

```
public class Prog100 {
    static Lectura leer = new Lectura();

    static void inicio() {
        int[] numeros = new int[7];
        System.out.println("Tamaño del arreglo: " + numeros.length);
        System.out.println("\nIngrese los valores: ");
        for (int i = 0; i < numeros.length; i++) {
            System.out.print("> numeros[" + i + "]: ");
            numeros[i] = leer.entero();
        }
        System.out.println("\nResultado:");
        for (int i = 0; i < numeros.length; i++) {
            System.out.println("* numeros[" + i + "]: " + numeros[i]);
        }
    }
}
```

1.^a EDICIÓN

FUNDAMENTOS DE PROGRAMACIÓN CON

Java

"Pero hágase todo
decentemente y con orden"
1 Corintios 14:40

Daniel Lévano Rodríguez

Flor Elizabeth Cerdán León

 **UPeU**
FONDO EDITORIAL

Fundamentos de programación con Java

Daniel Lévano Rodríguez
Flor Elizabeth Cerdán León

Título: Fundamentos de Programación con Java

Autores:

Mg. Daniel Lévano Rodríguez

Mg. Flor Elizabeth Cerdán León

.....
Alta Dirección:

Walter Murillo Antón, rector

Roussel Dávila Villavicencio, vicerrector académico

David Echevarría Miranda, vicerrector administrativo

Raúl Gómez Nicoll, vicerrector de bienestar universitario

.....
Consejo editorial:

Walter Murillo Antón

Edwin Sucapuca Sucapuca,

Jorge Moncada López

.....
Ediciones Universidad Peruana Unión

Director editor: Edwin Sucapuca Sucapuca

Diseño de portada/interior: Hank Cruz Bonifacio

© Fondo editorial

Universidad Peruana Unión

Carretera Central Km. 19.5, Ñaña, Lima, Perú

Teléfonos 016186300

fondoeditorial@upeu.edu.pe

www.upeu.edu.pe

Primera edición digital, marzo de 2025

ISBN: 978-9972-604-74-4

Hecho el depósito legal en la Biblioteca Nacional del Perú N°: 2025-02599

El contenido de esta publicación puede reproducirse respetando los derechos de los autores y la editorial.

Libro digital disponible en: <https://www.repositorio.upeu.edu.pe>

Dedicatoria

A nuestro amado hijo, Pablo Daniel, nuestro primogénito, por su valentía y creatividad. Comparte siempre tu conocimiento, pues al enseñar, crecerás aún más. Pon tus talentos al servicio de Dios y de los demás. Anhelamos verte en el cielo.

A nuestra querida hija, Flor Daneli, nuestra princesita, por su ternura y amabilidad. Nunca olvides que la verdadera sabiduría es el temor a Jehová. Usa tus dones con pasión y entrégalos al servicio de Dios y de los demás. Anhelamos verte en el cielo.

Con amor,
Papá y Mamá

Agradecimientos

A Dios, por sus interminables bendiciones y por darnos la oportunidad de predicar a través de este libro.

A la Universidad Peruana Unión, por su invaluable apoyo en la publicación de este material bibliográfico.

A nuestros estudiantes, cuyo deseo por aprender nos impulsa a seguir enseñando con pasión y compromiso.

Autores

Mg. Daniel Lévano Rodríguez

Ingeniero de Sistemas, Magíster en Ingeniería de Sistemas con mención en Ingeniería de Software y egresado del programa doctoral en Ingeniería de Sistemas. Cuento con 22 años de experiencia en el ámbito profesional, desempeñándome en diversas áreas de la ingeniería de software, la docencia universitaria y la investigación académica.

A lo largo de mi trayectoria, he asesorado y dirigido proyectos de investigación formativa en la línea de Ingeniería de Software, promoviendo el desarrollo de soluciones innovadoras y metodologías eficientes en el sector tecnológico. Como docente, he participado activamente en la formación de futuros profesionales, contribuyendo a la generación de conocimiento y a la aplicación de buenas prácticas en la industria del software.

Con un enfoque en la excelencia y la mejora continua, mi labor se alinea con las tendencias y avances tecnológicos, garantizando un impacto positivo en el desarrollo académico y profesional del sector.

Mg. Flor Elizabeth Cerdán León

Ingeniera de Sistemas, Magíster en Ingeniería de Sistemas y egresada del programa doctoral en Ingeniería de Sistemas. Con 17 años de experiencia profesional, he desarrollado mi carrera en las áreas de Ingeniería de Software y Machine Learning, integrando la innovación tecnológica con la investigación aplicada.

A lo largo de mi trayectoria, he dirigido y asesorado proyectos en estas disciplinas, impulsando el desarrollo de soluciones eficientes basadas en inteligencia artificial y metodologías avanzadas de desarrollo de software. Mi labor como docente de nivel universitario en el sector público y privado me ha permitido formar a nuevos profesionales, promoviendo el pensamiento crítico y el uso de buenas prácticas en la industria tecnológica.

Con un enfoque en la excelencia, la investigación y la mejora continua, mi objetivo es contribuir a la evolución de la Ingeniería de Software y el Machine Learning, alineando la educación con los avances tecnológicos y su impacto en el mundo real.

Contenido

Dedicatoria.....	02
Agradecimientos	03
Autores.....	04
1. Entorno de trabajo integrado	07
1.1. Descarga del IDE: NetBeans 24.....	07
1.2. Creando un proyecto en NetBeans.....	07
1.3. Crear Package	08
1.4. Crear una clase java	09
2. Algoritmo.....	10
2.1. Representación de algoritmos	10
2.2. Operadores.....	12
2.3. Ejercicios propuestos 01.....	13
3. Estructuras de control.....	15
3.1. Estructuras secuenciales.....	15
3.2. Ejercicios propuestos 02.....	18
3.3. Variables y tipos de datos	19
3.4. Práctica de autoevaluación 01.....	21
3.5. Estructura decisiva: if.....	22
3.6. Estructura decisiva: if-else.....	23
3.7. Estructuras decisivas compuestas: switch-case.....	30
3.8. Ejercicios propuestos 03.....	32
3.8.1 Secuencial.....	32
3.8.2 Decisivas	33
3.9. Práctica de autoevaluación 02.....	35
4. Manejo de cadenas.....	37
5. Estructuras repetitivas	39
5.1. Estructura while	39
5.2. Estructura do-while.....	50

5.3.	Estructura for	60
5.4.	Ejercicios propuestos 04	70
5.5.	Práctica de autoevaluación 03	72
6.	Modularidad	74
6.1.	Métodos vacíos.....	74
6.2.	Métodos que reciben parámetros	75
6.3.	Métodos que devuelven parámetros.....	80
6.4.	Métodos que reciben y devuelven parámetros	80
6.5.	Aplicación	83
7.	Arreglos	86
7.1.	Arreglos unidimensionales	86
7.2.	Aplicación	98
7.3.	Ejercicios propuestos 05	101
7.4.	Práctica de autoevaluación 04	102
8.	Archivos.....	103

Unidad I - Sesión 1

1. Entorno de trabajo integrado

Existen muchos Entornos de Trabajo Integrado (IDE), para este caso, se utilizará Apache Net Beans 24; sin embargo, puedes trabajar desde la versión 13 en adelante.

1.1. Descarga del IDE: **NetBeans 24**

Para descargar el Instalador: <https://netbeans.apache.org/>



Figura 1 - Página de descarga del Net Beans 24

1.2. Creando un proyecto en NetBeans

- Seleccionar: **File → New Project**

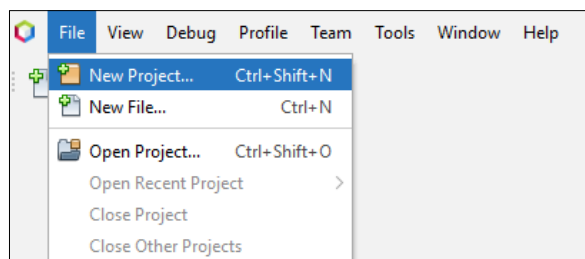


Figura 2 - Opción: New Project

- Seleccionar **Java with Ant → Java Application**

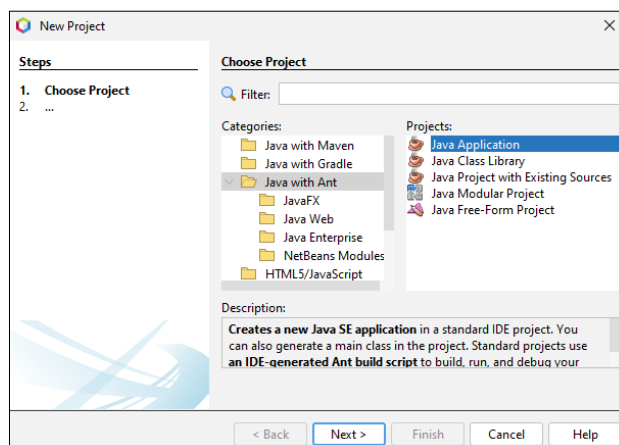


Figura 3 - Opción Java Application

- Asignar un **Project name**, **Project Location** y deshabilitar el check: **Create Main Class**:

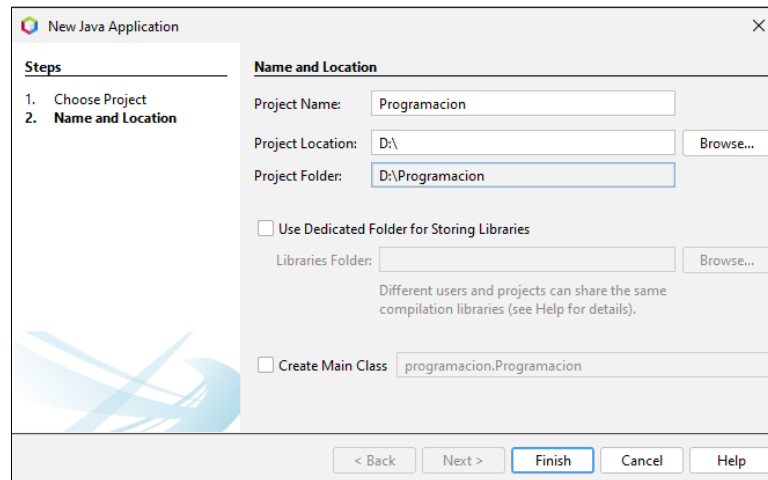


Figura 4 - Formulario de creación de un Java Application

1.3. Crear Package

- Clic derecho en la carpeta **Source Package** y seleccionar **New** → **Java Package**

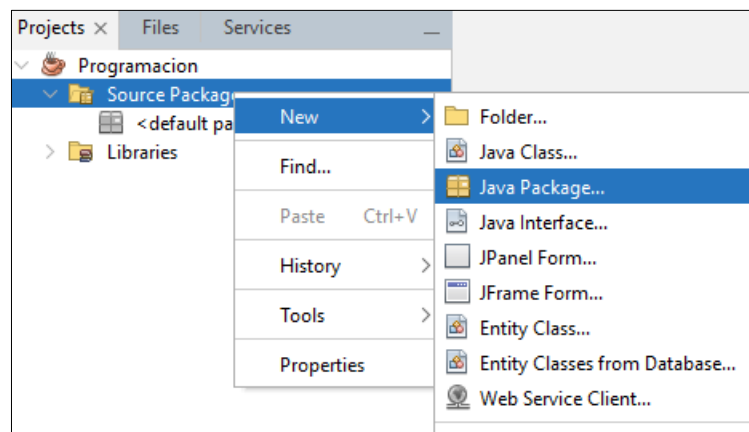


Figura 5 - Opción: Java Package

- Asignar un **Package name**, en este caso: **secuenciales**

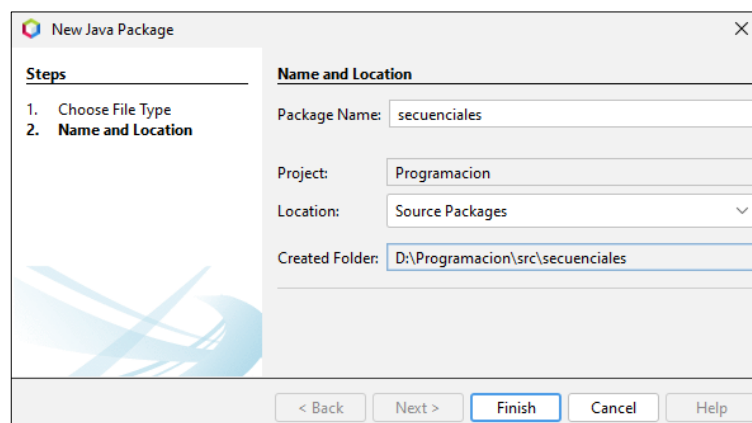


Figura 6 - Asignar nombre al Package

1.4. Crear una clase java

- Clic derecho en el package, luego seleccionar **New** → **Java Class**

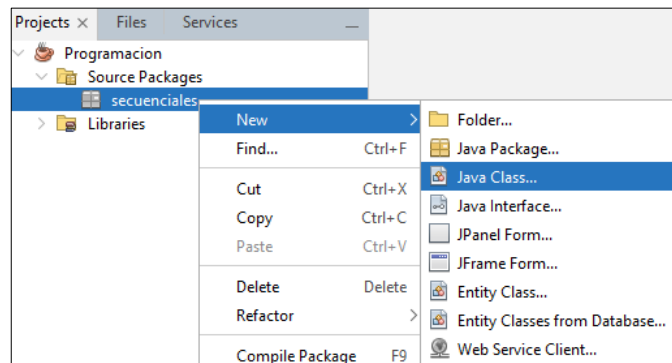


Figura 7 - Opción: Java Class

- Asignar un nombre a la clase, en este caso: Prog001

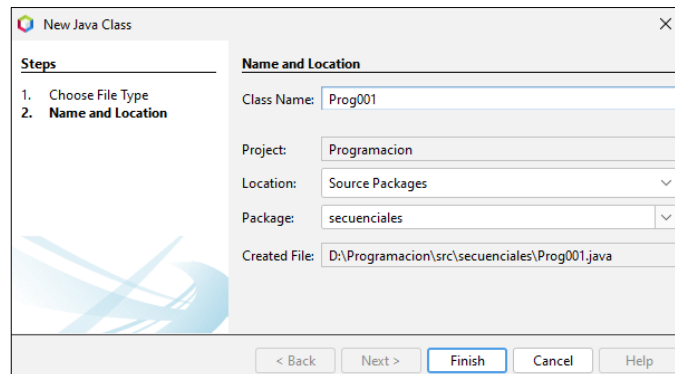


Figura 8 - Opción: Java Class

- Archivo: **Prog001.java** creado

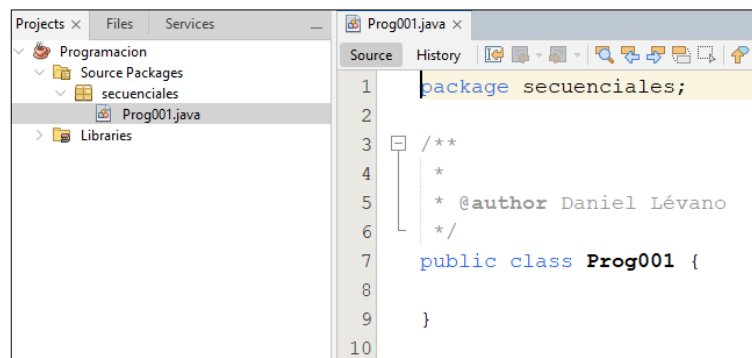


Figura 9 – Prog001.java

2. Algoritmo. ↻

Un algoritmo es “un conjunto de acciones para hallar la solución de un problema”.

- Problema 01: Arrojar un papel al tacho de basura.

Solución de Marlon: P1: Doblar el papel. P2: Pararse. P3: Caminar. P4: Tirar a la basura. P5: Regresar.
Solución de Luis: P1: Doblar papel. P2: Pararse. P3: Caminar. P4: Abrir el tacho. P5: Arrojar a la basura. P6: Regresar.
Solución de: Alexandra P1: Pararse. P2: Caminar. P3: Tirar el papel. P4: Regresar.

Explicación:

En el caso del Problema 01, se plantearon 3 soluciones.







Para este caso se observa que la solución de Marlon es más eficiente que la propuesta de Luis y la solución de Alexandra más eficiente que la presentada por Marlon.

2.1. Representación de algoritmos

En general, los algoritmos se pueden representar de 03 formas:

- Diagrama de flujo. Es la representación gráfica (bloques y/o símbolos.) de un algoritmo.

Tabla 1 - Elementos de un Diagrama de Flujo

Símbolo	Nombre
	inicio/fin
	Proceso/Actividad/Tarea. (Acciones)
	Input/Output (Entrada/Salida de datos)
	Flujo
	Conector
	Decisor

- Pseudocódigo. Es una forma de representar los pasos de un algoritmo utilizando el lenguaje humano. Se tiene como ejemplo el caso del Problema 01.
- Código de programación. Es la representación de un algoritmo utilizando un lenguaje de programación (LP); los ejercicios desarrollados en este libro, el LP utilizado es: Java.

Unidad I – Sesión 2

2.2. Operadores

A continuación, se presentarán los operadores más utilizados:

Tabla 2 - Operadores más utilizados

Categoría	Operador	Nombre
Aritméticos	+	Suma
	-	Resta
	*	Multiplicación
	/	División
	%	Módulo (Obtiene el residuo de una división)
Asignación	=	Asignación
Relacionales	==	Igualdad
	!=	Diferente
	>, >=, <, <=	Mayor que, Mayor o Igual que, Menor que, Menor o Igual que
Lógicos	&&	Y lógico
		O Lógico
	!	No

A continuación, se presenta un conjunto de ejercicios y la respuesta, escribe al lado derecho el proceso.

- Ejercicio 01

$$\begin{aligned} a &= 2 \\ a &= a + 3 \\ a &=? \end{aligned}$$

Respuesta: 5

- Ejercicio 02

$$\begin{aligned} a &= 2, b = 3, c = 4 \\ a &= a + b + c \\ b &= (a * b) + c \\ c &= b + c \\ c &=? \end{aligned}$$

Respuesta: 35

- Ejercicio 03

$$a = 1, b = 2, c = 3$$

$$a = ((a + b + c) * b)$$

$$b = \frac{b + c + 4}{b}$$

$$c = \frac{b + c + 4}{2}$$

$$c = b + c$$

$$c = ?$$

Respuesta: 9

- Ejercicio 04

$$a = 1, b = 2, c = 4, d = 5$$

$$a = ((a * b) - (b * c)) * c$$

$$b = \frac{(a + b) * (b + c)}{3}$$

$$c = ((c + d^2) * a) \% d$$

$$d = a + b^2 + c$$

$$d = ?$$

Respuesta: 1911

- Ejercicio 05

$$a = 2, b = 2, c = 4, d = 5$$

$$a = \left(\frac{(a * b^3) - b}{3} \right) * c$$

$$b = \frac{(a + b) * (b + c)}{\frac{a + b}{3}}$$

$$c = \left(\left(\frac{a + b}{\frac{d}{2}} + d^2 \right) * a \right) \% d$$

$$d = b^2 + c$$

$$d = ?$$

Respuesta: 326

2.3. Ejercicios propuestos 01

- Ejercicio 06

$$a = 1, b = 2, c = 3$$

$$a = ((a + b + c) * b) \% c$$

$$b = \frac{b + c + 4}{b} * c$$

$$c = \frac{b + c + 4}{2} \% 2$$

$$c = a + b + c$$

$$c = ?$$

Respuesta: _____

- Ejercicio 07

$$a = 2, b = 3, c = 4, d = 5$$
$$a = ((a * b) - (b * c)) * c$$
$$b = \frac{(a + b) * (b + c) * (c + d)}{3}$$
$$c = ((c + a + d + b) * 3) \% d$$
$$d = a + b + c$$
$$d = ?$$

Respuesta: _____

- Ejercicio 08

$$a = 1, b = 2, c = 3, d = 4$$
$$a = ((a * b) - (b * c)) * c$$
$$b = \frac{(a * b) + d}{3}$$
$$c = (a * b) \% 2$$
$$d = a + b + c + d$$
$$d = ?$$

Respuesta: _____

- Ejercicio 09

$$a = 1, b = 3, c = 5$$
$$a = \frac{(a + b + c) * b}{c}$$
$$b = \frac{(a + b)^c * c}{b}$$
$$c = ((a + b + c) * (a + b)) \% 2$$
$$c = ?$$

Respuesta: _____

3. Estructuras de control ↻

Dentro de las estructuras de control se puede identificar en general 3 tipos: Estructuras secuenciales, estructuras decisivas y estructuras repetitivas.

La estructura secuencial está conformada por una serie de instrucciones sucesivas; es decir, se ejecuta una sentencia después de otra; en Java el fin de una sentencia se identifica por el punto y coma “;”.

Las estructuras decisivas consisten en la ejecución de una acción siempre que se cumpla una determinada condición; estas pueden ser: simples o compuestas. En Java, dentro de las estructuras decisivas simples se tiene: `if`, `if-else` y `elseif`; y como estructura decisiva compuesta: `switch-case`.

Las estructuras repetitivas permiten la ejecución iterativa de una acción mientras se cumpla una condición. En Java se tienen: `while`, `do-while` y `for`.

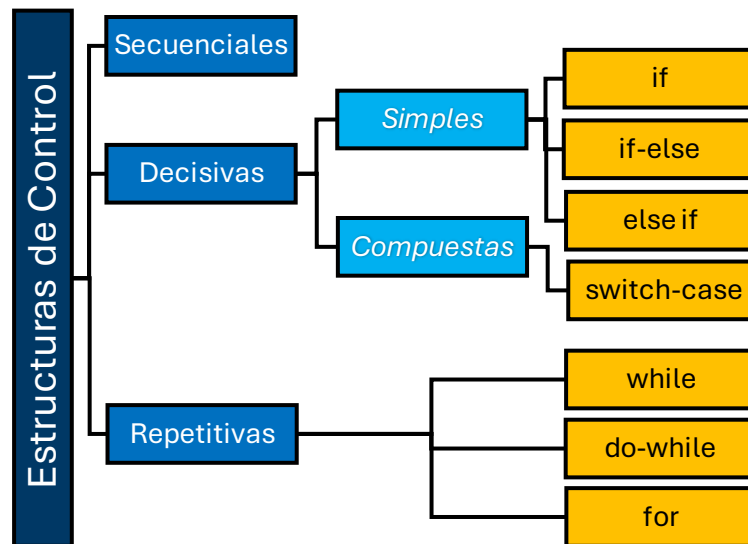


Figura 10 – Estructuras de control

3.1. Estructuras secuenciales

- **Prog001.java** Diseñe un programa que muestre un mensaje.

```
Datos de Salida:  
Jehová es mi pastor,  
nada me faltará.  
Salmos 23:1
```

```
package secuenciales;  
  
// Autor: Daniel Lévano  
public class Prog001 {  
    public static void main(String[] args ) {  
        System.out.println("Jehova es mi Pastor,");  
        System.out.println("nada me faltará");  
        System.out.println("Salmos 23:1");  
    }  
}
```

- **Prog002.java** (*Operadores*) Diseñe un programa que resuelva el siguiente ejercicio:

$$a = 1, b = 3, c = 5$$

$$a = \frac{(a + b + c) * b}{c}$$

$$b = \frac{(a + b) * c}{b}$$

$$c = ((a + b + c) * (a - b)) \% 2$$

$$c = ?$$

Datos de salida:

C = 0

```
package secuenciales;

// Autor: Daniel Lévano
public class Prog002 {
    public static void main(String[] args) {
        int a = 1, b = 3, c = 5;
        a = ((a + b + c) * b) / c;
        b = ((a + b) * c) / b;
        c = ((a + b + c) * (a - b)) % 2;
        System.out.println("c = " + c);
    }
}
```

- **Prog003.java** (*Operadores*) Diseñe un programa que resuelva el siguiente ejercicio:

$$a = a + 2$$

$$b = \frac{a + b}{a}$$

$$b = (b * a) + \frac{a}{b}$$

$$b = ?$$

Donde, a y b son ingresados por el teclado.

Datos de entrada: a: 5 b: 10	Datos de entrada: a: 10 b: 15
Datos de salida: Resultado: b = 17	Datos de salida: Resultado: b = 30

```
package secuenciales;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog003 {
    public static void main(String[] args) {
        Scanner leer = new Scanner(System.in);
        int a, b;
        System.out.print("a: ");
        a = leer.nextInt();
        System.out.print("b: ");
        b = leer.nextInt();
        a = a + 2;
        b = (a + b) / a;
        b = (b * a) + (a / b);
        System.out.println("Resultado: b = " + b);
        leer.close();
    }
}
```

- **Prog004.java** (*Operadores*) Mejore la interfaz del Prog003.java.

Datos de entrada:

Prog004.java

Ingrese valor de a: 5

Ingrese valor de b: 10

Datos de salida:

Resultado: b = 17

```
package secuenciales;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog004 {
    public static void main(String[] args) {
        Scanner leer = new Scanner(System.in);
        int a, b;
        System.out.println("Prog004.java");
        System.out.println("*****\n");
        System.out.print("Ingrese valor de a: ");
        a = Integer.parseInt(leer.nextLine());
        System.out.print("Ingrese valor de b: ");
        b = Integer.parseInt(leer.nextLine());
        a = a + 2;
        b = (a + b) / a;
        b = (b * a) + (a / b);
        System.out.println("\nResultado: b = " + b);
        leer.close();
    }
}
```

3.2. Ejercicios propuestos 02

- **Prog005.java (Operadores)** Desarrollar un programa que permita el ingreso de datos por el teclado y resultado el ejercicio.

Datos de entrada: a: 5 b: 10 c: 15	Datos de entrada: a: 29 b: 20 c: 20	Datos de entrada: a: 50 b: 30 c: 10
Datos de salida: c = 30	Datos de salida: c = 40	Datos de salida: c = 10

$$a = ((a + b + c) * b) \% c$$

$$b = \frac{b + c + 4}{b} * c$$

$$c = \frac{a + b + c + 4}{2} \% 2$$

$$c = a + b + c$$

$$c = ?$$

- **Prog006.java (Operadores)** Desarrollar un programa que permita el ingreso de datos por el teclado y resultado el ejercicio.

Datos de entrada: a: 5 b: 6 c: 7 d: 8	Datos de entrada: a: 5 b: 5 c: 5 d: 5	Datos de entrada: a: 10 b: -10 c: 20 d: -20
Datos de salida: d = -5155	Datos de salida: d = 169	Datos de salida: d = 2000

$$a = ((a * b) - (b * c)) * c$$

$$b = \frac{(a + b) * (b + c) * (c + d)}{3}$$

$$c = ((c + a + d + b) * 3) \% d$$

$$d = a + b + c$$

$$d = ?$$

- **Prog007.java (Operadores)** Desarrollar un programa que permita el ingreso de datos por el teclado y resultado el ejercicio.

Datos de entrada: a: 5 b: 10 c: 15	Datos de entrada: a: 1 b: 2 c: 3	Datos de entrada: a: -10 b: 10 c: 3
Datos de salida: c = 78	Datos de salida: c = -6	Datos de salida: c = 51

$$a = \frac{(a + b + c) * b}{c}$$

$$b = \frac{a + b * c}{b}$$

$$c = \frac{((a + b + c) * (a - b))}{2}$$

$$c = ?$$

3.3. Variables y tipos de datos

Si llegaste a este punto, te habrás dado cuenta de que programar no es tan difícil como parece, incluso es emocionante ver cómo construyes cada línea de código; sin embargo, hasta ahora solo hemos utilizado variables enteras definidas en Java como **int**. A continuación, te presento otros tipos de variables que también puedes utilizar:

Categoría	Subcategoría	Java	Descripción
Tipos de datos Primitivos	Tipos numéricos enteros	byte	Emplea un byte (8 bits) y puede almacenar valores del rango: [-128, 127] <i>Ejemplo:</i> byte a = 5;
		short	Emplea 2 bytes y puede almacenar valores del rango: [-32 768, 32 767] <i>Ejemplo:</i> short a = 30000;
		int	Emplea 4 bytes y puede almacenar valores del rango: [-2 ³¹ , 2 ³¹ -1] <i>Ejemplo:</i> int a = -100;
		long	Emplea 8 bytes y puede almacenar valores del rango: [-2 ⁶³ , 2 ⁶³ -1] <i>Ejemplo:</i> long a = 125;
	Tipos numéricos en punto flotante	float	Permite almacenar números decimales. Emplea 32 bits y puede almacenar valores del rango: [1.4x10 ⁻⁴⁵ , 3.4028235x10 ³⁸] <i>Ejemplo:</i> float a = 5.34f; Obs: Al asignar un número, se debe poner la letra F (Minúscula o Mayúscula) al final.
		double	Permite almacenar números decimales. Emplea 64 bits y puede almacenar valores del rango: [4.9x10 ⁻³²⁴ , 1.7976931348623157x10 ³⁰⁸] <i>Ejemplo:</i> double a = 46859.1234;
	Booleanos y caracteres	boolean	Solo admiten 2 valores, true o false ; <i>Ejemplo:</i> boolean resultado = true;
		char	Permite almacenar caracteres, es decir letras. Utiliza la codificación UTF-16 de Unicode. <i>Ejemplo:</i> char letra = 'a';
Tipos de datos estructurados	Cadenas	String	Permite almacenar una cadena de caracteres. Este tipo es una instancia de la clase String. <i>Ejemplo:</i> String cadena = "Dios es amor";
	Vectores	----	Son colecciones de datos del mismo tipo; pueden ser unidimensionales o multidimensionales, también es conocido como arrays (unidimensionales) o matrices (2 dimensiones) <i>Ejemplo:</i> int vector[] = new int[100];

- **TiposDeVariables.java** Desarrollar un programa que permita el ingreso de datos por el teclado y los muestre. Utilizar algunos tipos de variables.

```

package secuenciales;
import java.util.Scanner;

// Autor: Daniel Lévano
public class TiposDeVariables {
    public static void main(String[] args) {
        Scanner leer = new Scanner(System.in);
        int entero; // 5, 6, 8, -10, 0
        double decimal; // 4.6, 5.21, 6
        String cadena; //"Daniel Lévano", "F", "146"
        char letra; // 'a', 'B', '5'
        boolean valor = true;
        System.out.print("Valor Entero: ");
        entero = Integer.parseInt(leer.nextLine());
        System.out.print("Valor Decimal: ");
        decimal = Double.parseDouble(leer.nextLine());
        System.out.print("Cadena: ");
        cadena = leer.nextLine();
        System.out.print("Letra: ");
        letra = leer.nextLine().charAt(0);

        System.out.println("VALORES INGRESADOS");
        System.out.println("Valor Entero: " + entero);
        System.out.println("Valor Decimal: " + decimal);
        System.out.println("Cadena: " + cadena);
        System.out.println("Letra: " + letra);
        System.out.println("Booleano: " + valor);
        leer.close();
    }
}

```

Ahora, investiga otros tipos de variables y representalos como se ingresaría por el teclado utilizando la librería Scanner:

#	Tipo	Leer
1	Entero	<code>int entero;</code> <code>entero = Integer.parseInt(leer.nextLine());</code>
2		
3		
4		
5		
6		

3.4. Práctica de autoevaluación 01

Es necesario medir lo aprendido, entonces ahora es necesario resolver esta práctica de autoevaluación.

Indicaciones:

Crear un package: practica01

Tiempo asignado: 30 minutos

- **Preg01.java** Intercambiar valores de 2 variables

Datos de entrada:

A: 8

B: 9

Datos de salida:

A = 9

B = 8

- **Preg02.java** Dado n, hallar la suma de los n primeros números naturales.

Datos de entrada:

n: 100

Datos de salida:

Sumatoria = 5050

- **Preg03.java** Hallar el área de un triángulo.

Datos de entrada:

Base (cm): 11

Altura (cm): 7

Datos de salida:

Área = 38.5 cm²

- **Preg04.java** Hallar la hipotenusa de un triángulo rectángulo. Considere para la raíz cuadrada: `Math.sqrt()` y para la potencia `Math.pow()`

Datos de entrada:

Cateto1: 3

Cateto2: 4

Datos de salida:

Hipotenusa = 5

- **Preg05.java** Hallar la distancia entre 2 puntos

Datos de entrada:

X1: 7

Y1: 5

X2: 4

Y2: 1

Datos de salida:

Distancia = 5

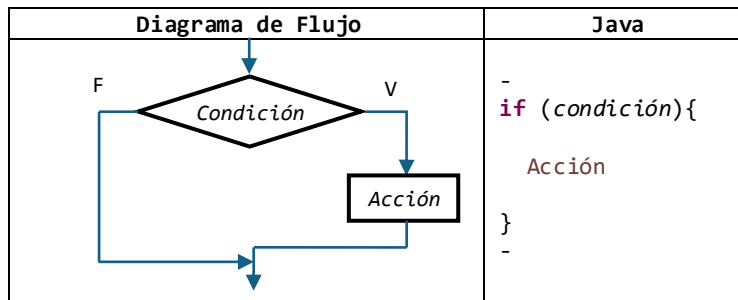
Recuerda: Josué tenía la misión de conquistar una tierra nueva. Al sentirse desfallecer Dios acudió en su ayuda y le dijo: "Mira que te mando que te esfuerces y seas valiente, no temas ni desmayes" Josué 1:9

La programación puede ser una tierra desconocida para ti, pero Dios te puede ayudar.

Unidad I - Sesión 3 y 4

3.5. Estructura decisiva: if

La estructura decisiva if, permite ejecutar una acción siempre que se cumpla una condición, de no cumplirse, sale de la estructura.



Recuerda: “Si oyereis hoy su voz, no endurezcáis vuestros corazones” Hebreos 3:15

Si estamos caminando en la vida y escuchamos el llamado del Señor, abramos nuestros corazones a Él.

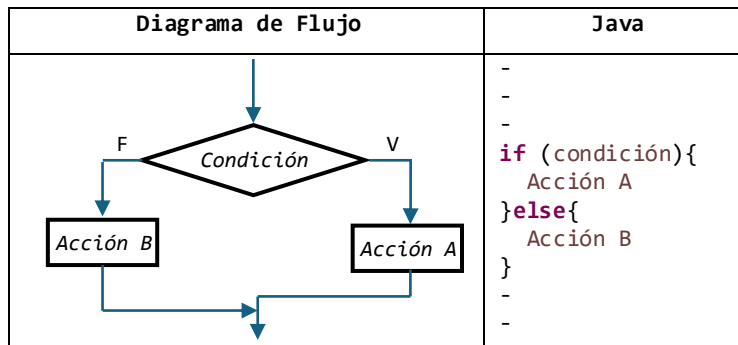
- **Prog008.java** Pablito y Daneli están juntando fondos para ir a un campamento; para ello, están vendiendo algunas manualidades de alta calidad y an diseñado una promoción: Si una persona compra 3 productos y el importe es mayor o igual a 1000 soles, se le dará un descuento del 10%.

Datos de entrada: Producto 1: 300 Producto 2: 200 Producto 3: 100	Datos de entrada: Producto 1: 500 Producto 2: 900 Producto 3: 100
Datos de salida: Total = 600.0	Datos de salida: Total = 1350.0

```
package decisivas;  
import java.util.Scanner;  
  
// Autor: Flor Cerdán  
public class Prog008 {  
    public static void main(String[] args) {  
        Scanner teclado = new Scanner(System.in);  
        double p1, p2, p3, total;  
        System.out.print("Producto 01: ");  
        p1 = Double.parseDouble(teclado.nextLine());  
        System.out.print("Producto 02: ");  
        p2 = Double.parseDouble(teclado.nextLine());  
        System.out.print("Producto 03: ");  
        p3 = Double.parseDouble(teclado.nextLine());  
        total = p1 + p2 + p3;  
        if(total >= 1000){  
            total = total * 0.9;  
        }  
        System.out.println("Total = " + total);  
        teclado.close();  
    }  
}
```

3.6. Estructura decisiva: if-else

La estructura decisiva if-else; ejecuta una acción siempre que se cumpla una condición; de no cumplirse, permite la ejecución de otra acción.



Recuerda: “Si quisierais y oyereis, comeréis el bien de la tierra; mas si rehusáis y fuerais rebeldes, seréis consumidos a espada; porque la boca de Jehová lo ha dicho”
Apocalipsis 3:20

- **Prog009.java** Diseñar un programa que muestre si un número ingresado por el teclado es par o impar.

Datos de entrada: Número: 100	Datos de entrada: Número: 77
Datos de salida: Resultado: PAR	Datos de salida: Resultado: IMPAR

```
package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog009 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int numero;
        System.out.print("Numero: ");
        numero = Integer.parseInt(teclado.nextLine());
        if (numero%2==0) {
            System.out.println("Resultado: Par");
        } else {
            System.out.println("Resultado: Impar");
        }
        teclado.close();
    }
}
```

- **Prog010.java** Diseñar un programa que muestre si un número ingresado por el teclado es positivo o negativo (Considere al Cero como positivo).

Datos de entrada: Número: -50	Datos de entrada: Número: 100	Datos de entrada: Número: 0
Datos de salida: Resultado: NEGATIVO	Datos de salida: Resultado: POSITIVO	Datos de salida: Resultado: POSITIVO

```
package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog010 {
```

```

public static void main(String[] args) {
    Scanner teclado = new Scanner(System.in);
    int numero;
    System.out.print("Numero: ");
    numero = Integer.parseInt(teclado.nextLine());

    if (numero>=0) {
        System.out.println("Resultado: Positivo");
    } else {
        System.out.println("Resultado: Negativo");
    }
    teclado.close();
}
}

```

- **Prog011.java** Diseñar un programa que muestre si un número ingresado por el teclado es positivo (si es mayor a cero), negativo (Menor a cero) o neutro (igual a cero).

Datos de entrada: Número: -50	Datos de entrada: Número: 100	Datos de entrada: Número: 0
Datos de salida: Resultado: NEGATIVO	Datos de salida: Resultado: POSITIVO	Datos de salida: Resultado: NEUTRO

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog011 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int numero;
        System.out.print("Numero: ");
        numero = Integer.parseInt(teclado.nextLine());

        if (numero>0) {
            System.out.println("Resultado: Positivo");
        } else {
            if (numero<0) {
                System.out.println("Resultado: Negativo");
            } else {
                System.out.println("Resultado: Neutro");
            }
        }
        teclado.close();
    }
}

```

Resuelve este ejercicio de una forma diferente:

- **Prog012.java** Diseñar un programa que muestre la categoría de un vendedor en base a las ventas que realizó, considere el siguiente cuadro:

Rango	Categoría
<4000 - Mas]	D
<2000 - 4000]	C
<1000 - 2000]	B
[0 - 1000]	A

Datos de entrada: Importe: 1000	Datos de entrada: Importe: 1500	Datos de entrada: Importe: 4000	Datos de entrada: Importe: 5000
Datos de salida: Resultado: A	Datos de salida: Resultado: B	Datos de salida: Resultado: C	Datos de salida: Resultado: D

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog012 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int importe;
        System.out.print("Importe: ");
        importe = Integer.parseInt(teclado.nextLine());
        if(importe > 4000){
            System.out.println("Categoría D");
        }else{
            if(importe > 2000 ){
                System.out.println("Categoría C");
            }else{
                if (importe > 1000) {
                    System.out.println("Categoría B");
                } else {
                    System.out.println("Categoría A");
                }
            }
        }
        teclado.close();
    }
}

```

Resuelve este ejercicio de una forma diferente:

- **Prog013.java** Desarrollar el ejercicio Prog012, utilizando else if.

Datos de entrada: Importe: 1000	Datos de entrada: Importe: 1500	Datos de entrada: Importe: 4000	Datos de entrada: Importe: 5000
Datos de salida: Resultado: A	Datos de salida: Resultado: B	Datos de salida: Resultado: C	Datos de salida: Resultado: D

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog013 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int importe;
        System.out.print("Importe: ");
        importe = Integer.parseInt(teclado.nextLine());
        if(importe > 4000){
            System.out.println("Categoría D");
        }else if(importe > 2000 ){
            System.out.println("Categoría C");
        }else if (importe > 1000) {
            System.out.println("Categoría B");
        } else {
            System.out.println("Categoría A");
        }
        teclado.close();
    }
}

```

- **Prog014.java** Hallar el importe a pagar en el estacionamiento, considerando que el costo fijo por las 3 primeras horas es de S/ 5.00 soles, y por cada hora adicional es de S/ 2.00, después de las 24 horas, la hora adicional es de S/ 1.00.

Datos de entrada: Cantidad de horas: 1	Datos de entrada: Cantidad de horas: 10	Datos de entrada: Cantidad de horas: 28
Datos de salida: Total a pagar: S/. 5.0	Datos de salida: Total a pagar: S/. 19.0	Datos de salida: Total a pagar: S/. 51.0

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog014 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int tiempo;
        System.out.print("Cantidad de horas: ");
        tiempo = Integer.parseInt(teclado.nextLine());
        System.out.print("Total a pagar: S/. ");
        if (tiempo<=3) {
            System.out.println("5.0" );
        }else if (tiempo<=24) {
            System.out.println((tiempo-3)*2+5);
        }else {
            System.out.println(5+42+tiempo-24);
        }
        teclado.close();
    }
}

```

- **Prog015.java** Identificar el mayor de 3 números ingresados por el teclado.

Datos de entrada: Número A: 20 Número B: 0 Número C: 15	Datos de entrada: Número A: 11 Número B: 15 Número C: 15	Datos de entrada: Número A: 0 Número B: 20 Número C: 150
Datos de salida: El mayor es 20	Datos de salida: El mayor es 15	Datos de salida: El mayor es 150

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog015 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int A, B, C, mayor;
        System.out.print("Número A: ");
        A = Integer.parseInt(teclado.nextLine());
        System.out.print("Número B: ");
        B = Integer.parseInt(teclado.nextLine());
        System.out.print("Número C: ");
        C = Integer.parseInt(teclado.nextLine());
        mayor = A;
        if (B > mayor) {
            mayor = B;
        }
        if (C > mayor) {
            mayor = C;
        }
        System.out.println("El mayor es " + mayor);
        teclado.close();
    }
}

```

- **Prog016.java** Indicar si tres longitudes forman un triángulo

Datos de entrada: Lado A: 7 Lado B: 10 Lado C: 5	Datos de entrada: Lado A: 5 Lado B: 5 Lado C: 20
Datos de salida: Sí	Datos de salida: No

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog016 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double ladoA, ladoB, ladoC;
        System.out.print("Lado A: ");
        ladoA = teclado.nextDouble();
        System.out.print("Lado B: ");
        ladoB = teclado.nextDouble();
        System.out.print("Lado C: ");
        ladoC = teclado.nextDouble();
        if ( ladoA + ladoB > ladoC &&
            ladoA + ladoC > ladoB &&
            ladoB + ladoC > ladoA) {

```

```

        System.out.println("Si");
    }else {
        System.out.println("No");
    }
    teclado.close();
}
}

```

- **Prog017.java** Clasificar triángulos según sus lados (Equilátero/Isósceles/Escaleno).

Datos de entrada: Lado A: 7 Lado B: 7 Lado C: 7	Datos de entrada: Lado A: 6 Lado B: 6 Lado C: 3	Datos de entrada: Lado A: 3 Lado B: 5 Lado C: 7
Datos de salida: Triángulo Equilátero	Datos de salida: Triángulo Isósceles	Datos de salida: Triángulo Escaleno

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog017 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        double ladoA, ladoB, ladoC;
        System.out.print("Lado A: ");
        ladoA = teclado.nextDouble();
        System.out.print("Lado B: ");
        ladoB = teclado.nextDouble();
        System.out.print("Lado C: ");
        ladoC = teclado.nextDouble();
        if (ladoA == ladoB && ladoB == ladoC) {
            System.out.println("Triángulo Equilátero");
        }else if (ladoA == ladoB || ladoA == ladoC || ladoB == ladoC) {
            System.out.println("Triángulo Isósceles");
        }else {
            System.out.println("Triángulo Escaleno");
        }
        teclado.close();
    }
}

```

- **Prog018.java** Una compañía de viajes cuenta con tres tipos de buses (1, 2 y 3), cada uno tiene un precio por kilómetro recorrido por persona, considerando los siguientes datos:

Tipo de Bus	Costo de km recorrido por persona	Capacidad máxima de pasajeros
1	2	20
2	2.5	25
3	3	30

Se requiere determinar el importe a pagar por persona y costo total del viaje considerando la cantidad de persona que viaja y la distancia. En caso exceda el número máximo de personas, el programa debe mostrar el mensaje: "Cantidad no permitida".

Datos de entrada: Tipo de Bus: 1 Distancia: 24 Cantidad de personas: 15	Datos de entrada: Tipo de Bus: 3 Distancia: 15 Cantidad de personas: 31
Datos de salida: Costo del pasaje por persona: 48 Costo total del viaje: 720	Datos de salida: Cantidad No Permitida

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog018 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int tipobus, distancia, personas;
        System.out.print("Tipo de Bus: ");
        tipobus = Integer.parseInt(teclado.nextLine());
        System.out.print("Distancia: ");
        distancia = Integer.parseInt(teclado.nextLine());
        System.out.print("Cantidad de Personas: ");
        personas = Integer.parseInt(teclado.nextLine());

        if (tipobus==1) {
            if (personas<=20) {
                System.out.println("Costo del viaje por persona: "
                    + 2*distancia);
                System.out.println("Costo total del viaje: "
                    + 2*distancia*personas);
            }else {
                System.out.println("Cantidad No Permitida");
            }
        }else if (tipobus == 2) {
            if (personas<=25) {
                System.out.println("Costo del viaje por persona: "
                    + 2.5*distancia);
                System.out.println("Costo total del viaje: "
                    + 2.5*distancia*personas);
            }else {
                System.out.println("Cantidad No Permitida");
            }
        }else if (tipobus == 3) {
            if (personas<=30) {
                System.out.println("Costo del viaje por persona: "
                    + 3*distancia);
                System.out.println("Costo total del viaje: "
                    + 3*distancia*personas);
            }else {
                System.out.println("Cantidad No Permitida");
            }
        }else {
            System.out.println("Tipo de Bus Incorrecto");
        }
        teclado.close();
    }
}

```

3.7. Estructuras decisivas compuestas: switch-case

Esta estructura permite ejecutar diversos tipos de acciones en base a una variable. En Java, frecuentemente, se utiliza las variables de tipo `int` y `char`.

La función `default` se ejecuta luego de no coincidir con alguna de las opciones.

La función `break` permite salir de la estructura: `switch-case`.

Recuerda: "Hay caminos que al hombre le parece derecho; pero su fin es camino de muerte"
Proverbios 14:12

En la vida tendremos muchos caminos por elegir, pero solo un camino puede salvarnos, ese camino es JESÚS.

- **Prog019.java** Elaborar un programa que permita mostrar el día de la semana en base a un número ingresado por el teclado.

Datos de entrada: Día: 1	Datos de entrada: Día: 5	Datos de entrada: Día: 40
Datos de salida: Domingo	Datos de salida: Jueves	Datos de salida: Fuera de Rango

```
package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog019 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int dia;
        System.out.print("Día: ");
        dia = Integer.parseInt(teclado.nextLine());
        switch (dia) {
            case 1:
                System.out.println("Domingo");
                break;
            case 2:
                System.out.println("Lunes");
                break;
            case 3:
                System.out.println("Martes");
                break;
            case 4:
                System.out.println("Miércoles");
                break;
            case 5:
                System.out.println("Jueves");
                break;
            case 6:
                System.out.println("Viernes");
                break;
            case 7:
                System.out.println("SÁBADO");
                break;
            default:
                System.out.println("Fuera de Rango");
                break;
        }
        teclado.close();
    }
}
```

- **Prog020.java** Elaborar un programa que permita mostrar el mes del año en base a un número ingresado por el teclado.

Datos de entrada: Mes: 1	Datos de entrada: Mes: 5	Datos de entrada: Mes: 40
Datos de salida: Enero	Datos de salida: Mayo	Datos de salida: Fuera de Rango

```

package decisivas;
import java.util.Scanner;

// Autor: Flor Cerdán

public class Prog020 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int mes;
        System.out.print("Mes: ");
        mes = teclado.nextInt();
        switch (mes) {
            case 1:
                System.out.println("Enero");
                break;
            case 2:
                System.out.println("Febrero");
                break;
            case 3:
                System.out.println("Marzo");
                break;
            case 4:
                System.out.println("Abril");
                break;
            case 5:
                System.out.println("Mayo");
                break;
            case 6:
                System.out.println("Junio");
                break;
            case 7:
                System.out.println("Julio");
                break;
            case 8:
                System.out.println("Agosto");
                break;
            case 9:
                System.out.println("Septiembre");
                break;
            case 10:
                System.out.println("Octubre");
                break;
            case 11:
                System.out.println("Noviembre");
                break;
            case 12:
                System.out.println("Diciembre");
                break;
            default:
                System.out.println("Fuera de Rango");
                break;
        }
        teclado.close();
    }
}

```

3.8. Ejercicios propuestos 03

3.8.1 Secuencial

- **Prog021.java** La universidad desea determinar el monto de un cheque que debe proporcionar a uno de sus docentes que tendrá que dictar una ponencia al extranjero; los gastos que cubre la universidad por cada día son:
 - Hotel = S/ 150.00
 - Alimentación = S/ 60.00
 - Movilidad = S/ 50.00
 - Otros gastos = S/ 100.00

Datos de entrada:

Cantidad de días: 7

Datos de salida:

Nro	DETALLE	IMPORTE
1.	Hotel	1050.0
2.	Alimentación	420.0
3.	Movilidad	350.0
4.	Otros Gastos	700.0
TOTAL		2520.0

- **Prog022.java** Un productor de leche lleva el registro de lo que produce en litros, pero cuando entrega le pagan en galones. Realice un programa en JAVA, que ayude al productor a saber cuánto recibirá por la entrega de su producción de un día.

Considere: 1 galón = 3.785 litros

Datos de entrada:

Cantidad de Litros Vendidos: 207
Costo por Litro: 1.7

Datos de Salida:

Cantidad de Galones: 54.7
Total a cobrar (S/.): 351.9

- **Prog023.java** Se requiere determinar el sueldo semanal de un trabajador con base en las horas que trabaja y el pago por hora que recibe. Para obtener la solución de este problema es necesario conocer las horas que labora cada trabajador por día y cuánto se le debe pagar por cada hora que trabaja. Considere jornada de 5 días a la semana.

Datos de entrada:

Cantidad de horas por día: 7
Costo por hora (S/): 25

Datos de salida:

Total a pagar: S/ 875.0

- **Prog024.java** Se requiere determinar el promedio que obtendrá un estudiante de la carrera de Ingeniería de Sistemas, considerando que realiza tres exámenes, de los cuales el primero y el segundo tienen una ponderación de 25%, mientras que el tercero de 50%.

Datos de entrada:

1er Examen (25%): 16
 2do Examen (25%): 19
 3er Examen (50%): 10

Datos de salida:

Promedio: 13.75

- Con la ayuda de un amigo(a), crea el enunciado de un ejercicio de estructura secuencial

3.8.2 Decisivas

- **Prog025.java** Ordenar 3 números de menor a mayor.

<p>Datos de entrada: Número A: 50 Número B: 20 Número C: 35</p>	<p>Datos de entrada: Número A: 20 Número B: 20 Número C: 80</p>	<p>Datos de entrada: Número A: 50 Número B: 25 Número C: 50</p>
<p>Datos de salida: 20 35 50</p>	<p>Datos de salida: 20 20 80</p>	<p>Datos de salida: 25 50 50</p>

- **Prog026.java** Un grupo de estudiantes está organizando un viaje de estudios para participar del X Congreso JavaConf, y requiere determinar cuánto deberá pagar cada estudiante y cuánto debe se debe pagar a la compañía de viajes en total por el servicio.

La forma de cobrar es la siguiente:

- Si son 100 estudiantes o más, el costo por cada estudiante es de S/ 65.00;
- De 50 a 99 estudiantes, el costo es de S/ 70.00,
- De 30 a 49, de S/ 95.00,
- Si son menos de 30, el costo de la renta del autobús es de S/ 4000.00, el cual debe dividirse entre en número de alumnos que viajan.

Datos de entrada: Cantidad de alumnos: 78 Datos de salida: Costo por alumno: S/ 70,0 Total a pagar: S/ 5460.0	Datos de entrada: Cantidad de alumnos: 21 Datos de salida: Costo por alumno: S/ 190,5 Total a pagar: S/ 4000.0
---	--

- **Prog027.java** El consultorio odontológico de la Dra. Karina Lévano, tiene como política cobrar la consulta con base al número de citas, de la siguiente forma:
 - Las tres primeras citas a S/ 200.00 c/u.
 - Las siguientes dos citas a S/ 150.00 c/u.
 - Las tres siguientes citas a S/ 100.00 c/u.
 - Las restantes a S/ 50.00 c/u, mientras dure el tratamiento.

Se requiere un programa que determine cuánto le costará a un paciente su tratamiento en base al número de citas.

Datos de entrada: Cantidad de citas: 5 Datos de salida: Total a pagar: S/ 900	Datos de entrada: Cantidad de citas: 12 Datos de salida: Total a pagar: S/ 1400
--	--

- Con la ayuda de un amigo(a), crea el enunciado de un ejercicio de estructura decisiva

Unidad I - Sesión 5

3.9. Práctica de autoevaluación 02

Crear un package: practica02

Tiempo: 40 minutos

Recuerda: “Examinaos a vosotros mismos si estáis en la fe; probaos a vosotros mismos. ¿O no os conocéis a vosotros mismos, que Jesucristo está en vosotros, a menos que estéis reprobados?”
Josué 1:9

Es importante la autoevaluación.

ESTRUCTURAS SECUENCIALES

- **Preg01.java** Diseñe un programa en java (considere solo variables enteras) que permita solucionar el siguiente ejercicio:

a, b, c

$$a = \frac{a + (b * c^2)}{a * c^2}$$

$$b = \frac{(a + b + c)}{a * b} + \frac{a}{c} - \frac{b}{a + c}$$

c = a + b + c

c = ?

Datos de entrada: a: 10 b: 20 c: 25	Datos de entrada: a: 8 b: 9 c: 10	Datos de entrada: a: 25 b: 30 c: 50
Datos de salida: c = 25	Datos de salida: c = 13	Datos de salida: c = 53

- **Preg02.java** Flor Daneli vende marcianos a S/ 1.50 soles al salir del “Club de Aventureros Orión” todos los domingos. Ella desea dar un regalo a su mamá por el Día de la Madre. Diseñe un programa que muestre la cantidad de marcianos que debe vender para alcanzar el importe del regalo. Considere que debe devolver por concepto de “Diezmo” el 10% de todas sus ventas a la iglesia.

Datos de entrada: Costo del Regalo: 50	Datos de entrada: Costo del Regalo: 78
Datos de salida: Cantidad: 38 marcianos Precio: S/ 1.5 Venta: S/ 57.0 Diezmo: S/ 5.7 Ahorro: S/ 51.3	Datos de salida: Cantidad: 58 marcianos Precio: S/ 1.5 Venta: S/ 87.0 Diezmo: S/ 8.7 Ahorro: S/ 78.3

ESTRUCTURAS DECISIVAS

- **Preg03.java** En la playa de estacionamiento “Mi Cochera”, el costo por las primeras 4 horas es de S/ 6.00 soles, luego por cada hora adicional tiene un costo de 2 soles. Diseñe un programa en java que permita el ingreso de un número de horas (Valor entero) y te muestre el importe a pagar.

Datos de entrada: Horas: 2	Datos de entrada: Horas: 8
Datos de salida: Importe a pagar = S/ 6.0	Datos de salida: Importe a pagar = S/ 14.0

- **Preg04.java** Pablito desea entregar un donativo en base a sus ganancias para la construcción de un templo cerca a su casa según el siguiente cuadro.

Rango	Porcentaje
[0 , 1000]	5%
<1000 , 1500]	7%
<1500 , 2000]	8%
<2000 , 5000]	10%
<5000 , Más]	15%

Datos de entrada: Ganancia: 2000	Datos de entrada: Ganancia: 5000	Datos de entrada: Ganancia: 1000
Datos de salida: Donación = S/ 160.0	Datos de salida: Donación = S/ 500.0	Datos de salida: Donación = S/ 50.0

Unidad II – Sesión 1

4. Manejo de cadenas

En Java, a nivel de consola, existe la dificultad de leer valores numéricos y cadenas en un mismo programa.

Considerando que se definió: `Scanner teclado = new Scanner(System.in)`; el ingreso de valores por el teclado para los ejercicios en adelante será de la siguiente manera:

Acción	Java
Leer Cadenas	<code>leer.nextLine()</code> ;
Leer Enteros	<code>Integer.parseInt(leer.nextLine())</code> ;
Leer Decimales	<code>Double.parseDouble(leer.nextLine())</code> ;
Leer Caracteres	<code>leer.nextLine().charAt(0)</code> ;

- **Prog028.java** Como leer cadenas y números en una misma clase.

```
Datos de entrada:
Nombre: Daneli
Edad: 7
Ocupación: Estudiante
Talla: 1.31

Datos de salida:
Los datos ingresados son:
- Nombre: Daneli
- Edad: 7
- Ocupación: Estudiante
- Talla: 1.31
```

```
package cadena;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog028 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String nombre, ocupacion;
        int edad;
        double talla;
        System.out.print("Nombre: ");
        nombre = teclado.nextLine();
        System.out.print("Edad: ");
        edad = Integer.parseInt(teclado.nextLine());
        System.out.print("Ocupación: ");
        ocupacion = teclado.nextLine();
        System.out.print("Talla: ");
        talla = Double.parseDouble(teclado.nextLine());
        System.out.println("Los datos ingresados son: ");
        System.out.println("- Nombre: " + nombre);
        System.out.println("- Edad: " + edad);
        System.out.println("- Ocupacion: " + ocupacion);
        System.out.println("- Talla: " + talla);
        teclado.close();
    }
}
```

- **Prog029.java** Uso de funciones `nextLine()`, `toUpperCase()`, `toLowerCase()`, `trim()`, `substring()`, `length()`, `charAt()`.

```

package cadena;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog029 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String cadena;
        System.out.println("**** MANEJO DE CADENAS ****");
        System.out.print("Leer Cadena con nextLine(): ");
        cadena = teclado.nextLine();
        System.out.println("Valor inicial ingresado es: " + cadena);
        System.out.println("Mayúscula con toUpperCase() : " + cadena.toUpperCase());
        System.out.println("Minúscula con toLowerCase() : " + cadena.toLowerCase());
        System.out.println("Elimina espacios con trim() : " + cadena.trim());
        System.out.println("subcadena con substring(int): " + cadena.substring(5));
        System.out.println("subcadena con substring(int,int): " + cadena.substring(5, 10));
        System.out.println("Tamaño de una cadena: " + cadena.length());
        System.out.println("Extraer letra: " + cadena.charAt(0));
        System.out.println("Extraer letra: " + cadena.charAt(1));
        System.out.println("Extraer letra: " + cadena.charAt(2));
        System.out.println("Extraer letra: " + cadena.charAt(3));
        System.out.println("Extraer letra: " + cadena.charAt(4));
        System.out.println("Extraer letra: " + cadena.charAt(5));
        System.out.println("Extraer letra: " + cadena.charAt(6));
        System.out.println("Ultima letra: " + cadena.charAt(cadena.length()-1));
        teclado.close();
    }
}

```

- **Prog030.java** Mostrar si dos cadenas son iguales o diferentes. Utilice de la función `equals()`.

Datos de entrada: Cadena 1: Pablo Cadena 2: Daneli	Datos de entrada: Cadena 1: Respeto Cadena 2: Respeto
Datos de salida: Son diferentes	Datos de salida: Son iguales

```

package cadena;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog030 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String cadena1, cadena2;
        System.out.print("Cadena 1: ");
        cadena1 = teclado.nextLine();
        System.out.print("Cadena 2: ");
        cadena2 = teclado.nextLine();
        if (cadena1.equals(cadena2)) {
            System.out.println("Son iguales");
        } else {
            System.out.println("Son diferentes");
        }
        teclado.close();
    }
}

```

- **Prog031.java** Indicar si dos cadenas son iguales, si Cadena 1 va antes que la Cadena 2 o si la Cadena 1 va después que la Cadena 2. Utilice la función `compareTo()`

Datos de entrada: Cadena 1: Adán Cadena 2: Moisés	Datos de entrada: Cadena 1: Adán Cadena 2: Adán	Datos de entrada: Cadena 1: Moisés Cadena 2: Adán
Datos de salida: Adán va antes de Moisés	Datos de salida: Son iguales	Datos de salida: Moisés va después que Adán

```

package cadena;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog031 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String cadena1, cadena2;
        System.out.print("Cadena 1: ");
        cadena1 = teclado.nextLine();
        System.out.print("Cadena 2: ");
        cadena2 = teclado.nextLine();
        teclado.close();
        System.out.println("CompareTo==> " + cadena1.compareTo(cadena2));

        if (cadena1.compareTo(cadena2)<0) {
            System.out.println(cadena1 + " Antes que " + cadena2);
        }else if (cadena1.compareTo(cadena2)>0) {
            System.out.println(cadena1 + " Después que " + cadena2);
        }else if (cadena1.compareTo(cadena2)==0) {
            System.out.println("Son iguales");
        }
    }
}

```

5. Estructuras repetitivas

Una estructura repetitiva permite ejecutar de forma iterativa (bucles) un bloque de código y tiene básicamente 3 componentes:

- Primero: Variable inicial
- Segundo: Condición de salida
- Tercero: Incremento o decremento (Cambio de valor en la variable involucrada en la condición de salida)

5.1. Estructura `while`

En esta estructura, mientras se cumpla una condición, se ejecutará una acción de forma repetitiva. Es importante que dentro de la estructura exista una sentencia que permita el

Recuerda:

“Jehová llamó a Samuel; y él respondió: Heme aquí. Y corriendo luego a Elí, dijo: Heme aquí; ¿Para qué me llamaste? Y Elí le dijo: Yo no he llamado; vuelve y acuéstate. Y él se volvió y se acostó. Y Jehová volvió a llamar otra vez a Samuel; levantándose Samuel, vino a Elí y dijo: Heme aquí; ¿Para qué me has llamado? Y él dijo: Hijo mío, yo no he llamado; vuelve y acuéstate. Jehová, pues, llamó la tercera vez a Samuel. Y él se levantó y vino a Elí, y dijo: Heme aquí; ¿para qué me has llamado? Entonces entendió Elí que Jehová llamaba al joven. Y dijo Elí a Samuel: Ve y acuéstate; y si te llamare, dirás: Habla, Jehová, porque tu siervo oye. Así se fue Samuel, y se acostó en su lugar. Y vino Jehová, y llamó como las otras veces: ¡Samuel, Samuel! Entonces Samuel dijo: Habla, porque tu siervo oye.”

1 Samuel 4 - 10

cambio de valor de la variable (Incremento o Decremento) para evitar que se genere un bucle infinito.

Diagrama de flujo	Java
<pre> graph TD A[Variable Inicial] --> B{Condición} B -- V --> C[Acción] C --> D[Inc/Dec] D --> B B -- F --> E[] style E fill:none,stroke:none </pre>	<pre> - - - Variable Inicial while(condición){ Acción Incremento/Decremento } - - </pre>

- **Prog032.java** (*while*) Mostrar números: 1, 2, 3, 4, ... ,100.

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog032 {
    public static void main(String[] args) {
        //Variable inicial
        int num=1;
        // condición de salida
        while(num<=100) {
            System.out.print(num + ", ");
            // Incremento/decremento
            num++;
        }
    }
}

```

- **Prog033.java** (*while*) Mostrar un mensaje 10 veces.

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog033 {
    public static void main(String[] args) {
        int num = 1;
        String mensaje = "Cristo viene pronto - Prepárate";
        while(num<=10) {
            System.out.println(mensaje);
            num++;
        }
    }
}

```

- **Prog034.java** (*while*) Mostrar números: 2, 4, 6, 8 100.

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog034 {

```

```

    public static void main(String[] args) {
        int num = 2;
        while (num <= 100) {
            System.out.println(num);
            num+=2;
        }
    }
}

```

- **Prog035.java** (*while*) Mostrar números: 1, 3, 5, 7 99.

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog035 {
    public static void main(String[] args) {
        int num = 1;
        while (num <= 99) {
            System.out.println(num);
            num+=2;
        }
    }
}

```

- **Prog036.java** (*while*) Mostrar números: 5,10, 15,, 100.

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog036 {
    public static void main(String[] args) {
        int num = 5;
        while (num <= 100) {
            System.out.println(num);
            num+=5;
        }
    }
}

```

- **Prog037.java** (*while*) Mostrar la secuencia: 1, 2, 3, ... ,N. Donde, N es ingresado por el teclado.

Datos de entrada:
N: 6

Datos de salida:
1
2
3
4
5
6

```

package repetitivas;

import java.util.Scanner;

```

```
// Autor: Daniel Lévano
public class Prog037 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int num = 1, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        while (num<=N) {
            System.out.println(num);
            num++;
        }
        teclado.close();
    }
}
```

- **Prog038.java** (*while*) Generar los número de [a , b]. Donde, a y b son ingresado por el teclado.

Datos de entrada:

a: 6
b: 12

Datos de salida:

[6 , 12]
6
7
8
9
10
11
12

```
package repetitivas;

import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog038 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a, b;
        System.out.print("Valor a: ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b: ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "]";
        while (a <= b) {
            System.out.println(a);
            a++;
        }
    }
}
```

- **Prog039.java** (*while*) Generar los número de [a , b]. Donde, a y b son ingresado por el teclado. Utilizar solo un *while*.

Datos de entrada: a: 6 b: 11	Datos de entrada: a: 11 b: 6
Datos de salida: [6 , 11] 6 7 8 9 10 11	Datos de salida: [11 , 6] 6 7 8 9 10 11

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog039 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a,b,temp;
        System.out.print("Valor a = ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b = ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "]";
        if(a>b) {
            temp = a;
            a = b;
            b = temp;
        }
        while (a<=b) {
            System.out.println(a);
            a++;
        }
    }
}

```

- **Prog040.java** (*while*) Mostrar números impares comprendidos entre [a, b], donde a y b, son ingresados por el teclado.

Datos de entrada: a: 6 b: 15	Datos de entrada: a: 15 b: 6
Datos de salida: [6 , 15] 7 9 11 13 15	Datos de salida: [15 , 6] 7 9 11 13 15

```

package repetitivas;
import java.util.Scanner;

```

```
// Autor: Daniel Lévano
public class Prog040 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a,b,temp;
        System.out.print("Valor a = ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b = ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "]";
        if(a>b) {
            temp = a;
            a = b;
            b = temp;
        }
        while (a<=b) {
            if (a%2==1) {
                System.out.println(a);
            }
            a++;
        }
    }
}
```

- **Prog041.java** (*while*) Mostrar la secuencia (*Serie Fibonacci*) de números: 0 1 1 2 3 5 8 13 de todos los números menores a N, donde N, es ingresado por el teclado.

Datos de entrada:

N: 50

Datos de Salida:

0 1 1 2 3 5 8 13 21 34

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog041 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a=0, b= 1, c, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        while (a<=N) {
            System.out.print(a + " ");
            c = a + b;
            a = b;
            b = c;
        }
        teclado.close();
    }
}
```

- **Prog042.java** (*while*) Dado un número N, ingresado por el teclado, generar la tabla de multiplicar de dicho número.

Datos de entrada:

N: 6

Datos de Salida:

```
6*1 = 6
6*2 = 12
6*3 = 18
6*4 = 24
6*5 = 30
6*6 = 36
6*7 = 42
6*8 = 48
6*9 = 54
6*10 = 60
6*11 = 66
6*12 = 72
```

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog042 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N, a = 1;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        while (a <= 12) {
            System.out.println(N + "*" + a + " = " + N * a);
            a++;
        }
    }
}
```

- **Prog043.java** (*while*) Dado un número N, ingresado por el teclado, generar el siguiente resultado.

Dato de Entrada:

N: 8

Resultado:

```
8    1
7    2
6    3
5    4
4    5
3    6
2    7
1    8
```

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog043 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int numero1;
```

```

        int numero2 = 1;
        System.out.print("N: ");
        //Valor inicial
        numero1 = Integer.parseInt(teclado.nextLine());
        while(N>=1) {
            System.out.println(numero1 + "\t" + numero2);
            numero1--;
            numero2++;
        }
        teclado.close();
    }
}

```

- **Prog044.java** (*while*) Mostrar el cuadrado de los números hasta N.

Dato de entrada:

N: 5

Datos de salida

Cuadrado de 1 = 1.0

Cuadrado de 2 = 4.0

Cuadrado de 3 = 9.0

Cuadrado de 4 = 16.0

Cuadrado de 5 = 25.0

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog044 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int num, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        num = 1;
        while(num<=N) {
            System.out.println("Cuadrado de " + num + " = " + Math.pow(num, 2));
            num++;
        }
    }
}

```

- **Prog045.java** (*while*) Hallar la sumatoria de $S = 1 + 2 + 3 + 4 + \dots + 100$.

Datos de salida

S = 5050

```

package repetitivas;

```

```
// Autor: Daniel Lévano
public class Prog045 {
    public static void main(String[] args) {
        int a = 1, S=0;
        while (a<=100) {
            S = S+a;
            a++;
        }
        System.out.println("S = " + S);
    }
}
```

- **Prog046.java** (*while*) Hallar la sumatoria de $S = 1 + 2 + 3 + 4 + \dots + N$, donde N es ingresado por el teclado.

Dato de entrada:
N: 50

Datos de salida
S = 1275

```
package repetitivas;
import java.util.Scanner;
// Autor: Daniel Lévano
public class Prog046 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a = 1, S=0, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        while (a<=N) {
            S = S+a;
            a++;
        }
        System.out.println("S = " + S);
        teclado.close();
    }
}
```

Observación: En acumuladores, cuando se trata de:

- Sumas y restas, el acumulador debe inicializar en CERO
- Multiplicaciones y divisiones, el acumulador debe inicializar en UNO.

- **Prog047.java** (*while*) Hallar la factorial de un número, considerando que $F = 1 * 2 * 3 * 4 * \dots * N$, donde N es ingresado por el teclado.

Dato de entrada:

N = 10

Datos de salida

F = 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 x
F = 3628800

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog047 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N, F = 1;
        System.out.print("N = ");
        N = Integer.parseInt(teclado.nextLine());
        System.out.print("F = ");
        while(N>=1) {
            System.out.print(N + " x ");
            F = F * N;
            N--;
        }
        System.out.println("\nF = " + F);
        teclado.close();
    }
}
```

- **Prog048.java** (*while*) Generar la tabla de multiplicar de 1 hasta N, donde N es ingresado por el teclado.

Dato de entrada:

N: 3

Datos de salida

1*1 = 1
1*2 = 2
1*3 = 3
1*4 = 4
1*5 = 5
1*6 = 6
1*7 = 7
1*8 = 8
1*9 = 9
1*10 = 10
1*11 = 11
1*12 = 12
2*1 = 2
2*2 = 4
2*3 = 6
2*4 = 8
2*5 = 10

```
2*6 = 12
2*7 = 14
2*8 = 16
2*9 = 18
2*10 = 20
2*11 = 22
2*12 = 24
3*1 = 3
3*2 = 6
3*3 = 9
3*4 = 12
3*5 = 15
3*6 = 18
3*7 = 21
3*8 = 24
3*9 = 27
3*10 = 30
3*11 = 33
3*12 = 36
```

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog048 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N, a = 1, b;
        System.out.print("N = ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        while (a <= N) {
            b=1;
            while(b<=12) {
                System.out.println(a + "*" + b + " = " + a*b);
                b++;
            }
            a++;
        }
    }
}
```

Unidad II - Sesión 2

5.2. Estructura do-while

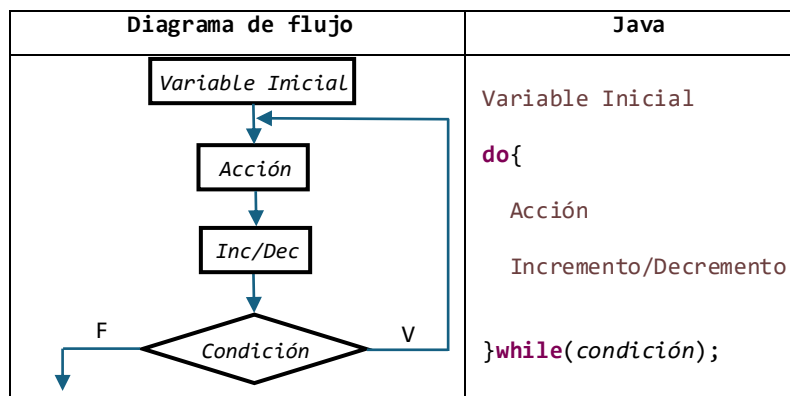
En esta estructura, primero se realiza la acción; luego si la condición se cumple, se volverá a ejecutar nuevamente la acción. Es importante que dentro de la estructura exista una sentencia que permita en cambio de valor de la variable inicial (Incremento o Decremento) para evitar se genere un bucle infinito.

La diferencia, que existe con la estructura while, radica en el número mínimo de iteraciones; en do-while el número mínimo de iteraciones es “uno” y en while es “cero”. Esto puedes demostrarlo asignando num = 1000 en los ejercicios: Prog031.java y Prog048.java.

Recuerda:

"Y Elías subió a la cumbre del Carmelo, y postrándose en tierra, puso su rostro entre las rodillas. Y dijo a su criado: Sube ahora, y mira hacia el mar. Y él subió, y miró, y dijo: No hay nada. Y él le volvió a decir: Vuelve siete veces. Y a la séptima vez dijo: Yo veo una pequeña nube como la palma de la mano de un hombre, que sube del mar."

1 Reyes 18:42-44



A continuación, se resolverá los ejercicios Prog032.java al Prog048.java ya desarrollados, utilizando la estructura do-while. Es recomendable que hagas una comparación de los códigos.

- **Prog049.java (do-while)** Mostrar números: 1, 2, 3, 4, ... ,100.

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog049 {
    public static void main(String[] args) {
        int num = 1;
        do {
            System.out.print(num + ", ");
            num++;
        } while (num<=100);
    }
}
```

- **Prog050.java** (*do-while*) Mostrar un mensaje 10 veces.

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog050 {
    public static void main(String[] args) {
        int num = 1;
        String mensaje = "Cristo viene pronto - Prepárate";
        do {
            System.out.println(mensaje);
            num++;
        } while (num<=10);
    }
}
```

- **Prog051.java** (*do-while*) Mostrar números: 2, 4, 6, 8 100.

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog051 {
    public static void main(String[] args) {
        int num = 2;
        do {
            System.out.println(num);
            num+=2;
        } while (num<=100);
    }
}
```

- **Prog052.java** (*do-while*) Mostrar números: 1, 3, 5, 7 99.

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog052 {
    public static void main(String[] args) {
        int num = 1;
        do {
            System.out.println(num);
            num+=2;
        } while (num<=99);
    }
}
```

- **Prog053.java** (*do-while*) Mostrar números: 5,10, 15, 100.

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog053 {
    public static void main(String[] args) {
        int num = 5;
        do { System.out.println(num);
            num+=5;
        } while (num<=100);
    }
}
```

- **Prog054.java** (*do-while*) Mostrar la secuencia: 1, 2, 3, ... ,N. Donde, N es ingresado por el teclado.

Datos de entrada:
N: 6

Datos de salida:
1
2
3
4
5
6

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog054 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int num = 1, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        do { System.out.println(num);
            num++;
        }while (num<=N);
        teclado.close();
    }
}
```

- **Prog055.java** (*do-while*) Generar los número de [a , b]. Donde, a y b son ingresado por el teclado.

Datos de entrada:
a: 6
b: 12

Datos de salida:
[6 , 12]

```
6
7
8
9
10
11
12
```

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog055 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a, b;
        System.out.print("Valor a: ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b: ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "]";
        do{
            System.out.println(a);
            a++;
        }while (a <= b);
    }
}
```

- **Prog056.java (do-while)** Generar los número de [a , b]. Donde, a y b son ingresado por el teclado. Utilizar solo un *do-while*.

Datos de entrada: a: 6 b: 10	Datos de entrada: a: 10 b: 6
Datos de salida: [6 , 12] 6 7 8 9 10	Datos de salida: [12 , 6] 6 7 8 9 10

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog056 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a,b,temp;
        System.out.print("Valor a = ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b = ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
    }
}
```

```

        System.out.println "[" + a + "," + b + "];
    if(a>b) {
        temp = a;
        a = b;
        b = temp;
    }
    do{
        System.out.println(a);
        a++;
    }while (a<=b);
}
}

```

- **Prog057.java (do-while)** Mostrar números impares comprendidos entre [a, b], donde a y b, son ingresados por el teclado.

Datos de entrada: a: 6 b: 15	Datos de entrada: a: 15 b: 6
Datos de salida: [6 , 15] 7 9 11 13 15	Datos de salida: [15 , 6] 7 9 11 13 15

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog057 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a,b,temp;
        System.out.print("Valor a = ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b = ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "];
        if(a>b) {
            temp = a;
            a = b;
            b = temp;
        }
        do{
            if (a%2==1) {
                System.out.println(a);
            }
            a++;
        }while (a<=b);
    }
}

```

- **Prog058.java (do-while)** Mostrar la secuencia (*Serie Fibonacci*) de números: 0 1 1 2 3 5 8 13 de todos los números menores a N, donde N, es ingresado por el teclado.

Datos de entrada:

N: 50

Datos de Salida:

0 1 1 2 3 5 8 13 21 34

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog058 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a=0, b= 1, c, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());

        do{
            System.out.print(a + " ");
            c = a + b;
            a = b;
            b = c;
        }while (a<=N);
        teclado.close();
    }
}
```

- **Prog059.java (do-while)** Dado un número N, ingresado por el teclado, generar la tabla de multiplicar de dicho número.

Datos de entrada:

N: 6

Datos de salida:

6*1 = 6
6*2 = 12
6*3 = 18
6*4 = 24
6*5 = 30
6*6 = 36
6*7 = 42
6*8 = 48
6*9 = 54
6*10 = 60
6*11 = 66
6*12 = 72

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog059 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N, a = 1;
        System.out.print("N = ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        do{
            System.out.println(N + "*" + a + " = " + N * a);
            a++;
        }while (a <= 12);
    }
}

```

- **Prog060.java (do-while)** Dado un número N, ingresado por el teclado, generar el siguiente resultado.

Dato de entrada:

N: 8

Datos de salida:

Resultado:

8	1
7	2
6	3
5	4
4	5
3	6
2	7
1	8

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog060 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N;
        int N2 = 1;
        System.out.print("N: ");
        //Valor inicial
        N = Integer.parseInt(teclado.nextLine());
        do{
            System.out.println(N + "\t" + N2);
            N2++;
            N--;
        }while(N>=1);
        teclado.close();
    }
}

```

- **Prog061.java (do-while)** Mostrar el cuadrado de los números hasta N.

Dato de entrada:

N: 5

Datos de salida

Cuadrado de 1 = 1.0

Cuadrado de 2 = 4.0

Cuadrado de 3 = 9.0

Cuadrado de 4 = 16.0

Cuadrado de 5 = 25.0

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog061 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int num, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        num = 1;
        do{
            System.out.println("Cuadrado de " + num + " = " + Math.pow(num,2));
            num++;
        }while(num<=N);
    }
}
```

- **Prog062.java (do-while)** Hallar la sumatoria de $S = 1 + 2 + 3 + 4 + \dots + 100$.

Datos de salida

S = 5050

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog062 {
    public static void main(String[] args) {
        int a = 1, S=0;
        do{
            S = S+a;
            a++;
        }while (a<=100);
        System.out.println("S = " + S);
    }
}
```

- **Prog063.java (do-while)** Hallar la sumatoria de $S = 1 + 2 + 3 + 4 + \dots + N$, donde N es ingresado por el teclado.

Dato de entrada:
N: 50

Datos de salida
S = 1275

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog063 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a = 1, S=0, N;
        System.out.print("N = ");
        N = Integer.parseInt(teclado.nextLine());
        while (a<=N) {
            S = S+a;
            a++;
        }
        System.out.println("S = " + S);
        teclado.close();
    }
}
```

- **Prog064.java (do-while)** Hallar la factorial de un número, considerando que $F = 1 * 2 * 3 * 4 * \dots * N$, donde N es ingresado por el teclado.

Dato de entrada:
N = 10

Datos de salida
F = 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 x
F = 3628800

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog064 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N, F = 1;
        System.out.print("N = ");
        N = Integer.parseInt(teclado.nextLine());
        System.out.print("F = ");
        do {
            System.out.print(N + " x ");
            F = F * N;
            N--;
        }
    }
}
```

```

        }while(N>=1);
        System.out.println("\nF = " + F);
        teclado.close();
    }
}

```

- **Prog065.java** (do-while) Generar la tabla de multiplicar de 1 hasta N, donde N es ingresado por el teclado.

Dato de entrada:

N: 3

Datos de salida

```

1*1 = 1
1*2 = 2
1*3 = 3
1*4 = 4
1*5 = 5
1*6 = 6
1*7 = 7
1*8 = 8
1*9 = 9
1*10 = 10
1*11 = 11
1*12 = 12
2*1 = 2
2*2 = 4
2*3 = 6
2*4 = 8
2*5 = 10
2*6 = 12
2*7 = 14
2*8 = 16
2*9 = 18
2*10 = 20
2*11 = 22
2*12 = 24
3*1 = 3
3*2 = 6
3*3 = 9
3*4 = 12
3*5 = 15
3*6 = 18
3*7 = 21
3*8 = 24
3*9 = 27
3*10 = 30
3*11 = 33
3*12 = 36

```

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog065 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N, a = 1, b;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
    }
}

```

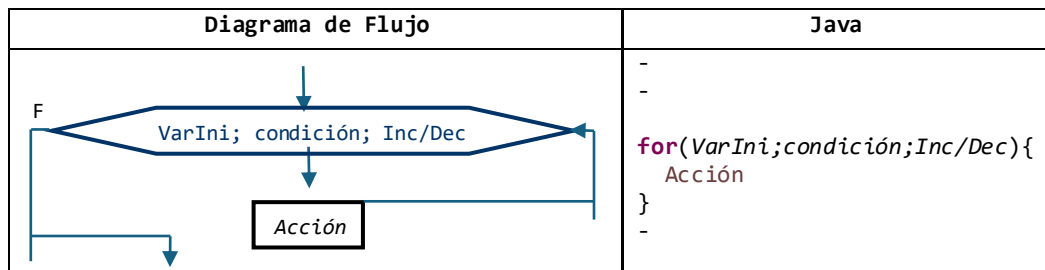
```

teclado.close();
do{
    b=1;
    do {
        System.out.println(a + "*" + b + " = " + a*b);
        b++;
    }while(b<=12);
    a++;
}while (a <= N);
}

```

5.3. Estructura for

Esta estructura es conocida por poseer una condición tripartita, pues permite tener los 3 elementos de toda estructura repetitiva: *Variable inicial, condición de salida y el incremento o decremento*; en el encabezado de su estructura.



Abraham intercede repetidamente por Sodoma

"Y se acercó Abraham y dijo: **¿Destruirás también al justo con el impío?** Quizá haya cincuenta justos dentro de la ciudad: ¿destruirás también y no perdonarás al lugar por amor a los cincuenta justos que estén dentro de él? 25 Lejos de ti el hacer tal, que hagas morir al justo con el impío, y que sea el justo tratado como el impío; nunca tal hagas. El Juez de toda la tierra, ¿no ha de hacer lo que es justo?

Entonces respondió Jehová: **Si hallare en Sodoma cincuenta justos dentro de la ciudad, perdonaré a todo este lugar por amor a ellos.**

Y Abraham replicó y dijo: He aquí ahora que he comenzado a hablar a mi Señor, aunque soy polvo y ceniza. **Quizá faltarán de cincuenta justos cinco; ¿destruirás por aquellos cinco toda la ciudad?** Y dijo: **No la destruiré, si hallare allí cuarenta y cinco.**

Y volvió a hablarle, y dijo: **Quizá se hallarán allí cuarenta.** Y respondió: **No lo haré por amor a los cuarenta.**

Y dijo: No se enoje ahora mi Señor, si hablare: **quizá se hallarán allí treinta.** Y respondió: **No lo haré si hallare allí treinta.**

Y dijo: He aquí ahora que he emprendido el hablar a mi Señor: **quizá se hallarán allí veinte.** **No la destruiré,** respondió, por amor a los veinte.

Y volvió a decir: No se enoje ahora mi Señor, si hablare solamente una vez: **quizá se hallarán allí diez.** **No la destruiré, respondió, por amor a los diez**

1 Reyes 18:42-44

- ¿Cuántos bucles tiene esta historia?

- ¿Cuál es la acción repetitiva?

- ¿Cuál sería la condición dentro del bucle?

A continuación, se mostrará los ejercicios desarrollados con la estructura `while` y `do-while`, utilizando la estructura `for`. Es recomendable que hagas una comparación de los códigos.

- **Prog066.java** (for) Mostrar números: 1, 2, 3, 4, ... ,100.

```
Datos de salida:  
1, 2, 3, 4, 5 , ... , 98 , 99 , 100 ,
```

```
package repetitivas;  
  
// Autor: Daniel Lévano  
public class Prog066 {  
    public static void main(String[] args) {  
        for (int num = 1; num <= 100; num++) {  
            System.out.print(num + ", ");  
        }  
    }  
}
```

- **Prog067.java** (for) Mostrar un mensaje 7 veces.

```
Datos de salida:  
Cristo viene pronto - Prepárate  
Cristo viene pronto - Prepárate  
Cristo viene pronto - Prepárate  
Cristo viene pronto - Prepárate  
Cristo viene pronto - Prepárate  
Cristo viene pronto - Prepárate  
Cristo viene pronto - Prepárate
```

```
package repetitivas;  
  
// Autor: Daniel Lévano  
public class Prog067 {  
    public static void main(String[] args) {  
        String mensaje = "Cristo viene pronto - Prepárate";  
        for (int num = 1; num <= 7; num++) {  
            System.out.println(mensaje);  
        }  
    }  
}
```

- **Prog068.java** (for) Mostrar números: 2, 4, 6, 8 100.

```
Datos de salida:  
2  
4  
6  
...  
...  
96  
98  
100
```

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog068 {
    public static void main(String[] args) {
        for (int num = 2; num <= 100; num+=2) {
            System.out.println(num);
        }
    }
}

```

- **Prog069.java** (for) Mostrar números: 1, 3, 5, 7 99.

```

Datos de salida:
1
3
5
...
...
97
99

```

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog069 {
    public static void main(String[] args) {
        for (int num = 1; num <= 99; num+=2) {
            System.out.println(num);
        }
    }
}

```

- **Prog070.java** (for) Mostrar números: 5,10, 15,, 100.

```

Datos de salida:
5
10
15
...
...
95
100

```

```

package repetitivas;

// Autor: Daniel Lévano
public class Prog070 {
    public static void main(String[] args) {
        for (int num = 5; num <= 100; num+=5) {
            System.out.println(num);
        }
    }
}

```

- **Prog071.java** (for) Mostrar la secuencia: 1, 2, 3, ... ,N. Donde, N es ingresado por el teclado.

Datos de entrada:

N: 6

Datos de salida:

1
2
3
4
5
6

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog071 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        for (int num = 1; num <= N; num++) {
            System.out.println(num);
        }
        teclado.close();
    }
}
```

- **Prog072.java** (for) Generar los número de [a , b]. Donde, a y b son ingresado por el teclado.

Datos de entrada:

a: 6

b: 12

Datos de salida:

[6 , 12]
6
7
8
9
10
11
12

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog072 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a, b;
        System.out.print("Valor a: ");
        a = Integer.parseInt(teclado.nextLine());
```

```

        System.out.print("Valor b: ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "];
        for (int num = a; num <= b; num++) {
            System.out.println(num);
        }
    }
}

```

- **Prog073.java** (for) En base al ejercicio anterior, generar los número de [a , b]. Donde, a y b son ingresado por el teclado de tal manera que muestre el rango de números así **a<b** o **a>b**. Utilizar solo una estructura repetitiva.

Datos de entrada: a: 6 b: 11	Datos de entrada: a: 11 b: 6
Datos de salida: [6 , 11] 6 7 8 9 10 11	Datos de salida: [11 , 6] 6 7 8 9 10 11

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog073 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a,b,temp;
        System.out.print("Valor a = ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b = ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "];
        if(a>b) {
            temp = a;
            a = b;
            b = temp;
        }
        for (int num = a; num <= b; num++) {
            System.out.println(num);
        }
    }
}

```

- **Prog074.java** (for) Mostrar números impares comprendidos entre [a, b], donde a y b, son ingresados por el teclado.

Datos de entrada: a: 6 b: 15	Datos de entrada: a: 15 b: 6
---	---

Datos de salida: [6 , 15] 7 9 11 13 15	Datos de salida: [15 , 6] 7 9 11 13 15
---	---

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog074 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int a,b,temp;
        System.out.print("Valor a: ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Valor b: ");
        b = Integer.parseInt(teclado.nextLine());
        teclado.close();
        System.out.println "[" + a + "," + b + "]";
        if(a>b) {
            temp = a;
            a = b;
            b = temp;
        }
        for (int num = a; num <= b; num++) {
            if (num%2==1) {
                System.out.println(num);
            }
        }
    }
}

```

- **Prog075.java** (for) Mostrar la secuencia (*Serie Fibonacci*) de números: 0 1 1 2 3 5 8 13 de todos los números menores a N, donde N, es ingresado por el teclado.

Datos de entrada: N: 50
Datos de salida: 0 1 1 2 3 5 8 13 21 34

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog075 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int b= 1, c, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());

        for (int a = 0; a <= N; ) {
            System.out.print(a + " ");
            c = a + b;
            a = b;
            b = c;
        }
        teclado.close();
    }
}

```

- **Prog076.java** (for) Dado un número N, ingresado por el teclado, generar la tabla de multiplicar de dicho número.

Datos de entrada:

N: 6

Datos de salida:

6*1 = 6
 6*2 = 12
 6*3 = 18
 6*4 = 24
 ...
 ...
 6*9 = 54
 6*10 = 60
 6*11 = 66
 6*12 = 72

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog076 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        for (int a = 1; a <= 12; a++) {
            System.out.println(N + "*" + a + " = " + N * a);
        }
    }
}
```

- **Prog077.java** (for) Dado un número N, ingresado por el teclado, generar el siguiente resultado.

Dato de Entrada:

N: 8

Datos de salida:

Resultado:

8	1
7	2
6	3
5	4
4	5
3	6
2	7
1	8

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog077 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N;
        int N2 = 1;
        System.out.print("N: ");
        //N es Valor inicial
        N = Integer.parseInt(teclado.nextLine());
        for (; N>=1; N--) {
            System.out.println(N + "\t" + N2);
            N2++;
        }
        teclado.close();
    }
}

```

- **Prog078.java** (for) Mostrar el cuadrado de los números hasta N.

Dato de entrada:

N: 5

Datos de salida

Cuadrado de 1 = 1.0

Cuadrado de 2 = 4.0

Cuadrado de 3 = 9.0

Cuadrado de 4 = 16.0

Cuadrado de 5 = 25.0

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog078 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        for (int num = 1; num <= N; num++) {
            System.out.println("Cuadrado de " + num + " = " + Math.pow(num, 2));
        }
    }
}

```

- **Prog079.java** (for) Hallar la sumatoria de $S = 1 + 2 + 3 + 4 + \dots + 100$.

Datos de salida
S = 5050

```
package repetitivas;

// Autor: Daniel Lévano
public class Prog079 {
    public static void main(String[] args) {
        int S=0;
        for (int a = 1; a <= 100; a++) {
            S = S+a;
        }
        System.out.println("S = " + S);
    }
}
```

- **Prog080.java** (for) Hallar la sumatoria de $S = 1 + 2 + 3 + 4 + \dots + N$, donde N es ingresado por el teclado.

Dato de entrada:
N: 50

Datos de salida
S = 1275

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog080 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int S=0, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        for (int a = 1; a <= N; a++) {
            S = S+a;
        }
        System.out.println("S = " + S);
        teclado.close();
    }
}
```

- **Prog081.java** (for) Hallar la factorial de un número, considerando que $F = 1 * 2 * 3 * 4 * \dots * N$, donde N es ingresado por el teclado.

Dato de entrada:

N = 10

Datos de salida

F = 10 x 9 x 8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 x
F = 3628800

```
package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog081 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int F=1, N;
        System.out.print("N: ");
        N = Integer.parseInt(teclado.nextLine());
        System.out.print("F = ");
        for (; N>=1; N--) {
            System.out.print(N + " x ");
            F *= N;
        }
        System.out.println("\nF = " + F);
        teclado.close();
    }
}
```

- **Prog082.java** (for) Generar la tabla de multiplicar de 1 hasta N, donde N es ingresado por el teclado.

Dato de entrada:

N: 3

Datos de salida

```
1*1 = 1
1*2 = 2
1*3 = 3
1*4 = 4
1*5 = 5
1*6 = 6
1*7 = 7
1*8 = 8
1*9 = 9
1*10 = 10
1*11 = 11
1*12 = 12
2*1 = 2
2*2 = 4
2*3 = 6
2*4 = 8
2*5 = 10
2*6 = 12
2*7 = 14
2*8 = 16
```

```

2*9 = 18
2*10 = 20
2*11 = 22
2*12 = 24
3*1 = 3
3*2 = 6
3*3 = 9
3*4 = 12
3*5 = 15
3*6 = 18
3*7 = 21
3*8 = 24
3*9 = 27
3*10 = 30
3*11 = 33
3*12 = 36

```

```

package repetitivas;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Prog082 {
    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        int N ;
        System.out.print("N = ");
        N = Integer.parseInt(teclado.nextLine());
        teclado.close();
        for (int a = 1; a <=N ; a++) {
            for (int b = 1; b <= 12; b++) {
                System.out.println(a + "*" + b + " = " + a*b);
            }
        }
    }
}

```

5.4. Ejercicios propuestos 04

Desarrollar los ejercicios propuestos utilizando la estructura repetitiva de su preferencia.

- **Prog083.java** Codificar la operación, donde n es ingresado por el teclado.

$$S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n-1} + \frac{1}{n}$$

Datos de entrada:

N: 9

Datos de salida:

S = 2.82896

- **Prog084.java** Codificar la operación, donde n es ingresado por el teclado.

$$M = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \dots + \frac{n-1}{n} + \frac{n}{n+1}$$

Datos de entrada:

N: 9

Datos de salida:

S = 7.07103

- **Prog085.java** Codificar la operación, donde n es ingresado por el teclado.

$$S = \frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \frac{4}{4!} + \dots + \frac{n-1}{(n-1)!} + \frac{n}{n!}$$

Datos de entrada:

N: 6

Datos de salida:

S = 2.71666

- **Prog086.java** Codificar la operación, donde n es ingresado por el teclado.

$$S = \frac{2!}{3!} + \frac{3!}{4!} + \frac{4!}{5!} + \dots + \frac{n!}{(n+1)!} + \frac{(n+1)!}{(n+2)!}$$

Datos de entrada:

N: 6

Datos de salida:

S = 1.21785

- **Prog087.java** Hallar la sumatoria de la secuencia (*Serie Fibonacci*) de números: 0 1 1 2 3 5 8 13 de todos los números menores a N, donde N, es ingresado por el teclado.

Datos de entrada:

N: 50

Datos de salida:

S = 0 + 1 + 1 + 2 + 3 + 5 + 8 + 13 + 21 + 34

S = 88

Enunciado:

Solución:

5.5. Práctica de autoevaluación 03

Crear un package: practica03

Tiempo: 60 minutos

- **Preg01.java** Hallar la sumatoria:

$$S = \frac{3!}{5!} + \frac{4!}{6!} + \frac{5!}{7!} + \dots + \frac{(n+2)!}{(n+4)!}$$

Datos de entrada:

N: 6

Datos de salida:

S = 1.21785

David pide ser examinado por Dios:

"Examíname, oh Dios, y conoce mi corazón; pruébame y conoce mis pensamientos. Y ve si hay en mí camino de perversidad, y guíame en el camino eterno."

Salmo 139:23, 24

Es importante ser evaluados para identificar oportunidades de mejora.

- **Preg02.java** Separar un número en los dígitos que lo componen

Datos de entrada: N: 1561	Datos de entrada: N: 25	Datos de entrada: N: 25500
Datos de salida: 1 5 6 1	Datos de salida: 2 5	Datos de salida: 2 5 5 0 0

- **Preg03.java** Mostrar si un número es capicúa. Un número es capicúa si se lee igual de izquierda a derecha o de derecha a izquierda.

Datos de entrada: N: 1561	Datos de entrada: N: 252	Datos de entrada: N: 1001
Datos de salida: No es capicúa	Datos de salida: Si es capicúa	Datos de salida: Si es capicúa

- **Preg04.java** Mostrar si un número es perfecto. Un número es perfecto cuando la suma de sus divisores da el mismo número.

Datos de entrada: N: 1561	Datos de entrada: N: 496	Datos de entrada: N: 28
Datos de salida: No es capicúa	Datos de salida: Si es perfecto	Datos de salida: Si es perfecto

Unidad II - Sesión 3 Y 4

6. Modularidad

Permite la agrupación de código para facilitar la reutilización del mismo. En Java se les conoce como métodos y se dispone de 4 tipos: Métodos vacíos, métodos que reciben parámetros, métodos que devuelven parámetros y, finalmente, los métodos que reciben y devuelven parámetros.

El cuerpo humano

"Porque de la manera que en un cuerpo tenemos muchos miembros, pero no todos los miembros tienen la misma función."

Romanos 12:4

6.1. Métodos vacíos

- **Prog088.java** (*M. Vacío*) Diseñe un método que muestre un mensaje de saludo.

```
Datos de salida:  
Bienvenidos a la Universidad  
Esperamos sea de su agrado...  
Dios los bendiga...
```

```
package modularidad;  
  
// Autor: Flor Cerdán  
public class Prog087 {  
    public static void saludo() {  
        System.out.println("Bienvenidos a la Universidad");  
        System.out.println("Esperamos sea de su agrado...");  
        System.out.println("Dios los bendiga...");  
    }  
  
    public static void main(String[] args) {  
        saludo();  
    }  
}
```

- **Prog089.java** (*M. Vacío*) Diseñar un programa que tenga los siguientes métodos: bienvenida(), mensaje() y despedida()

```
Datos de salida:  
Bienvenidos al sistema...  
  
Mira que te mando que te esfuerces y seas valiente;  
no temas ni desmayes, porque Jehová tu Dios  
estará contigo en dondequiera que vayas. Josué 1:9  
  
Gracias por su visita...
```

```

package modularidad;

// Autor: Flor Cerdán
public class Prog089 {
    public static void bienvenida() {
        System.out.println("Bienvenidos al sistema...");
    }

    public static void mensaje() {
        System.out.println("\nMira que te mando que te "
            + "esfuerces y seas valiente; "
            + "no temas ni desmayes, "
            + "porque Jehová tu Dios "
            + "estará contigo en "
            + "dondequiera que vayas. Josué 1:9");
    }

    public static void despedida() {
        System.out.println("\nGracias por su visita...");
    }

    public static void main(String[] args) {
        bienvenida();
        mensaje();
        despedida();
    }
}

```

6.2. Métodos que reciben parámetros

- **Prog090.java** (*M. Reciben Parámetros*) Diseñar un método que permita recibir un nombre, e imprima un saludo personalizado:

<p>Datos de entrada: Nombre: Flor Elizabeth</p> <p>Datos de salida: Bienvenido(a) Flor Elizabeth</p>
--

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog090 {

    public static void saludo(String nombre) {
        System.out.println("Bienvenido(a) " + nombre);
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        String nombre;
        System.out.print("Nombre: ");
        nombre = teclado.nextLine();
        saludo(nombre);
    }
}

```

```

        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog091.java** (M. Reciben Parámetros) Diseñar un método llamado potencia(int a, int b)

Datos de entrada:

Num: 5
Potencia: 6

Datos de salida:

5 elevado a la 6 es = 15625

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog091 {
    public static void potencia(int a, int b) {
        int pot = 1;
        for(int i = 1; i<= b ; i++) {
            pot *= a;
        }
        System.out.println(a + " elevado a la " + b + " es = " + pot);
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        int a,b;
        System.out.print("Num: ");
        a = Integer.parseInt(teclado.nextLine());
        System.out.print("Potencia: ");
        b = Integer.parseInt(teclado.nextLine());
        potencia(a,b);
        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog092.java** (*M. Reciben Parámetros*) Diseñar un método llamado **factorial(int num)**, donde num es ingresado por el teclado.

Datos de entrada:

Num = 5

Datos de salida:

5! = 120

```
package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog092 {
    public static void factorial(int num) {
        int fac = 1;
        for(int i = 1; i<= num ; i++) {
            fac *= i;
        }
        System.out.println(num + "! = " + fac);
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        int num;
        System.out.print("Num: ");
        num = Integer.parseInt(teclado.nextLine());
        factorial(num);
        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}
```

- **Prog093.java** (*M. Reciben Parámetros*) Diseñar un método llamado analizar(int num). que muestre:
 - Par o impar
 - Positivo, negativo o neutro
 - Primo o no primo

Datos de entrada:

Numero: 100

Datos de salida:

Par
Positivo
No es Primo

Datos de entrada:

Numero: -7

Datos de salida:

Impar
Negativo
Primo

Datos de entrada:

Numero: 31

Datos de salida:

Impar
Positivo
Primo

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog093 {
    public static void analizar(int num) {
        int a = 0;
        if (num % 2 == 0) {
            System.out.println("Par");
        } else {
            System.out.println("Impar");
        }
        if (num < 0) {
            System.out.println("Negativo");
        } else {
            if (num > 0) {
                System.out.println("Positivo");
            } else {
                System.out.println("Neutro");
            }
        }
        for (int i = 1; i <= num && a<=2; i++) {
            if (num % i == 0) {
                a++;
            }
        }
        if (a <= 2) {
            System.out.println("Primo");
        } else {
            System.out.println("No es Primo");
        }
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        int num;
        System.out.print("Numero: ");
        num = Integer.parseInt(teclado.nextLine());
        analizar(num);
        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog094.java** (*M. Reciben Parámetros*) Diseñar el método decimalabinario(num), que permita convertir un número de base 10 a su equivalente binario.

Datos de entrada: Numero: 12	Datos de entrada: Numero: 7	Datos de entrada: Numero: 29
Datos de salida: 1100	Datos de salida: 111	Datos de salida: 11101

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog094 {
    public static void decimalbinario(int num) {
        int residuo = 0, potencia = 0, resultado = 0;
        do {
            residuo = num % 2;
            num = num / 2;
            resultado += residuo * (int)Math.pow(10, potencia);
            potencia++;
        }while(num>0);
        System.out.println(resultado);
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        int num;
        System.out.print("Numero: ");
        num = Integer.parseInt(teclado.nextLine());
        decimalbinario(num);
        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog095.java** (*M. Reciben Parámetros*) Diseñar el método decimalaooctal(num), que permita convertir un número de base 10 a su equivalente en base 8.

Datos de entrada: Numero: 12	Datos de entrada: Numero: 7	Datos de entrada: Numero: 29
Datos de salida: 1100	Datos de salida: 111	Datos de salida: 11101

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog095 {
    public static void decimaloocal(int num) {
        int residuo = 0, potencia = 0, resultado = 0;
        do {
            residuo = num % 8;
            num = num / 8;
            resultado += residuo * (int)Math.pow(10, potencia);
            potencia++;
        }while(num>0);
        System.out.println(resultado);
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        int num;
        System.out.print("Numero: ");
        num = Integer.parseInt(teclado.nextLine());
        decimaloocal(num);
    }
}

```

```
        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}
```

6.3. Métodos que devuelven parámetros

- **Prog096.java** (*M. Devuelven Parámetros*) Diseñar un método `saludo()`, que permita devolver un mensaje de bienvenida al programa.

```
Datos de salida:
BIENVENIDOS AL SISTEMA
2021
```

```
package modularidad;

// Autor: Flor Cerdán
public class Prog096 {
    public static String saludo() {
        String mensaje = "Bienvenidos al Sistema";
        return mensaje;
    }

    public static int anioactual() {
        return 2021;
    }

    public static void main(String[] args) {
        System.out.println(saludo().toUpperCase());
        System.out.println(anioactual());
    }
}
```

6.4. Métodos que reciben y devuelven parámetros

- **Prog097.java** (*M. Reciben y Devuelven Parámetros*) Diseñar el método `factorial(int num)`, que devuelva su resultado.

Datos de entrada: Numero: 6	Datos de entrada: Numero: 10
Datos de salida: 6!= 720	Datos de salida: 10!= 3628800

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog097 {
    public static int factorial(int num) {
        int fac = 1;
        for (int i = 1; i <= num; i++) {
            fac*=i;
        }
        return fac;
    }

    public static void inicio() {
        Scanner teclado = new Scanner(System.in);
        int num;
        System.out.print("Numero: ");
        num = Integer.parseInt(teclado.nextLine());
        System.out.println(num + "! = " + factorial(num));
        teclado.close();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog098.java** (*M. Reciben y Devuelven Parámetros*) Diseñar el método `analizarprimo(int num)`, que devuelva si un número es “Primo” o “No Primo”.

Datos de entrada: Numero: 11	Datos de entrada: Numero: 20
Datos de salida: 11 es Primo	Datos de salida: 20 es No Primo

```

package modularidad;
import java.util.Scanner;

// Autor: Flor Cerdán
public class Prog098 {
    public static String primo(int num) {
        String resultado="";
        int contceros=0;
        for(int i=1; i<=num && contceros<=2 ; i++) {
            if(num%i==0) {
                contceros++;
            }
        }
        if(contceros<=2) {
            resultado = "Primo";
        }else {

```

```
        resultado = "No Primo";
    }
    return resultado;
}

public static void inicio() {
    Scanner teclado = new Scanner(System.in);
    int num;
    System.out.print("Numero: ");
    num = Integer.parseInt(teclado.nextLine());
    System.out.println(num + " es " + primo(num));
    teclado.close();
}

public static void main(String[] args) {
    inicio();
}
}
```

Unidad II – Sesión 5

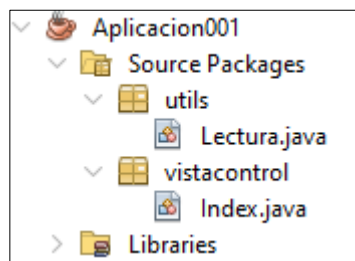
6.5. Aplicación

- Diseñe un menú de opciones para desarrollar ejercicios matemáticos.

```
**** MENU PRINCIPAL ****
* 1. Factorial          *
* 2. Mayor de Edad     *
* 3. Par - Impar       *
* 4. Primo              *
* 5. Salir              *
Seleccione opción [1-5]:
```

Solución:

Para ellos se debe crear un nuevo proyecto llamado: **Aplicación001**, y un package llamado: **utils** que contenga la clase **Lectura.java** y otro package llamado: **vistacontrol** que contenga la clase **Index.java**, como se muestra a continuación:



Lectura.java

```
package utils;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Lectura {
    private Scanner teclado = new Scanner(System.in);

    public String cadena() {
        return teclado.nextLine();
    }

    public String cadenaMayuscula() {
        return cadena().toUpperCase();
    }

    public String cadenaMinuscula() {
        return cadena().toLowerCase();
    }

    public int entero() {
        return Integer.parseInt(cadena());
    }

    public double decimal() {
        return Double.parseDouble(cadena());
    }

    public char letra() {
```

```

        return cadena().charAt(0);
    }

    public void cerrar() {
        teclado.close();
    }
}

```

```

package vistacontrol;
import utils.Lectura;

// Autor: Flor Cerdán
public class Index {

    private static Lectura Leer = new Lectura();

    public static void factorial(){
        System.out.println("*** FACTORIAL ***");
        int fac=1, numero;
        System.out.print("Número: ");
        numero = Leer.entero();
        for (int i = 1; i <= numero; i++) {
            fac*=i;
        }
        System.out.println("Resultado");
        System.out.println("> " + numero + " != " + fac);
    }

    public static void mayordeedad(){
        System.out.println("*** MAYOR DE EDAD ***");
        int edad;
        System.out.print("Edad: ");
        edad = Leer.entero();
        System.out.println("Resultado: ");
        if (edad>=18) {
            System.out.println("> Mayor de Edad");
        }else if (edad>=0) {
            System.out.println("> Menor de Edad");
        }else{
            System.out.println("<< No es una edad válida >>");
        }
    }

    public static void parimpar(){
        System.out.println("*** PAR O IMPAR ***");
        int numero;
        System.out.print("Número: ");
        numero = Leer.entero();
        System.out.println("Resultado: ");
        if (numero%2==1) {
            System.out.println("> Impar");
        }else if (numero == 0) {
            System.out.println("> Neutro");
        }else{
            System.out.println("> Par");
        }
    }

    public static void primo(){
        System.out.println("*** PRIMO O NO PRIMO ***");
        String resultado="";
        int contceros=0, num;
        System.out.print("Número: ");
    }
}

```

```

        num = Leer.entero();
        for (int i = 1; i <= num && contceros <= 2; i++) {
            if (num % i == 0) {
                contceros++;
            }
        }
        if (contceros <= 2) {
            resultado = "Primo";
        } else {
            resultado = "No Primo";
        }
        System.out.println(num + " es " + resultado);
    }

    public static void salir(){
        System.out.println("Gracias por su visita");
    }

    public static void error(){
        System.out.println("Opción incorrecta");
    }

    public static void menu(){
        System.out.println("*** MENU PRINCIPAL ***");
        System.out.println("1. Factorial");
        System.out.println("2. Mayor de Edad");
        System.out.println("3. Par - Impar");
        System.out.println("4. Primo");
        System.out.println("5. Salir");
        System.out.print("Seleccione una opción [1-5]: ");
    }

    public static void inicio(){
        int opcion;
        do {
            menu();
            opcion = Leer.entero();
            switch (opcion) {
                case 1:
                    factorial();
                    break;
                case 2:
                    mayordeedad();
                    break;
                case 3:
                    parimpar();
                    break;
                case 4:
                    primo();
                    break;
                case 5:
                    salir();
                    break;
                default:
                    error();
            }
        } while (opcion!=5);
        Leer.cerrar();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

Unidad III - Sesión 1

7. Arreglos

Son estructuras que permiten almacenar varios valores en una sola variable. Todos los valores deben ser del mismo tipo de variable. En Java la sintaxis para definir una variable es:

```
TipoVariable[] variable = new TipoVariable[Tamaño];
```

7.1. Arreglos unidimensionales

- **Lectura.java** Antes de continuar, su proyecto debe tener la clase **Lectura.java** dentro de un package llamado: **utils**. A continuación, mostramos el código de la clase **Lectura.java**:

```
package utils;
import java.util.Scanner;

// Autor: Daniel Lévano
public class Lectura {
    private Scanner teclado = new Scanner(System.in);

    public String cadena() {
        return teclado.nextLine();
    }

    public String cadenaMayuscula() {
        return cadena().toUpperCase();
    }

    public String cadenaMinuscula() {
        return cadena().toLowerCase();
    }

    public int entero() {
        return Integer.parseInt(cadena());
    }

    public double decimal() {
        return Double.parseDouble(cadena());
    }

    public char letra() {
        return cadena().charAt(0);
    }

    public void cerrar() {
        teclado.close();
    }
}
```

- **Prog099.java** (Arreglos) Diseñar un algoritmo que permita el ingreso de 5 valores a un arreglo y muestre los valores asignados.

Dato de salida:
 > numeros[0]: 1
 > numeros[1]: 10
 > numeros[2]: 50
 > numeros[3]: 11000
 > numeros[4]: 33

```
package arreglos;

// Autor: Daniel Lévano
public class Prog099 {
    public static void inicio() {
        int[] numeros = new int[5];
        numeros[0] = 1;
        numeros[1] = 10;
        numeros[2] = 50;
        numeros[3] = 11000;
        numeros[4] = 33;
        System.out.println("> numeros[0]: " + numeros[0]);
        System.out.println("> numeros[1]: " + numeros[1]);
        System.out.println("> numeros[2]: " + numeros[2]);
        System.out.println("> numeros[3]: " + numeros[3]);
        System.out.println("> numeros[4]: " + numeros[4]);
    }

    public static void main(String[] args) {
        inicio();
    }
}
```

- **Prog100.java** (Arreglos) Diseñar un algoritmo que permita el ingreso de 7 valores por el teclado a un arreglo y muestre el resultado.

<p>Dato de entrada: Tamaño: 7 Ingresar valores: > numeros[0]: 10 > numeros[1]: -2 > numeros[2]: 30 > numeros[3]: -40 > numeros[4]: 50 > numeros[5]: 0 > numeros[6]: 70</p>	<p>Dato de salida: Resultado: * numeros[0]: 10 * numeros[1]: -2 * numeros[2]: 30 * numeros[3]: -40 * numeros[4]: 50 * numeros[5]: 0 * numeros[6]: 70</p>
---	---

```
package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog100 {
    private static Lectura Leer = new Lectura();

    public static void inicio() {
        int[] numeros = new int[7];
        System.out.println("Tamaño del arreglo: " + numeros.length);
        System.out.println("\nIngrese los valores: ");
        for (int i = 0; i < numeros.length; i++) {
```

```

        System.out.print("> numeros[" + i + "]: ");
        numeros[i] = Leer.entero();
    }
    System.out.println("\nResultado:");
    for (int i = 0; i < numeros.length; i++) {
        System.out.println("* numeros[" + i + "]: " + numeros[i]);
    }
    Leer.cerrar();
}

public static void main(String[] args) {
    inicio();
}
}

```

- **Prog101.java (Arreglos)** Utilizando métodos, diseñe un programa que permita el ingreso de valores a un arreglo y muestre los valores ingresados.

<p>Dato de entrada: > ¿Cuántos elementos desea ingresar?: 5</p> <p>Agregar valores al arreglo: > numeros[0]: 10 > numeros[1]: -50 > numeros[2]: 10 > numeros[3]: 500 > numeros[4]: 0</p>	<p>Dato de salida:</p> <p>Valores del arreglo: * numeros[0]: 10 * numeros[1]: -50 * numeros[2]: 10 * numeros[3]: 500 * numeros[4]: 0</p>
---	--

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog101 {
    // Variables Globales
    private static Lectura Leer = new Lectura();
    private static int[] numeros = new int[1000];
    private static int cantidad;

    public static void escribirarreglo() {
        System.out.println("\nAgregar valores al arreglo: ");
        for (int i = 0; i < cantidad; i++) {
            System.out.print("> numeros[" + i + "]: ");
            numeros[i] = Leer.entero();
        }
    }

    public static void imprimirarreglo() {
        System.out.println("\nValores del arreglo:");
        for (int i = 0; i < cantidad; i++) {
            System.out.println("* numeros[" + i + "]: " + numeros[i]);
        }
    }

    public static void inicio() {
        System.out.print("> ¿Cuántos elementos desea ingresar?: ");
        cantidad = Leer.entero();
        escribirarreglo();
        imprimirarreglo();
        Leer.cerrar();
    }
}

```

```

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog102.java (Arreglos)** Desarrolle un programa que permita el ingreso de valores a un arreglo y muestre la sumatoria de sus elementos. Utilice métodos.

Dato de entrada:

> Cantidad de elementos: 5

Agregar valores al arreglo:

> numeros[0]: 10

> numeros[1]: 20

> numeros[2]: 30

> numeros[3]: 10

> numeros[4]: 20

Dato de salida:

* Resultado (Sumatoria): 90

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog102 {

    // Variables Globales
    private static Lectura Leer = new Lectura();
    private static int[] numeros = new int[1000];
    private static int cantidad;

    public static void escribirarreglo() {
        System.out.println("\nAgregar valores al arreglo: ");
        for (int i = 0; i < cantidad; i++) {
            System.out.print("> numeros[" + i + "]: ");
            numeros[i] = Leer.entero();
        }
    }

    public static void sumatoria() {
        int suma = 0;
        for (int i = 0; i < cantidad; i++) {
            suma += numeros[i];
        }
        System.out.println("\n* Resultado (Sumatoria): " + suma);
    }

    public static void inicio() {
        System.out.print("> Cantidad de elementos: ");
        cantidad = Leer.entero();
        escribirarreglo();
        sumatoria();
        Leer.cerrar();
    }

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog103.java (Arreglos)** Desarrolle un programa que permita el ingreso de valores a un arreglo y muestre el factorial de cada elemento. Utilice métodos.

```

Dato de entrada:
Cantidad de elementos: 5

Agregar valores al arreglo:
numeros[0]= 10
numeros[1]= 12
numeros[2]= 5
numeros[3]= 6
numeros[4]= 4

Dato de salida:
Resultado:
10!= 3628800
12!= 479001600
5!= 120
6!= 720
4!= 24

```

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog103 {

    // Variables Globales
    private static int[] numeros = new int[1000];
    private static int cantidad;
    private static Lectura Leer = new Lectura();

    public static void escribirarreglo(){
        System.out.println("\nAgregar valores al arreglo: ");
        for (int i = 0; i < cantidad; i++) {
            System.out.print("numeros[" + i + "]: " );
            numeros[i] = Leer.entero();
        }
    }

    public static void imprimirarreglo(){
        System.out.println("\nResultado: ");
        for (int i = 0; i < cantidad; i++) {
            System.out.println(numeros[i] + "!= " + factorial(numeros[i]));
        }
    }

    public static int factorial(int numero){
        int fac = 1;
        for (int i = 1; i <= numero; i++) {
            fac*=i;
        }
        return fac;
    }

    public static void inicio(){
        System.out.print("Cantidad de Elementos: ");
        cantidad = Leer.entero();
        escribirarreglo();
        imprimirarreglo();
        Leer.cerrar();
    }
}

```

```

    public static void main(String[] args) {
        inicio();
    }
}

```

- **Prog104.java (Arreglos)** Desarrolle un programa que permita el ingreso de valores a un arreglo y muestre los valores ingresados indicando si el número es PAR/IMP/NEUTRO y POSITIVO/NEGATIVO/NEUTRO

Dato de entrada:
Cantidad: 5
Agregar valores al arreglo:
numeros[0]: 10
numeros[1]: 5
numeros[2]: -10
numeros[3]: -5
numeros[4]: 0

Dato de salida:
10: Par - Positivo
5: Impar - Positivo
-10: Par - Negativo
-5: Impar - Negativo
0: Neutro - Neutro

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog104 {

    // Variables Globales
    private static int[] numeros = new int[1000];
    private static int cantidad;
    private static Lectura Leer = new Lectura();

    public static void escribirarreglo() {
        System.out.println("\nAgregar Valores al arreglo");
        for (int i = 0; i < cantidad; i++) {
            System.out.print("Numeros[" + i + "]: ");
            numeros[i] = Leer.entero();
        }
    }

    public static void imprimirarreglo() {
        System.out.println("\nResultado: ");
        for (int i = 0; i < cantidad; i++) {
            System.out.println(numeros[i] + ": "
                + parimpar(numeros[i])
                + " - "
                + positivonegativo(numeros[i]));
        }
    }

    public static String parimpar(int numero) {
        String resultado;
        if (numero % 2 == 1) {
            resultado = "Impar";
        } else if (numero == 0) {
            resultado = "Neutro";
        } else {
            resultado = "Par";
        }
    }
}

```

```

    }
    return resultado;
}

public static String positivonegativo(int numero) {
    String resultado;
    if (numero > 0) {
        resultado = "Positivo";
    } else if (numero < 0) {
        resultado = "Negativo";
    } else {
        resultado = "Neutro";
    }
    return resultado;
}

public static void inicio() {
    System.out.println("Cantidad de elementos: ");
    cantidad = Leer.entero();
    escribirarreglo();
    imprimirarreglo();
    Leer.cerrar();
}

public static void main(String[] args) {
    inicio();
}
}

```

- **Prog105.java (Arreglos)** Eliminar elementos de un arreglo.

```

Dato de entrada:
Cantidad: 5
Numeros[0]: 10
Numeros[1]: 4
Numeros[2]: -6
Numeros[3]: 12
Numeros[4]: 100

Dato de salida:
Los valores del arreglo son:
Numeros[0]= 10
Numeros[1]= 4
Numeros[2]= -6
Numeros[3]= 12
Numeros[4]= 100

Dato de entrada:
Indique la posición a eliminar: 2

Dato de salida:
Los valores del arreglo son:
Numeros[0]= 10
Numeros[1]= 4
Numeros[2]= 12
Numeros[3]= 100

```

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog105 {

```

```

// Variables Globales
private static int[] numeros = new int[1000];
private static int cantidad;
private static Lectura Leer = new Lectura();

public static void escribirarreglo() {
    System.out.println("\nAgregar valores al arreglo: ");
    for (int i = 0; i < cantidad; i++) {
        System.out.print("Numeros[" + i + "]: ");
        numeros[i] = Leer.entero();
    }
}

public static void listararreglo() {
    System.out.println("\nValores del Arreglo: ");
    for (int i = 0; i < cantidad; i++) {
        System.out.println("Numeros[" + i + "]:" + numeros[i]);
    }
}

public static void eliminarelemento() {
    int posicion;
    listararreglo();
    System.out.print("\nIndique la posición a eliminar: ");
    posicion = Leer.entero();
    for (int i = posicion; i < cantidad - 1; i++) {
        numeros[i] = numeros[i + 1];
    }
    numeros[cantidad - 1] = 0;
    cantidad--;
}

public static void inicio() {
    System.out.print("Cantidad de elementos: ");
    cantidad = Leer.entero();
    escribirarreglo();
    eliminarelemento();
    listararreglo();
    Leer.cerrar();
}

public static void main(String[] args) {
    inicio();
}
}

```

- **Prog106.java** Ingresar números a un arreglo mientras este sea diferente de cero, y mostrar el resultado.

Datos de entrada

```

10
20
15
50
10
15
0

```

Datos de salida

```
Numeros[0]= 10
Numeros[1]= 20
Numeros[2]= 15
Numeros[3]= 50
Numeros[4]= 10
Numeros[5]= 15
Numeros[6]= 0
```

```
package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog106 {

    // Variables Globales
    private static Lectura Leer = new Lectura();
    private static int[] numeros = new int[100];
    private static int cantidad = 0;

    public static void ingresarvalores() {
        do {
            numeros[cantidad] = Leer.entero();
            cantidad++;
        } while (numeros[cantidad-1]!=0);
    }

    public static void mostrararreglo() {
        for (int i = 0; i < cantidad; i++) {
            System.out.println("Numeros[" + i + "] = " + numeros[i]);
        }
    }

    public static void main(String[] args) {
        ingresarvalores();
        mostrararreglo();
        Leer.cerrar();
    }
}
```

- **Prog107.java** Generar la serie fibanacci de todos los números menores a N, y registrarlos en un arreglo.

Datos de entrada

Límite: 10

Datos de salida

```
fibonacci[0]= 0
fibonacci[1]= 1
fibonacci[2]= 1
fibonacci[3]= 2
fibonacci[4]= 3
fibonacci[5]= 5
fibonacci[6]= 8
```

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog107 {

    // Variables Globales
    private static Lectura Leer = new Lectura();
    public static int[] fibonacci = new int[200];
    public static int posicion = 0;

    public static void fibo() {
        int a = 0, b = 1, c;
        int limite;
        System.out.print("Límite: ");
        limite = Leer.entero();
        while (a <= limite) {
            fibonacci[posicion] = a;
            posicion++;
            c = a + b;
            a = b;
            b = c;
        }
    }

    public static void listararreglo() {
        for (int i = 0; i < posicion; i++) {
            System.out.println("fibonacci[" + i + "]= " + fibonacci[i]);
        }
    }

    public static void main(String[] args) {
        fibo();
        listararreglo();
        Leer.cerrar();
    }
}

```

- **Prog108.java** Ingresar valores a un arreglo mientras este sea diferente de cero, luego ingresar por el teclado un valor y mostrar cuántas veces se repite ese elemento en el arreglo.

Datos de entrada

```

10
15
20
15
10
150
0

```

Datos de salida

```

numero[0]= 10
numero[1]= 15
numero[2]= 20
numero[3]= 15
numero[4]= 10
numero[5]= 150
numero[6]= 0

```

Datos de entrada

Buscar: 10

Datos de salida

Resultado: 2 veces

```
package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog108 {

    // Variables Globales
    private static Lectura Leer = new Lectura();
    public static int[] numeros = new int[100];
    public static int posicion = 0;

    public static void ingresarvalores() {
        do {
            numeros[posicion] = Leer.entero();
            posicion++;
        } while (numeros[posicion] - 1 != 0);
    }

    public static void listararreglo() {
        for (int i = 0; i < posicion; i++) {
            System.out.println("numero[" + i + "] = " + numeros[i]);
        }
    }

    public static void buscarelemento() {
        int contador = 0;
        System.out.print("\nBuscar: ");
        int elemento = Leer.entero();
        for (int i = 0; i < posicion; i++) {
            if (elemento == numeros[i]) {
                contador++;
            }
        }
        System.out.println("\nResultado: " + contador + " veces");
    }

    public static void main(String[] args) {
        ingresarvalores();
        listararreglo();
        buscarelemento();
        Leer.cerrar();
    }
}
```

- **Prog109.java** Permitir el ingreso de una cadena por teclado, copiar cada letra en un arreglo de char y mostrar el resultado.

Datos de entrada

El Señor es mi pastor

Datos de salida

E l S e ñ o r e s m i p a s t o r

```

package arreglos;
import utils.Lectura;

// Autor: Daniel Lévano
public class Prog109 {

    // Variables Globales
    private static Lectura Leer = new Lectura();
    private static char[] aCaracteres = new char[200];

    public static void leerTexto() {
        System.out.println("Escriba una palabra: ");
        String texto = Leer.cadena();
        descomponer(texto);
    }

    public static void descomponer(String texto) {
        aCaracteres = new char[texto.length()];
        for (int pos = 0; pos < texto.length(); pos++) {
            aCaracteres[pos] = texto.charAt(pos);
        }
    }

    public static void imprimir() {
        for (int x = 0; x < aCaracteres.length; x++) {
            System.out.print(aCaracteres[x] + " ");
        }
    }

    public static void main(String[] args) {
        LeerTexto();
        imprimir();
        Leer.cerrar();
    }
}

```

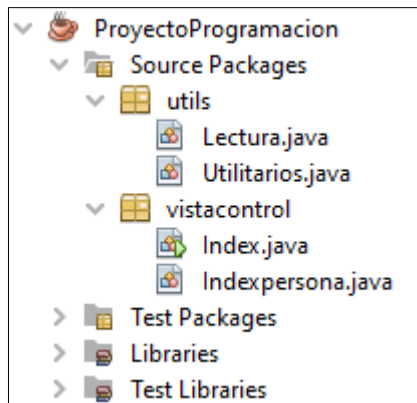
Unidad III - Sesión 2 y 3

7.2. Aplicación

- Utilizando conceptos de arreglos, desarrollar el CRUD de Persona. Considere: DNI, Nombre, Fecha de Nacimiento.

```
*** MENU ***  
1. Agregar Persona  
2. Eliminar Persona  
3. Modificar Persona  
4. Listar Persona  
5. Salir
```

Arquitectura de la Solución:



Lectura.java

```
package utils;  
import java.util.Scanner;  
  
// Autor: Daniel Lévano y Flor Cerdán  
public class Lectura {  
    private Scanner teclado = new Scanner(System.in);  
  
    public String cadena() {  
        return teclado.nextLine();  
    }  
  
    public String cadenaMayuscula() {  
        return cadena().toUpperCase();  
    }  
  
    public String cadenaMinuscula() {  
        return cadena().toLowerCase();  
    }  
  
    public int entero() {  
        return Integer.parseInt(cadena());  
    }  
  
    public double decimal() {  
        return Double.parseDouble(cadena());  
    }  
  
    public char letra() {
```

```

        return cadena().charAt(0);
    }

    public void cerrar() {
        teclado.close();
    }
}

```

Utilitarios.java

```

package utils;

// Autor: Daniel Lévano y Flor Cerdán
public class Utilitarios {
    public static void lineasenblanco() {
        System.out.println("\n\n");
    }

    public static void salir() {
        System.out.println("Gracias por su visita");
    }

    public static void error() {
        System.out.println("Error, opción inválida");
    }
}

```

Indexpersona.java

```

package vistacontrol;
import utils.Lectura;
import utils.Utilitarios;

// Autor: Daniel Lévano y Flor Cerdán
public class Indexpersona {

    // Variables Globales
    public static Lectura Leer = new Lectura();
    public static String[] dni = new String[100];
    public static String[] nombre = new String[100];
    public static String[] fecha = new String[100];
    public static int cantidad = -1;

    public static void cargardatosdeinstalacion() {
        cantidad++;
        dni[cantidad] = "00000001";
        nombre[cantidad] = "Mario";
        fecha[cantidad] = "10/10/2010";

        cantidad++;
        dni[cantidad] = "00000020";
        nombre[cantidad] = "Carlos";
        fecha[cantidad] = "01/10/2020";

        cantidad++;
        dni[cantidad] = "00000111";
        nombre[cantidad] = "Samuel";
        fecha[cantidad] = "20/01/2000";

        cantidad++;
        dni[cantidad] = "22222222";
        nombre[cantidad] = "Rosa";
        fecha[cantidad] = "30/12/2012";

        cantidad++;
        dni[cantidad] = "12345678";
        nombre[cantidad] = "Roberto";
        fecha[cantidad] = "15/05/2005";
    }

    public static void agregar() {
        System.out.println("-- REGISTRAR NUEVA PERSONA --");
        cantidad++;
        System.out.print("> DNI: ");
        dni[cantidad] = Leer.cadena();
        System.out.print("> Nombre: ");
    }
}

```

```

        nombre[cantidad] = Leer.cadena();
        System.out.print("> Fecha de Nacimiento: ");
        fecha[cantidad] = Leer.cadena();
    }

    public static void eliminar() {
        System.out.println("-- ELIMINAR PERSONAS --");
        listar();
        System.out.print("Indique la posición a eliminar: ");
        int posicion = Leer.entero();
        posicion--;
        for (int i = posicion; i < cantidad; i++) {
            dni[i] = dni[i + 1];
            nombre[i] = nombre[i + 1];
            fecha[i] = fecha[i + 1];
        }
        dni[cantidad] = null;
        nombre[cantidad] = null;
        fecha[cantidad] = null;
        cantidad--;
        System.out.println("< Eliminado correctamente >");
    }

    public static void editar() {
        System.out.println("-- EDITAR PERSONAS --");
        listar();
        System.out.print("Indique la posición a editar: ");
        int posicion = Leer.entero();
        posicion--;
        System.out.println("> DNI anterior: " + dni[posicion]);
        System.out.print("* Nuevo DNI: ");
        dni[posicion] = Leer.cadena();
        System.out.println("> Nombre anterior: " + nombre[posicion]);
        System.out.print("* Nuevo Nombre: ");
        nombre[posicion] = Leer.cadena();
        System.out.println("> Fecha de nacimiento anterior: " + fecha[posicion]);
        System.out.print("* Nuevo Fecha de nacimiento: ");
        fecha[posicion] = Leer.cadena();
        System.out.println("< Editado correctamente >");
    }

    public static void listar() {
        System.out.println("Nro\tDNI\t\tNOMBRE\tFECHA DE NACIMIENTO");
        int num = 0;
        for (int i = 0; i <= cantidad; i++) {
            num++;
            System.out.println(num + "\t" + dni[i] + "\t" + nombre[i] + "\t" + fecha[i]);
        }
    }

    public static void menu() {
        System.out.println("*** PERSONAS ***");
        System.out.println("1. Agregar Personas");
        System.out.println("2. Eliminar Personas");
        System.out.println("3. Editar Personas");
        System.out.println("4. Listar Personas");
        System.out.println("5. Salir");
        System.out.print("Seleccione opción [1-5]: ");
    }

    public static void inicio() {
        int opcion;
        cargardatosdeinstalacion();
        do {
            Utilitarios.Lineasenblanco();
            menu();
            opcion = Leer.entero();
            switch (opcion) {
                case 1 -> agregar();
                case 2 -> eliminar();
                case 3 -> editar();
                case 4 -> listar();
                case 5 -> Utilitarios.salir();
                default -> Utilitarios.error();
            }
        } while (opcion != 5);
        Leer.cerrar();
    }
}

```

Index.java

```
package vistacontrol;

// Autor: Daniel Lévano y Flor Cerdán
public class Index {
    public static void inicio() {
        Indexpersona.inicio();
    }

    public static void main(String[] args) {
        inicio();
    }
}
```

7.3. Ejercicios propuestos 05

- **Prog110.java (Arreglos)** Desarrolle un programa que permita el ingreso de valores mientras este sea diferente de CERO, y muestre como resultado el mayor, menor y promedio de los valores ingresados.

```
Datos de entrada
Ingreso de valores
numeros[0]= 2
numeros[1]= 5
numeros[2]= 6
numeros[3]= -3
numeros[4]= 0

Datos de salida
Mayor= 6
Menor= -3
Promedio = 2
```

- **Prog111.java** Ingresar números a un arreglo mientras este sea diferente de cero, imprimir el resultado y mostrar su equivalente binario.

```
Datos de entrada
10
20
15
50
10
155
0

Datos de salida
Numeros[0] = 10 = 1010
Numeros[1] = 20 = 10100
Numeros[2] = 15 = 1111
Numeros[3] = 50 = 110010
Numeros[4] = 10 = 1010
Numeros[5] = 155 = 10011011
Numeros[6] = 0 = 0
```

- **Prog112.java** Permitir el ingreso de una cadena por teclado, copiar cada palabra en una celda del arreglo y mostrar el resultado.

Datos de entrada
Cadena: El Señor es mi Pastor

Datos de salida
Cadena[0]= El
Cadena[1]= Señor
Cadena[2]= es
Cadena[3]= mi
Cadena[4]= Pastor

7.4. Práctica de autoevaluación 04

Crear un package: practica04

Tiempo: 60 minutos

- **Preg01.java** Dado un arreglo A de N elementos ingresados por el teclado, copiar sus valores a otro arreglo B.

Datos de entrada

A[10]

5	8	10	6	-20	100
---	---	----	---	-----	-----

Datos de salida

B[10]

100	-20	6	10	8	5
-----	-----	---	----	---	---

- **Preg02.java** Desarrolle un programa que permita el ingreso de valores a un arreglo y muestre el equivalente en binario, base 4 y octal.

Datos de entrada

Cantidad: 5

Ingreso de valores:

numeros[0]= 10

numeros[1]= 16

numeros[2]= 14

numeros[3]= 7

numeros[4]= 12

Datos de salida

Resultado:

NUM	BASE 8	BASE 4	BINARIO
10	12	22	1010
16	20	100	10000
14	16	32	1110
7	7	13	111
12	14	30	1100

- **Preg03.java** Ingresar valores a un arreglo mientras este sea diferente de cero y mostrar un reporte.

```

Datos de entrada
10
20
15
50
10
155
0

Datos de salida
Reporte:

Cantidad de valores: 7
Sumatoria total: XX

Cantidad de Pares: XX
Sumatoria de Pares: XX

Cantidad de Impares: XX
Sumatoria de Impares: XX

Cantidad de Primos: XX
Sumatoria de Primos: XX

Cantidad de Capicúas: XX
Sumatoria de Capicúas: XX

```

8. Archivos

En programación, los datos pueden ser volátiles y no volátiles.

- **Datos volátiles:** Son aquellos que se almacenan temporalmente en la Memoria RAM y se pierden cuando el programa termina; este es el caso de todos los ejercicios desarrollados hasta este punto.
- **Datos no volátiles:** Son aquellos que se almacenan en el disco duro o repositorios en la nube; estos pueden ser recuperados incluso después de terminar el programa o apagar el computador.

Para este caso se utilizará los siguientes métodos:

Métodos	Explicación
File	Permite manipular archivos y directorios
FileReader - BufferedReader	Leer archivos de texto
FileWriter - BufferedWriter	Escribir archivos de texto
PrintWriter	Escribir texto con formato
Scanner	Leer datos de un archivo

- **Prog113.java** Crear un archivo: Versiculo.txt y escribir un mensaje.

```
package archivos;
import java.io.*;

// Autor: Daniel Lévano y Flor Cerdán
public class Prog113 {
    public static void main(String[] args) {
        try {
            String nombearchivo = "Versiculo.txt";
            FileWriter archivo = new FileWriter(nombearchivo);
            PrintWriter escritor = new PrintWriter(archivo);
            escritor.println("Jehová es mi pastor, ");
            escritor.println("nada me faltará");
            escritor.println("Salmos 23:1");
            escritor.close();
            System.out.println("< Archivo: " + nombearchivo + " creado >");
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

- **Prog114.java** Abrir un archivo de texto y mostrar el contenido línea por línea.

```
package archivos;
import java.io.*;
// Autor: Daniel Lévano y Flor Cerdán

public class Prog114 {
    public static void main(String[] args) {
        try {
            String nombearchivo = "Versiculo.txt";
            BufferedReader lector = new BufferedReader(new FileReader(nombearchivo));
            String linea;
            while ((linea = lector.readLine()) != null) {
                System.out.println(linea);
            }
            lector.close();
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

```

// Clase Lectura
public class Lectura {
    // Constructor
    public Lectura() {
        // Inicialización de variables
    }

    // Método para leer un número entero
    public int entero() {
        // Lógica para leer un número entero
    }
}

// Clase Prueba
public class Prueba {
    // Método main
    public static void main(String[] args) {
        // Creación de un objeto Lectura
        static Lectura leer = new Lectura();

        // Método inicio
        static void inicio() {
            int[] numeros = new int[7];
            System.out.println("Tamaño del arreglo: " + numeros.length);
            System.out.println("\nIngrese los valores: ");
            for (int i = 0; i < numeros.length; i++) {
                System.out.print("> numeros[" + i + "]: ");
                numeros[i] = leer.entero();
            }

            System.out.println("\nResultado:");
            for (int i = 0; i < numeros.length; i++) {
                System.out.println("* numeros[" + i + "]: " + numeros[i]);
            }
        }
    }
}

```

CRISTO VIENE PRONTO

¡Prepárate!