

# Tienda de plantas

## Examen desarrollo Convocatoria Extraordinaria

DWEC 2DAW 17Jun25

### Necesidad de negocio

Esta actividad consiste en realizar un desarrollo para una tienda cuya actividad es la venta de plantas.

### 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 utilizará el fichero ***tiendaPlantas.html***. Este fichero se proporciona al alumno como material del ejercicio
- Todo el tratamiento del UI se realizará mediante Javascript. Dentro de ***tiendaPlantas.html*** se enlazará con el fichero o ficheros ***.js*** que el alumno genere. No se podrá poner código de Javascript dentro de ***tiendaPlantas.html***.
- El fichero ***tiendaPlantas.html*** **NO** podrá modificarse, **SALVO** la parte entre **`<script>`** y **`</script>`**
- El alumno decidirá, de forma lógica y profesional, sobre detalles no especificados
- 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:
  - ***tiendaPlantas.html***: es el esqueleto sobre el que construir el desarrollo de este ejercicio
  - ***index.html***: Ejemplo de lectura de 2 ficheros
  - Ficheros que contienen artículos y que se importarán en el punto 1 de las tareas especificadas más adelante. Estos ficheros están en formato **CSV** y son:
    - ***Listado de Plantas.csv***: Contiene todos los productos (plantas) que vende la tienda
    - ***Listado Clientes.csv***: Contiene el listado de los clientes de la tienda
    - ***Listado de Ventas.csv***: Contiene las ventas que se han realizado desde el 1 de enero hasta el 31 de mayo de 2025

# Estructuras de datos

Se crearán 3 estructuras de datos, una con las plantas, otra con los clientes y otra de la ventas realizadas.

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:

- **planta** para las plantas que vende la tienda
- **cliente** para los clientes de la tienda
- **venta** para las ventas realizadas por la tienda

## Tipos de objetos

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

Nombre de propiedad	Tipo	Formato
<b>idPlanta</b>	String	5 dígitos <ul style="list-style-type: none"><li>• Dígitos: <b>[0-9]</b></li></ul> <i>Ejemplos: 64453, 00278, 83022</i>
<b>nombrePlanta</b>	String	Varias palabras compuestas de caracteres (letras o dígitos) separadas por espacio <ul style="list-style-type: none"><li>• Letras: <b>[a-z]</b> y <b>[A-Z]</b>, sin acentos ni “Ñ” (igual que en el alfabeto inglés)</li><li>• Dígitos: <b>[0-9]</b></li></ul> <i>Ejemplos: Hosta sieboldiana, Impatiens walleriana, Heuchera</i>
<b>tamanoMaceta</b>	Integer	Número entero mayor que 0. Se representará en pantalla sin decimales
<b>existencias</b>	Integer	Número entero mayor que 0. Se representará en pantalla sin decimales
<b>precio</b>	Integer	Mayor que <b>0</b> . Se representará en pantalla con 2 decimales
<b>plazoEntrega</b>	Integer	Número entero mayor que <b>0</b>
<b>foto</b>	String	Cualquier String que admita el navegador como nombre de fichero en Windows 10/11

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

Nombre de propiedad	Tipo	Formato
<b>idCliente</b>	String	5 dígitos <ul style="list-style-type: none"><li>• Dígitos: <b>[0-9]</b></li></ul> <i>Ejemplos: 64443, 02678, 80922</i>
<b>nombre</b>	String	Una palabra compuesta de caracteres (letras) sin espacios <ul style="list-style-type: none"><li>• Letras: <b>[a-z]</b> y <b>[A-Z]</b>, con acentos y “Ñ”</li></ul> <i>Ejemplos: Luis, Andrés, Pedro</i>
<b>apellido</b>	String	Una palabra compuesta de caracteres (letras) sin espacios <ul style="list-style-type: none"><li>• Letras: <b>[a-z]</b> y <b>[A-Z]</b>, con acentos y “Ñ”</li></ul> <i>Ejemplos: Atienza, Pérez, Muñoz</i>
<b>telefono</b>	String	Un texto que empieza por “+”, seguido de 2 cifras de prefijo de país, un espacio y 9 dígitos, separados de 3 en 3 por espacios <ul style="list-style-type: none"><li>• Dígitos: <b>[0-9]</b></li></ul> <i>Ejemplos: +56 617 654 321, +34 687 339 182, +34 678 123 987</i>
<b>codTarjeta</b>	String	Un código compuesto de 5 caracteres (letras y/o números) sin espacios <ul style="list-style-type: none"><li>• Letras: <b>[a-z]</b> y <b>[A-Z]</b>, sin acentos ni “Ñ” (igual que en el alfabeto inglés)</li></ul> <i>Ejemplos: C5FZM, VZK14, 4JMTK</i>

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

Nombre de propiedad	Tipo	Formato
<b>idVenta</b>	String	5 dígitos • Dígitos: <b>[0-9]</b> <i>Ejemplos: 64443, 02678, 80922</i>
<b>elementosVendidos</b>	Array de objetos	Dentro del tipo de objeto <b>venta</b> se incluirá un array de objetos. Cada uno de estos objetos tendrá 2 propiedades: • <b>idPlanta</b> (tiene que coincidir con algún <b>idPlanta</b> del tipo de objeto <b>planta</b> ) • <b>unidades</b> . En el array, habrá un objeto con estas 2 propiedades por cada planta vendida que contenga en el fichero <b>Listado de Ventas.csv</b> con el mismo <b>idVenta</b> .
<b>idPlanta</b>	String	5 dígitos (tiene que coincidir con algún <b>idPlanta</b> del tipo de objeto <b>planta</b> ) • Dígitos: <b>[0-9]</b> <i>Ejemplos: 6443, 0278, 8022</i>
<b>unidades</b>	Integer	Número entero mayor que 0. Se representará en pantalla sin decimales
<b>idCliente</b>	String	5 dígitos (tiene que coincidir con algún <b>idCliente</b> del tipo de objeto <b>cliente</b> ) • Dígitos: <b>[0-9]</b> <i>Ejemplos: 64443, 02678, 80922</i>
<b>fechaPedido</b>	String	Formato <b>dd/mm/2025</b> (dd = día; mm = día del mes) • dd: 2 dígitos del 01 al 31 <b>[0-9]</b> • mm: 2 dígitos del 01 al 12 <b>[0-9]</b>

## Tareas

### 1 - Importación de datos y creación de estructuras de datos

Como primer paso, en la sección **Importar** (**id="importar"**), se importarán los datos de los 3 ficheros en formato **CSV** proporcionados conteniendo los elementos de los tipos **planta**, **cliente** y **venta**. Los elementos de cada uno de estos tipos se introducirán en la estructura creada para tal fin (array de objetos; uno por cada tipo de elemento).

Para ello, se utilizará la función **importar()** que se ejecutará a consecuencia de hacer click en el botón **Importar** (**id="importar-button"**), una vez que el usuario seleccione los ficheros correspondientes.

No se importará la línea de cabecera de cada fichero.

El usuario no realizará ninguna acción del resto de secciones de **tiendaPlantas.html** sin antes haber realizado la importación de los 3 ficheros.

Nombre función	Argumentos	Devuelve	Qué realiza
<b>importar()</b>	(A elección del alumno)	<b>0</b> : Resultado correcto (se ha realizado la importación correctamente correctamente) <b>-1</b> : Se ha producido algún problema (indeterminado)	<ul style="list-style-type: none"> <li>• Leerá los 3 ficheros con formato <b>CSV</b> proporcionados, que contienen los elementos a importar de <b>plantas</b>, <b>clientes</b> y <b>ventas</b></li> <li>• Los ficheros se seleccionarán a través de los <b>input type="file"</b> que aparecen en el fichero <b>tiendaPlantas.html</b></li> <li>• La importación se realizará a través del botón con <b>id="importar-button"</b>, una vez seleccionados los 3 ficheros</li> <li>• Una vez leído el contenido de cada fichero, se crearán los objetos o se añadirán propiedades a objetos ya creados en la estructura correspondiente</li> <li>• Al realizar la importación, no se analizará el formato de la información. Simplemente, se crearán/añadirán los objetos y se pondrá el valor de las propiedades correspondientes según la lectura de los ficheros</li> <li>• <b>El usuario no realizará ninguna acción sin antes haber realizado la importación de los 3 ficheros</b></li> </ul>

## 2 - Listado de todos los clientes de la tienda

Se ofrecerá al usuario la lista de todas las plantas disponibles en la tienda en la sección **Listado de clientes** (*id="listado-clientes"*), según el formato de columnas que refleja en la tabla de dicha sección (*id="listado-clientes-tabla"*).

La forma de crear la tabla o de actualizarla será a través del evento que se produzca cuando el usuario presione el botón **Generar listado de clientes** (*id="listado-clientes-button"*). En ese momento, se llamará a la función **listadoClientes()**.

Para la comprobación de si un **idPlanta** tiene formato correcto o no, se llamará a la función **comprobarId()**.

En la consola del navegador, se generará un mensaje dependiendo del código que devuelva la función **listadoClientes()** tras ejecutarse:

- Si **listadoClientes()** devuelve **0**, el mensaje en consola será **"Listado de clientes completado con éxito"**
- Si **listadoClientes()** devuelve **-1**, el mensaje en consola será **"Listado de clientes completado con problemas"**

Nombre función	Argumentos	Devuelve	Qué realiza
<b>listadoClientes()</b>	(Ninguno)	<b>0</b> : Resultado correcto (se ha creado o actualizado la tabla con las plantas correctamente) <b>-1</b> : Se ha producido algún problema (indeterminado)	Mostrará una lista completa de todos los clientes de la tienda en formato tabla. A tener en cuenta: <ul style="list-style-type: none"><li>• Las columnas que aparecerán serán las señaladas dentro del elemento tabla con <i>id="listado-clientes-tabla"</i></li><li>• El fondo de cada fila de la tabla será del siguiente color:<ul style="list-style-type: none"><li>• <b>#7AF0CB</b> – Si el campo <b>idCliente</b> tiene un formato correcto</li><li>• <b>#F075D2</b> – Si el campo <b>idCliente</b> tiene formato incorrecto</li></ul></li><li>• La lista se creará o se actualizará an generarse el evento producido cuando el usuario presione el botón <b>Generar listado de clientes</b> (<i>id="listado-clientes-button"</i>)</li><li>• Para comprobar si un <b>idCliente</b> tiene formato correcto se llamará a la función <b>comprobarId()</b></li></ul>

Nombre función	Argumentos	Devuelve	Qué realiza
<b>comprobarId()</b>	<b>id (String)</b>	<b>0</b> : Formato correcto. El <b>Id</b> pasado como argumento cumple con el formato de <b>idPlanta</b> , <b>idCliente</b> o <b>idVenta</b> (es el mismo formato en los 3 casos) <b>-1</b> : Formato incorrecto. El <b>Id</b> no cumple con el formato de <b>idPlanta</b> , <b>idCliente</b> o <b>idVenta</b> (indeterminado)	Comprobará que el argumento <b>id</b> cumple con el formato de <b>idPlanta</b> , <b>idCliente</b> o <b>idVenta</b> , que es el mismo para los 3: 5 dígitos <ul style="list-style-type: none"><li>• Dígitos: <b>[0-9]</b></li></ul> Ejemplos: 64443, 02678, 80922

## 3 – Mostrar los datos de una venta concreta

En la sección **Mostrar datos de venta** (*id="mostrar-venta"*), el usuario podrá ver los detalles de una venta realizada por un cliente. Una venta viene definida por todas las plantas vendidas bajo ese **idVenta** a un cliente determinado.

El usuario elegirá un *idVenta* rellenando el campo **Introducir ID de venta** (*id="mostrar-venta-id"*). Al pulsar el botón **Mostrar venta** (*id="mostrar-venta-button"*), se recogerá la información del *input type="text"* correspondiente (*id="mostrar-venta-id"*) y se mostrarán los datos de la venta a través de 2 tablas de esta sección llamando a la función **mostrarVenta()**. Dichas 2 tablas se completarán con la siguiente información:

- Tabla con *id="mostrar-venta-tabla-base"*
  - Nombre del cliente
  - Apellido del cliente
  - Teléfono
  - Número de unidades totales de la venta (suma de las unidades de cada planta que el cliente ha comprado)
  - Días desde la fecha de la venta hasta la fecha actual. Se calculará la diferencia entre la fecha en la que se ha realizado la venta hasta la fecha de hoy (fecha en la que se ejecute este código)
- Tabla con *id="mostrar-venta-tabla-detalle"*. Se mostrará una fila por cada tipo de planta vendida con las columnas
  - Nombre de la planta
  - Número de unidades vendidas de ese tipo de planta
  - Precio de la planta por unidad

Ejemplo de cómo mostrar una venta (formato aproximado):

Nombre cliente	Apellido cliente	Teléfono	Unidades totales	Días fecha venta hasta hoy
Javier	Fernández	+34 634 211 556	8	15

Nombre Planta	Número de unidades	Precio por unidad
Celosia argentea	2	5,95 €
Dahlia hortensis	3	5,95 €
Gaillardia aristata	3	7,95 €

En la consola del navegador, se generará un mensaje dependiendo del código que devuelva la función **mostrarVenta()** tras ejecutarse:

- Si **mostrarVenta()** devuelve **0**, el mensaje en consola será **“Función MostarVenta completada con éxito”**
- Si **mostrarVenta()** devuelve **-1**, el mensaje en consola será **“Función MostarVenta completada con problemas”**

Nombre función	Argumentos	Devuelve	Qué realiza
<b><i>mostrarVenta()</i></b>	<b><i>idVenta</i></b>	<b>0:</b> Resultado correcto. Se han creado/actualizado las tablas con la venta correctamente <b>-1:</b> Se ha producido algún problema (indeterminado)	Mostrará los detalles de una venta concreta a un cliente, incluyendo los datos y el formato explicado anteriormente. Procedimiento: <ul style="list-style-type: none"> <li>• El usuario introducirá un código de venta en el <b><i>input type="text"</i></b> de la sección titulado <b><i>Introducir ID de venta</i></b>. Este dato se pasará como argumento a esta función</li> <li>• Si el id de venta introducido por el usuario está vacío o no se corresponde con un <b><i>idVenta</i></b> correspondiente a una venta realizada, no se realizará ninguna acción. Terminará la función devolviendo <b>0</b> sin realizar ninguna acción</li> <li>• Si el id de venta introducido por el usuario sí se corresponde con un <b><i>idVenta</i></b> correspondiente a una venta realizada, se buscarán/calcularán los datos que hay que presentar en pantalla y se presentarán con el formato explicado anteriormente en las tablas con <b><i>id="mostrar-venta-tabla-base"</i></b> y <b><i>id="mostrar-venta-tabla-detalle"</i></b></li> </ul>

## Entregables

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

- ***tiendaPlantas.html*** incluyendo las modificaciones realizadas por el alumno
- Todos los ficheros Javascript (**.js**) que el alumno haya generado