

Desarrollo Web en Entorno Cliente

# **UD 02. Sintaxis Javascript ES6 - Actividades 02**

---

Actualizado Octubre 2020

## 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. Actividad 1	3
2. Actividad 2	3
3. Actividad 3	4
4. Actividad 4	5
5. Autores (en orden alfabético)	5

## UD02. SINTAXIS JAVASCRIPT ES6 - ACTIVIDADES 02

 **Importante 1:** no intentes copiar ejercicios ni tan siquiera “ver un poco” código de otros compañeros. Es el mayor error de quien empieza a programar, ya que luego no sabe resolver problemas por sí mismo y da una falsa sensación de aprendizaje.

 **Importante 2:** si en programación algo no sale a la primera... es totalmente normal. Es parte del aprendizaje. ¿Cómo crees que aprendieron los mejores programadores?

### 1. ACTIVIDAD 1

Realiza una aplicación web que compruebe si una solución de un Sudoku es correcta o no.

Una sugerencia de diseño para la aplicación web es que debe tener 9x9 campos de texto y un botón “Comprobar”. Recomendando un valor por defecto de un Sudoku válido para hacer pruebas

La explicación de las reglas del Sudoku las tenéis aquí <https://es.wikipedia.org/wiki/Sudoku>

Internamente el programa realizará la comprobación de si el Sudoku es o no correcto en una función definida como

**function esSudokuCorrecto(miArrayBi)**

que devolverá true si es correcto, false en caso contrario.

Esta función se podrá probar en [www.aprendeaprogramar.org](http://www.aprendeaprogramar.org)

### 2. ACTIVIDAD 2

Realiza una aplicación web que solicite una cadena de texto.

El programa debe decir cuántas veces ocurre cada uno de estos patrones sin distinguir mayúsculas y minúsculas: “00” “101”, “ABC”, “HO”.

Un carácter puede formar parte de más de un patrón encontrado. Por ejemplo:

En la cadena “000” el patrón “00” aparece dos veces (una empieza en la posición 0 y otra empieza en la posición 1).

Internamente el programa realizará la cuenta de patrones con una función definida como

**function numeroPatrones(texto)**

que devolverá un número entero con el número de patrones encontrados.

Esta función se podrá probar en [www.aprendeaprogramar.org](http://www.aprendeaprogramar.org)

### 3. ACTIVIDAD 3

Buscaminas:



Realiza una aplicación web que reciba en código mediante un array bidimensional (de longitud variable) un escenario de Buscaminas, donde haya un 0 donde no hay minas y un -1 donde si hay. Para cada casilla que no tenga una mina, diga cuantas minas adyacentes hay (en diagonal, horizontal y vertical).

Internamente el programa realizará las acciones con una función definida como

**function contandoMinas(miCampo)**

que devolverá un array bidimensional con el número de minas adyacentes en cada posición. Esta función se podrá probar en [www.aprendeaprogramar.org](http://www.aprendeaprogramar.org)

Más información de cómo funciona el Buscaminas

<https://es.wikipedia.org/wiki/Buscaminas>

#### Ejemplo Entrada

0 0 -1 0

0 -1 -1 0

#### Ejemplo Salida

1 3 -1 2

1 -1 -1 2

## 4. ACTIVIDAD 4

Crea un fichero HTML básico que cargue el siguiente código javascript:

```
const inicioDeJornada = "07:30";
const finalDeJornada = "17:45";
function agendarReunión(horaDeInicio,duracionEnMinutos) {
  //ToDo...
  return true;
}
console.assert(agendarReunión("7:00",15)==false, 'Fallo comprobando
agendarReunión("7:00",15)==false');
console.assert(agendarReunión("07:15",30)==false, 'Fallo comprobando
agendarReunión("07:15",30)==false');
console.assert(agendarReunión("7:30",30)==true, 'Fallo comprobando
agendarReunión("7:30",30)==true');
console.assert(agendarReunión("11:30",60)==true, 'Fallo comprobando
agendarReunión("11:30",60)==true');
console.assert(agendarReunión("17:00",45)==true, 'Fallo comprobando
agendarReunión("17:00",45)==true');
console.assert(agendarReunión("17:30",30)==false, 'Fallo comprobando
agendarReunión("17:30",30)==false');
console.assert(agendarReunión("18:00",15)==false, 'Fallo comprobando
agendarReunión("18:00",15)==false');
console.assert(agendarReunión("15:30",180)==false, 'Fallo comprobando
agendarReunión("15:30",180)==false');
console.log("Test Done!");
```

Rellena el contenido de la función "**agendarReunión**" de modo que devuelva true si la reunión ocurre dentro del horario laboral y false en caso contrario.

Idea una estrategia para poder trabajar con los distintos tipos básicos involucrados.

Los assert del final te indicarán si tu solución es correcta.

## 5. AUTORES (EN ORDEN ALFABÉTICO)

A continuación ofrecemos en orden alfabético el listado de autores que han hecho aportaciones a este documento:

- García Barea, Sergi