

UNIDAD OCHO

ACTIVIDADES (cadenas)

1. Realizar un estudio de interfaz e implementar una clase *Cadena* con las funcionalidades vistas en clase de la clase *String* de *Java* y cualesquiera otras que consideres interesantes. Por ejemplo:
 - a. Invertir una cadena
 - b. Transformar una cadena en ella misma pero cambiando por mayúscula su primer carácter.
 - c. Eliminar todos los espacios en blanco de una cadena de caracteres.
 - d. Formar los plurales de palabras (Añadir **s** si acaba en vocal y **es** si acaba en consonante.)

2. Escriba un programa que pida al usuario una palabra de *n* caracteres. Calcule una matriz que contenga en cada fila la palabra rotada hacia la izquierda un lugar.

Por ejemplo, si el usuario introduce la palabra 'ROSA', el programa calculará e imprimirá por pantalla la matriz:

```
R O S A
A R O S
S A R O
```

3. Realizar un subprograma para comprobar si un número es o no capicúa.
4. Realizar un programa que calcule los números *omirps* comprendidos entre 2 y un número leído por teclado. Decimos que un número es *omirp* si cumple estas tres condiciones:
 - a. Ser primo.
 - b. No ser capicúa.
 - c. El mismo número escrito al revés, también debe ser primo.

5. Verificar si una cadena de la clase *String*, es un NIF correcto o no. Si lo es, se mostrará por consola su parte numérica; si no lo es se mostrará "NIF no valido".

Se tendrá en cuenta que un NIF tienen 9 caracteres, los 8 primeros dígitos y, a continuación, una letra (no importa que sea mayúscula o minúscula).

RECOMENDACIONES:

- a. Usar el método `length()` de `java.lang.String` para conocer el número de caracteres de una cadena de texto.
- b. Usar el método estático `isLetter(char c)` de `java.lang.Character` para comprobar que un carácter es una letra.
- c. Usar el método estático `isDigit(char c)` de `java.lang.Character` para comprobar que un carácter es un dígito.
- d. Usar el método `substring(int inicio, int fin)` de `java.lang.String` para obtener la parte numérica del NIF.