

Ejercicios Expresiones Regulares

[Descargar estos ejercicios](#)

Índice

- [Ejercicio 1](#)
- [Ejercicio 2](#)
- [Ejercicio 3](#)
- [Ejercicio 4](#)
- [Ejercicio 5](#)
- [Ejercicio 6](#)

Ejercicio 1

Crea un programa que reciba una matrícula de coche y valide si cumple con el formato actual de España: 4 números seguidos de 3 letras.
Las letras no pueden incluir vocales.

Ejemplo válido: 1234BCD

Ejemplo inválido: 12AB345, 1234AEI

Ejercicio 2

Diseña una aplicación que evalúe si una contraseña es segura.
Para ser válida debe contener:

- Al menos 8 caracteres
- Al menos una letra mayúscula
- Al menos una letra minúscula
- Al menos un número
- Al menos un carácter especial (como @, #, !, etc.)

Ejercicio 3

Dado un párrafo o texto largo, utiliza una expresión regular para **extraer todos** los correos electrónicos que contenga. Los correos deben tener el formato clásico:

`usuario@dominio.extensión`

Ejercicio 4

Crea un validador de códigos postales Españoles. Debe aceptar únicamente códigos de 5 dígitos, donde los dos primeros estén entre 01 y 52.

Ejemplo válido: 46010, 28013

Ejemplo inválido: 99000, 523456

Ejercicio 5

Diseña un programa que verifique si una cadena representa un número de tarjeta de crédito válido en formato típico. Debe aceptar tarjetas con o sin espacios, agrupadas en bloques de 4 dígitos. El patrón debe:

- Aceptar exactamente 16 dígitos
- Permitir espacios opcionales entre cada 4 dígitos
- Rechazar si hay letras o si los bloques son incorrectos

Ejemplos válidos:

1234 5678 9012 3456

1234567890123456

Ejercicio 6

Crea un programa que valide nombres de usuario para una web. Las reglas son:

- Deben empezar con una letra
- Pueden contener letras, números, guiones bajos o puntos

- Longitud mínima: 5 caracteres
- Longitud máxima: 15 caracteres
- No pueden terminar en punto ni guion bajo

Ejemplos válidos: `juan_87`, `maria.rosa`, `Carlos1990`

Ejemplos inválidos: `123mario`, `luis_`, `.pepe`, `ana-too-long-user-name`