloStarWars**. Tendrá TODAS las propiedades de **articulo** más:

Nombre de propiedad	Tipo	Formato
protagonista	String	Una palabra (caracteres sin espacios) compuesta de caracteres (letras o dígitos) <ul style="list-style-type: none"> Letras: [a-z] y [A-Z], sin acentos ni “Ñ” (como en el alfabeto inglés) Ejemplos: <i>PrincesaLeia, DarthSidious, DaganGera</i>

La adquisición de las propiedades de **articulo** se realizará a través de prototipos.

Tareas

1 - Importación de datos y creación de estructuras de datos

Como primer paso, se importarán los datos de los ficheros proporcionados conteniendo los artículos de los tipos **Asterix**, **CajaSorpresa** y **StarWars**. Los artículos de cada uno de estos tipos estarán en una estructura distinta.

Para ello, se utilizará la función **importar()** que se ejecutará a consecuencia de hacer click en el botón **Importar** (**id="importar-boton"**).

No se importará la línea de cabecera de cada fichero.

Nombre función	Argumentos	Devuelve	Qué realiza
importar()	(A elección del alumno)	0 : Resultado correcto (se ha realizado la importación correctamente) -1 : Se ha producido algún problema (indeterminado)	<ul style="list-style-type: none"> Leerá los 3 ficheros con formato CSV proporcionados, que contienen los artículos a importar de Asterix, CajaSorpresa y StarWars Los ficheros se seleccionarán a través de los input type="file" que aparecen en el fichero catalogo.html La importación se realizará a través del botón con id="importar-boton", una vez seleccionados los 3 ficheros Una vez leído el contenido de cada fichero se crearán los objetos (uno por cada artículo) en la estructura correspondiente Al realizar la importación, no se analizará el formato de la información. Simplemente, se crearán los objetos y se pondrá el valor de las propiedades correspondientes El usuario no realizará ninguna acción sin antes haber realizado la importación

2 - Listado de todos los artículos

Se ofrecerá al usuario la lista de todos los artículos presentes en la sección **Vista Artículos** (**id="vista-articulos"**), según el formato de columnas que refleja en la tabla de dicha sección (**id="vista-articulos-tabla"**).

La forma de crear la tabla o de actualizarla será al producirse el evento generado al pulsar la tecla “\$”. En ese momento, se llamará a la función **vistaArticulos()**.

En la documentación que acompaña a este ejercicio, se incluye un ejemplo de utilización de eventos de teclado.

Nombre función	Argumentos	Devuelve	Qué realiza
vistaArticulos()	(Ninguno)	0 : Resultado correcto (se ha creado o actualizado la tabla de artículos correctamente) -1 : Se ha producido algún problema (indeterminado)	Mostrará una lista completa de todos los artículos existentes de los 3 tipos (articuloAsterix , articuloCajaSorpresa y articuloStarWars). A tener en cuenta: <ul style="list-style-type: none"> Las columnas que aparecerán serán las señaladas dentro del elemento tabla con id="vista-articulos-tabla" El fondo de cada celda de la tabla será del siguiente color: <ul style="list-style-type: none"> #F0E97A – Si el campo tiene un formato correcto #EF6902 – Si el campo tiene formato incorrecto La lista se creará o se actualizará cuando el usuario presione la tecla "\$"

3 - Alta de nuevo artículo

En la sección **Alta Artículo** (**id="alta-articulo"**) se podrá añadir un nuevo artículo al catálogo.

Para ello, se rellenarán los **input type="text"** correspondientes con la información del nuevo artículo.

Una vez rellenos, el usuario podrá pulsar el botón **Alta artículo** (**id="alta-articulo-boton"**), lo que originará llamar a la función **altaArticulo()**.

Para conocer el resultado del alta, se escribirá un mensaje en un elemento tipo <p> debajo del botón de esta sección con el mensaje:

- “Alta realizada correctamente”: si el alta ha sido correcta (la función **altaArticulo()** devuelve **0**)
- “Faltan argumentos”: si la función **altaArticulo()** devuelve **-1**
- “Argumento con formato incorrecto”: si la función **altaArticulo()** devuelve **-2**
- “El tipo de artículo no es válido”: si la función **altaArticulo()** devuelve **-3**

Cada vez que se pulse el botón **Alta artículo**, este mensaje se borrará (si existe) por haber quedado obsoleto.

Nombre función	Argumentos	Devuelve	Qué realiza
altaArticulo()	tipoArticulo newEspecifico newSku newNombreArticulo newPrecio newNumArticulos newImagen	0 : Alta correcta -1 : Faltan argumentos -2 : Algún argumento con formato incorrecto -3 : tipoArticulo no es ni Asterix ni CajaSorpresa ni StarWars	Dará de alta un nuevo artículo con el objeto del tipo que marque tipoArticulo (Asterix , CajaSorpresa o StarWars). Tendrá en cuenta: <ul style="list-style-type: none"> El argumento newEspecifico hará referencia a tipoObjeto, tematica o protagonista, dependiendo del valor de tipoArticulo El resto de argumentos corresponden con las propiedades del tipo de objeto articulo Comprobará que no quedará ninguna propiedad vacía al crear el nuevo artículo Comprobará que no existe otro artículo con el mismo sku Se rellenarán los campos de la sección titulada Alta Artículo (id="alta-articulo") con los datos del nuevo artículos Se ejecutará la función cuando el usuario, una vez rellenos los datos, haga click en el botón Alta Artículo (id="alta-articulo-boton")

4 - Añadir unidades a un artículo existente

La sección **Añadir unidades a un producto existente** (**id="add-units"**) permitirá añadir unidades de un determinado artículo en el almacén.

Para ello, se rellenarán los **input type="text"** correspondientes al SKU del artículo y el número de unidades a añadir.

Una vez rellenos, el usuario podrá pulsar el botón **Añadir unidades** (**id="add-units-boton"**), lo que originará llamar a la función **anadirUnidades()**.

Para conocer el resultado de la operación, se escribirá un mensaje en un elemento tipo `<p>` debajo del botón de esta sección con el mensaje:

- “Unidades añadidas correctamente”: si el alta ha sido correcta (la función **anadirUnidades()** devuelve **0**)
- “No existe el SKU introducido”: si la función **anadirUnidades()** devuelve **-1**
- “Número de artículos < 0”: si la función **anadirUnidades()** devuelve **-2**

Cada vez que se pulse el botón **Añadir unidades**, este mensaje se borrará (si existe) por haber quedado obsoleto.

Nombre función	Argumentos	Devuelve	Qué realiza
anadirUnidades()	sku num	0 : Resultado correcto (se han añadido las unidades) -1 : No existe sku -2 : num < 0	Añadirá al artículo con propiedad sku el número de unidades num . Contendrá las siguientes tareas: <ul style="list-style-type: none">• Comprobará que existe un artículo con ese sku• Comprobará que num > 0• Añadirá num al número de unidades existente (numArticulos)

5 - Ver imágenes de artículos

En esta sección, tenemos la posibilidad de ver las imágenes de los artículos de un determinado tipo (**Asterix**, **CajaSorpresa** o **StarWars**).

Para mostrar las imágenes se utilizará la función **verImagenes()** de forma **asíncrona**. La creación de cada elemento imagen (**img**) se realizará a través de una **Promise** (Promesa).

Se llamará a esta función cada vez que el ratón entre en uno de los elementos con **class="cuadro"** (utilizar evento **mouseenter**). Se procederá de la siguiente manera:

- Si el ratón entra en el área del elemento con texto “**Asterix**” (**id="imagenes-asterix"**), se llamará a la función **verImagenes()** con el argumento “**Asterix**”, para que aparezcan todas las imágenes con tipo de objeto **articuloAsterix**
- Si el ratón entra en el área del elemento con texto “**CajaSorpresa**” (**id="imagenes-cajasorpresa"**), se llamará a la función **verImagenes()** con el argumento “**CajaSorpresa**”, para que aparezcan todas las imágenes con tipo de objeto **articuloCajaSorpresa**
- Si el ratón entra en el área del elemento con texto “**StarWars**” (**id="imagenes-starwars"**), se llamará a la función **verImagenes()** con el argumento “**StarWars**”, para que aparezcan todas las imágenes con tipo de objeto **articuloStarWars**

Cada imagen tendrá un tamaño de 200 x 200 pixels.

Nombre función	Argumentos	Devuelve	Qué realiza
verImagenes()	tipoArticulo	(No se tendrán en cuenta)	<p>Mostrará las imágenes solamente de los artículos del tipo tipoArticulo (Asterix, CajaSorpresa o StarWars).</p> <p>Funcionará del siguiente modo:</p> <ul style="list-style-type: none"> • Aparecerán las imágenes de todos los artículos existentes del tipo tipoArticulo dentro del elemento con class="contenedor-imagenes" <ul style="list-style-type: none"> • Si tipoArticulo =" Asterix" → aparecerán todas las imágenes con tipo de objeto articuloAsterix • Si tipoArticulo =" CajaSorpresa" → aparecerán todas las imágenes con tipo de objeto articuloCajaSorpresa • Si tipoArticulo =" StarWars" → aparecerán todas las imágenes con tipo de objeto articuloStarWars • Las imágenes permanecerán hasta que se vuelva a llamar a esta función. En ese momento, se quitarán todas las imágenes y se volverán a mostrar de la misma manera que en el punto anterior • Las imágenes se encuentran en el directorio images (ver Consideraciones técnicas)

Entregables

El alumno entregará el al Aula Virtual un fichero **.zip** con el formato "**nombre apellido.zip**" que contendrá:

- **catálogo.html** incluyendo las modificaciones realizadas por el alumno
- Todos los ficheros Javascript (**.js**) que el alumno haya generado
- El directorio incluyendo las imágenes de los artículos (**images**)