

EVALUACIÓN	PARCIAL 1	GRUPOS	Mar. Matutino	FECHA	28/11/2023
MATERIA	PROGRAMACIÓN 1				
CARRERA	AP/ATI				
CONDICIONES	- Puntos: 50 puntos - Duración: 2 Horas - Sin material - Otros: Consultas solamente sobre interpretación de letra.				

El BPS desea generar una aplicación con la que centralizar los datos de pensiones otorgadas para poder llevar un seguimiento.

El estado tiene información disponible de los usuarios y pensiones para usar en nuestra aplicación.

```

class Persona{
    constructor(pCI, pNombre, pApellido) {
        This.CI = pCI;
        this.nombre = pNombre;
        this.apellido = pApellido;
    }
}

class Pension{
    constructor(pTipo, pMonto) {
        this.tipo= pTipo;
        this.monto = pMonto;
    }
}

class Sistema{
    constructor(){
        this.personas = [
            new Persona ("38512457","Juan","Perez"),
            new Persona ("52345127","Ana","Silva"),
            // ...
        ];
        this.pensiones= [
            new Pension("Vejez",12000),
            new Pension("Invalidez",10000),
            // ...
        ];
    }
}

```

- Crear la interfaz HTML (*solamente el contenido de dentro del body*) y la funcionalidad javascript para crear la interfaz para registrar el pago de una pensión a una persona. Se seleccionará la persona, la pensión a pagar, el mes de pago (1 al 12) y el año (2023 en adelante). Los datos de la persona y la pensión se deberán obtener de combos desplegables cargados dinámicamente en

JavaScript, el combo de la persona debe mostrar nombre y apellido y el de la pensión el nombre de la pensión. **En esta parte, no es necesario almacenar información, solo crear la interfaz.** (5pts.)

- b. Crear la/s clase/s y array/s necesarios para manejar la información a utilizar en la aplicación. Lo que se plantee dentro del constructor de la clase Sistema se debe aclarar con un comentario JavaScript, a modo de ejemplo se puede escribir lo siguiente: **//En Sistema (5pts.)**
- c. Crear un método de la clase Sistema **obtenerCantidadPagosDeUnaPension(pCi, pTipoPension)** que reciba como parámetro un número de ci y un tipo de pensión y devuelva la cantidad de pagos de esa pensión realizados a esa persona. (5 pts.)
- d. Crear la funcionalidad para, utilizando el HTML creado, hacer un registro de pago de pensión. Cada pago se almacenará en un array "pagos", con objetos Pago que contendrán el número de cédula (u objeto), el id de la pensión (u objeto), el mes y el año. Además, el pago tendrá un id que será un valor numérico único autoincremental. Se deberá validar que el mes esté entre 1 y 12, que el año sea mayor o igual a 2023 y que no exista ya un pago de ese tipo de pensión a esa persona en ese mes y ese año. En caso de error, mostrar un mensaje genérico en un párrafo. Recordar definir este párrafo en el HTML (15 pts.)
- e. Crear un método de la clase Sistema **obtenerCantidadPensionesPagasEnMesAnio(pMes, pAnio)** que reciba como parámetro un mes y un año y devuelva el total de pensiones pagas ese mes y ese año. (7 pts.)
- f. Crear un método de la clase Sistema **obtenerTotalPagoPension(pTipoPension)** que reciba como parámetro un tipo de pensión y devuelva el total del dinero pago de la misma. (8pts.)
- g. Crear una función que arme una tabla con todos los tipos de pensiones y muestre el monto individual de cada una y el monto total de pagos de esa pensión sin importar la fecha. (5pts.)

Pensión	Monto individual	Monto total pago
Vejez	12000	240.000
Invalidez	10000	320.000
....		...

Se valorará la utilización de nombres de variables y funciones descriptivos, la optimización de la solución y que se respeten las estructuras que se solicita utilizar. En caso de no poder resolver alguna de las partes se puede suponer como completada cuando se solicite. Identificar claramente qué parte del ejercicio se está resolviendo.