

# UD02.EXAME Práctico REC

DAM1-Programación 2023-24  
2023/11/29

1. Estadística de estaturas (3)	2
2. Suma de primos (3)	2
3. Soldados en Formación (4)	3

- Puedes utilizar apuntes y materiales que consideres pero deberás realizar los programas individualmente. En caso contrario se retirará el examen.
- Realiza programas bien estructurados, legibles, con comentarios, líneas en blanco, identificadores adecuados, etc.
- Cuida la interacción con el usuario, presentando la información de forma clara y ordenada.
- Utiliza los casos de prueba de ejemplo para probar tu programa y definir la salida por pantalla.

- Crea un **paquete** de nombre **examenrecud02** donde agrupar los ficheros de código fuente. Si es necesario crea un nuevo proyecto/carpeta Java.
- Crea dentro del paquete ficheros **.java** por cada ejercicio con el nombre indicado.
- Incluye al inicio de cada programa un **comentario con tu nombre y apellidos**.
- **Comprime y entrega la carpeta del paquete en el Aula Virtual al terminar.**
- **Tiempo estimado:** 2 horas

# 1. Estadística de estaturas (3)

## EstadisticaEstaturas.java

Implementa una aplicación que realice una estadística de la estatura del alumnado de un aula. Se introducirán datos por teclado de estaturas en centímetros hasta que una entrada sea negativa, y se mostrará al terminar:

- el número total de alumnos
- la media de estatura (en metros con hasta 2 decimales)
- la altura más alta
- la altura más baja

La aplicación controlará que no se introduzcan letras u otros caracteres no numéricos, permitiendo al usuario repetir la entrada hasta que sea correcta y sin contabilizar las entradas erróneas.

# 2. Suma de primos (3)

## SumaPrimosEntre.java

Implementa una función que devuelva la suma de los números primos comprendidos entre dos números enteros positivos, min y max, excluyendo éstos.

```
static int sumaPrimosEntre(int min, int max);
```

Si min es mayor que max o no hay números primos en el rango la función devolverá cero.

Puedes utilizar el siguiente código para probar tu función:

```
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.*;

public class SumaPrimosEntreTest {

    @Test
    void testSumaPrimosEntre() {
        assertEquals(0, SumaPrimosEntre.sumaPrimosEntre(1, 1));
        assertEquals(0, SumaPrimosEntre.sumaPrimosEntre(1, 2));
        assertEquals(0, SumaPrimosEntre.sumaPrimosEntre(2, 1));
        assertEquals(0, SumaPrimosEntre.sumaPrimosEntre(2, 2));
        assertEquals(0, SumaPrimosEntre.sumaPrimosEntre(30, 20));
        assertEquals(5, SumaPrimosEntre.sumaPrimosEntre(1, 5));
        assertEquals(17, SumaPrimosEntre.sumaPrimosEntre(0, 10));
        assertEquals(17, SumaPrimosEntre.sumaPrimosEntre(1, 10));
        assertEquals(31, SumaPrimosEntre.sumaPrimosEntre(5, 15));
        assertEquals(52, SumaPrimosEntre.sumaPrimosEntre(20, 30));
        assertEquals(77, SumaPrimosEntre.sumaPrimosEntre(1, 20));
        assertEquals(1060, SumaPrimosEntre.sumaPrimosEntre(1, 100));
        assertEquals(3167, SumaPrimosEntre.sumaPrimosEntre(100, 200));
        assertEquals(76127, SumaPrimosEntre.sumaPrimosEntre(1, 1000));
        assertEquals(200923, SumaPrimosEntre.sumaPrimosEntre(1000, 2000));
    }
}
```

### 3. Soldados en Formación (4)

#### FormacionSoldados.java

Un general quiere disponer sus tropas en formaciones cuadradas, es decir, de modo que haya tantos soldados por fila como por columna. Para ello necesita un programa que, dado un número de soldados, forme con ellos el cuadrado más grande posible. Con los soldados restantes intentará formar otro cuadrado, más pequeño pero también lo más grande posible. y así sucesivamente hasta que no queden suficientes soldados para formar un cuadrado, es decir, menos de 4, en cuyo caso les ordenará formar en línea.

El programa deberá aceptar un número de soldados y mostrar por pantalla el tamaño de las formaciones resultantes.

El programa deberá también dibujar las formaciones representando cada soldado por un símbolo, por ejemplo un asterisco (\*), indicando antes de cada formación el número de soldados que contiene.

Ejemplos:

<pre>Ingrese el número de soldados: 7 Formación cuadrada: 4 soldados ** **  Formación en línea: 3 soldados ***</pre>	<pre>Ingrese el número de soldados: 15 Formación cuadrada: 9 soldados *** *** ***  Formación cuadrada: 4 soldados ** **  Formación en línea: 2 soldados **</pre>
<pre>Ingrese el número de soldados: 70 Formación cuadrada: 64 soldados ***** ***** ***** ***** ***** ***** *****  Formación cuadrada: 4 soldados ** **  Formación en línea: 2 soldados **</pre>	<pre>Ingrese el número de soldados: 72 Formación cuadrada: 64 soldados ***** ***** ***** ***** ***** ***** *****  Formación cuadrada: 4 soldados ** **  Formación cuadrada: 4 soldados ** **</pre>

Ejercicio inspirado en: [Escudos del ejército romano - ¡Acepta el reto!](#)

Idea para ampliar: [Unidad militar - Wikipedia, la enciclopedia libre](#)