

## Practica 5

---

Course: *Introducción a las ciencias de la computación* – Professor: *Vilchis Salazar Jose Antonio*  
Due date: *24 Septiembre, 2019*

### Objetivo

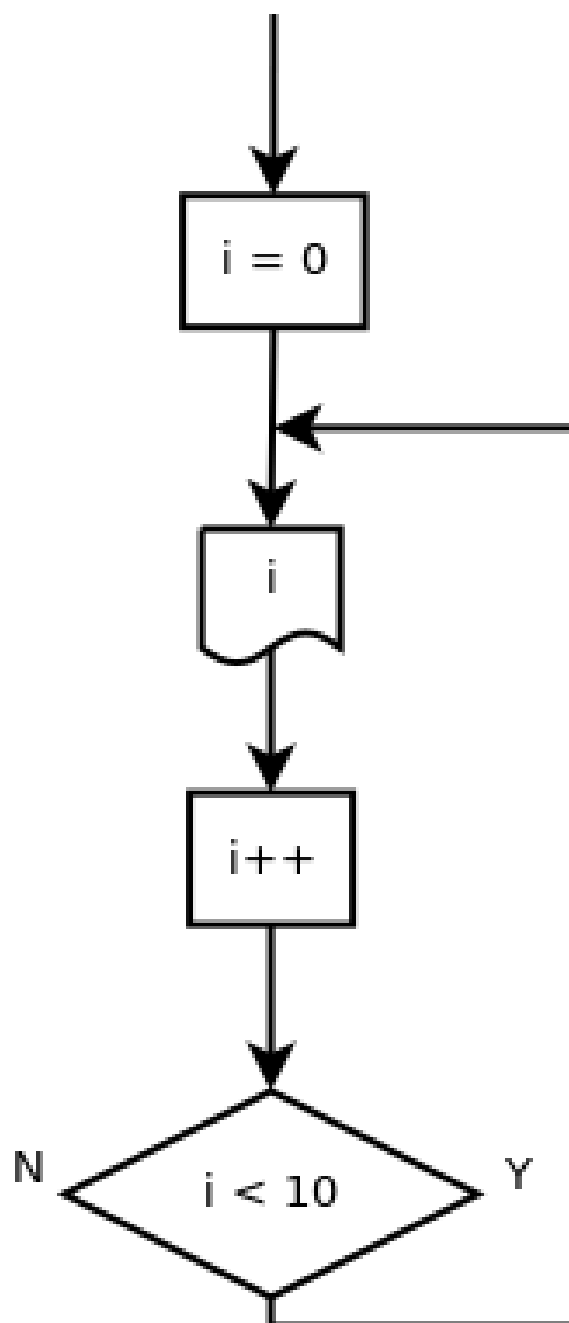
Que el alumno aprenda el uso de las estructuras de control do-while y switch. Y que aprenda su representación en diagramas de flujo.

**Do While.** En Java, además de for y while, también existe una estructura de control llamada do while que también es un ciclo, sin embargo, tiene la peculiaridad de que supone que el ciclo se ejecuta invariablemente al menos una vez.

Listing 1: Sintaxis de do while

```
1 package icc.prueba;
2
3 public class Prueba {
4     public static void main(String[] args){
5         int i = 0;
6         boolean b = true;
7         do{
8             System.out.println(i);
9             i++;
10            if(i >= 10){
11                b = false;
12            }
13        }while(b);
14    }
15 }
```

La forma en la que se representa do while de forma no reducida en diagramas es:

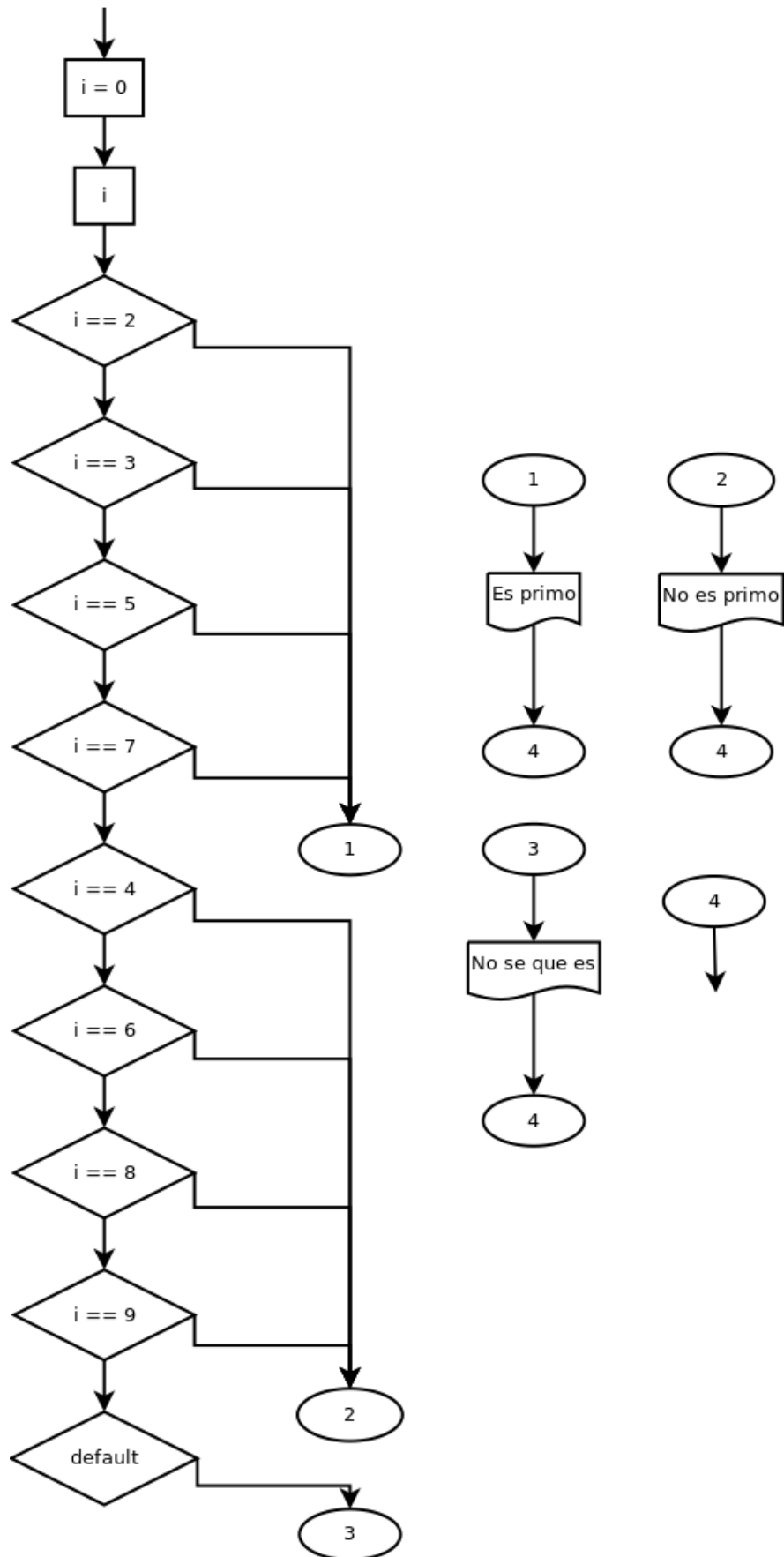


**Switch.** En Java, existe la estructura de control Switch, esta permite extender la funcionalidad del if-elseif-es para casos múltiples, sin embargo, tiene la limitante de que el flujo de programa se debe mantener con UNA sola variable, esta variable solo puede tener el tipo de los primitivos enteros, tipos enumerados, objetos String y objetos de las clases wrapper de los primitivos enteros.

## Listing 2: Sintaxis de switch

```
1 package icc.prueba;
2
3 public class Prueba {
4     public static void main(String[] args){
5         int i = 4;
6         switch (i) {
7             case 2:
8             case 3:
9             case 5:
10            case 7:
11                System.out.println("Es primo");
12                break;
13            case 4:
14            case 6:
15            case 8:
16            case 9:
17                System.out.println("No es primo");
18                break;
19            default:
20                System.out.println("No se que es");
21                break;
22        }
23    }
24 }
```

La forma en la que se representa switch de forma no reducida en diagramas es:



## Desarrollo

En la clase Scanner se tiene la familia de métodos *hasNext*, la cual permite conocer el tipo específico del siguiente token en el flujo de datos del Scanner, estos métodos devuelve un valor boolean, donde *true* representa que el tipo preguntado es el tipo del siguiente token.

Listing 3: Ejemplo de uso para el flujo de datos con hasNext

```
1 package icc.prueba;
2
3 public class Prueba {
4     public static void main(String[] args){
5         int i;
6         String s;
7         Scanner scn = new Scanner(System.in);
8         if(scn.hasNextInt()){
9             i = scn.nextInt();
10        } else {
11            s = scn.next();
12        }
13    }
14 }
```

## Actividades

Lee la documentación de do-while, switch, y scanner.

Implementa un menú fuera de main. El único tipo de ciclo que se puede usar es do-while.

Haz funcionar Prueba.java sin modificarlo

La funcionalidad de la clase a implementar esta separada en dos métodos.

Un método llamado run, el cual no recibe ni devuelve nada, quien se encarga de darle las opciones al usuario sobre lo que puede hacer. Debe darle dos opciones: si presiona 1 debe de darle la opción de convertir un número entero a binario; y si presiona 0 debe de salir de la aplicación. Si el usuario presiona otra cosa debe de avisarle de que fue una opción invalida. RESTRICCIONES: No se puede utilizar un Scanner dentro de este método.

Un método llamado getInt, este devuelve un entero y recibe dos cadenas, una cadena que es un mensaje al usuario sobre lo que debe hacer, y una cadena que es un mensaje al usuario cuando se equivoque. Este método debe asegurarse de que el usuario ingrese un número tipo int, si el usuario ingresa algo que no sea un int debe imprimir el mensaje de error y volver a pedirle que ingrese un int; esto se repetirá hasta que el usuario ingrese un int.

### **Preguntas**

1. Semánticamente ¿cuál es la diferencia entre if y switch?
  2. ¿Es necesaria la cláusula default para la correcta ejecución de switch?
  3. ¿Es posible simular el do-while con while? Si es posible escribe un código de ejemplo de como seria, si no lo es escribe porque.
  4. ¿Qué pasa si se pregunta un hasNextDouble() pero se manda un int? ¿Qué devuelve hasNextDouble()?
-