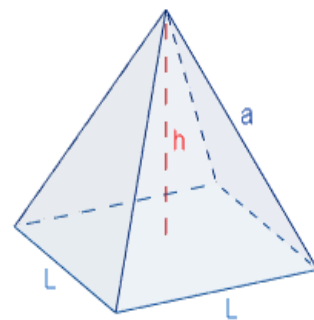


Boletín Ejercicios Tema 3

1. Ejercicio: Área de una pirámide

- a) Crea una función llamada **areaPiramide** que tenga dos argumentos de entrada, el lado de la pirámide **L** y la altura de la misma **h**. La función debe devolver un número con el área de la pirámide redondeada a cinco decimales. Si a la función le pasamos un lado o altura negativos debe lanzar un error con el texto "Los parámetros de entrada deben tener valores positivos"



El área de una pirámide viene dada por la fórmula:

$$A = L \cdot (L + \sqrt{4h^2 + L^2})$$

(Donde **L** es el lado y **h** la altura)

Debes utilizar todas las posibilidades que te ofrece la clase **Math** para implementar la fórmula.

La sintaxis de la función debe ser:

areaPiramide(lado, altura) □ Devuelve un número con el área redondeada a cinco decimales. Los argumentos **lado** y **altura** son números que representan el lado y la altura de la pirámide

Ejemplo de uso:

areaPiramide(6.8, 9) □ Debería devolver 177.08303

A efectos de testeo puedes usar la siguiente web:

<https://www.problemasyecuaciones.com/geometria3D/volumen/piramide/cuadrada/calculadora-area-volumen-formulas.html>

- b) Usando Jasmine debes testear la función del apartado anterior con los siguientes valores de entrada (lado y altura) y valores esperados (área). Los datos de la tabla deben estar encapsulados en alguna estructura (array, literal de objetos, etc.) Debes tener en cuenta que tu función **areaPiramide** hace un redondeo a cinco decimales, mientras que los valores esperados están redondeados a tres (debes usar el oportuno matcher de Jasmine):

VALORES DE ENTRADAS		VALOR ESPERADO
Lado (L)	Altura (h)	Área (redondeada a tres decimales)
6.8	9	177.083
7.1	9.4	193.092
7.4	9.8	209.793

Comprueba también con Jasmine:

- Que la función devuelve un dato de tipo number
- Que la función lanza un error cuando el lado es negativo
- Que la función lanza un error cuando la altura es negativa

2. Filtrar números primos mayores que 11.

Crea una función llamada **filtrarPrimosMayoresOnce** que tenga como argumento de entrada un array de números. Tu función debe devolver un array que contenga solo los números primos del array de entrada mayores que 11. Además, el array devuelto deberá estar ordenado de menor a mayor. Para el filtrado debes utilizar el método **filter** que ofrecen los objetos de tipo Array.

Se recuerda al alumno que los números primos son aquellos que solo son divisibles por uno y por ellos mismos (2, 3, 5, 7, 11, 13, 17, 19, 23, 29, etc.)

Sintaxis:

filtrarPrimosMayoresOnce (array) □ Debe devolver array conteniendo sólo primos mayores que 11 y ordenados de menor a mayor. El argumento array es un array de números.

Ejemplo de uso:

filtrarPrimosMayoresOnce ([6,11,18,43,8,5,45,53,9,7,24,23]) □ Debería devolver el array [23, 43, 53]

Haz un testeo usando Jasmine usando los siguientes datos:

```
let datos = [  
  { entrada: [6, 11, 18, 43, 8, 5, 45, 53, 9, 7, 24, 23], salida: [23, 43, 53] },  
  { entrada: [6, 5, 24, 47, 8, 11, 18, 41, 9, 2, 35, 19], salida: [19, 41, 47] },  
  { entrada: [4, 5, 45, 47, 6, 7, 27, 43, 10, 11, 35, 23], salida: [23, 43, 47] },  
  { entrada: [9, 11, 20, 23, 6, 3, 24, 17, 8, 5, 14, 47], salida: [17, 23, 47] },  
  { entrada: [9, 2, 45, 29, 8, 7, 18, 19, 6, 5, 12, 13], salida: [13, 19, 29] }  
];
```

Comprueba también desde Jasmine que la función devuelve un array.

3. Calcular el número de días entre dos fechas.

Crea una función llamada **numeroDiasFechas** que tenga como argumento de entrada dos argumentos de tipo string que representan una fecha en formato "dd/mm/yyyy" (por ejemplo, el 17 de abril de 1973 se representaría como "17/04/1973"). Tu función debe devolver el número de días transcurridos entre esas dos fechas. Si alguna de las fechas de entrada es incorrecta (p.e. "32/10/2024") debe lanzar un error con el texto "Alguna(s) de las fechas de entrada es incorrecta"

Sintaxis:

numeroDiasFechas (fechaDesde, fechaHasta) □ Devuelve el número de días transcurridos entre las dos fechas especificadas como argumentos. Los argumentos fechaDesde y fechaHasta son dos fechas dadas como string con el formato "dd/mm/yyyy" (por ejemplo el 17 de abril de 1973 se especificaría como "17/04/1973")

Ejemplo de uso:

numeroDiasFechas ("05/05/1975", "10/05/1975") □ Debería devolver el número 5 (han transcurrido 5 días entre esas dos fechas)

Haz un testeo con Jasmine usando los siguientes datos:

Fecha desde	Fecha hasta	Nº días transcurridos
"9/11/2021"	"9/11/2021"	0
"28/02/2020"	"1/3/2020"	2
"28/02/2021"	"1/3/2021"	1
"17/04/1973"	"14/11/1979"	2402

Comprueba también desde Jasmine:

- Que la función devuelve un número.
- Que lanza un error cuando una de las fechas de entradas no es correcta

4. Clase Reserva

Implementa la clase **Reserva**, que permita crear posteriormente reservas para un hotel indicando:

- El nombre del cliente que hace la reserva (usando el formato CSV separados por punto y coma "Apellido1;Apellido2; Nombre de Pila", por ejemplo "Franco;Salvatierra; Luis Fernando")
- El DNI , será un dato de tipo string que contendrá tanto los dígitos como la letra del DNI(por ejemplo "44958629E")
- La fecha de entrada, que se indicará como un dato de tipo String con el formato "dd/mm/yyyy". Por ejemplo si el cliente llega el 27 de octubre de 2021 se indicará como "27/10/2021"

- La fecha de salida, que se indicará como un dato de tipo String con el formato "dd/mm/yyyy". Por ejemplo si el cliente sale el 1 de noviembre de 2021 se indicará como "01/11/2021"

Por ejemplo, con los datos anteriores se usaría el constructor:

```
let reserva1= new Reserva("Franco;Salvatierra;Luis Fernando", "44958629E", "27/10/2021", "01/11/2021");
```

- Implementa una propiedad getter llamada **codigoCliente** que devuelva el código del cliente, el cual estará formado por la primera inicial del nombre de pila más el primer apellido más los tres últimos dígitos del DNI (y todo en mayúsculas). Por ejemplo, con los datos del supuesto debería devolver "LFRANCO629"
- Implementa una propiedad getter llamada **numeroDiasEstancia** que devuelva el número de días de la estancia, a partir de la fecha de entrada y de salida.
- Implementa un método de instancia llamado **modificarFechas** que permita modificar la fecha de entrada y la de salida (tendrá por tanto dos argumentos).

Por ejemplo, si la reserva será del 2 de noviembre de 2021 al 7 de noviembre de 2021 lo haremos mediante:

```
reserva1.modificarFechas("2/11/2021","7/11/2021");
```

Hay que tener en cuenta las siguientes consideraciones:

- La fecha de salida debe ser posterior a la de entrada (en caso contrario se lanzará el error "Fecha de salida debe ser posterior a la de entrada")
 - Entre la fecha de entrada y salida debe transcurrir al menos un día (en caso contrario se lanzará el error "Estancia mínima debe ser de un día")
- Implementa un método de instancia llamado **costeEstancia()** que devuelva el coste de la estancia, teniendo en cuenta que el coste diario de lunes a viernes son 24€, los sábados 36€ y los domingos 43€.
- e) Somete tu clase a las especificaciones de Jasmine proporcionadas por el profesor.
- Dicho test de Jasmine realiza internamente las siguientes operaciones:
- Crea una reserva para un cliente llamado Juan Antonio García Ortiz, con DNI 44958625A, que llega al hotel el 27 de febrero de 2020 y se va el 3 de marzo de ese mismo año.
 - Comprobar que la propiedad getter **codigoCliente** devuelve JGARCÍA625
 - Comprobar que el número de días de estancia (lo devuelto por la propiedad getter **numeroDiasEstancia**) sea igual a 5
 - Comprobar que el coste esperado de la estancia (el valor devuelto al invocar al método **costeEstancia()**) sea de 151
 - Comprobar que las propiedades internas **fechaEntrada** y **fechaSalida** no se vean modificadas al invocar al método **costeEstancia()**
 - Comprobar que al modificar la fecha de entrada a 28 de febrero de 2020 y la de salida a 1 de marzo de 2020 se haya realizado correctamente (esto se hace comprobando que la propiedad getter devuelva 2).

- Comprobar que si se intenta modificar las fechas de la estancia y la de salida es anterior a la de entrada se lance un error con el texto "Fecha de salida debe ser posterior a la de entrada"
- Comprobar que si se intenta modificar las fechas de la estancia y entre la entrada y salida no han transcurrido al menos un día se lance el error "Estancia mínima debe ser un día"