

**NOMBRE ALUMNO/A** \_\_\_\_\_

Curso 2018/2019	<b>PORTADA</b>  <b>Gestión de Eventos y Formularios</b>  Grupo: 2º DAW	<b>CALIFICACIÓN</b>
Evaluación: 1º		
Fecha: 05/12/18		

MÓDULO	UNIDAD DE TRABAJO	ACTIVIDAD
DEW	UT5	A12

La tarea consiste en la implementación de varios scripts en lenguaje Javascript. . Se recomienda estructurar el código lo más posible facilitando su mantenimiento y reutilización. Está prohibido el uso de variables globales salvo que se indique explícitamente lo contrario.

Todos los eventos se definirán mediante el modelo de eventos del W3C

- Queremos implementar una página que muestre un pequeño juego que consiste en adivinar una palabra a partir de sus letras. En el campo de texto “Letras” (no editable) se mostrarán las letras de una palabra desordenada, mientras que en el campo de texto “Palabra” (editable) podremos intentar adivinar de qué palabra se trata:

### Adivina la palabra oculta

Letras	<input type="text" value="RNASNADET"/>
Palabra	<input type="text"/>
<input type="button" value="Nueva palabra"/> <input type="button" value="Ver solución"/> <input type="button" value="Finalizar juego"/>	

Además tenemos 3 botones que nos permiten:

- Generar una nueva palabra (inicialmente deshabilitado).
- Ver la solución (palabra ordenada)
- Finalizar el juego.

Cada vez que introduzcamos una letra en el campo “Palabra”, además de pasar la misma a mayúsculas, se comprobará si la palabra introducida coincide con la original. En caso afirmativo, se mostrará un mensaje indicativo, además de habilitar el botón para generar una nueva palabra y deshabilitar el que nos permite ver la solución de la palabra (pues la acabamos de mostrar).

## Adivina la palabra oculta

Letras

Palabra

Has acertado la palabra SANTANDER

Al pulsar “Nueva palabra” se producirán las siguientes acciones:

- Se generará una nueva palabra que se mostrará en la caja de texto “Letras” (la palabra se tomará aleatoriamente de entre al menos 10 valores.).
- Se borrará el contenido de “Palabra”.
- Se restablecerán los botones a su estado inicial.
- Se ocultará el mensaje indicativo.

## Adivina la palabra oculta

Letras

Palabra

En cualquier momento podemos pulsar el botón “Ver solución”. Este botón muestra un mensaje con la palabra ordenada, cambiando de nuevo el estado de los botones. Nótese que el formato del mensaje es distinto al del mensaje anterior.

## Adivina la palabra oculta

Letras

Palabra

La palabra correcta es LAREDO

Por último, si mostramos el botón “Finalizar juego”, se mostrará un mensaje (de formato distinto al demás) que nos mostrará el porcentaje de aciertos, es decir, el total de partidas que hemos acertado frente al total de partidas jugadas, expresado de manera porcentual.

Nótese que una vez que hemos pulsado este botón el resto de botones se deshabilitan, finalizando efectivamente el juego.

## Adivina la palabra oculta

Letras

Palabra

Porcentaje de aciertos: 50%

Para resolver el ejercicio deben usarse los siguientes archivos:

- **adivinapalabra.js**: Contiene la clase **PalabraOculta**, que contiene la lógica necesaria para manejar el juego. Será necesario determinar antes de comenzar a programar qué métodos y propiedades se necesitan (aunque puede ser adecuado focalizarnos en las propiedades y los métodos principales, para luego ir añadiendo los métodos a medida que los necesitemos).
- **palabraoculta\_prueba.js**: Contiene la prueba de la clase **PalabraOculta**
- **ejercicio1.js**: Contiene los manejadores de eventos y el código de acceso al DOM.

NOTA: Podemos establecer el estilo de un elemento del DOM accediendo a su propiedad **style** y a continuación especificando el nombre de la propiedad CSS (en notación **camelCase**) a la que queremos dar valor. Por ejemplo, si queremos ocultar un elemento podemos hacerlo mediante el código siguiente:

```
document.getElementById("elemento").style.display="none"
```

Si queremos mostrar el mismo elemento podemos usar la sentencia  
`document.getElementById("elemento").style.display="block" | "inline"`

## Ejemplo de funcionamiento

Al arrancar la aplicación, nos sale en el formulario la palabra desordenada.  
En consola podemos ver la palabra que es correcta, también la palabra separadas por “,” que es el array desordenado. Y por último la palabra sin las “,” que es la que se muestra finalmente al usuario.

Adivina la palabra oculta

Letras

Palabra

Inspector

Filtrar salida

SERVIDOR

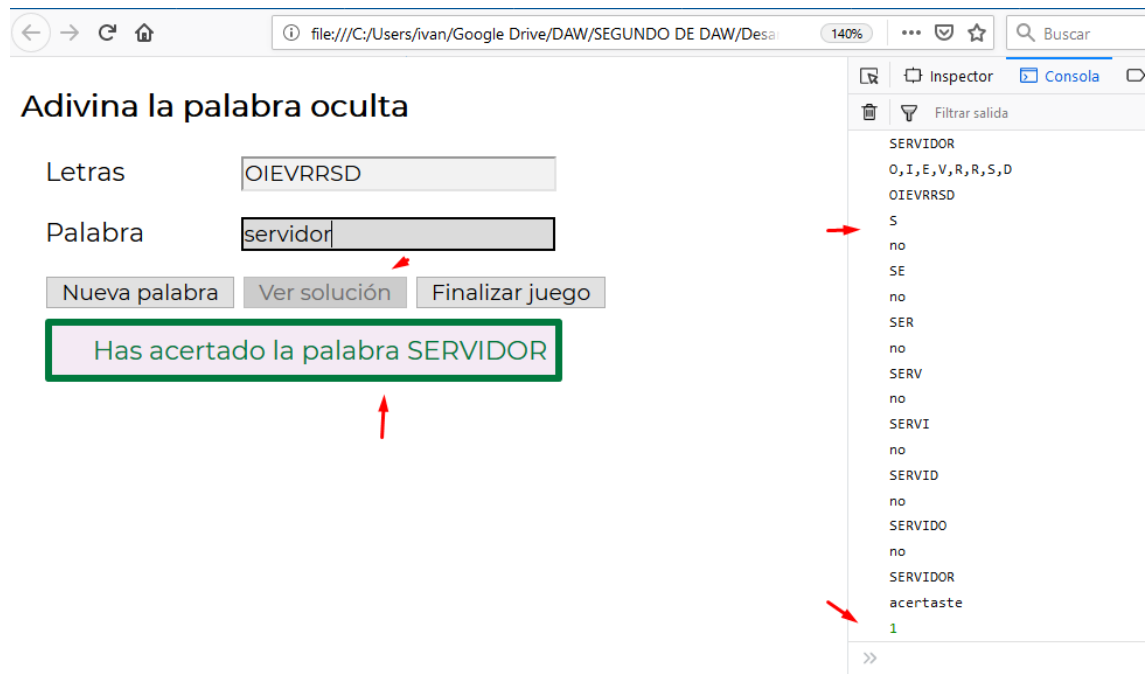
O,I,E,V,R,R,S,D

OIEVRRSD

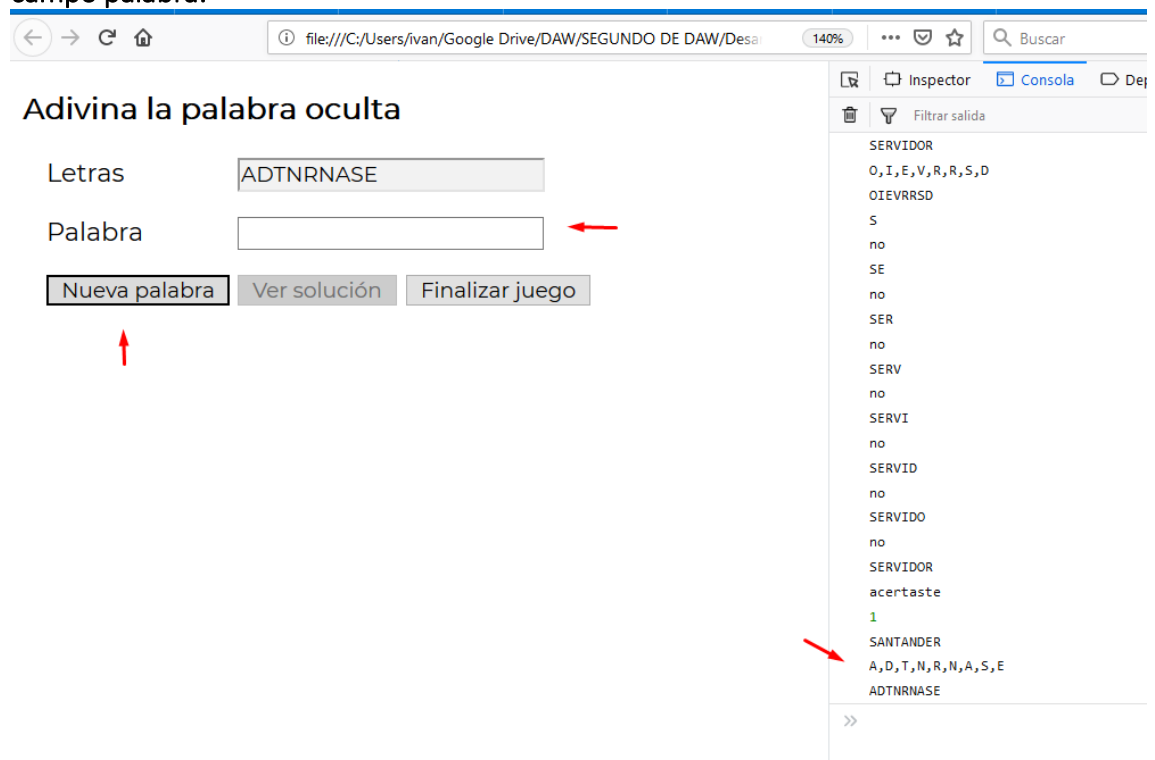
>> |

Cuando rellenamos el campo “Palabra” , hacemos una comprobación por cada letra que se introduce en el campo tal y como dice el enunciado, además se deshabilita el botón ver solución.

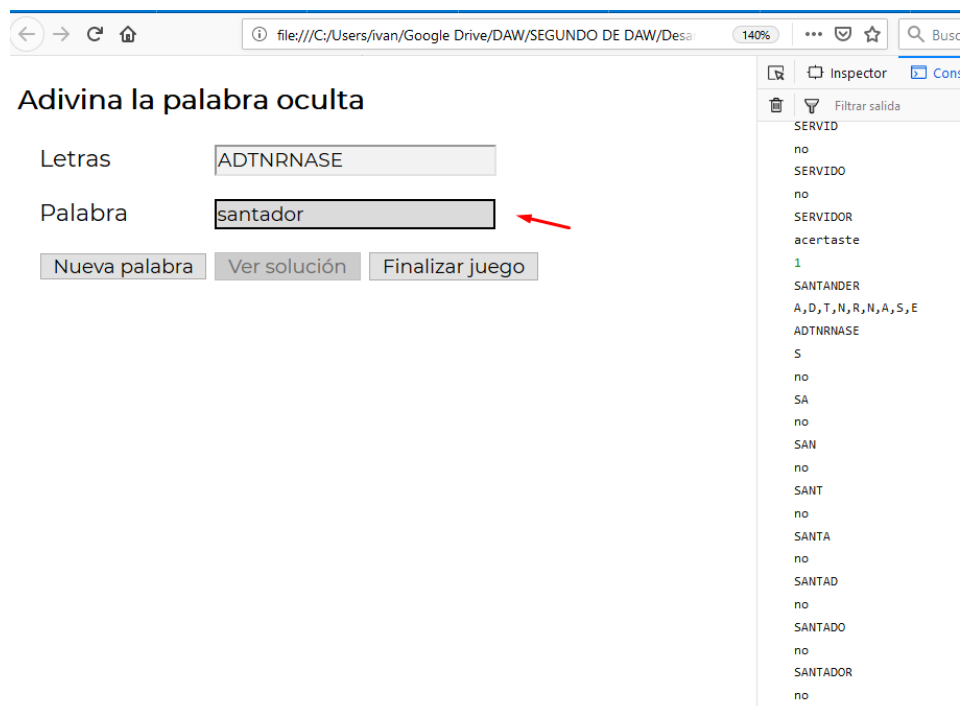
Vemos a la derecha las comprobaciones por cada letra, una vez hallamos acertado se notifica al usuario que la palabra ha sido acertada y se suma 1 punto al contador, **dicho punto lo vemos en consola**



Si le damos a “Nueva Partida”, se genera una palabra nueva, además se limpia el campo palabra.

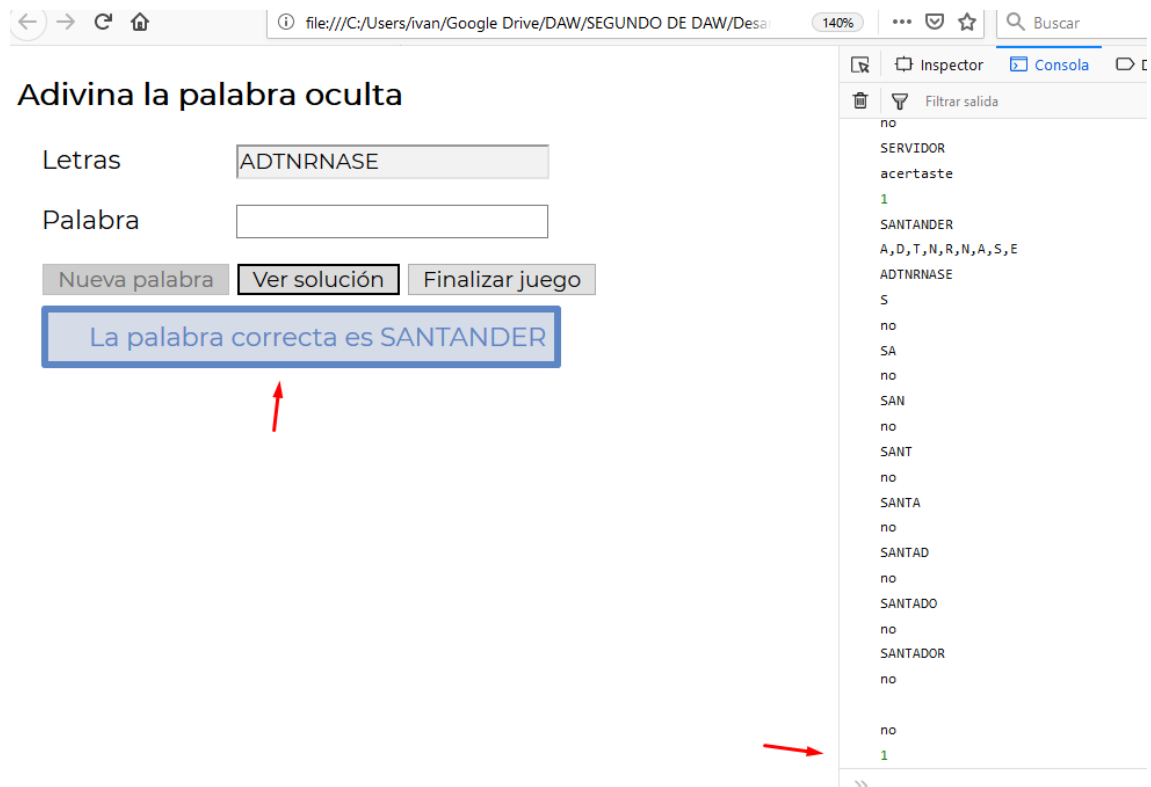


En este caso vamos a suponer que el usuario no sabe la palabra.



Para ver la solución borramos el campo “palabra” y se procedera a la activación del botón “ver solución”.

Si damos a “ver solución” se le dice al usuario que palabra es la correcta y se suma 1 al contador de los fallos.



Finalizamos el juego, y nos muestra el porcentaje de aciertos en este caso 50- 50 puesto que adivinamos una y fallamos otra.

← → ↺ 🏠

file:///C:/Users/ivan/Google Drive/DAW/SEGUNDO DE DAW/Desa

# Adivina la palabra oculta

Letras

ADTNRNASE

Palabra

Nueva palabra

Ver solución

Finalizar juego

Porcentaje de ciertos 50 %

Probaremos con otro porcentaje en este caso acertamos 3 y fallamos 1

← → ↺ 🏠

file:///C:/Users/ivan/Google Drive/DAW/SEGUNDO DE DAW/Desa

140%

⋮ 🛡️ ⭐

# Adivina la palabra oculta

Letras

IBLBOA

Palabra

Nueva palabra

Ver solución

Finalizar juego

Porcentaje de ciertos 75 %

🔍 Inspecto

🗑️ 🔍 Filtrar sa

B

no

BI

no

BIL

no

BILB

no

BILBA

no

BILBAO

acertaste

3

BILBAO

I,B,L,B,O,A

IBLBOA

B

no

BI

no

BIL

no

no

1

4

- Modificar el ejercicio anterior para añadir una línea de botones de radio que nos permitan escoger la longitud de la palabra.

## Adivina la palabra oculta

**Dificultad** ☒ 4 letras ☐ 6 letras ☐ 8 letras

**Letras**

**Palabra**

La palabra se tomará de manera aleatoria de un array de 3 filas por 10 columnas, donde cada fila contendrá las palabras de una de las longitudes (por ejemplo, la primera fila contiene 10 palabras de 4 letras, la segunda fila contiene 10 palabras de 6 letras y la tercera fila contiene 10 palabras de 8 letras).

Además, vamos a guardar una cookie sin caducidad con el valor del mejor porcentaje de partidas hasta la fecha. Esto implica que al pulsar el botón “Finalizar partida” realizaremos las siguientes acciones:

- Leeremos el valor de la cookie.
- Compararemos con el porcentaje obtenido en la partida actual. Si el porcentaje obtenido es más alto que el porcentaje leído, se actualizará el valor de la cookie y además de mostrar la información de la partida actual se informará al usuario de que ha batido el record. En caso contrario, simplemente informaremos de la puntuación obtenida.

## Ejemplo de Funcionamiento

Al iniciar la aplicación tendremos que elegir si queremos jugar con 4,6 u 8 palabras. El juego se inicia, en consola vemos que ha sido el array 1, nos sale la palabra original, la palabra desordenada en un array y la palabra sin “,”

**Adivina la palabra oculta**  
☐ 4 Letras ☒ 6 Letras ☐ 8 Letras  
**Letras**   
**Palabra**

```
console.table()
adivinapalabra.js:41:9
0      1      2      3      4      5      6      7      8
DUDE   COVE   ARFO   ARDA   FUFU   GAÑA   HULA   AHCB   C
AFINAN AFARAS ABALAS ABACAS ABALAN AGORAS AGRAVO AGOTAN A
VAQUEE TANQUE TEQUES TIUQUE JUQUEO QUINAS QUESOS QUISTO P

El array es 1
Original AGORAS
A,A,S,R,G,O
AASRGO
```

### Acertamos la palabra

**Adivina la palabra oculta**  
☐ 4 Letras ☒ 6 Letras ☐ 8 Letras  
**Letras**   
**Palabra**

Has acertado la palabra AGORAS

Jugamos con una de 8 letras, se notifica en consola que es el array 2. En este caso no sabemos la palabra le damos a ver solución y nos sale la correcta.

The screenshot shows a web browser window with the address bar displaying a file path. The page title is "Adivina la palabra oculta". There are three radio buttons for word length: "4 Letras", "6 Letras", and "8 Letras" (which is selected). Below these are input fields for "Letras" (containing "UIIQTE") and "Palabra". At the bottom of the form are three buttons: "Nueva palabra", "Ver solución", and "Finalizar juego". A blue message box at the bottom of the page says "La palabra correcta es TIUQUE". To the right, the browser's developer console is open, showing a list of words being checked: "no", "AG", "no", "AGO", "no", "AGOR", "no", "AGORA", "no", "AGORAS", "acertaste", "undefined", "1". Below this, a table is displayed with the result of a console.table() call:

0	1	2
DUDE	COVE	ARFO
AFINAN	AFARAS	ABALES
VAQUEE	TANQUE	TEQUES

Below the table, the console shows the text: "El array es 2", "Original TIUQUE", "U,U,I,Q,T,E", and "UIIQTE".

Después de jugar varias partidas finalizamos el juego y nos notifica el porcentaje de aciertos.

The screenshot shows the same web browser window. The "8 Letras" radio button is still selected. The "Letras" input field now contains "QUUEOJ" and the "Palabra" input field contains "juqueo". The "Finalizar juego" button is highlighted. Below the form, a red message box says "Porcentaje de aciertos 66 %". At the bottom, a green message box says "Has acertado la palabra JUQUEO".



3. Una empresa de envíos quiere implementar un sistema que nos permita gestionar sus paquetes. Para ello tenemos que crear un formulario que nos permita validar la información de cada paquete teniendo en cuenta los siguientes campos (los campos antepuestos de (\*) son obligatorios).:
- a. (\*) Fecha de envío: Obligatorio y con formato dd/mm/aaaa
  - b. (\*) CP Origen: Formado por 5 dígitos.
  - c. (\*) CP Destino: Formado por 5 dígitos.
  - d. Repartidor: Código formado por 2 letras en mayúscula, un símbolo ·\$ y 4 dígitos (Ej: ES\$3302).
  - e. (\*) Peso del paquete (g): Un número de 50 a 5000.
  - f. (\*) Dimensiones (3 campos) (cm): Ancho por alto por largo. 3 números entre 10 y 100.
  - g. Peso volumétrico (no editable). Se rellena automáticamente a partir de las dimensiones [(A x A x L)/5000].
  - h. (\*) Tarjeta bancaria: 16 dígitos que pueden introducirse de manera consecutiva o en grupos de 4 separados por espacios.
  - i. Código de descuento: 5 letras seguidas de dos dígitos.

Además el formulario contará con una casilla de verificación de “Acepto las condiciones” (inicialmente desactivada). Hasta que no se active dicha casilla no se activará el botón de envío.

Ten en cuenta que el formulario no se enviará a ningún script de servidor. Por tanto, en lugar de un elemento submit, puedes usar un botón de validación. Al pulsar en el mismo se realizará la validación de todos los campos de acuerdo a las directrices especificadas.

En caso de haber algún error, se mostrará en una capa (div) de mensajes (inicialmente oculta) el mensaje o los mensajes correspondientes. En caso de no haber ningún error, también se mostrará un mensaje indicativo en la capa de que todos los campos son correctos. Si se desea pueden utilizarse entradas de formulario de HTML5, pero deberán realizarse las validaciones oportunas en previsión de que el navegador no implemente el tipo de entrada

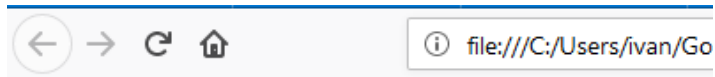
## Ejemplo de Funcionamiento

Rellenando el formulario van saltando errores si no se respeta los campos.

The screenshot shows a web browser window with the address bar displaying 'file:///C:/Users/ivan/Google Drive/DAW/SEGUNDO DE DAW/Desa'. The page title is 'Formulario'. The form contains the following fields and elements:

- Fecha de Envío:** A date picker showing '14 / 01 / 2019'.
- CP Origen:** A text input field containing '35600'.
- CP Destino:** A text input field containing 'd', with a red arrow pointing to it, indicating an error.
- Repartidor:** A text input field.
- Peso Paquete:** A text input field with a spinner.
- Dimensiones:** Three text input fields labeled 'ancho', 'alto', and 'largo', each with a spinner.
- Peso Volumétrico:** A text input field with a spinner.
- Tarjeta Bancaria:** A text input field.
- Código de descuento:** A text input field containing 'ABCDE12'.
- Checkboxes:** A checkbox labeled 'Acepto las condiciones de servicio'.
- Validation Message:** A red box with the text 'Este Campo solo puede llevar números' and a red arrow pointing to the 'CP Destino' field.

Si el código del repartidor sigue la expresión regular se notifica al usuario de que es correcto.



## Formulario

Fecha de Envío 14 / 01 / 2019 ✕

CP Origen 35600

CP Destino 34611

Repartidor ES\$6718

Peso Paquete

Dimensiones ancho alto

Peso Volumétrico

Tarjeta Bancaria

Código de descuento ABCDE12

☐ Acepto las condiciones de servicio

Número de repartidor Correcto

El peso volumétrico se calcula una vez que se haya rellenado el último campo (largo).

Fecha de Envío 14 / 01 / 2019 ✕

CP Origen 35600

CP Destino 34611

Repartidor ES\$6718

Peso Paquete 7

Dimensiones 30 30 15

Peso Volumétrico 2.7

Tarjeta Bancaria

Código de descuento ABCDE12

☐ Acepto las condiciones de servicio

Número de repartidor Correcto

## Introducimos número de tarjeta correcto

file:///C:/Users/ivan/Google Drive/DAW/

### Formulario

Fecha de Envío 14 / 01 / 2019

CP Origen 35600

CP Destino 34611

Repartidor ES\$6718

Peso Paquete 7

Dimensiones 30 30 15

Peso Volumétrico 2.7

Tarjeta Bancaria 7654317826156761

Código de descuento ABCDE12

☐ Acepto las condiciones de servicio

Número de tarjeta Correcto

## Introducimos el código de descuento, y si es correcto se notifica.

file:///C:/Users/ivan/Google Drive/DAW/SEGUNDO DE DAW/Desa

### Formulario

Fecha de Envío 14 / 01 / 2019

CP Origen 35600

CP Destino 34611

Repartidor ES\$6718

Peso Paquete 7

Dimensiones 30 30 15

Peso Volumétrico 2.7

Tarjeta Bancaria 7654317826156769

Código de descuento HJVBC78

☐ Acepto las condiciones de servicio

Código de descuento válido

## Por último al pulsar “Acepto las condiciones de servicio” se habilita el botón enviar.

file:///C:/Users/ivan/Google Driv

### Formulario

Fecha de Envío 14 / 01 / 2019

CP Origen 35600

CP Destino 34611

Repartidor ES\$6718

Peso Paquete 7

Dimensiones 30 30

Peso Volumétrico 2.7

Tarjeta Bancaria 7654317826156769

Código de descuento HJVBC78

☒ Acepto las condiciones de servicio

Enviar

Código de descuento válido

4. Vamos a gestionar una lista de países haciendo uso de una interfaz Web. Para ello se van a implementar los siguientes elementos:
- Clase **País** (archivo **pais.js**). Encapsula las propiedades de un país.
    - Contiene propiedades públicas que permiten almacenar los datos correspondientes a un país:
      - Código único de 2 letras
      - Nombre
      - Población en millones de habitantes.
    - Además se implementarán los siguientes métodos:
      - Constructor con parámetros
      - Método **toString()**. Devuelve una representación en formato cadena de las propiedades concatenadas.

En el archivo **pais\_prueba.js** se instanciarán 3 países de ejemplo y se mostrarán sus propiedades por la consola de depuración.

- Clase **ListaPaíses** (archivo **lista\_paises.js**). Permite manejar un listado de países. Incluye:
  - Una propiedad privada que permite almacenar un listado de países.
  - Métodos que permitan:
    - Devolver el número de elementos del array.
    - Devolver el elemento que está en la posición i-esima.
    - Añadir un elemento al principio del array.
    - Añadir un elemento al final del array.
    - Comprobar si existe un elemento con un código determinado.
    - Borrar el elemento con un código determinado.
    - Borrar un elemento al principio del array (devolviendo el nombre del mismo).
    - Borrar un elemento al final del array (devolviendo el nombre del mismo).
    - Muestra la posición en la que se encuentra un elemento dado su nombre.
    - Devolver el país de mayor población
    - Devolver el país de menor población.

En el archivo **lista\_paises\_prueba.js** se instanciará un objeto de la clase **ListaPaíses** y se realizarán pruebas que permitan comprobar si los métodos funcionan correctamente. Toda la información correspondiente a las pruebas se mostrará en la consola de depuración.

- Crea una página HTML que llamarás **ejercicio4.html**. La página tendrá un script asociado llamado **ejercicio4.js** que contendrá las funciones de manejo de eventos y carga de la página. Está permitido el uso de una única variable global con el listado de países (tipo **ListaPaíses**). La página HTML contendrá un formulario que constará de los siguientes elementos:
  - Campos de texto con las etiquetas Código, Nombre y Población.
    - Aunque el campo código se introduzca en minúsculas, una vez introducido se convertirá automáticamente a mayúsculas.
  - Un botón que permita **Añadir País**
    - Al pulsar el botón deberíamos comprobar (si es posible hacer uso de expresiones regulares):
      - Que el código tiene dos letras.
      - Que el campo población es numérico.
      - Que todos los campos se han introducido
      - Que no existe el país con el código introducido en el campo de texto

En caso de que las validaciones sean correctas se guardará el país, actualizando el listado de países y borrando los campos de texto. Si alguna de las validaciones es incorrecta se mostrará la información en la capa pertinente.

- Un botón que permita **Eliminar País**.
  - Al pulsar el botón se eliminará país cuyo código coincide con el introducido en el campo código. Si no se introduce ningún código, se eliminará el primer país. En cualquier caso, se actualizará la información del listado de países.
    - Si no hay países se mostrará el mensaje en la capa de información.
  - Una capa con el identificador información que mostrará retroalimentación de la última acción que se ha realizado (si se ha insertado/borrado un país), así como los correspondientes errores que se vayan produciendo. Dicha capa debería tener un formato diferenciado.
  - Una capa con el identificador listado que mostrará el listado de países actualizado (o un mensaje indicativo si no hay ningún país). Dicha capa debería tener un formato diferenciado.

Ten en cuenta que el formulario no va a tener botón de envío (*submit*) ni atributos *action* ni *method*, pues no lo enviaremos a ningún script de servidor (toda la información de este ejercicio es gestionada en local).

Si se desea pueden utilizarse entradas de formulario de HTML5, pero deberán realizarse las validaciones oportunas en previsión de que el navegador no implemente el tipo de entrada.

## Ejemplo de Funcionamiento

Instanciamos los 3 países de ejemplo que pone el enunciado.

✓ El nombre del país es ESPAÑA, el código del país es ES, el número de habitantes es 46.659.302	pais.js:14:9
✓ El nombre del país es FRANCIA, el código del país es FR, el número de habitantes es 67.795.823	pais.js:14:9
✓ El nombre del país es ALEMANIA, el código del país es DEU, el número de habitantes es 82.800.000	pais.js:14:9

Observamos algunos ejemplos de los descritos en el enunciado (**no puedo hacer todos al mismo tiempo pero se pueden comprobar cada método uno por uno**)

El número de elementos del array son 2

lista\_paises.js:11:9

console.table()

lista\_paises.js:12:9

(índice)	codigo	nombre	poblacion
0	EEUU	Estados Unidos	60000
1	ES	España	450000
2	FR	Francia	600000
3	CH	China	1300000000

► Object { codigo: "CH", nombre: "China", poblacion: "1300000000" }

lista\_paises.js:18:9

4

lista\_paises.js:39:13

El número de elementos del array son 4

lista\_paises.js:11:9

console.table()

lista\_paises.js:12:9

(índice)	codigo	nombre	poblacion
0	EEUU	Estados Unidos	60000
1	ES	España	450000
2	FR	Francia	600000
3	CH	China	1300000000

La posición de ES es 1

lista\_paises.js:87:9

EL código del País CH existe

lista\_paises.js:50:17

El país con mayor población es China, con una población de 1300000000

lista\_paises.js:107:9

El país con menor población es Estados Unidos con una población de 60000

lista\_paises.js:119:9

## Probamos el formulario

### Añadimos a un País

The screenshot shows a web browser with a file path: `file:///C:/Users/ivan/Google Drive/DAW/SEGUNDO DE DAW/Desarrollo web`. The page title is "Añadir un País". The form contains three input fields: "Codigo del País" (AU), "Nombre del País" (Australia), and "Poblacion" (25000000). Below the inputs are four buttons: "Enviar", "Guardar Países", "Ver Cookies", and "Eliminar País". A red arrow points to the "Nombre del País" field.

The browser's developer console shows the following log:

```
Object { codigo: "CH", nombre: "China", poblacion: 1300000000 } lista_paises.js:18:9
4 lista_paises.js:39:13
El número de elementos del array son 4 lista_paises.js:11:9
console.table() lista_paises.js:12:9
```

(indice)	codigo	nombre	poblacion
0	EEUU	Estados Unidos	60000
1	ES	España	450000
2	FR	Francia	600000
3	CH	China	1300000000

La posición de ES es 1 lista\_paises.js:87:9  
El código del País CH existe lista\_paises.js:50:17  
El país con mayor población es China, con una población de 1300000000 lista\_paises.js:107:9  
El país con menor población es Estados Unidos con una población de 60000 lista\_paises.js:119:9  
5 lista\_paises.js:39:13  
El número de elementos del array son 5 lista\_paises.js:11:9  
console.table() lista\_paises.js:12:9

(indice)	codigo	nombre	poblacion
0	EEUU	Estados Unidos	60000
1	ES	España	450000
2	FR	Francia	600000
3	CH	China	1300000000
4	AU	Australia	25000000

A red arrow points to the "CH" row in the second table.

Añadimos a un país que ya existe.

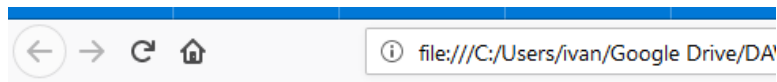
The screenshot shows the same web application as before. The "Codigo del País" field now contains "AU". A red arrow points to this field. Below the form, a red error message is displayed: "El código del país ya existe". A red arrow points to this message.

The browser's developer console shows the following log:

```
2 FR
3 CH
La posición de ES es 1
El código del País CH existe
El país con mayor población es China, con una población de 1300000000
El país con menor población es Estados Unidos con una población de 60000
5
El número de elementos del array
console.table()
(indice) codigo
0 EEUU
1 ES
2 FR
3 CH
4 AU
El número de elementos del array
console.table()
(indice) codigo
0 EEUU
1 ES
2 FR
3 CH
4 AU
```

A red arrow points to the "4 AU" row in the second table.

También existen validaciones como por ejemplo la “población”



## Añadir un País

Codigo del País   
Nombre del País   
Poblacion

La población debe ser de tipo numérico. Debe rellenar todos los campos

## Eliminamos a un país (Francia)

**Añadir un País**

Codigo del País   
Nombre del País   
Poblacion

**Console**

Filtrar salida ☐ Registros persistentes

(indice)	codigo	nombre	poblacion
0	EEUU	Estados Unidos	60000
1	ES	España	450000
2	FR	Francia	600000
3	CH	China	1300000000
4	AU	Australia	2500000

El número de elementos del array son 4

(indice)	codigo	nombre	poblacion
0	EEUU	Estados Unidos	60000
1	ES	España	450000
2	CH	China	1300000000
3	AU	Australia	2500000

5. Modifica el ejercicio anterior para añadir un botón que permita **Guardar Países**. Al pulsar el botón se guardará el listado de países en una cookie con caducidad de un día (en el formato que se prefiera). Al cargar la página, habrá que consultar si existe la cookie con el listado de países, actualizando tanto el array como la información que se muestra en pantalla.

## Ejemplo de funcionamiento

(Este apartado se ha realizado en el mismo archivo que el 4 )

*Ojo con la caducidad que tiene puesta la cookie!*

Añadimos un país (México), una vez añadido pulsamos “Guardar Países”, se nos muestra en consola la información guardada en la cookie.

**Añadir un País**

Codigo del País

Nombre del País

Poblacion

```
console.table() lista_paises.js:12:9
(indice) codigo nombre poblacion
0 EEUU Estados Unidos 60000
1 ES España 450000
2 FR Francia 600000
3 CH China 1300000000

La posición de ES es 1 lista_paises.js:87:9
El código del País CH existe lista_paises.js:50:17
El país con mayor población es China, con una población de lista_paises.js:107:9
1300000000
El país con menor población es Estados Unidos con una lista_paises.js:119:9
poblacion de 60000
5 lista_paises.js:39:13
El número de elementos del array son 5 lista_paises.js:11:9
console.table() lista_paises.js:12:9
(indice) codigo nombre poblacion
0 EEUU Estados Unidos 60000
1 ES España 450000
2 FR Francia 600000
3 CH China 1300000000
4 MX Mexico 600000

({ "listaPaises": [{"codigo": "EEUU", "nombre": "Estados Unidos", "poblacion": "60000"}, {"codigo": "ES", "nombre": "España", "poblacion": "450000"}, {"codigo": "FR", "nombre": "Francia", "poblacion": "600000"}, {"codigo": "CH", "nombre": "China", "poblacion": "1300000000"}, {"codigo": "MX", "nombre": "Mexico", "poblacion": "600000"}] }) ejercicio4.js:31:5
```

Cerramos y abrimos el navegador, le damos a “ver cookies” y nos muestra efectivamente el país México almacenada en la cookie.

**Añadir un País**

Codigo del País

Nombre del País

Poblacion

```
console.table() lista_paises.js:12:9
(indice) codigo nombre poblacion
0 EEUU Estados Unidos 60000
1 ES España 450000
2 FR Francia 600000
3 CH China 1300000000
4 MX Mexico 600000

({ "listaPaises": [{"codigo": "EEUU", "nombre": "Estados Unidos", "poblacion": "60000"}, {"codigo": "ES", "nombre": "España", "poblacion": "450000"}, {"codigo": "FR", "nombre": "Francia", "poblacion": "600000"}, {"codigo": "CH", "nombre": "China", "poblacion": "1300000000"}, {"codigo": "MX", "nombre": "Mexico", "poblacion": "600000"}] }) ejercicio4.js:31:5

Cookies ejercicio4.js:43:5
[{"codigo": "EEUU", "nombre": "Estados Unidos", "poblacion": "60000"}, {"codigo": "ES", "nombre": "España", "poblacion": "450000"}, {"codigo": "FR", "nombre": "Francia", "poblacion": "600000"}, {"codigo": "CH", "nombre": "China", "poblacion": "1300000000"}, {"codigo": "MX", "nombre": "Mexico", "poblacion": "600000"}] ejercicio4.js:46:5

({ "listaPaises": [{"codigo": "EEUU", "nombre": "Estados Unidos", "poblacion": "60000"}, {"codigo": "ES", "nombre": "España", "poblacion": "450000"}, {"codigo": "FR", "nombre": "Francia", "poblacion": "600000"}, {"codigo": "CH", "nombre": "China", "poblacion": "1300000000"}, {"codigo": "MX", "nombre": "Mexico", "poblacion": "600000"}] }) ejercicio4.js:47:5

(5) [-] ejercicio4.js:50:5
0: Object { codigo: "EEUU", nombre: "Estados Unidos", poblacion: "60000" }
1: Object { codigo: "ES", nombre: "España", poblacion: "450000" }
2: Object { codigo: "FR", nombre: "Francia", poblacion: "600000" }
3: Object { codigo: "CH", nombre: "China", poblacion: "1300000000" }
4: Object { codigo: "MX", nombre: "Mexico", poblacion: "600000" }
length: 5
__proto__: Array []
```



6. Modifica el ejercicio anterior para que la funcionalidad se logre a través de las funciones de la API WebStorage.