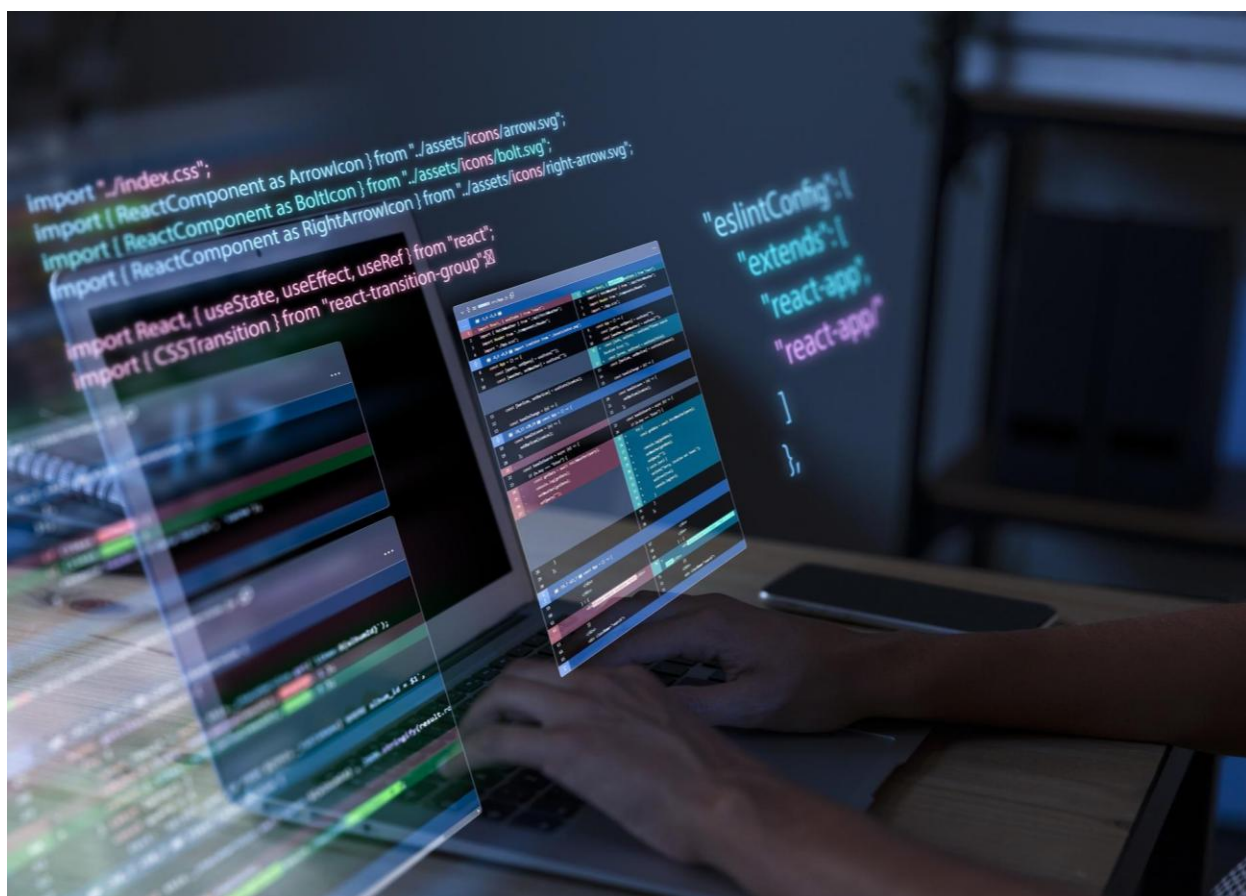


# Ejercicios

## Funciones (III)

---



**Autor:** Marcela Martín

**Fecha:** noviembre 2025

1. Escribe un programa que solicite una cadena al usuario y comprobar si contiene al menos una ocurrencia de una sílaba de dos caracteres, por ejemplo 'ma'.

- Utilizar una función para buscar la sílaba (buscaSilaba()) que devuelve true/false, a la que se le pasa como parámetro la cadena de texto y sílaba, siendo esta un String de dos posiciones.

Añade las validaciones que consideres oportunas.

2. Escribe una función que decida si dos números enteros positivos son amigos. Dos números a y b son amigos si la suma de los divisores propios (distintos de él mismo) de a es igual a b. Y viceversa. Para probar se pueden usar los números 220 y 284, que son amigos. Añade las validaciones que consideres oportunas.

3. Diseña un programa que valide si una contraseña proporcionada por el usuario cumple con varios criterios de seguridad:

- Mínimo 8 caracteres.
- Al menos una mayúscula.
- Al menos una minúscula.
- Al menos un dígito numérico.

Implementa las siguientes funciones para resolver el ejercicio:

- public static boolean tieneLongitudMinima(String password, int min): Verifica si la longitud de la contraseña es mayor o igual a min.
- public static boolean contieneMayuscula(String password): Recorre la contraseña con un bucle y usa un condicional para verificar si existe al menos una mayúscula.
- public static boolean contieneMinuscula(String password): Similar a la anterior, verifica si existe al menos una minúscula.
- public static boolean contieneDigito(String password): Similar a las anteriores, verifica si existe al menos un dígito.
- public static boolean esContrasenaValida(String password): Llama a las cuatro funciones anteriores y retorna true solo si todas retornan true.
- public static void iniciarValidacion(): Pide la contraseña, llama a esContrasenaValida(), e imprime el resultado usando un condicional.

Utiliza la clase Character y los métodos 'isLowerCase(), isUpperCase() e isDigit()' para las diferentes validaciones.

4. Crea un programa que analice una cadena de texto para limpiar caracteres no deseados y proporcionar estadísticas básicas sobre su contenido.

Implementa las siguientes funciones para resolverlo:

- `public static String limpiarTexto(String texto)`: Recibe una cadena. Utiliza un bucle y condicionales para crear y retornar una nueva cadena que solo contenga letras, números y espacios, eliminando todos los caracteres especiales (puntuación, símbolos, etc.).
- `public static int contarPalabras(String textoLimpio)`: Recibe la cadena ya limpia. Retorna el número de palabras que contiene (puedes asumir que las palabras están separadas por un espacio simple).
- `public static int contarVocales(String textoLimpio)`: Recibe la cadena limpia. Utiliza un bucle y condicionales (o switch) para contar y retornar el número total de vocales (a, e, i, o, u), ignorando mayúsculas/minúsculas.
- `public static void analizarTexto(String textoOriginal)`:
  - o Llama a `limpiarTexto()` para obtener la versión limpia.
  - o Llama a `contarPalabras()` y `contarVocales()` con la cadena limpia.
  - o Imprime el texto original, el texto limpio, el conteo de palabras y el conteo de vocales.

5. Diseña un programa que tome una palabra o frase y determine si es un palíndromo (se lee igual de izquierda a derecha que de derecha a izquierda), además de mostrar la cadena original y su versión invertida.

Implementa las siguientes funciones para resolverlo:

- `public static String prepararCadena(String original)`: Recibe la cadena, la convierte a minúsculas y elimina todos los espacios en blanco (sin utilizar métodos de String que los eliminen). Retorna la cadena preparada.
- `public static String invertirCadena(String preparada)`: Recibe la cadena preparada y retorna la cadena invertida (Sin utilizar métodos de String)

- `public static boolean esPalindromo(String preparada, String invertida)`: Recibe la cadena preparada (limpia) y la invertida. Retorna true/false dependiendo de si los parámetros son iguales o no.
- `public static void procesarPalindromo(String entradaUsuario)`:
  - Llama a `prepararCadena()` con la entrada del usuario.
  - Llama a `invertirCadena()` con la cadena preparada.
  - Llama a `esPalindromo()` con ambas cadenas procesadas.
  - Imprime la entrada original, la cadena invertida y el resultado final