



El futuro digital  
es de todos

MinTIC

**Ciclo: Programación Básica**

**Introducción a JAVA**

»  
**Misión TIC 2022**

xxx



**Misión  
TIC 2022**



El futuro digital  
es de todos

MinTIC



# Programación Básica



## Presentación del formador



El futuro digital  
es de todos

MinTIC



# Programación Básica

x  
x  
x

- **Introducción al ciclo II – Programación Básica**
- **Introducción a Java**
  - **Historia y Generalidades**
  - **Instalación JDK e IDE Desarrollo**
  - **Sintaxis – IDE Netbeans (Consola – JOptionPane)**





El futuro digital  
es de todos

MinTIC



# Programación Básica



## Introducción al ciclo II – Programación Básica



El tripulante que haya culminado con éxito este ciclo estará en la capacidad de:

- **Desarrollar** de manera individual un programa orientado por objetos en Java, monousuario, para resolver los requerimientos planteados por un tercero, que use una base de datos relacional, con una arquitectura MVC y con una interfaz gráfica.
- **Diseñar y utilizar** una base de datos relacional simple usando un sistema gestor de base de datos y conectarla desde un programa usando un Framework o ORM
- **Diseñar** una aplicación utilizando objetos, expresando dicho diseño mediante diagramas
- **Diseñar y ejecutar** un conjunto de pruebas unitarias para un programa
- **Entender y utilizar** los conceptos de interfaz y de herencia como mecanismos de reutilización
- **Utilizar** una arquitectura MVC para desacoplar la lógica de negocio de la interfaz de usuario
- **Utilizar** librerías y frameworks que apoyen el desarrollo de software, en particular el desarrollo de la interfaz gráfica.





## » Programación Básica Ejes temáticos

**Temáticas estratégicas**

**Introducción JAVA – (Programación estructurada)**

**Programación Orientada a Objetos (POO)**

**Diseño Orientado a Objetos – Interfaz gráfica**

**Bases de Datos**



El futuro digital  
es de todos

MinTIC

## » Programación Básica

## Detalle temas

X  
X  
X

Semana	Unidad de aprendizaje	Contenido	Tipo	Actividad de aprendizaje
1	Introducción a Java	Estructuras de Control en java	Sesiones de videoconferencia (TEAMS)	Reto 1 (trabajo individual)
		Estructuras de Datos en java		
		Funciones en java		
		Taller		
2	Programación Orientada a Objetos	Clases, objetos y encapsulamiento	Sesiones de videoconferencia (TEAMS)	Reto 2 (trabajo individual)
		Constructores y visibilidad		
		Modificadores de clases		
		Taller		
3		Herencia e interfaces	Sesiones de videoconferencia (TEAMS)	Reto 3 (trabajo individual)
		Polimorfismo		
		Taller		
4	Diseño Orientado a Objetos	Principios de modelamiento orientado a objetos	Sesiones de videoconferencia (TEAMS)	Reto 4 (trabajo individual)
		Lenguaje unificado de modelamiento		
		Vista de casos de uso y Diagrama de Clases		
		Taller historia de usuario		
5	Bases de Datos Relacionales	Modelamiento de bases de datos	Sesiones de videoconferencia (TEAMS)	Reto 4 (trabajo individual)
		Lenguaje SQL		
		Taller		
6		Mapeo objeto relacional	Sesiones de videoconferencia (TEAMS)	Reto 5 (trabajo individual)
		JDBC		
		Taller		
7	Proyecto	Proyecto	Sesiones de videoconferencia (TEAMS).	Reto 5 (trabajo individual)



El futuro digital  
es de todos

MinTIC

# Herramientas







El futuro digital  
es de todos

MinTIC



# Programación Básica



## Introducción a Java



### Historia y Generalidades



### Instalación JDK e IDE Desarrollo



### Sintaxis – IDE Netbeans (Consola – JOptionPane)



El futuro digital  
es de todos

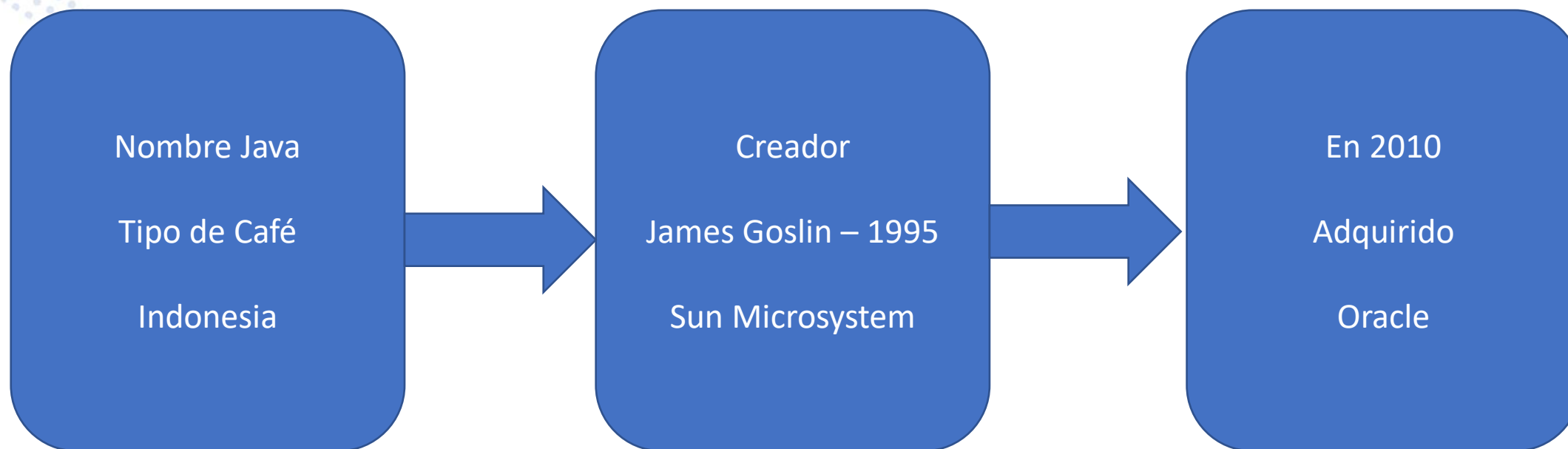
MinTIC

## » Introducción a Java



### Historia y Generalidades

x  
x  
x



Lenguaje Orientado a Objetos - Multiplataforma



El futuro digital  
es de todos

MinTIC

## » Introducción a Java



### Instalación JDK - IDE

x  
x  
x

Instalación JDK (Java Development Kit)



Instalación IDE ( Netbeans, Eclipse, IntelliJ, IDEA )

[https://www.youtube.com/watch?v=Stx3MNV3AHE&list=PLCTD\\_CpMeEKT-qEHGqZH3fkBgXH4GOTF&index=2](https://www.youtube.com/watch?v=Stx3MNV3AHE&list=PLCTD_CpMeEKT-qEHGqZH3fkBgXH4GOTF&index=2)



El futuro digital  
es de todos

MinTIC

## » Introducción a Java



## Sintaxis – IDE Netbeans (Primer programa)

XXXX

ProgramaHolaMundo - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

The screenshot displays the NetBeans IDE interface with the following components:

- Projects View:** Shows a project named 'ProgramaHolaMundo' containing a 'Source Packages' folder with 'programaholamundo' and a file 'ProgramaHolaMundo.java'.
- Source Editor:** Displays the code for 'ProgramaHolaMundo.java':

```
10  * @author SERGIO
11  */
12  import java.util.Scanner;
13
14  public class ProgramaHolaMundo {
15
16      /**
17       * @param args the command line arguments
18       */
19      public static void main(String[] args) {
20          // Crear instancia de Scanner para entrada por consola
21          Scanner entrada=new Scanner(System.in);
22          System.out.println("Hola Mundo");
23
24      }
25
26  }
```
- main - Navigator:** Shows the 'main' method of the 'ProgramaHolaMundo' class.
- Output View:** Shows the execution results:

```
run:
Hola Mundo
BUILD SUCCESSFUL (total time: 2 seconds)
```

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

## » Introducción a Java

### Sintaxis – IDE Netbeans (Primer programa)

NetBeans IDE 8.2 interface showing the project structure and the source code of the first Java program.

The project structure (left pane) includes:

- Ejemplo1
  - Source Packages
    - ejemplo1
  - Libraries
- Ejemplo2\_JOptionPane
  - Source Packages
    - ejemplo2\_joptionpne
      - Ejemplo2\_JOptionPane.java
  - Libraries
- JavaApplication1
  - Source Packages
  - Libraries

The source code (main editor) shows the following code:

```
7
8
9 import javax.swing.JOptionPane;
10
11 /**
12  * @author SERGIO
13  */
14 public class Ejemplo2_JOptionPane {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         JOptionPane.showMessageDialog(null, "Hola Mundo");
21     }
22
23 }
24
```

The 'Mensaje' dialog box displays the message: **Hola Mundo** with an 'Aceptar' button.

The 'main - Navigator' pane shows the project structure, including the 'main(String[] args)' method.

The 'Output' pane shows the command: **run:**





El futuro digital  
es de todos

MinTIC

## » Introducción a Java

x  
x  
x

### Estructuras de Control en Java

-  Tipos de Datos
-  Estructuras de Control
-  Estructura de Datos
-  Funciones





### Tipos de Datos en Java



## Tipos de Datos Primitivos en Java

### ¿Qué son los tipos de datos primitivos en Java?

Como ya hemos comentado **Java** es un lenguaje de tipado estático. Es decir, se define el tipo de dato de la variable a la hora de definir esta. Es por ello que todas las variables tendrán un tipo de dato asignado.

El lenguaje **Java** da de base una serie de tipos de datos primitivos.

- byte
- short
- int
- long
- float
- double
- boolean
- char





## » Introducción a Java



### Tipos de Datos en Java



#### byte

Representa un tipo de dato de 8 bits con signo. De tal manera que puede almacenar los valores numéricos de -128 a 127 (ambos inclusive).

#### short

Representa un tipo de dato de 16 bits con signo. De esta manera almacena valores numéricos de -32.768 a 32.767.

#### int

Es un tipo de dato de 32 bits con signo para almacenar valores numéricos. Cuyo valor mínimo es  $-2^{31}$  y el valor máximo  $2^{31}-1$ .

#### long

Es un tipo de dato de 64 bits con signo que almacena valores numéricos entre  $-2^{63}$  a  $2^{63}-1$ .

#### float

Es un tipo dato para almacenar números en coma flotante con precisión simple de 32 bits.

#### double

Es un tipo de dato para almacenar números en coma flotante con doble precisión de 64 bits.

#### boolean

Sirve para definir tipos de datos booleanos. Es decir, aquellos que tienen un valor de true o false. Ocupa 1 bit de información.

#### char

Es un tipo de datos que representa a un carácter Unicode sencillo de 16 bits.



El futuro digital  
es de todos

MinTIC

# Tipos de Datos en Java

Ejemplo: Tipos de Datos

## Ejercicios



Tipos\_Datos - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects Files Services

- JavaApplication1
  - Source Packages
  - Libraries
- Tipos\_Datos
  - Source Packages
  - tipos\_datos
    - Tipos\_Datos.java
  - Test Packages
  - Libraries
  - Test Libraries

Tipos\_Datos - Navigator

Members

- Tipos\_Datos
  - main(String[] args)

Start Page JavaApplication1.java Tipos\_Datos.java

Source History

```
6 package tipos_datos;
7
8 /**
9  * @author SERGIO
10  */
11 public class Tipos_Datos {
12
13
14     public static void main(String[] args) {
15         byte var_byte;
16         var_byte=127;
17         System.out.println("Varibla byte: "+var_byte);
18
19         short var_short;
20         var_short=32000;
21         System.out.println("Varibla short: "+var_short);
22
23         int var_int;
24         var_int=125000;
25         System.out.println("Varibla int: "+var_int);
26     }
27 }
```

Output - Tipos\_Datos (run)

```
Varibla byte: 127
Varibla short: 32000
Varibla int: 125000
Varibla long: 125000000
Varibla float: 7500000.5
Varibla double: 7.500000055E7
Varibla boolean: true
Varibla char: c
BUILD SUCCESSFUL (total time: 0 seconds)
```

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

# Tipos de Datos en Java

Ejemplo: Tipos de Datos

## Ejercicios



Tipos\_Datos - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

- JavaApplication1
  - Source Packages
  - Libraries
- Tipos\_Datos
  - Source Packages
    - tipos\_datos
      - Tipos\_Datos.java
  - Test Packages
  - Libraries
  - Test Libraries

Tipos\_Datos - Navigator

Members

- Tipos\_Datos
  - main(String[] args)

```
27 long var_long;
28 var_long=125000000;
29 System.out.println("Varibla long: "+var_long);
30
31 float var_float;
32 var_float=(float)7500000.5;
33 System.out.println("Varibla float: "+var_float);
34
35 double var_double;
36 var_double=75000000.55;
37 System.out.println("Varibla double: "+var_double);
38
39 boolean var_boolean;
40 var_boolean=true;
41 System.out.println("Varibla boolean: "+var_boolean);
42
43 char var_char;
44 var_char='c';
45 System.out.println("Varibla char: "+var_char);
46
47 }
```

Output - Tipos\_Datos (run)

```
Varibla byte: 127
Varibla short: 32000
Varibla int: 125000
Varibla long: 125000000
Varibla float: 7500000.5
Varibla double: 7.500000055E7
Varibla boolean: true
Varibla char: c
BUILD SUCCESSFUL (total time: 0 seconds)
```

Mision  
TIC2022





El futuro digital  
es de todos

MinTIC

## » Introducción a Java



### Tipos de Datos en Java

x  
x  
x

#### Videotutoriales recomendados:

[https://www.youtube.com/watch?v=JQEYWxB2gbM&list=PLCTD\\_CpMeEKT-qEHGqZH3fkBgXH4GOTF&index=7](https://www.youtube.com/watch?v=JQEYWxB2gbM&list=PLCTD_CpMeEKT-qEHGqZH3fkBgXH4GOTF&index=7)

[https://www.youtube.com/watch?v=zZjNNxaLXdw&list=PLCTD\\_CpMeEKT-qEHGqZH3fkBgXH4GOTF&index=8](https://www.youtube.com/watch?v=zZjNNxaLXdw&list=PLCTD_CpMeEKT-qEHGqZH3fkBgXH4GOTF&index=8)



## » Introducción a Java



### Estructuras de Control



#### Condicionales

- if - (Condicional)
- switch (Selección múltiple)



#### Ciclos:

- for (Ciclo PARA)
- while (Ciclo Hacer Mientras)
- do while (Ciclo Hacer Hasta)



El futuro digital  
es de todos

MinTIC

## » Introducción a Java

x  
x  
x



### Estructuras de Control



#### Condicionales

- if - (Condicional)
- switch (Selección múltiple)



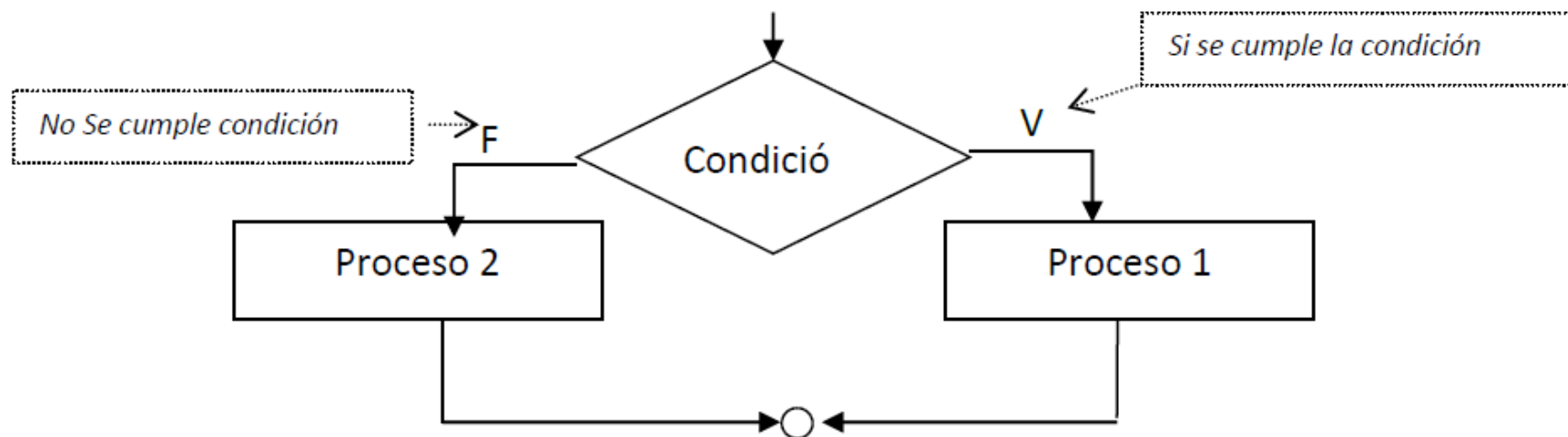
## » Introducción a Java



### Estructuras de Control - Condicional (if)

x  
x  
x

Estructura lógica en la cual se realiza la división del flujo del proceso de acuerdo a una condición ó pregunta.



Esta estructura se utiliza dentro de un proceso, cuando se presenta una decisión, condición ó pregunta, si se cumple la condición, es decir si es verdadera, se realiza el proceso 1, en caso de no cumplirse la condición, es decir si es falsa se realiza el proceso 2.



## » Introducción a Java



### Estructuras de Control - Condicional (if)

x  
x  
x

En pseudocódigo, la estructura del condicional sería:

**SI** Condición **ENTONCES**

Proceso 1

**SINO**

Proceso 2

**FIN SI**





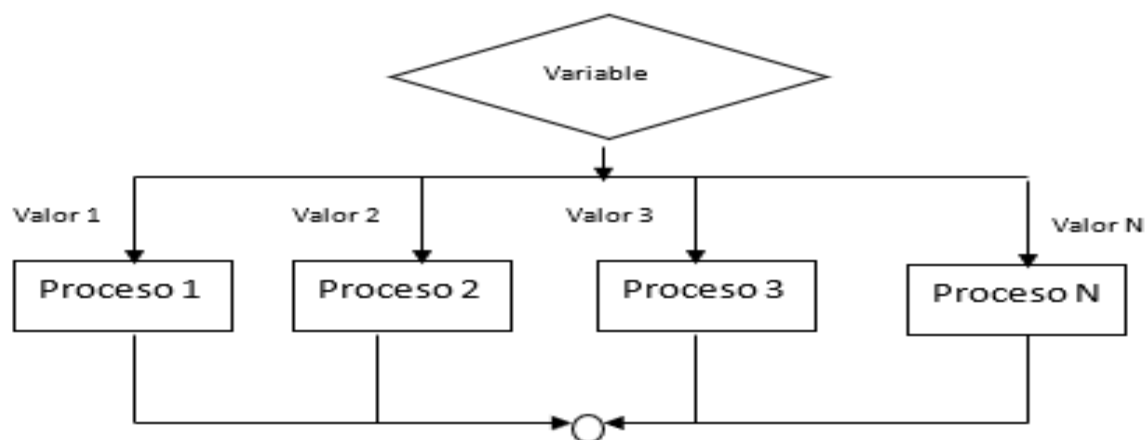
## » Introducción a Java



### Estructuras de Control - Selección múltiple (switch)

#### Selección múltiple (CASE o SWITCH)

Estructura que permite la ejecución de una tarea ó proceso dependiendo del valor de una variable, permite varias opciones o ramificaciones para el diagrama de flujo. Gráficamente sería:



Se tiene los siguientes componentes para esta estructura:

- Una variable que tiene valores predefinidos. (Ej: Sexo, Estado civil, estrato, etc.)
- Proceso 1, Proceso 2, Proceso 3 y Proceso N, que son las tareas o actividades que se ejecutan de acuerdo con cada valor válido de la variable.



## » Introducción a Java



### Estructuras de Control - Selección múltiple (switch)

x  
x  
x

En pseudocódigo:

**SELECCIONE** Variable

**EN CASO DE** Valor1: Proceso 1

**EN CASO DE** Valor2: Proceso 2

**EN CASO DE** Valor3: Proceso 3

.....

**EN CASO DE** Valor N: Proceso N

**FIN SELECCION**



## Estructuras de control

### Ejemplo: Condicionales

## Ejercicios



Dada la siguiente información de un usuario del servicio de agua:

- Documento de identidad
- Estado (A=activo, S=Suspendido)
- Estrato(1,2,3,4,5)

Se pide calcular el valor a pagar por tarifa básica, de acuerdo con las siguientes observaciones:

- Si el usuario está suspendido, el valor de tarifa básica es 0.
- Si el usuario está activo, el valor de la tarifa básica depende del estrato:

Estrato	Tarifa Básica
1	\$10.000
2	\$15.000
3	\$30.000
4	\$50.000
5	\$65.000

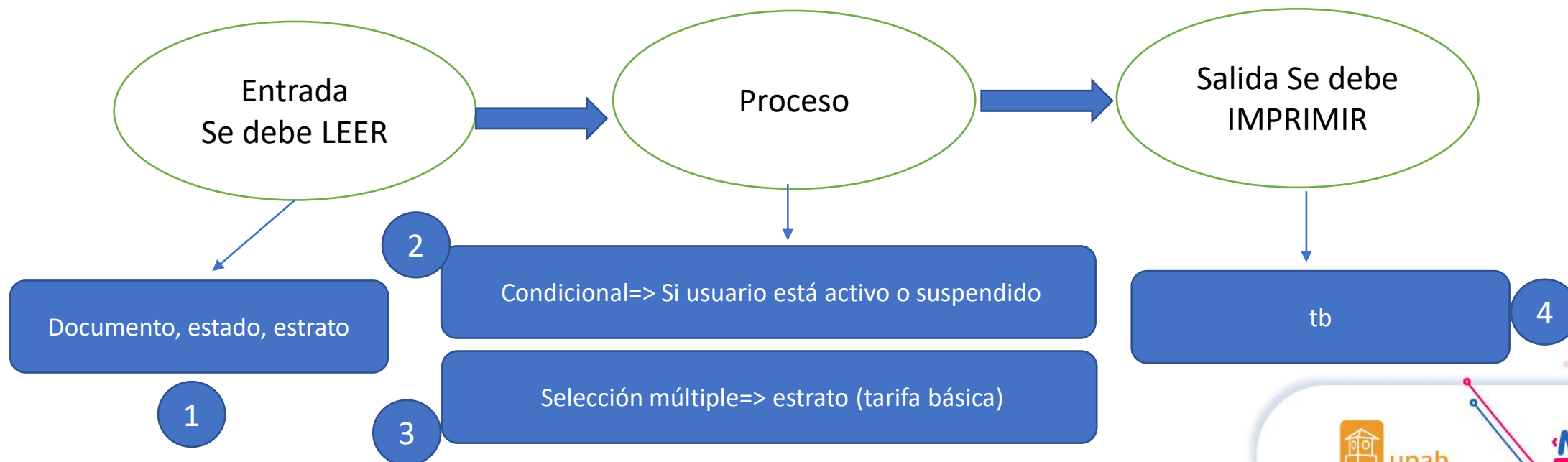


## Estructuras de control

## Ejercicios



Análisis → Cálculo de la tarifa básica





## Estructuras de control

Variables del programa

Identificación de variables

ENTRADA:  
documento => long  
estado => char  
estrato => int

SALIDA:  
tb => float o long

## Ejercicios







El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Tarifa Básica (Código)

# Ejercicios



Calcula\_Tarifa\_basica - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

The screenshot displays the NetBeans IDE interface. On the left, the 'Projects' and 'Navigator' panels show the project structure. The 'Projects' panel lists 'Calcula\_Tarifa\_basica' with sub-items 'Source Packages' (containing 'calcula\_tarifa\_basica' and 'Calcula\_Tarifa\_basica.java') and 'Test Packages'. The 'Navigator' panel shows the 'main(String[] args)' method. The main editor window displays the source code of 'Calcula\_Tarifa\_basica.java'. The code includes a package declaration, an import for 'Scanner', and a 'main' method that initializes a 'Scanner' object and prints the results of a calculation. The 'Output' panel at the bottom shows the execution results, including the input values and the calculated 'Tarifa Básica'.

```
4  * and open the template in the editor.
5  */
6  package calcula_tarifa_basica;
7
8  /**
9   *
10  * @author SERGIO
11  */
12  import java.util.Scanner;
13
14  public class Calcula_Tarifa_basica {
15
16      /**
17       * @param args the command line arguments
18       */
19      public static void main(String[] args) {
20          /* Instancia de Scanner para entrada por consola */
21          Scanner entrada=new Scanner(System.in);
```

Output:

```
run:
Documento de identidad:
91254478
Estado (A=Activo,S=Suspendido) :
A
Estrato (1,2,3,4,5) :
3
Tarifa Básica: 30000.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

# Estructuras de control

# Ejercicios



## Ejercicio: Tarifa Básica (Código)

Calcula\_Tarifa\_basica - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

The screenshot displays the NetBeans IDE interface. The left sidebar shows the project structure for 'Calcula\_Tarifa\_basica', including 'Source Packages' and 'Test Packages'. The main editor window shows the source code for 'Calcula\_Tarifa\_basica.java'. The code defines a 'main' method that takes command-line arguments and performs calculations for a basic tariff. The 'Output' window at the bottom shows the execution results, including the input values and the calculated tariff.

```
17  * @param args the command line arguments
18  */
19  public static void main(String[] args) {
20      /* Intancia de Scanner para entrada por consola */
21      Scanner entrada=new Scanner(System.in);
22      /* Definición de variables */
23      long documento;
24      char estado;
25      int estrato;
26      float tb=0;
27
28      /* Entrada de Datos*/
29      System.err.println("Documento de identidad: ");
30      documento=entrada.nextLong();
31      System.err.println("Estado (A=Activo,S=Suspendido): ");
32      estado=entrada.next().charAt(0);
33      System.err.println("Estrato (1,2,3,4,5): ");
34      estrato=entrada.nextInt();
```

Output:

```
run:
Documento de identidad:
91254478
Estado (A=Activo,S=Suspendido):
A
Estrato (1,2,3,4,5):
3
Tarifa Básica: 30000.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Tarifa Básica (Código)

# Ejercicios



Calcula\_Tarifa\_basica - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

- Calcula\_Tarifa\_basica
  - Source Packages
    - calcula\_tarifa\_basica
      - Calcula\_Tarifa\_basica.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication 1
  - Source Packages
  - Libraries

Start Page JavaApplication1.java Calcula\_Tarifa\_basica.java

Source History

```
35      /*Calcular tarifa básica */
36      if (estado=='S')
37          tb=0;
38      else
39          switch (estrato)
40          {
41              case 1: tb=10000;break;
42              case 2: tb=15000;break;
43              case 3: tb=30000;break;
44              case 4: tb=50000;break;
45              case 5: tb=65000;break;
46          }
47      System.out.println("Tarifa Básica: "+tb);
48
49
50
51  }
52
```

calcula\_tarifa\_basica.Calcula\_Tarifa\_basica main

Output

Calcula\_Tarifa\_basica (run) Calcula\_Tarifa\_basica (run) #2

```
run:
Documento de identidad:
91254478
Estado (A=Activo, S=Suspendido) :
A
Estrato (1,2,3,4,5) :
3
Tarifa Básica: 30000.0
BUILD SUCCESSFUL (total time: 8 seconds)
```

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

## » Introducción a Java

x  
x  
x



### Estructuras de Control



#### Ciclos:

- For (Ciclo PARA)
- While (Ciclo Hacer Mientras)
- Do while (Ciclo Hacer Hasta)





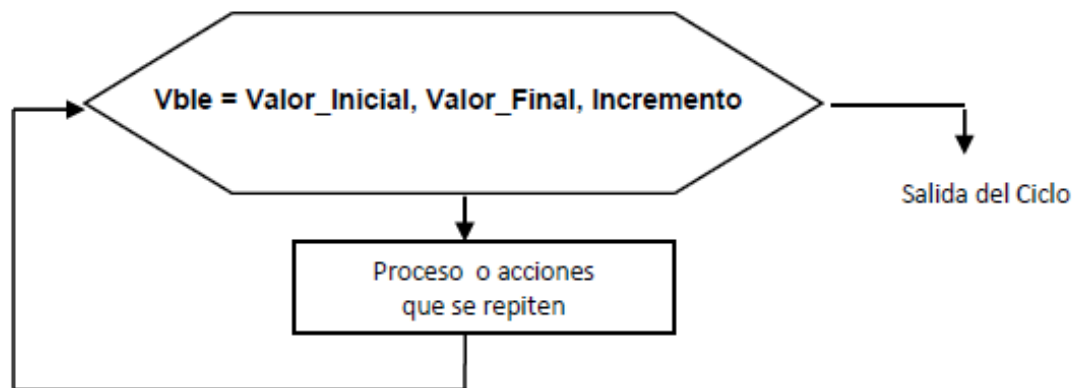
## » Introducción a Java



### Estructuras de Control - Ciclo PARA ( for )

#### ☑ Repetición Iterativa o Bucle PARA

Estructura en donde un proceso ó actividad se repite varias veces, lo cual se le conoce con el nombre de CICLO. Como característica fundamental de esta estructura es que se conoce el número de repeticiones que se van a realizar, también es usada para series o sucesiones. Veamos su representación gráfica:



Donde,





## » Introducción a Java



### Estructuras de Control - Ciclo PARA ( for)



**Vble** = Encargada de tomar valores de acuerdo a cada ocurrencia de la repetición. Puede darse cualquier nombre a la variable (tener en cuenta las reglas para identificadores), aunque se utiliza con mucha frecuencia, los nombre de I, J, K entre otros. Las iteraciones estarán controladas por esta variable.

**Valor\_Inicial** = Constante o variable con el valor inicial del ciclo, esta asignación solo se realiza una vez, cuando se inicializa el ciclo, es la primera ocurrencia.

**Valor\_Final** = Constante o variable con el valor final del ciclo, esta determina la condición para terminar el ciclo, es la última ocurrencia.

**Incremento** = Constante o variable que representa el intervalo de aumento de la variable del ciclo, cuando el incremento es 1 se puede omitir, si es un decremento se representa con un valor negativo.



## » Introducción a Java



### Estructuras de Control - Ciclo PARA ( for)

x  
x  
x

En pseudocódigo, la estructura sería:

```
PARA Vble= Valor_Inicial HASTA Valor_Final INCREMENTO INCR  
    Proceso que se Repite  
FIN PARA
```

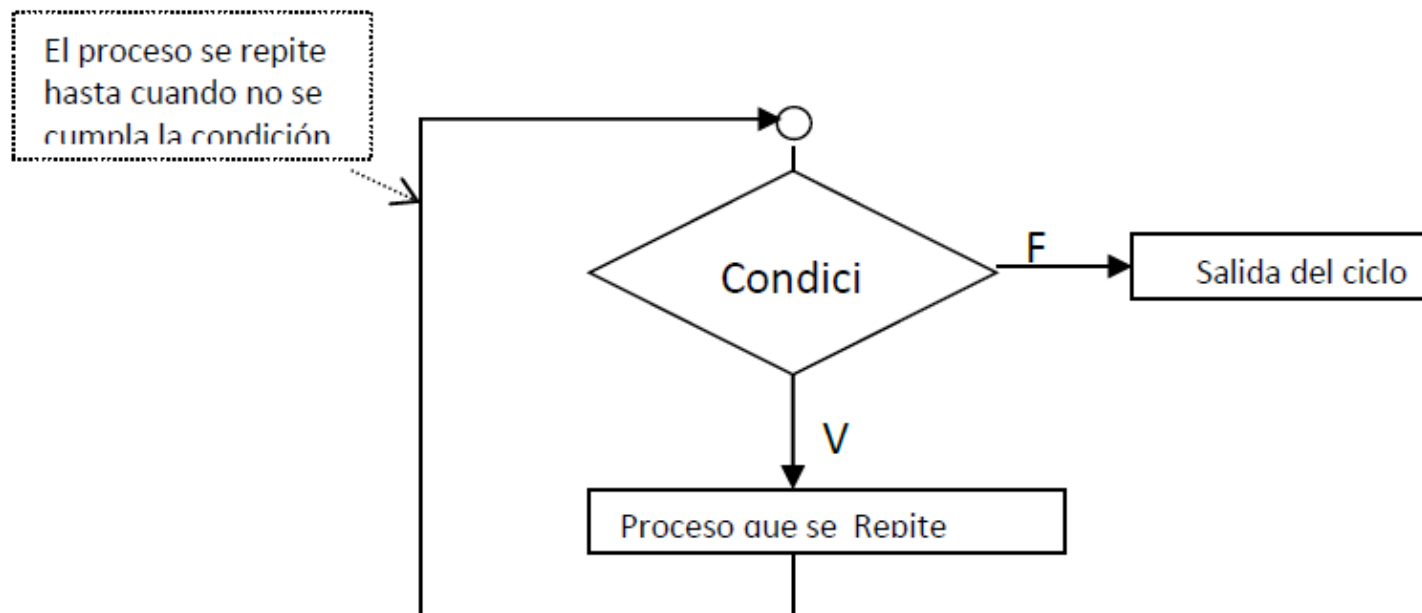


## » Introducción a Java



### Estructuras de Control - Ciclo MIENTRAS (while)

x  
x  
x





El futuro digital  
es de todos

MinTIC

## » Introducción a Java



### Estructuras de Control - Ciclo MIENTRAS (while)

x  
x  
x

En pseudocódigo:

**MIENTRAS** Condición **HAGA**

Proceso que se repite

**FIN MIENTRAS**



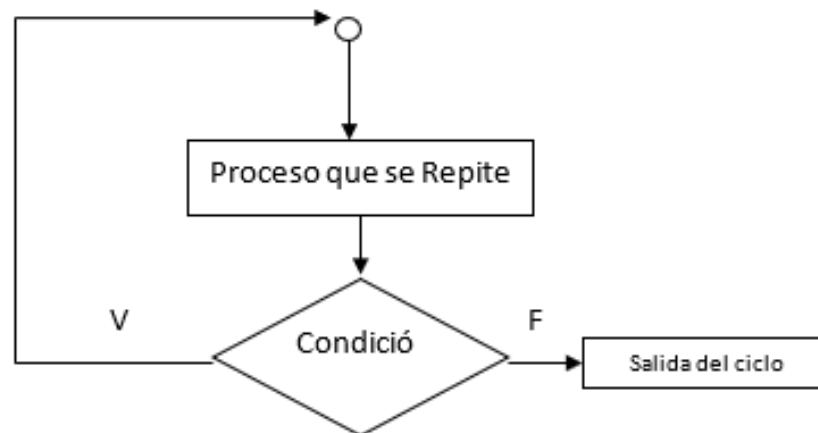
## » Introducción a Java



### Estructuras de Control - Ciclo HACER MIENTRAS (do while)

#### ☑ Repetición Condicional - HACER MIENTRAS (DO - WHILE)

Estructura en donde aparece nuevamente un proceso que se repite, CICLO, pero esta vez su repetición depende del cumplimiento de una condición, es similar a la estructura MIENTRAS se diferencia en que la condición va después del proceso, es decir este bucle siempre realiza por lo menos una iteración antes de evaluar la condición. Gráficamente sería:



En esta estructura el proceso se repite mientras la condición sea verdadera.





El futuro digital  
es de todos

MinTIC

## » Introducción a Java



### Estructuras de Control - Ciclo HACER MIENTRAS ( Do while)

x  
x  
x

En pseudocódigo:

**HACER**

Proceso que se repite

**MIENTRAS** Condición



## Estructuras de control

Ejemplo: Ciclos (FOR)

## Ejercicios



Dada la información sobre los **N vendedores**:

- **Cédula**
- **Tipo Vendedor** (1: Puerta a Puerta 2=Ejecutivo de ventas)
- **Valor ventas del mes**

Se pide calcular el **valor de comisiones** de cada vendedor, de acuerdo con las siguientes observaciones:

**Información para liquidar comisiones:** Si el vendedor es puerta a puerta (1) gana por comisiones el 20% del valor de sus ventas y si es ejecutivo de ventas (2) gana por comisiones el 30% del valor de sus ventas del mes

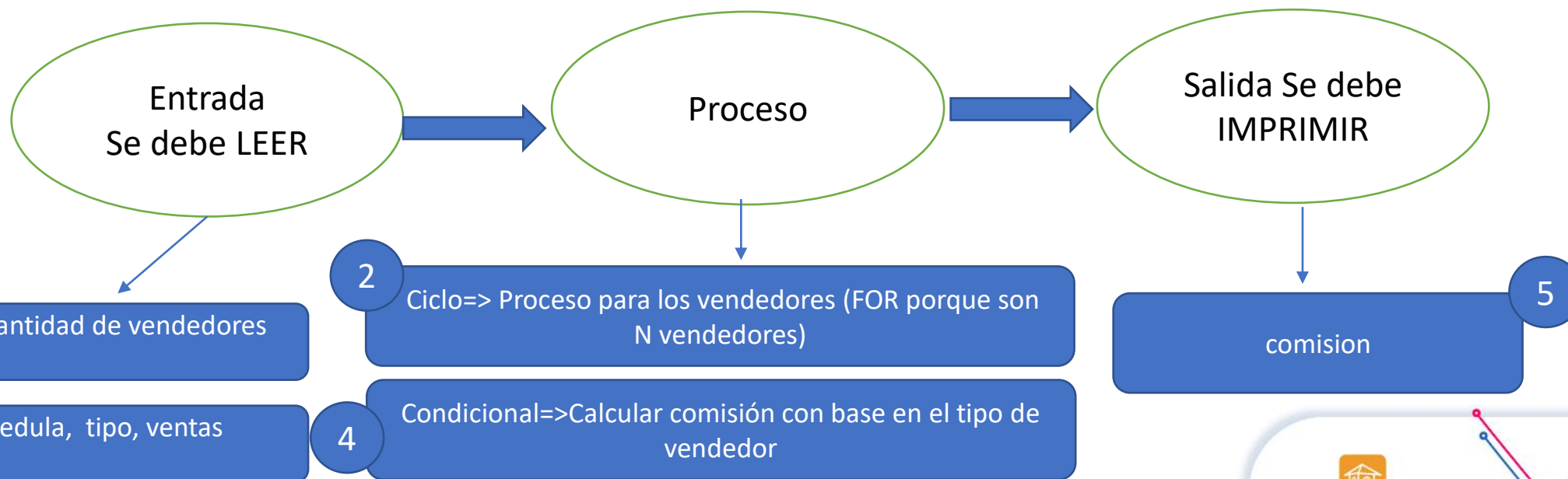


## Estructuras de control

## Ejercicios



Análisis → Cálculo de comisiones





## Estructuras de control

Variables del programa

Identificación de variables

ENTRADA:  
N, tipo => int  
cedula => long  
ventas => double o float

SALIDA:  
comision => double

## Ejercicios





El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Comisiones

# Ejercicios



calculo\_comisiones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

NetBeans IDE interface showing the project structure and the source code for the 'calculo\_comisiones' application.

**Projects:**

- calculo\_comisiones
  - Source Packages
    - calculo\_comisiones
      - Calculo\_comisiones.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication 1
  - Source Packages
    - javaapplication 1
      - JavaApplication 1.java
  - Libraries

**main - Navigator:**

- Calculo\_comisiones
  - main(String[] args)

**Source Code (Calculo\_comisiones.java):**

```
5  package calculo_comisiones;  
6  
7  
8  /**  
9   *  
10   * @author SERGIO  
11   */  
12 import java.util.Scanner;  
13  
14 public class Calculo_comisiones {  
15  
16     /**  
17      * @param args the command line arguments  
18      */  
19     public static void main(String[] args) {  
20         /* Instancia Scanner para lectura por consola */  
21         Scanner entrada=new Scanner(System.in);  
22         /*Definición de variables */  
23         long cedula;
```

**Output:**

```
Calcula_Tarifa_basica (run) x calculo_comisiones (run) x  
Comisión: 200000.0  
Cédula:  
2  
Tipo(1=Puerta a Puerta, 2=Ejecuyivo):  
2  
Valor ventas:  
3000000  
Comisión: 900000.0  
BUILD SUCCESSFUL (total time: 19 seconds)
```

Mision  
TIC2022





El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Comisiones

# Ejercicios



calculo\_comisiones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

NetBeans IDE 8.2 interface showing the project structure and the execution of the 'calculo\_comisiones' program.

**Projects:**

- calculo\_comisiones
  - Source Packages
    - calculo\_comisiones
      - Calculo\_comisiones.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries

**main - Navigator:**

- Members
  - Calculo\_comisiones
    - main(String[] args)

**Source:**

```
20  /* Instancia Scanner para lectura por consola */
21  Scanner entrada=new Scanner(System.in);
22  /*Definición de variables */
23  long cedula;
24  int N,tipo;
25  double comision,ventas;
26  /* Entrada de datos */
27  System.err.println("Cantidad de vebdedores: ");
28  N=entrada.nextInt();
29  /* Ciclo para calacular comisiones */
30  for (int i=1;i<=N;i++)
31  {
32      System.err.println("Cédula: ");
33      cedula=entrada.nextLong();
34      System.err.println("Tipo(1=Puerta a Puerta, 2=Ejecuyivo): ");
35      tipo=entrada.nextInt();
36      System.err.println("Valor ventas: ");
37      ventas=entrada.nextDouble();
```

**Output:**

```
Calcula_Tarifa_basica (run) x calculo_comisiones (run) x
Comisión: 200000.0
Cédula:
2
Tipo(1=Puerta a Puerta, 2=Ejecuyivo):
2
Valor ventas:
3000000
Comisión: 900000.0
BUILD SUCCESSFUL (total time: 19 seconds)
```

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Comisiones

# Ejercicios



calculo\_comisiones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

NetBeans IDE 8.2 interface showing the project structure and the execution of the `Calculo_comisiones` program.

**Project Structure:**

- calculo\_comisiones
  - Source Packages
    - calculo\_comisiones
      - Calculo\_comisiones.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries

**main - Navigator:**

- Members
  - Calculo\_comisiones
    - main(String[] args)

**Source Code (Calculo\_comisiones.java):**

```
30 for (int i=1; i<=N; i++)
31 {
32     System.err.println("Cédula: ");
33     cedula=entrada.nextLong();
34     System.err.println("Tipo(1=Puerta a Puerta, 2=Ejecuyivo): ");
35     tipo=entrada.nextInt();
36     System.err.println("Valor ventas: ");
37     ventas=entrada.nextDouble();
38     if (tipo==1)
39         comision=ventas*0.20;
40     else
41         comision=ventas*0.30;
42     System.out.println("Comisión: "+comision);
43 }
44 }
45 }
46 }
47 }
```

**Output:**

```
Calcula_Tarifa_basica (run) x calculo_comisiones (run) x
Comisión: 200000.0
Cédula:
2
Tipo(1=Puerta a Puerta, 2=Ejecuyivo):
2
Valor ventas:
3000000
Comisión: 900000.0
BUILD SUCCESSFUL (total time: 19 seconds)
```



El futuro digital  
es de todos

MinTIC

## Estructuras de control

Ejemplo: Ciclos (WHILE)

## Ejercicios



Dada la información sobre una lista de vendedores:

- Cédula
- Tipo Vendedor (1: Puerta a Puerta 2=Ejecutivo de ventas)
- Valor ventas del mes

La lista termina cuando la cédula ingresada sea 999

Se pide calcular el valor de comisiones de cada vendedor, de acuerdo con las siguientes observaciones:

Información para liquidar comisiones: Si el vendedor es puerta a puerta (1) gana por comisiones el 20% del valor de sus ventas y si es ejecutivo de ventas (2) gana por comisiones el 30% del valor de sus ventas del mes

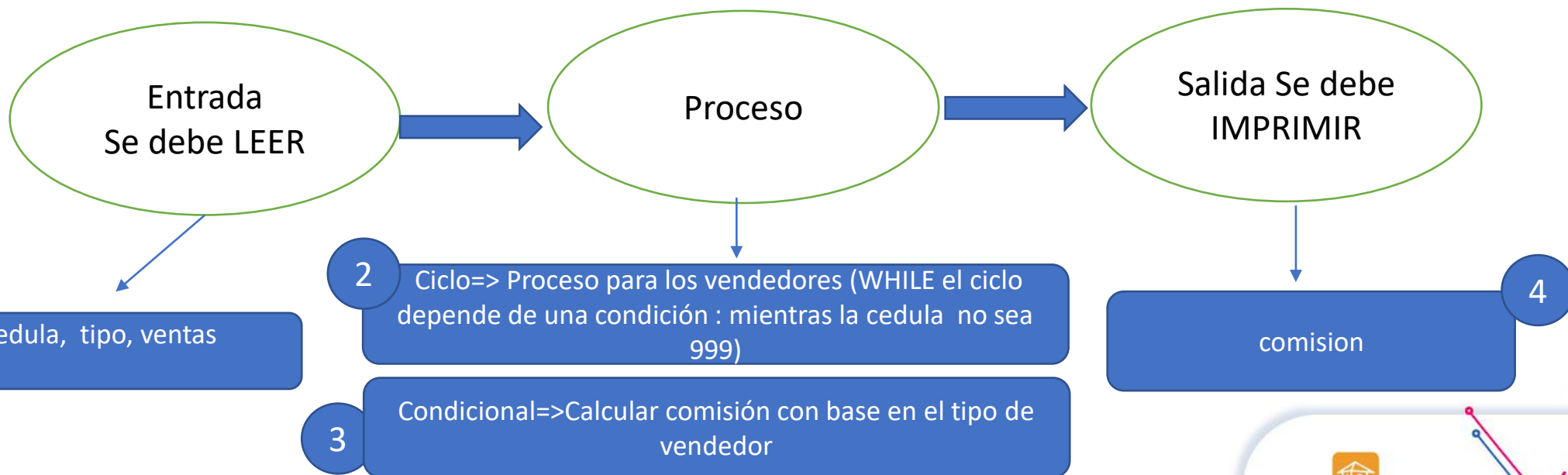


## Estructuras de control

## Ejercicios



Análisis → Cálculo de comisiones (WHILE)





## Estructuras de control

Variables del programa

Identificación de variables

ENTRADA:  
tipo => int  
cedula => long  
ventas => double o float

SALIDA:  
comision => float o long

## Ejercicios







El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Comisiones (WHILE)

# Ejercicios



calculo\_comisiones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

- calculo\_comisiones
  - Source Packages
  - Test Packages
  - Libraries
  - Test Libraries
  - JavaApplication1

Navigator

Members

- Calculo\_comisiones
  - main(String[] args)

Start Page JavaApplication1.java Calculo\_comisiones.java

Source History

```
5  /*
6  package calculo_comisiones;
7
8  /**
9   *
10  * @author SERGIO
11  */
12 import java.util.Scanner;
13
14 public class Calculo_comisiones {
15
16     /**
17     * @param args the command line arguments
18     */
19     public static void main(String[] args) {
20         /* Instancia Scanner para lectura por consola */
21         Scanner entrada=new Scanner(System.in);
22         /*Definición de variables */
```

Output - calculo\_comisiones (run)

```
Valor ventas:
200
Comisión: 60.0
Cédula Vendedor:
3
Tipo (1=Puerta a Puerta, 2=Ejecutivo):
2
Valor ventas:
400
Comisión: 120.0
BUILD SUCCESSFUL (total time: 15 seconds)
```



El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Comisiones (WHILE)

# Ejercicios



calculo\_comisiones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

NetBeans IDE 8.2 interface showing the source code for the 'calculo\_comisiones' project. The code implements a while loop to calculate commissions based on sales and type.

```
22  /* Definición de variables */
23  long cedula;
24  int tipo;
25  double comision, ventas;
26  /* Leer primer vendedor para entrar al while */
27  System.err.println("Cédula Vendedor: ");
28  cedula = entrada.nextInt();
29  /* Ciclo para calcular comisiones */
30  while (cedula != 999)
31  {
32      System.err.println("Tipo (1=Puerta a Puerta, 2=Ejecutivo): ");
33      tipo = entrada.nextInt();
34      System.err.println("Valor ventas: ");
35      ventas = entrada.nextDouble();
36      if (tipo == 1)
37          comision = ventas * 0.20;
38      else
39          comision = ventas * 0.30;
```

The output window shows the execution results:

```
Valor ventas:
200
Comisión: 60.0
Cédula Vendedor:
3
Tipo (1=Puerta a Puerta, 2=Ejecutivo):
2
Valor ventas:
400
Comisión: 120.0
BUILD SUCCESSFUL (total time: 15 seconds)
```



El futuro digital  
es de todos

MinTIC

# Estructuras de control

## Ejercicio: Comisiones (WHILE)

# Ejercicios



calculo\_comisiones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

```
30 while (cedula!=999)
31 {
32     System.err.println("Tipo(1=Puerta a Puerta, 2=Ejecuyivo): ");
33     tipo=entrada.nextInt();
34     System.err.println("Valor ventas: ");
35     ventas=entrada.nextDouble();
36     if (tipo==1)
37         comision=ventas*0.20;
38     else
39         comision=ventas*0.30;
40     System.out.println("Comisión: "+comision);
41     /*Leer siguiente */
42     System.err.println("Cédula: ");
43     cedula=entrada.nextLong();
44 }
45 }
46 }
47 }
```

Output - calculo\_comisiones (run)

```
Valor ventas:
200
Comisión: 60.0
Cédula Vendedor:
3
Tipo(1=Puerta a Puerta, 2=Ejecuyivo):
2
Valor ventas:
400
Comisión: 120.0
BUILD SUCCESSFUL (total time: 15 seconds)
```

Mision  
TIC2022



## » Introducción a Java



### Estructuras de Datos – Vectores y Marices

x  
x  
x

#### vector

Representación gráfica de una LISTA

MEMORIA

	Catalina	Silvia	Sergio	Iván	Paula		

Lista: Conjunto consecutivo, secuencial de elementos



## » Introducción a Java



### Estructuras de Datos – Vectores y Marices

x  
x  
x

MEMORIA

		Columnas				
	Filas	1	2	3	4	
		5	6	7	8	
		9	10	11	12	

Matriz

Matriz de Números: En esta matriz se almacenan números, que serán utilizados en el proceso. En este caso, la matriz maneja la dimensión horizontal ó fila y la dimensión vertical o columna.

Número de Filas: 3

Número de Columnas: 4





## Acumuladores ( Contador – Sumador)

## Ejercicio



Dados 6 números enteros que se almacenan en un **vector**,  
calcular:

- Cuáles y cuantos son pares
- Cuáles y cuantos son impares
- La suma de los números pares
- La suma de los números impares

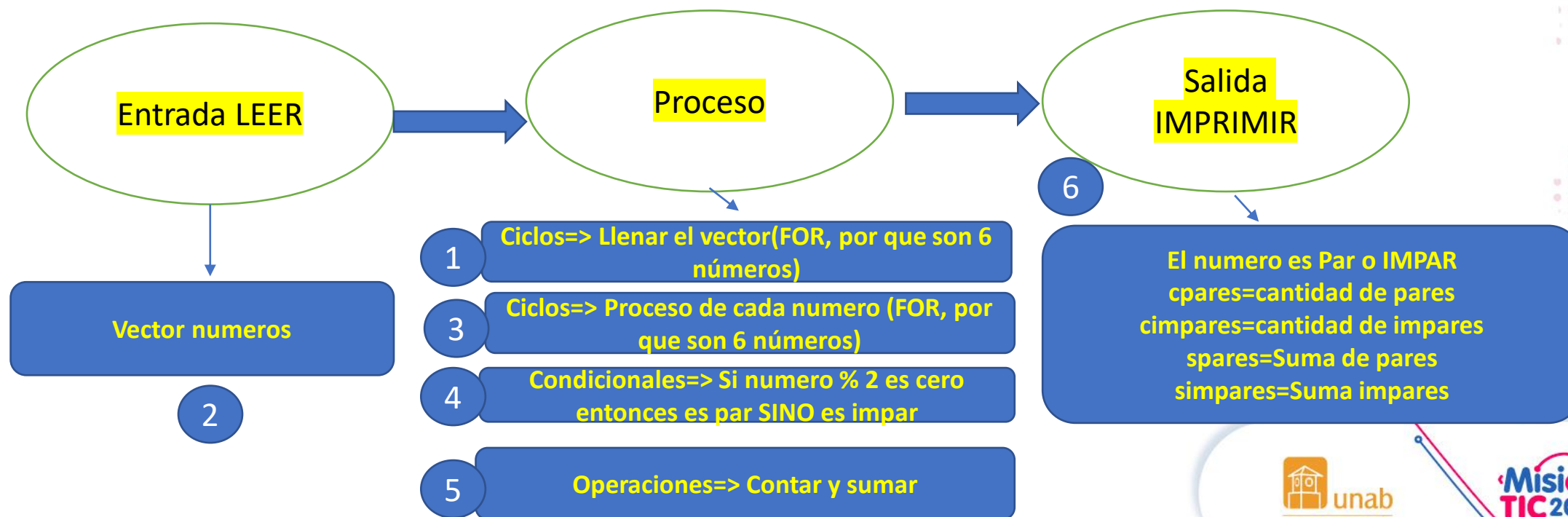


# Acumuladores

## Ejercicios



Análisis → Ejercicio pares e impares





## Estructuras de control

Variables del programa

Identificación de variables

ENTRADA:  
vector números => int

SALIDA:  
cp, ci, sp, si => int

## Ejercicios





El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



vector.es\_java - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects Files Services

- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries
- vector.es\_java
  - Source Packages
    - vector.es\_java
      - Vector.es\_java.java
  - Test Packages
  - Libraries
  - Test Libraries

main - Navigator

Members

- Vector.es\_java
  - main(String[] args)

Source

```
5  /*
6  package vector.es_java;
7
8  /**
9   *
10  * @author SERGIO
11  */
12  import java.util.Scanner;
13
14  public class Vector.es_java {
15
16      /**
17       * @param args the command line arguments
18       */
19      public static void main(String[] args) {
20          /* Instancia Scanner para entrada de datos por consola */
21          Scanner entrada=new Scanner(System.in);
22          /* Definición de variables */
23          for (inti = 1; i <= 6; i++) >
```

Output

Calcula\_Tarifa\_basica (run) x vector.es\_java (run) x

```
El número es IMPAR : 3
El número es PAR : 4
El número es IMPAR : 5
El número es PAR : 6
Cantidad de pares : 3
Cantidad de impares : 3
Suma de pares : 12
Suma de pares : 9
BUILD SUCCESSFUL (total time: 8 seconds)
```



El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



vectores\_java - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects Files Services

- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries
- vectores\_java
  - Source Packages
    - vectores\_java
      - Vectores\_java.java
  - Test Packages
  - Libraries
  - Test Libraries

main - Navigator

Members

<empty>

Vectores\_java

- main(String[] args)

Start Page JavaApplication1.java Vectores\_java.java

Source History

```
22  /* Definición de variables */
23  long cp,ci,sp,si;
24  long numeros[];
25  numeros=new long[10];
26  for (int i=1;i<=6;i++)
27  {
28      System.out.println("Ingrese número: ");
29      numeros[i]=entrada.nextLong();
30
31  }
32  cp=0;ci=0;sp=0;si=0;
33  for (int i=1;i<=6;i++)
34  {
35      if (numeros[i] % 2==0)
36      {
37          System.out.println("El número es PAR : "+numeros[i]);
38          cp=cp+1;
39          sp=sp+numeros[i];
```

vectores\_java.Vectores\_java main for (inti = 1; i <= 6; i++)

Output

Calcula\_Tarifa\_basica (run) vectores\_java (run)

```
El número es IMPAR : 3
El número es PAR : 4
El número es IMPAR : 5
El número es PAR : 6
Cantidad de pares : 3
Cantidad de impares : 3
Suna de pares : 12
Suma de pares : 9
BUILD SUCCESSFUL (total time: 8 seconds)
```

Mision  
TIC2022





El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



vectores\_java - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects Files Services

- JavaApplication1
  - Source Packages
    - javaapplication1
      - JavaApplication1.java
  - Libraries
- vectores\_java
  - Source Packages
    - vectores\_java
      - Vectores\_java.java
  - Test Packages
  - Libraries
  - Test Libraries

main - Navigator

Members

<empty>

Vectores\_java

- main(String[] args)

Start Page JavaApplication1.java Vectores\_java.java

Source History

```
35 if (numeros[i] % 2 == 0)
36 {
37     System.out.println("El número es PAR : "+numeros[i]);
38     cp=cp+1;
39     sp=sp+numeros[i];
40 }
41 else
42 {
43     System.out.println("El número es IMPAR : "+numeros[i]);
44     ci=ci+1;
45     si=si+numeros[i];
46 }
47
48
49 }
50 System.out.println("Cantidad de pares : "+cp);
51 System.out.println("Cantidad de impares : "+ci);
52 System.out.println("Suma de pares : "+sp);
53 System.out.println("Suma de impares : "+si);
```

vectores\_java.Vectores\_java main for (inti = 1; i <= 6; i++)

Output

Calcula\_Tarifa\_basica (run) vectores\_java (run)

```
El número es IMPAR : 3
El número es PAR : 4
El número es IMPAR : 5
El número es PAR : 6
Cantidad de pares : 3
Cantidad de impares : 3
Suma de pares : 12
Suma de impares : 9
BUILD SUCCESSFUL (total time: 8 seconds)
```



## » Introducción a Java



### Modularidad - Funciones

Uno de los aspectos fundamentales de la programación moderna, base de los nuevos paradigmas, es sin duda alguna la **modularidad**, entendida como la generación de **módulos** o **segmentos funcionales** e **independientes** que permitan una mejor organización y compresión de un programa. Este aspecto se basa en la aplicación de dos técnicas propias de la ingeniería del software, denominadas **Acoplamiento de módulos** y **Cohesión de módulos** que definen unas guías en la definición de un módulo.



## » Introducción a Java



### Cohesión de módulos

La técnica de la ingeniería del software, denominada Cohesión de Módulos busca medir el grado de **relación** o **dependencia** que existe entre las **actividades propias** de un **proceso o módulo**.



### Acoplamiento de módulos

La técnica La técnica del Acoplamiento de Módulos que se aplica después de la cohesión, tiene como objetivo la generación de **módulos independientes** dentro de un proceso, en los cuales, **cada uno de ellos defina sus propias variables** y la **comunicación entre ellos se haga a través de parámetros**, o sea, variables que un módulo envía y el otro recibe en variables propias



## Funciones

# Ejercicios

La empresa de teléfonos de la ciudad necesita realizar su proceso de facturación en forma automática, contando con los **N abonados**, de los cuales conoce el **código**, **estrato**, que puede ser (1, 2, 3, 4, 5), cantidad de **impulsos del mes**. Además la empresa nos informa que para la liquidación de la factura se debe tener en cuenta el valor de la tarifa básica, de acuerdo al estrato, que depende de la siguiente tabla:



Estrato	tarifa Básica
1	\$10.000
2	\$15.000
3	\$20.000
4	\$25.000
5	\$30.000

Además se debe calcular el valor de los impulsos, con base en la cantidad de impulsos del mes, conociendo que cada impulso tiene un valor de \$100. Con esta información, se desea:

- ☒ **Valor a pagar de cada abonado.**
- ☒ **Valor total a pagar(Todos los abonados)**

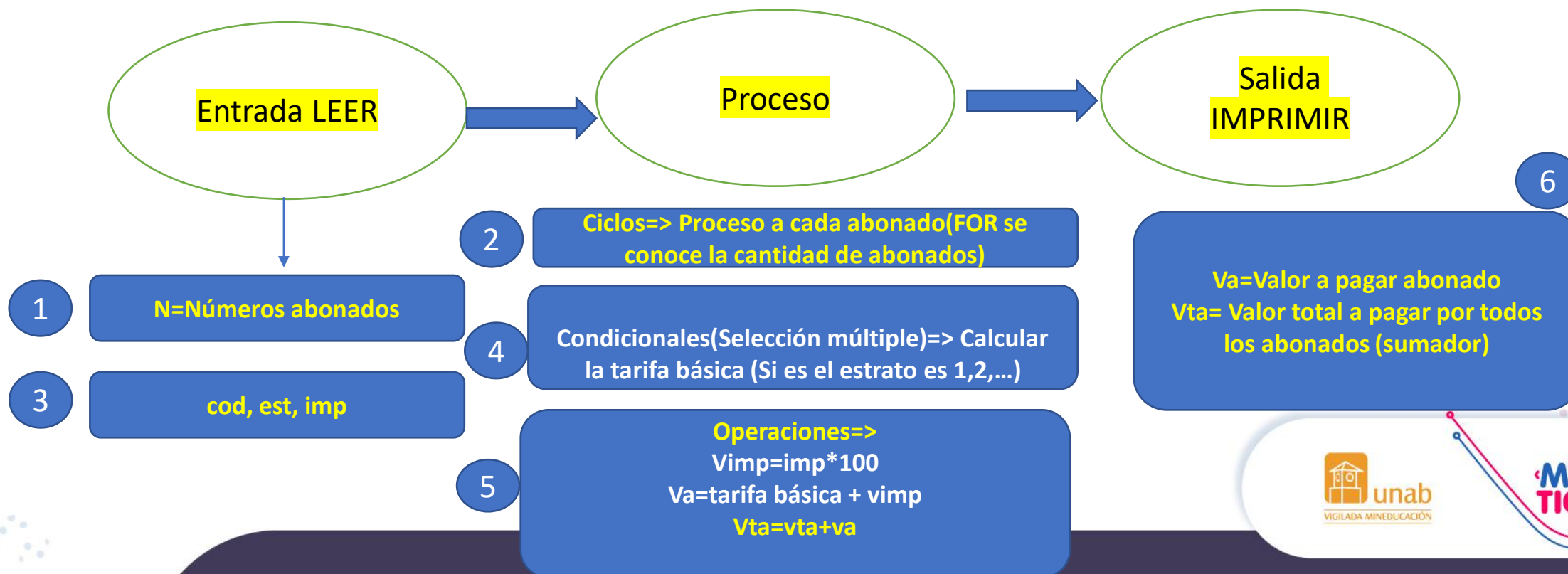


## Funciones

## Ejercicios



Análisis → Ejercicio funciones







## Funciones

## Ejercicios



Modularidad

Parámetros de entrada

Est, imp



**MODULO (FUNCION)**  
**Factura\_abonado:**

Cálculo de la tarifa básica  
Cálculo del valor impulsos  
Cálculo del valor a pagar



Parámetros de salida

Va=Valor a pagar

**FUNCIÓN** retorna o regresa un solo valor

Parte de la

4

5



## Estructuras de control

Variables del programa

Identificación de variables

ENTRADA:  
N, est =>int  
ced, imp => long

SALIDA:  
Va, vta=> double

## Ejercicios





El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



facturacion\_telefono\_funciones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects × Files Services

- facturacion\_telefono\_funciones
  - Source Packages
    - facturacion\_telefono\_funciones
      - Facturacion\_telefono\_funciones.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication1

main - Navigator ×

Members

<empty>

- Facturacion\_telefono\_funciones
  - factura\_abonado(int estrato, long impulsos) : double
  - main(String[] args)

Start Page × JavaApplication1.java × Facturacion\_telefono\_funciones.java ×

Source History

```
6 package facturacion_telefono_funciones;
7
8 /**
9  *
10  * @author SERGIO
11  */
12 import java.util.Scanner;
13
14 public class Facturacion_telefono_funciones {
15
16     /**
17      * @param args the command line arguments
18      */
19     static double factura_abonado(int estrato, long impulsos)
20     {
21         double tb=0, va;
22         switch (estrato)
23         {
```

facturacion\_telefono\_funciones.Facturacion\_telefono\_funciones > main > for (int i = 1; i <= N; i++)

Output - facturacion\_telefono\_funciones (run) ×

```
Impulsos mes:
100
Valor factura: 20000.0
Cedula:
2
Estrato (1, 2, 3, 4, 5):
4
Impulsos mes:
200
Valor factura: 45000.0
Valor total facturación: 65000.0
BUILD SUCCESSFUL (total time: 15 seconds)
```



El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



facturacion\_telefono\_funciones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects Files Services

- facturacion\_telefono\_funciones
  - Source Packages
    - facturacion\_telefono\_funciones
      - Facturacion\_telefono\_funciones.java
  - Test Packages
  - Libraries
  - Test Libraries
  - JavaApplication1

main - Navigator

Members

- Facturacion\_telefono\_funciones
  - factura\_abonado(int estrato, long impulsos) : double
  - main(String[] args)

Start Page JavaApplication1.java Facturacion\_telefono\_funciones.java

Source History

```
16 /**
17  * @param args the command line arguments
18  */
19 static double factura_abonado(int estrato, long impulsos)
20 {
21     double tb=0, va;
22     switch (estrato)
23     {
24         case 1:tb=10000;break;
25         case 2:tb=15000;break;
26         case 3:tb=20000;break;
27         case 4:tb=25000;break;
28         case 5:tb=30000;break;
29     }
30     va=tb+impulsos*100;
31     return va;
32 }
33
```

facturacion\_telefono\_funciones.Facturacion\_telefono\_funciones main for (inti = 1; i <= N; i++)

Output - facturacion\_telefono\_funciones (run)

```
Impulsos mes:
100
Valor factura: 20000.0
Cedula:
2
Estrato(1,2,3,4,5):
4
Impulsos mes:
200
Valor factura: 45000.0
Valor total facturación: 65000.0
BUILD SUCCESSFUL (total time: 15 seconds)
```

Mision  
TIC2022





El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



facturacion\_telefono\_funciones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

<default config>

Projects × Files Services

- facturacion\_telefono\_funciones
  - Source Packages
    - facturacion\_telefono\_funciones
      - Facturacion\_telefono\_funciones.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication1

main - Navigator ×

Members

- Facturacion\_telefono\_funciones
  - factura\_abonado(int estrato, long impulsos) : double
  - main(String[] args)

Start Page × JavaApplication1.java × Facturacion\_telefono\_funciones.java ×

Source History

```
34 public static void main(String[] args) {
35     // Instancia Scanner
36     Scanner entrada=new Scanner(System.in);
37     //Definición de variables
38     int est,N;
39     long imp,ced;
40     double va=0,vta=0;
41     System.out.println("Cantidad de Abonados: ");
42     N=entrada.nextInt();
43     for (int i=1;i<=N;i++)
44     {
45         System.out.println("Cedula: ");
46         ced=entrada.nextLong();
47         System.out.println("Estrato(1,2,3,4,5): ");
48         est=entrada.nextInt();
49         System.out.println("Impulsos mes: ");
50         imp=entrada.nextLong();
```

facturacion\_telefono\_funciones.Facturacion\_telefono\_funciones > main > for (int i = 1; i <= N

Output - facturacion\_telefono\_funciones (run) ×

```
Impulsos mes:
100
Valor factura: 20000.0
Cedula:
2
Estrato(1,2,3,4,5):
4
Impulsos mes:
200
Valor factura: 45000.0
Valor total facturación: 65000.0
BUILD SUCCESSFUL (total time: 15 seconds)
```





El futuro digital  
es de todos

MinTIC

# Estructura de Datos

Construcción → Programa

# Ejercicios



facturacion\_telefono\_funciones - NetBeans IDE 8.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Projects X Files Services

- facturacion\_telefono\_funciones
  - Source Packages
    - facturacion\_telefono\_funciones
      - Facturacion\_telefono\_funciones.java
  - Test Packages
  - Libraries
  - Test Libraries
- JavaApplication1

main - Navigator X

Members

- Facturacion\_telefono\_funciones
  - factura\_abonado(int estrato, long impulsos) : double
  - main(String[] args)

Source

```
42 entrada.nextLine();
43 for (int i=1;i<=N;i++)
44 {
45     System.out.println("Cedula: ");
46     ced=entrada.nextLong();
47     System.out.println("Estrato(1,2,3,4,5): ");
48     est=entrada.nextInt();
49     System.out.println("Impulsos mes: ");
50     imp=entrada.nextLong();
51     // Llamada a la función
52     va=factura_abonado(est,imp);
53     System.out.println("Valor factura: "+va);
54     vta+=va;
55 }
56 System.out.println("Valor total facturación: "+vta);
57 }
58
59 }
```

facturacion\_telefono\_funciones.Facturacion\_telefono\_funciones > main > for (inti = 1; i <= N; i+)

Output - facturacion\_telefono\_funciones (run) X

```
Impulsos mes:
100
Valor factura: 20000.0
Cedula:
2
Estrato(1,2,3,4,5):
4
Impulsos mes:
200
Valor factura: 45000.0
Valor total facturación: 65000.0
BUILD SUCCESSFUL (total time: 15 seconds)
```



El futuro digital  
es de todos

MinTIC

**Ciclo: Programación Básica**

**Introducción a JAVA**

»  
**Misión TIC 2022**

xxx



**Misión  
TIC 2022**