

## Ejercicios clase 3

### Ejercicio 1 – Jugando con String

Pedirle al usuario que ingrese un texto por teclado.

El programa debe implementar distintos comportamientos para:

- Mostrar la cantidad de caracteres que tiene el texto.
- Mostrar la primera mitad de la cadena.
- Mostrar el último carácter.
- Mostrar el texto en forma inversa.
- Mostrar cada carácter separado por un guión.
- Verificar si el texto contiene la palabra “hola” (sin importar mayúsculas o minúsculas).

### Ejercicio 2 – Comprimir cadenas (desafío)

Pedirle al usuario que ingrese un texto y comprimirlo de la siguiente manera: contar cuántas veces aparecen consecutivamente las mismas letras, sin distinguir entre mayúsculas y minúsculas.

```
Entrada:  aAaBbcccccaab  
Salida:   a3b2c4a2b1
```

### Ejercicio 3 – Cafetera

Queremos simular una cafetera para probar distintas operaciones.

La cafetera debe tener dos datos:

- la capacidad máxima que soporta en mililitros,
- y la cantidad actual de café que tiene en ese momento.

Necesitamos:

- Que se pueda crear una cafetera de 3 maneras:
  1. Vacía de 1000 ml.
  2. Vacía, pero con una capacidad máxima indicada por el usuario.
  3. Con capacidad máxima y cantidad inicial. (Si se pasa, que se llene hasta el máximo).
- Que tenga **métodos** para:
  - **Llenar** la cafetera al máximo.
  - **Servir una taza** de cierto tamaño. (Si no alcanza, servir lo que quede).
  - **Vaciar** la cafetera.
  - **Agregar café** en ml. (Si sobra porque se pasa, devolver cuánto no entró).
- Que siempre se controle que la cafetera nunca tenga menos de 0 ml ni más de su capacidad máxima.
- Que la capacidad mínima permitida para una cafetera sea de 250 ml.

Finalmente, crear un programa de prueba:

- Primero crear una cafetera vacía por defecto.
- Después una cafetera de medio litro.
- Y otra de 750 ml con 500 ml iniciales.
- Probar todas las funciones de la cafetera con un único método `testearCafetera(Cafetera c)`.

### Ejercicio 4 – Liga de la Justicia Informática

Queremos crear un programa que simule una **batalla de superhéroes**.

La clase se debe llamar `SuperHeroe` y debe tener los siguientes datos:

- Nombre (texto).
- Fuerza (número entre 0 y 100).
- Resistencia (número entre 0 y 100).
- Superpoderes (número entre 0 y 100).

Reglas importantes:

- Cada vez que se asigne un valor a fuerza, resistencia o superpoderes, el número debe estar siempre en el rango 0 a 100.
  - Si se intenta poner un valor menor a 0, el valor debe quedar en 0.
  - Si se intenta poner un valor mayor a 100, el valor debe quedar en 100.
- El **constructor** debe recibir todos los valores y usará los setters (privados) para aplicar estas validaciones.

- Crear el método público `mostrarInfo()` que imprima en consola todos los datos del superhéroe de forma clara.

Además, la clase debe tener el método público `competir(SuperHeroe otro)`. Este método comparará al superhéroe actual contra el que se recibe como parámetro y devolverá:

- **"TRIUNFO"**: si el superhéroe gana en al menos 2 de las 3 categorías (fuerza, resistencia, superpoderes).
- **"EMPATE"** : si están iguales en al menos 2 de las 3 categorías.
- **"DERROTA"**: en cualquier otro caso.

Por último, crear una clase `PruebaLiga` con un `main()` que haga lo siguiente:

1. Crear un superhéroe llamado "Batman" con fuerza 90, resistencia 70 y superpoderes 0.
2. Crear un superhéroe llamado "Superman" con fuerza 95, resistencia 60 y superpoderes 70.
3. Mostrar la información de ambos superhéroes con el método `mostrarInfo()`.
4. Hacer que Batman compita contra Superman y mostrar el resultado (debería ser "DERROTA").
5. Hacer que Superman compita contra Batman y mostrar el resultado (debería ser "TRIUNFO").