



Programación con Java

TAREA 06

Índice.....	1
Introducción.....	2
Métodos y arrays.....	3
Programas.....	3
Ejercicio 01:.....	3
Ejercicio 02:.....	3
Ejercicio 03:.....	4
Ejercicio 04:.....	4
Ejercicio 05:.....	5
Ejercicio 06:.....	5
Ejercicio 07:.....	6
Ejercicio 08:.....	7
Ejercicio 09:.....	8
Ejercicio 10:.....	10
Ejercicio 11:.....	11
Ejercicio 12:.....	12
Webgrafía.....	13

Con las prácticas que haremos aprenderemos a crear métodos y arrays. Los métodos se utilizan para asignarles unas instrucciones concretas donde habrá un valor que se asignará en otro método (normalmente el main), cuando llamemos al método creado por nosotros, al que le hemos asignado unos parámetros que necesita del main [Ejemplo para aclarar la explicación: metodo(**op1, op2**)], tendremos que darle estos parámetros para que haga las instrucciones que tiene definidas y nos dé un resultado. En cambio, los arrays son colecciones de datos donde guarda varios datos del mismo tipo, en los que podremos buscar valores o índices concretos, entre más funciones que veremos.

Métodos y arrays

Programas

Ejercicio 01:

Para empezar a practicar con los métodos, tendremos que hacer un programa para calcular el área de un cuadrado, un triángulo y un círculo; pero tendremos que hacerlo definiendo un método para calcular cada tipo de área.

main del programa y primer
método para mostrar un texto según la figura (con el área)

métodos para definir el área de las figuras

```
import javax.swing.JOptionPane;

public class Ejercicio01 {

    Run | Debug
    public static void main(String[] args) {
        String figuraM = JOptionPane.showInputDialog("¿De que figura quieres calcular su área?");
        String figura = figuraM.toLowerCase();
        forma(figura);
    }

    public static void forma(String figur) {
        if (figur.equals("cuadrado")) {
            String costado = JOptionPane.showInputDialog("Inserte el tamaño de un lado de tu cuadrado");
            JOptionPane.showMessageDialog(null, "El área del caudrado es : " + Cuadrado(costado));
        } else if (figur.equals("triangulo")) {
            String base = JOptionPane.showInputDialog("Inserte el tamaño de la base del triangulo");
            String altura = JOptionPane.showInputDialog("Inserte la altura del triangulo");
            JOptionPane.showMessageDialog(null, "El área del caudrado es : " + Triangulo(base, altura));
        } else if (figur.equals("circulo")) {
            String radio = JOptionPane.showInputDialog("Inserte el tamaño del radio");
            JOptionPane.showMessageDialog(null, "El área del caudrado es : " + Circulo(radio));
        } else {
            JOptionPane.showMessageDialog(null, "¿Has escrito bien el nombre de la figura? \n Las opciones"+
                " son: \n - Cuadrado \n - Triangulo \n - Circulo");
        }
    }
}
```

```
public static int Cuadrado(String lado) {
    int lado1 = Integer.parseInt(lado);
    int AreaH = lado1 * lado1;
    return AreaH;
}

public static double Triangulo(String bas, String altur) {
    int base = Integer.parseInt(bas);
    int altura = Integer.parseInt(altur);
    double AreaT = (base * altura) / 2;
    return AreaT;
}

public static double Circulo(String radi) {
    final double pi = 3.14;
    int radio = Integer.parseInt(radi);
    double AreaC = (Math.pow(radio, 2) * pi);
    return AreaC;
}
```

Ejercicio 02:

En el segundo programa hemos tenido que generar una cantidad de números aleatorios según quiera el usuario, igual que el rango máximo de cada.

```
import javax.swing.JOptionPane;

public class Ejercicio02 {

    Run | Debug
    public static void main(String[] args) {
        String can = JOptionPane.showInputDialog(message:"Cual es la cantidad de numeros aleatorios que se generaran");
        int canti = Integer.parseInt(can);
        generador(canti);
    }
}
```

main del programa

```
public static void generador(int cantidad) {
    int numero = 0;
    StringBuilder juntas = new StringBuilder(numero);
    for (int i = 0; i < cantidad; i++) {
        String limi = JOptionPane.showInputDialog(message:"El rango de numeros aleatorios ira desde el 1 hasta: ");
        int limite = Integer.parseInt(limi);
        numero = (int) Math.round(Math.random() * limite);
        if (numero == 0) {
            numero += 1;
        }
        juntas.append(numero+" ");
    }
    JOptionPane.showMessageDialog(parentComponent:null, juntas);
}
```

métodos para definir el área de las figuras

Ejercicio 03:

Para el ejercicio03, crearemos de nuevo un programa que nos indicará si es primo o no a través de un método booleano.

```
public class Ejercicio03 {  
    Run | Debug  
    public static void main(String[] args) {  
        String numero = JOptionPane.showInputDialog(message:"Ingresa un numero para saber si es primo:");  
        int num = Integer.parseInt(numero);  
  
        if (true == esprimo(num)) {  
            JOptionPane.showMessageDialog(parentComponent:null, "¡Exacto! El número " + num + " es primo ");  
        } else {  
            JOptionPane.showMessageDialog(parentComponent:null, "Que lástima, el número " + num + " no es primo ");  
        }  
    }  
  
    public static boolean esprimo(int num) {  
        if (num <= 1) {  
            return false;  
        }  
        for (int produc = (num - 1); produc >= 2; produc--) {  
            if (num % produc == 0) {  
                return false;  
            }  
        }  
        return true;  
    }  
}
```

Ejercicio 04:

En este programa tendremos que programar una calculadora de números factoriales con un método a parte.

```
public class Ejercicio04 {  
    Run | Debug  
    public static void main(String[] args) {  
        String numero = JOptionPane.showInputDialog(message:"Ingresa un numero para saber su factorial");  
        int num = Integer.parseInt(numero);  
        int resultado = factorial(num);  
  
        JOptionPane.showMessageDialog(parentComponent:null, "El factorial de " + num + " es : " + resultado);  
    }  
  
    public static int factorial(int fac) {  
        int resultado = 1;  
        for (int operador = fac; operador >= 1; operador--) {  
            resultado *= operador;  
        }  
        return resultado;  
    }  
}
```

Ejercicio 05:

Parecido al exterior, en este haremos una “calculadora” pero para convertir un número a base binaria.

```
public class Ejercicio05 {
    Run | Debug
    public static void main(String[] args) {
        String decima = JOptionPane.showInputDialog(message:"Ingresa un numero para conocerlo en base binaria");
        int decimal = Integer.parseInt(decima);
        System.out.println("El numero "+decimal+" en base binaria es:");
        int binario[] = rellenarbin(decimal);
        escribirbin(binario);
    }
}
```

main

```
public static int[] rellenarbin(int numero) {
    int restos[] = new int[numero];
    for (int i = 0; i < restos.length; i++) {
        restos[i] = numero % 2;
        numero /= 2;
    }
    return restos;
}

public static int[] escribirbin(int list[]) {
    for (int i = (list.length - 1); i >= 0; i--) {
        System.out.print(list[i]+" ");
    }
    return list;
}
```

métodos

Ejercicio 06:

En este ejercicio tenemos que contar el número qbe dígitos que tiene, pero tenemos que “controlarlo” por lo que, primero reemplazamos la posibilidad de que contenga caracteres no numéricos después, con un nuevo método boolean, verificamos que el número ingresado sea positivo y al ser mayor nos contará los dígitos que contiene.

```
public class Ejercicio06 {
    Run | Debug
    public static void main(String[] args) {
        String numero = JOptionPane.showInputDialog(message:"Ingresa un numero para saber su numero de cifras");
        String sinletras = numero.replaceAll(regex:"[\\D]", replacement:"");
        if (conteocifras(sinletras) == false) {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Ingresa un número superior a 0");
        } else {
            JOptionPane.showMessageDialog(parentComponent:null,
                "El numero " + sinletras + " contiene " + (sinletras.length()) + " dígitos");
        }
    }

    public static boolean conteocifras(String num) {
        int numero = Integer.parseInt(num);
        if (numero <= 0) {
            return false;
        } else {
            return true;
        }
    }
}
```

Ejercicio 07:

Para la ejecución del programa 7, haremos un conversor de euros a otra moneda, en nuestro caso: “*dólares, yenes o libras*”. Entonces lo que haremos será crear un método para convertir los euros en la moneda que insertemos, para ello definiremos como variables finales, para que no tengan posibilidad de variar, a los *divisores* y dentro del “*switch*” haremos la operación y la mostraremos para así en el main solo nombrar al método.

```
public class Ejercicio07 {  
    Run | Debug  
    public static void main(String[] args) {  
        String euro = JOptionPane.showInputDialog(message:"¿Cual es la cantidad de euros que quiere convertir?");  
        double euros = Double.parseDouble(euro);  
        String moneda = JOptionPane  
            .showInputDialog(message:"¿A que moneda quiere convertirlo?\n - Dolares\n - Libras\n - Yenes");  
        moneda = moneda.toLowerCase();  
        conversor(euros, moneda);  
    }  
}
```

main

```
public static void conversor(double euros, String moneda) {  
    double dolares = 1.07;  
    double yenes = 161.48;  
    double libras = 0.85;  
  
    switch (moneda) {  
        case "dolares":  
            JOptionPane.showMessageDialog(parentComponent:null,  
                "La cantidad de " + euros + "€ en dolares es: " + euros / dolares + "$");  
            break;  
  
        case "yenes":  
            JOptionPane.showMessageDialog(parentComponent:null,  
                "La cantidad de " + euros + "€ en yenes es: " + euros / yenes + "¥");  
            break;  
  
        case "libras":  
            JOptionPane.showMessageDialog(parentComponent:null,  
                "La cantidad de " + euros + "€ en libras es: " + euros / libras + "£");  
            break;  
        default:  
            JOptionPane.showMessageDialog(parentComponent:null, message:"Ingrese valores validos");  
            break;  
    }  
}
```

conversor

Ejercicio 08:

Para empezar con los arrays, crearemos este código donde tendremos que definir el tamaño de una array en 10 después de esto, habrá que asignarle unos valores a cada índice del array en un método y en otro imprimirlo en pantalla. Para poder poner en un solo cuadro de texto todos los valores del array, he hecho un “*StringBuilder*” para ir añadiendo cada valor en el mismo valor con un bucle y luego llamarlo en un “*JOptionPane.showMessageDialog()*”.

main y método de relleno

```
public static void main(String[] args) {
    int coleccion[] = new int[10];
    rellenar(coleccion);
    JOptionPane.showMessageDialog(parentComponent:null, imprimir(coleccion).toString());
}

public static void rellenar(int valor[]) {
    for (int i=0; i<10; i++) {
        String num = JOptionPane.showInputDialog("Cual será el valor asignado en la posición "+(i+1)+" :");
        valor[i] = Integer.parseInt(num);
    }
}
```

método StringBuilder para crear un único String

```
public static StringBuilder imprimir(int colec[]) {
    StringBuilder todosnum = new StringBuilder(str: " - - COLECCIÓN DE NÚMEROS - - \n");
    for (int junto : colec) {
        todosnum.append(junto).append(str: " ");
    }
    return todosnum;
}
```


Ejercicio 09:

En el programa del ejercicio 09, tendremos que crear un array del tamaño que el usuario quiera. Ahora en este array, iremos definiendo valores aleatorios del 0 hasta un rango que inserte el usuario pero no sea superior a 10. Por último habrá que sumar todos los números generados, y mostrar por pantalla todos los números del array y la suma de estos.

```
import javax.swing.JOptionPane;

public class Ejercicio09 {
    Run | Debug
    public static void main(String[] args) {
        String tamaño = JOptionPane.showInputDialog("Cantidad de numeros aleatorios generado:");
        while (tamaño.equals("")) {
            tamaño = JOptionPane.showInputDialog("Ingresa algun número por favor:");
        }
        int tamaño = Integer.parseInt(tamaño);
        int numaleatorios[] = new int[tamaño];
        rellenar(numaleatorios, tamaño);
        StringBuilder numesp = new StringBuilder("");
        StringBuilder sumacion = new StringBuilder("");
        sumar(numaleatorios, sumacion, imprimir(numaleatorios, numesp), tamaño);
    }

    public static StringBuilder imprimir(int[] valor, StringBuilder numesp) {
        int i = 0;
        for (int junto : valor) {
            numesp.append("Indice [\"+i+\"]: \"+junto).append("\n");
            i++;
        }
        return numesp;
    }
}
```

main y
método para
imprimir los
arrays

```
public static int sumar(int[] valor, StringBuilder sumacion, StringBuilder numesp, int tam) {
    int suma = 0;
    int i = 0;
    for (int numero : valor) {
        suma += numero;
    }
    for (int numero : valor) {
        if (i < tam - 1) {
            sumacion.append(numero + " ");
            i++;
        } else {
            sumacion.append(numero + " = ");
        }
    }
    JOptionPane.showMessageDialog(null, "La suma de tus numeros es: \n" + sumacion + suma+"\n\nY tu colección: \n"+numesp);
    return suma;
}
```

método para sumar los
números del array y
mostrarlos

```
public static void rellenar(int valor[], int tam) {
    for (int i = 0; i < tam; i++) {
        String numtxt = JOptionPane.showInputDialog(
            "Valor máximo para la generación del numero " + (i + 1) + ":");
        int nume = Integer.parseInt(numtxt);
        while (nume > 10) {
            numtxt = JOptionPane.showInputDialog(
                "El rango de generación dado en el ejercicio es de 0[1]-9[10].\n Elija un valor válido");
            nume = Integer.parseInt(numtxt);
        }
        int aleatorio = (int) (Math.random() * nume);
        if(tam<9) {
            while (repes(valor, aleatorio)) {
                aleatorio = (int) (Math.random() * nume);
            }
            valor[i] = aleatorio;
        } else {
            aleatorio = (int) (Math.random() * nume);
            valor[i] = aleatorio;
        }
    }
}
```

método para rellenar el array con números aleatorios entre el rango [0-9], pero preguntando al usuario si quiere “acortarlo”. Con repes, no se repetirán los números, pero si el tamaño es mayor al rango posible, se repetirán.

```
public static boolean repes(int[] valor, int aleatorio) {
    for (int preguntorepe : valor) {
        if (preguntorepe == aleatorio) {
            return true;
        }
    }
    return false;
}
```

método para comprobar que no se repitan los números

Ejercicio 10:

En este programa tendremos que crear una array con el tamaño introducido por teclado y tendremos que indicar cual entre ellos es el mayor. Para rellenar el array, lo haremos con números primos aleatorios entre un rango que nosotros le indiquemos, tendremos que ir controlando que efectivamente se generan números primos.

```
public class Ejercicio10 {
    Run | Debug
    public static void main(String[] args) {
        String numtxt = JOptionPane.showInputDialog(message:"Ingrese la cantidad de numeros primos que quiere generar: ");
        int tamaño = Integer.parseInt(numtxt);
        int numeros[] = new int[tamaño];
        StringBuilder numerosprim = new StringBuilder(str:"");
        rellenaymuestra(numeros, tamaño, numerosprim);
        String losprimos = numerosprim.toString();
        elmayor(losprimos, tamaño);
    }

    public static void rellenaymuestra(int col[], int tam, StringBuilder losprimos) {
        int i = 0;
        while (i < tam) {
            String destxt = JOptionPane.showInputDialog(message:"Valor máximo para el intervalo de numeros aleatorio: ");
            int max = Integer.parseInt(destxt);
            int mates = (int) (Math.random() * max);

            while (!primo(mates) || repes(col, mates)) {
                mates = (int) (Math.random() * max);
            }
            if (primo(mates)) {
                col[i] = mates;
                losprimos.append(mates + " ");
                i++;
            }
        }
        JOptionPane.showMessageDialog(parentComponent:null, "Estos son los numeros primos generados: \n - - " + losprimos + " - - \n");
    }
}
```

main con método para rellenarlo con números primos aleatorios no repetidos y mostrarlo

```
public static boolean repes(int[] valor, int aleatorio) {
    for (int preguntorepe : valor) {
        if (preguntorepe == aleatorio) {
            return true;
        }
    }
    return false;
}

public static boolean primo(int num) {
    if (num <= 1) {
        return false;
    }
    for (int divisor = (num - 1); divisor >= 2; divisor--) {
        if (num % divisor == 0) {
            return false;
        }
    }
    return true;
}

public static void elmayor(String primos, int tama) {
    String[] nums = primos.split(regex:" ");
    int[] numeros = new int[tama];
    int elgrande = 0;
    for (int i = 0; i < tama; i++) {
        int intnum = Integer.parseInt(nums[i]);
        numeros[i] = intnum;
        if (elgrande < numeros[i]) {
            elgrande = numeros[i];
        }
    }
    JOptionPane.showMessageDialog(parentComponent:null, "Y entre ellos el mayor és: " + elgrande);
}
```

3 métodos: el primero verifica que ese número no haya sido dado antes, el segundo verifica que el número es primo y con el último buscamos el número más grande de la array

Ejercicio 11:

En la resolución de este programa definiremos el tamaño de 3 arrays introduciendolo por teclado, el primer array se llenará con números aleatorios, para llenar el segundo tendremos que hacerlo generando una dependencia a los números generados en el primer array, yo entendí que esta dependencia sería con los índices, pero solo lo multiplicaremos el valor del primer array por `Math.random()` y se lo asignaremos al segundo, para el último array lo llenaremos con la multiplicación de los valores de los arrays anteriores en el índice en el que se situará el resultado. [Como entendí otra cosa, Ejercicio11V2 es el código para el ejercicio y Ejercicio11V1 es el primer código que hice]

```
public class Ejercicio11V2 {
    Run | Debug
    public static void main(String[] args) {
        String tamaño = JOptionPane.showInputDialog("¿Cual será el tamaño del array?");
        int tam = Integer.parseInt(tamaño);

        StringBuilder coleccion1 = new StringBuilder();
        StringBuilder coleccion2 = new StringBuilder();
        StringBuilder coleccion3 = new StringBuilder();
        int array[] = new int[tam];
        int aleatorio[] = new int[tam];
        int multiplicado[] = new int[tam];

        rellenararray1(array, tam, coleccion1);
        rellenararray2(array, aleatorio, tam, coleccion2);
        rellenararray3(array, aleatorio, multiplicado, tam, coleccion3);

        JOptionPane.showMessageDialog(null,
            "La primera colección contiene: \n - - " + coleccion1 + " - - \n La segunda colección contiene: \n - - "
            + coleccion2 + " - - \n Y las multiplicaciones dan: \n - - " + coleccion3 + " - - ");
    }
}
```

main

```
public static void rellenararray1(int[] array, int tam, StringBuilder coleccion1) {
    int min = 0;
    int max = 10;
    for (int i = 0; i < tam; i++) {
        int alea = (int) ((Math.random() * (max - min)) + min);
        min += 10;
        max += 10;
        array[i] = alea;
        coleccion1.append(alea + " ");
    }
}

public static void rellenararray2(int[] array, int[] aleatorio, int tam,
    StringBuilder coleccion2) {
    for (int i = 0; i < tam; i++) {
        aleatorio[i] = (int) (Math.random() * (1 + array[i]) - 1);
        coleccion2.append(aleatorio[i] + " ");
    }
}

public static void rellenararray3(int[] array, int[] aleatorio, int[] multiplicado, int tam,
    StringBuilder coleccion3) {
    for (int i = 0; i < tam; i++) {
        multiplicado[i] = array[i] * aleatorio[i];
        coleccion3.append(multiplicado[i] + " ");
    }
}
```

3 métodos
independientes
para rellenar
cada array con
sus condiciones

- - [[Link al Ejercicio11V1](#)] - -

Ejercicio 12:

Para terminar la tarea 06, crearemos un programa que genere números aleatorios entre el 1 y el 300, y los vaya asignando a un array con un tamaño que se ha introducido por teclado. Cuando el array está lleno de números aleatorios, con otra entrada de tipo “*JOptionPane*”, buscaremos y mostraremos los números del array que terminan en el número que se ha introducido en el “*JOptionPane*”.

```
public class Ejercicio12 {
    Run | Debug
    public static void main(String[] args) {
        StringBuilder tusnumeros = new StringBuilder("");
        String tamaño = JOptionPane.showInputDialog("Ingrese la cantidad de numeros que se generará");
        int tam = Integer.parseInt(tamaño);
        int col[] = new int[tam];

        rellenar(tam, col, tusnumeros);

        String sospecharde = JOptionPane.showInputDialog("Indique el número que se buscará");
        int sospechoso = Integer.parseInt(sospecharde);

        while (!numerocorrecto(sospechoso)) {
            sospecharde = JOptionPane.showInputDialog("Indique un número entre [0-9]");
            sospechoso = Integer.parseInt(sospecharde);
        }

        int sos[] = new int[tam];
        StringBuilder bonitos = new StringBuilder("");

        detective(sospechoso, col, sos);

        boolean nohaypapeles = false;
        for (int numero : sos) {
            if (numero != 0) {
                nohaypapeles = true;
                break;
            }
        }
        if (nohaypapeles) {
            mostrar(sos, bonitos, sospechoso);
        } else {
            JOptionPane.showMessageDialog(null,
                "No hay números que terminen en " + sospechoso + "\nTus números generados:\n" + tusnumeros);
        }
    }
}
```

main

```
public static void rellenar(int tam, int[] col, StringBuilder tusnumeros) {
    for (int i = 0; i < tam; i++) {
        int aleatorio = (int) ((Math.random() * 300) + 1);
        col[i] = aleatorio;
        tusnumeros.append(aleatorio + " ");
    }
}

public static void mostrar(int[] sos, StringBuilder bonitos, int sospechoso) {
    for (int i = 0; i < sos.length; i++) {
        if (sos[i] != 0) {
            if (i < (sos.length - 1)) {
                bonitos.append(sos[i] + ", ");
            } else {
                bonitos.append(sos[i] + " ");
            }
        }
    }
    JOptionPane.showMessageDialog(null,
        "Los numeros encontrados terminados en " + sospechoso +
        "\n son: \n - - " + bonitos + " - - ");
}
```

métodos para rellenar y mostrar el contenido del array

```
public static void detective(int sospechoso, int[] col, int[] sos) {
    int numero = 0;
    int i = 0;
    for (int detecta : col) {
        numero = detecta % 10;
        if (numero == sospechoso) {
            sos[i] = detecta;
            i++;
        }
    }
}

public static boolean numerocorrecto(int sospechoso) {
    if (sospechoso < 0) {
        return false;
    } else if (sospechoso >= 10) {
        return false;
    }
    return true;
}
```

métodos para buscar el número ingresado y controlar el número que ingresas

- <https://parzibyte.me/blog/2019/02/20/invertir-cadena-string-java/>
- [https://es.stackoverflow.com/questions/59870/como-eliminar-caracteres-que-n
o-seas-letras-ni-n%C3%BAmoros-de-un-string](https://es.stackoverflow.com/questions/59870/como-eliminar-caracteres-que-no-sean-letras-ni-n%C3%BAmoros-de-un-string)
- [https://learn.microsoft.com/es-es/dotnet/api/system.string.contains?view=net-
8.0](https://learn.microsoft.com/es-es/dotnet/api/system.string.contains?view=net-8.0)
- El copiloto de confianza como ayuda siempre, importante:
<https://chat.openai.com/>