

# Examen Java Script

## Ejercicio 1 (4 puntos)

En el fichero **ejercicio1.js** se facilita la función **comprobarPatrones(elementoAComprobar, tipoComprobacion)** que comprueba si se cumple un patrón de los que tiene almacenados. Es necesario indicar que patrón que se va escoger para realizar la comprobación con las siguientes etiquetas:

- **numeroRegistro:** Si es un número de registro
- **nombre:** Si es un nombre
- **numeroSS:** Si es un número de la seguridad social
- **dirección:** Si es una dirección: direccion

Complétala siguiendo las siguientes indicaciones:

1. El número de registro debe cumplir:
  - a. Comenzar con 3 letras mayúsculas
  - b. Continuar con 3 números.
2. El nombre debe cumplir:
  - a. Sólo debe contener letras
  - b. Comenzar con una palabra que tenga la primera letra en mayúsculas y el resto en minúsculas y que representa al primer apellido.
  - c. Continúa con un espacio y una o dos letras en mayúsculas seguidas de un punto.

Ejemplo:

Fernández J. -> Sería correcto

López Ariadna M. -> No sería correcto

Castillo JL. -> Sería correcto

3. El número de la seguridad social debe cumplir:
  - a. Ser un número entre el 0 y el 999999999.
4. La dirección debe cumplir:
  - a. Comenzar por C/ o Av.
  - b. El nombre de la vía sólo debe contener letras.
  - c. El nombre de la vía puede tener mínimo una palabra o más separadas por espacios.
  - d. La primera palabra del nombre de la vía ha de comenzar por mayúsculas.
  - e. Debe contener el número del edificio separado del nombre de la vía por una coma.
  - f. El número del edificio sólo puede contener números.
  - g. No hay un máximo de número de edificio.

Ejemplo:

C/Federico lorca, 50 -> Es correcta

Av.Argentina, 123 -> Es correcta

C/lorenzo, 3 -> No es correcta

Av.Gimenez Losantos, 40 -> No es correcta

C/Almirante 45 -> No es correcta

C/Almirante -> No es correcta

5. La función devolverá el valor true si el patrón se cumple y false si el patrón no se cumple.

```
const comprobarPatrones = function (elementoAComprobar, tipoComprobacion) {  
  let patronNumeroRegistro; // Contiene el patrón para comprobar el número de registro  
  let patronNombreCompleto; // Contiene el patrón para comprobar el nombre  
  let patronNumeroSS; // Contiene el patrón para comprobar el número de la seguridad  
social  
  let patronDireccion; // Contiene el patrón para comprobar la dirección  
  
  let patrones = new Map([  
    ['numeroRegistro', patronNumeroRegistro],  
    ['nombre', patronNombreCompleto],  
    ['numeroSS', patronNumeroSS],  
    ['direccion', patronDireccion]  
  ]);  
  
  return //Devolver el valor true si se cumple el patrón y false si no se cumple  
}
```

### Calificaciones Ejercicio 1

Elemento a calificar	Puntuación
patronNumeroRegistro	0.5 puntos
patronNombreCompleto	1.25 puntos
patronNumeroSS	0.25 puntos
patronDireccion	1.5 puntos
salida de la función	0.5 puntos

## Ejercicio 2 (2 puntos)

Crea un objeto Paciente que guardará los datos de los pacientes de un hospital.

Este objeto contiene 4 propiedades que tendrán los siguientes nombres:

- **numeroRegistro**: Es el número de registro asignado al paciente,
- **nombreCompleto**: Es el nombre completo del paciente,
- **numeroSS**: Es el número de la seguridad social del paciente,
- **dirección**: Es la dirección del paciente,

Además, este objeto tiene los siguientes métodos:

- 4 métodos para modificar estas propiedades que se muestran a continuación:
  - **modificarNumeroRegistro(nuevoNumeroRegistro)**,
  - **modificarNombreCompleto(nuevonombreCompleto)**,
  - **modificarNumeroSS(nuevoNumeroSS)**,
  - **modificarDireccion(nuevaDireccion)**
- 4 métodos para mostrar los valores de las propiedades que se muestran a continuación:
  - **imprimirNumeroRegistro()**,
  - **imprimirNombreCompleto()**,
  - **imprimirNumeroSS()**,
  - **imprimirDirección()**
- 1 método para realizar las comprobaciones oportunas de los valores introducidos:
  - **comprobarPatrones(elementoAComprobar, tipoComprobacion)**: Este método es el programado en el ejercicio anterior.

El método **comprobarPatrones** se debe utilizar para comprobar cualquier dato que se introduzca en el objeto. Si el dato cumple el patrón se introducirá, si no lo cumple no se introducirá.

### Nota.

Si no habéis sido capaces de crear algún patrón poner ./ para que deje pasar todo.

**Extra:** (+0.5 puntos) Tratar de introducir los valores en las propiedades a través de los métodos modificar.

## Calificaciones Ejercicio 2

Elemento a calificar	Puntuación
crearObjeto	0.75 puntos
métodos modificar	1 puntos
métodos imprimir	0.25 puntos
introducir valores a las propiedades con métodos modificar	+0.5 puntos

## Ejercicio 3 (4 puntos)

Fichero a utilizar:

ejercicio3.js

En el fichero **ejercicio3.js** se nos facilita el siguiente mapa:

```
const registroPacientes = new Map([
  [`AAA024`, `Fernández M. (321790059) -> C/Recoletos, 50`],
  [`BCD827`, `Ruíz P. (100973253) -> C/Esquerdo izquierdo, 103`],
  [`YUN835`, `Benítez E. (154811767) -> Av.Argentina, 5`]
]);
```

Cada uno de los elementos del mapa se corresponde con los datos de un paciente del hospital siguiendo el siguiente formato:

**[numeroRegistro, `nombreCompleto (numeroSS) -> direccion`]**

Es decir, para el primer elemento del mapa tendremos:

**numeroRegistro:** AAA024,  
**nombreCompleto:** Fernández M.  
**numeroSS:** 321790059  
**direccion:** C/Recoletos, 50

Que serán los datos de un paciente.

Se pide crear un paciente por cada uno de los elementos de este mapa introduciendo los datos del paciente correctamente utilizando el objeto Paciente creado en el ejercicio previo.

### Nota.

Si no se ha creado correctamente el objeto **Paciente** del ejercicio anterior, extrae los datos y muéstralos en un mapa donde para cada elemento la clave sea `Paciente num\_paciente` y el valor un string con el formato mostrado a continuación:

Output:

```
{Paciente 1 =>
  `numeroRegistro: AAA024,
  nombreCompleto: Fernández M.,
  numeroSS: 321790059 ,
  direccion: C/Recoletos, 50`
Paciente 2 =>
  `numeroRegistro: BCD827 ,
  nombreCompleto: Ruíz P.,
  numeroSS: 100973253,
  direccion: C/Esquerdo izquierdo, 103`
Paciente 3 =>
  `numeroRegistro: YUN835,
  nombreCompleto: Benítez E.,
```

```
numeroSS: 154811767
direccion: Av.Argentina, 5`
}
```

### Calificaciones Ejercicio 3

Elemento a calificar	Puntuación
extraer datos numeroRegistro	0,5 puntos
extraer nombreCompleto	0,5 puntos
extraer numeroSS	0,5 puntos
extraer direccion	0,5 puntos
creación todos Pacientes	1 punto
Claridad y sencillez del código	1 punto

### Criterios de entrega

Los requisitos de entrega para los documentos del examen son:

- Tipo de archivo: archivo comprimido con todos los archivos generados durante el examen
- Nombre del archivo comprimido: Llinatge1Nom\_ExamenJS