

# *PROGRAMACIÓN FRONTEND Y BACKEND*

## **Java 8**

### **Funciones (EJERCICIOS)**

**Edición: Septiembre 2019**

1. Crea una función que te dibuje una fila de 20 asteriscos usando un bucle. Llámala desde el main 3 veces.
2. Crea una función que reciba una cadena y 2 números enteros por parámetro y te imprima si la longitud de la cadena está entre los 2 números o no. Llámala al menos 2 veces, una en la que sí esté en el rango, y otra en la que no.
3. Crea una función que reciba un carácter y un número, dibuja un cuadrado usando ese carácter cuyo ancho y alto sea igual al número recibido. Prueba la función.
4. Crea una función con 2 parámetros. El primero será una cadena y representa al nombre de una persona, mientras que el segundo contendrá al resto de parámetros recibidos (...), que serán cadenas con los trabajos que la persona ha realizado.

Imprime por consola el nombre de la persona seguido de la lista de trabajos realizados por la misma. Si no hay ningún trabajo, indica que no ha trabajado nunca. Prueba la función varias veces, al menos una de ellas no le pases ningún trabajo.

5. Crea una función que reciba un número y devuelva un booleano indicando si el número recibido es primo o no. Llama a la función varias veces y muestra el resultado devuelto. Recuerda que un número primo es aquel que solo es divisible (resto == 0) entre 1 y él mismo.
6. Crea una función que reciba un array de números y devuelva su media.
7. Crea una función que reciba un array de cadenas por parámetro y devuelva una de las cadenas al azar. **Pista:** puedes generar un número aleatorio entre 0 y la última posición del array usando **new Random().nextInt(array.length)**.

A continuación pide al usuario que adivine la palabra seleccionada (tiene 3 intentos, usa un bucle). Si la acierta felicítale, y si no, dile cual era la correcta.

8. Crea una función que reciba un número indeterminado de parámetros enteros y devuelva el máximo. Ten en cuenta que si recibes cero números, debes devolver cero también.
9. Crea una función que reciba una cadena y una subcadena por parámetro y devuelva cuantas veces aparece la subcadena dentro de la cadena. Para ello debes saber que como segundo parámetro a **indexOf** le puedes pasar la posición a partir de la cual buscar (empezaríamos en cero). Cuando te devuelva -1 es que ya no la ha encontrado (utiliza un bucle).

Por ejemplo, si recibimos "cocinando cocos con chocolate" y "co" devolveremos el número 4.

10. Crea una función que reciba una cadena, un separador (char) y un número n. La función insertará el separador dentro de la cadena cada n caracteres (ten en cuenta que al insertar el separador, la siguiente posición se incrementa en 1).

Por ejemplo, si recibimos "Hay una mosca en mi sopa", '\*' y 3, devolveremos "Hay\* un\*a m\*osc\*a e\*n m\*i s\*opa\*"

11. Crea una función que reciba una cadena con una serie de números separados por punto y coma ";", y devuelva la media de dichos números.
12. Crea una función que reciba un array de cadenas y te lo devuelva ordenado alfabéticamente de la 'z' a la 'a'.

13. Crea una función que reciba un array de números y un número. y te devuelva cuantas veces aparece ese número en el array.

14. Crea una función que a partir de una fecha (LocalDateTime) que reciba por parámetro, te devuelva una cadena con la fecha en el siguiente formato:

Si recibimos la siguiente fecha "04/07/2019", devolvería "Jueves, 4 de julio de 2019".

15. Crea una función que reciba una fecha en formato dd-mm-yyyy, conviértela a fecha (LocalDate) utilizando la función subString o split para extraer el día, mes y año, o usa el DateFormatter, súmale 2 años, 3 meses y 5 días, y muestra la fecha resultante.