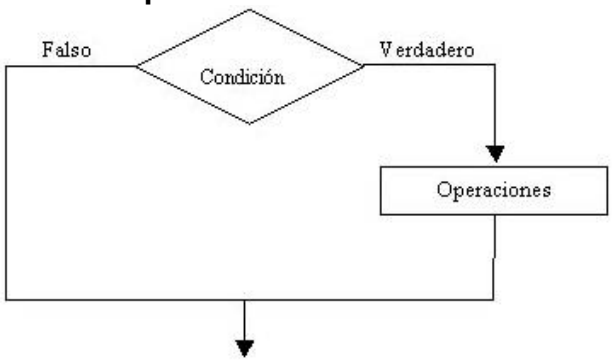
# MISCELÁNEA DE EJERCICIOS EJE TEMÁTICO 1 ESTRUCTURAS DE CONTROL EN LENGUAJE JAVA

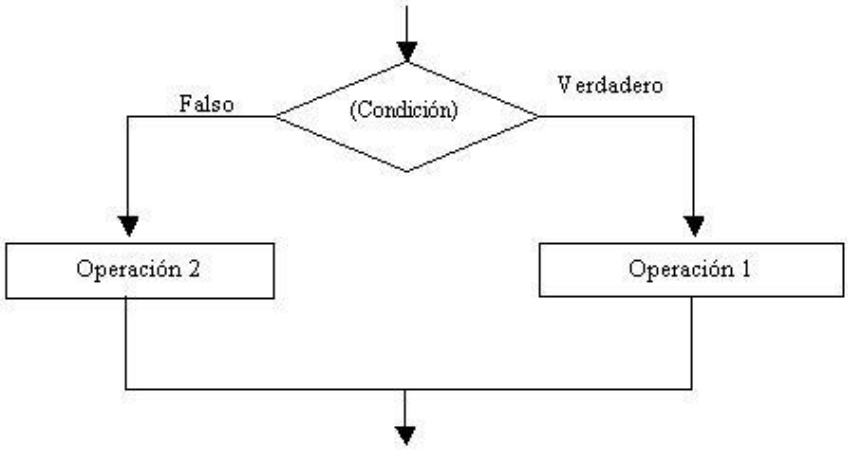
## Estructuras condicionales simples y compuestas

### Estructura condicional simple.



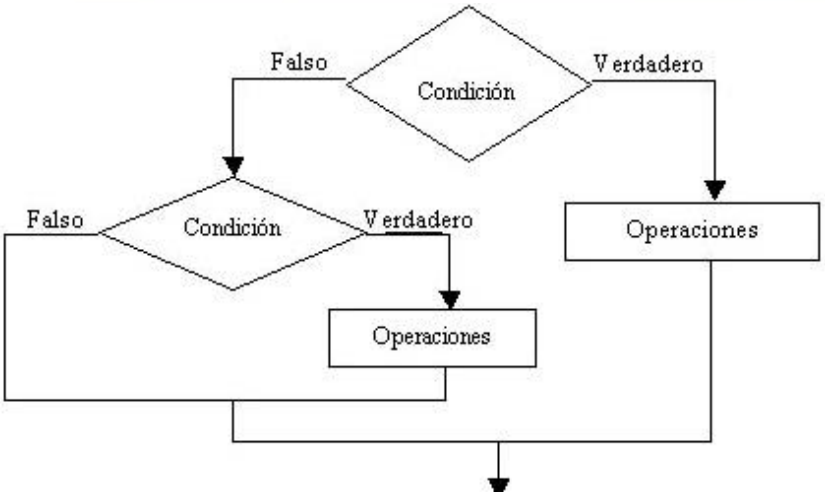

1. Ingresar el sueldo de una persona, si supera los 3000 pesos mostrar un mensaje en pantalla indicando que debe abonar impuestos.

### Estructura condicional compuesta.

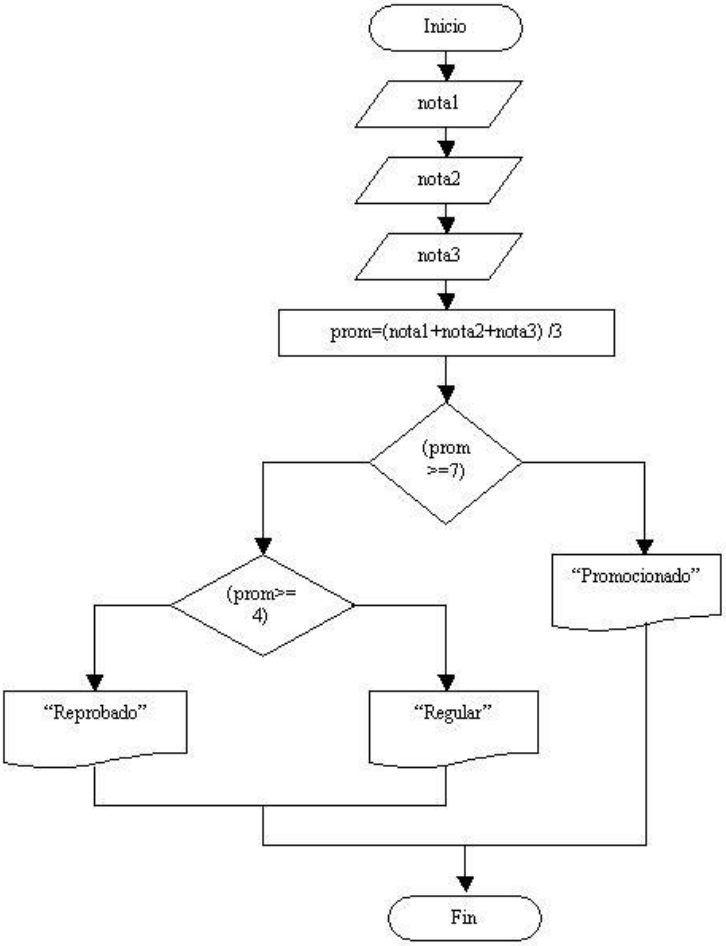



2. Realizar un programa que solicite ingresar dos números distintos y muestre por pantalla el mayor de ellos.

## Estructuras condicionales anidadas



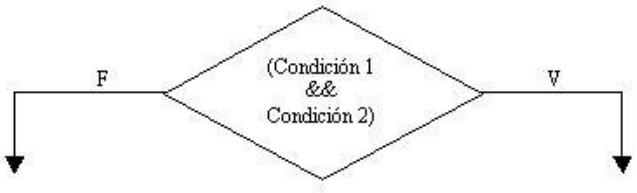

3. Confeccionar un programa que pida por teclado tres notas de un alumno, calcule el promedio e imprima alguno de estos mensajes:
   Si el promedio es >=7 mostrar "Promocionado".
   Si el promedio es >=4 y <7 mostrar "Regular".
   Si el promedio es <4 mostrar "Reprobado".



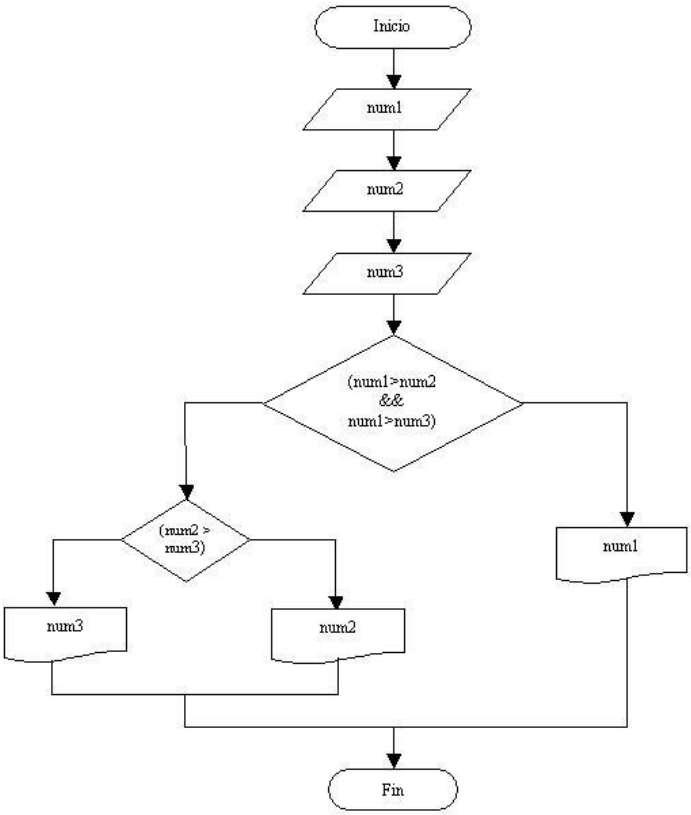
4. Un postulante a un empleo, realiza un test de capacitación, se obtuvo la siguiente información: cantidad total de preguntas que se le realizaron y la cantidad de preguntas que contestó correctamente. Se pide confeccionar un programa que ingrese los dos datos por teclado e informe el nivel del mismo según el porcentaje de respuestas correctas que ha obtenido, y sabiendo que:
   Nivel máximo: Porcentaje>=90%.
   Nivel medio: Porcentaje>=75% y <90%.
   Nivel regular: Porcentaje>=50% y <75%.
   Fuera de nivel: Porcentaje<50%.

**Condiciones compuestas con operadores lógicos**

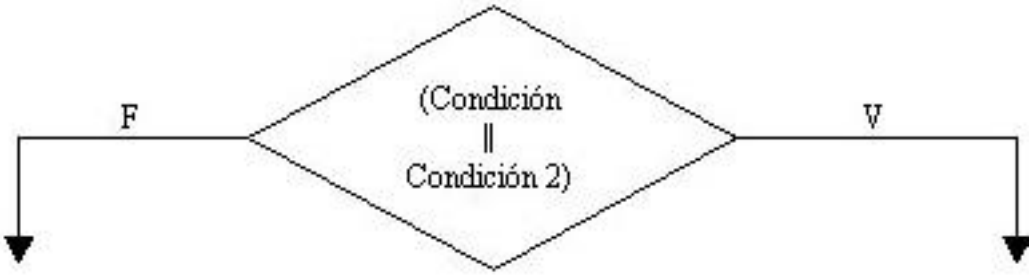
- Operador **and &&**




5. Confeccionar un programa que lea por teclado tres números distintos y nos muestre el mayor.

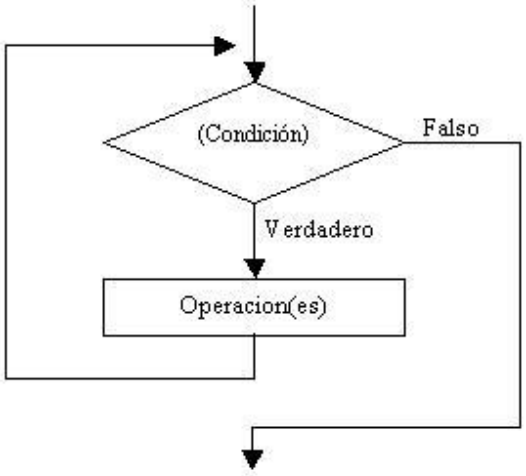


- Operador **or ||**

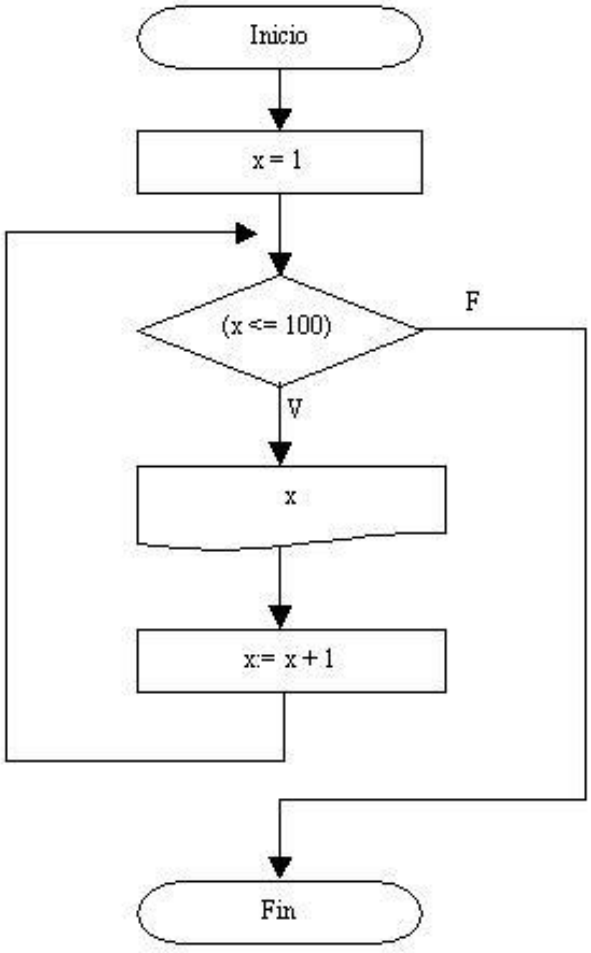



6. De un operario se conoce su sueldo y los años de antigüedad. Se pide confeccionar un programa que lea los datos de entrada e informe:
   a) Si el sueldo es inferior a 1000000 y su antigüedad es igual o superior a 10 años, otorgarle un aumento del 20 %, mostrar el sueldo a pagar.
   b)Si el sueldo es inferior a 1000000 pero su antigüedad es menor a 10 años, otorgarle un aumento de 5 %.
   c) Si el sueldo es mayor o igual a 1000000 mostrar el sueldo en pantalla sin cambios.

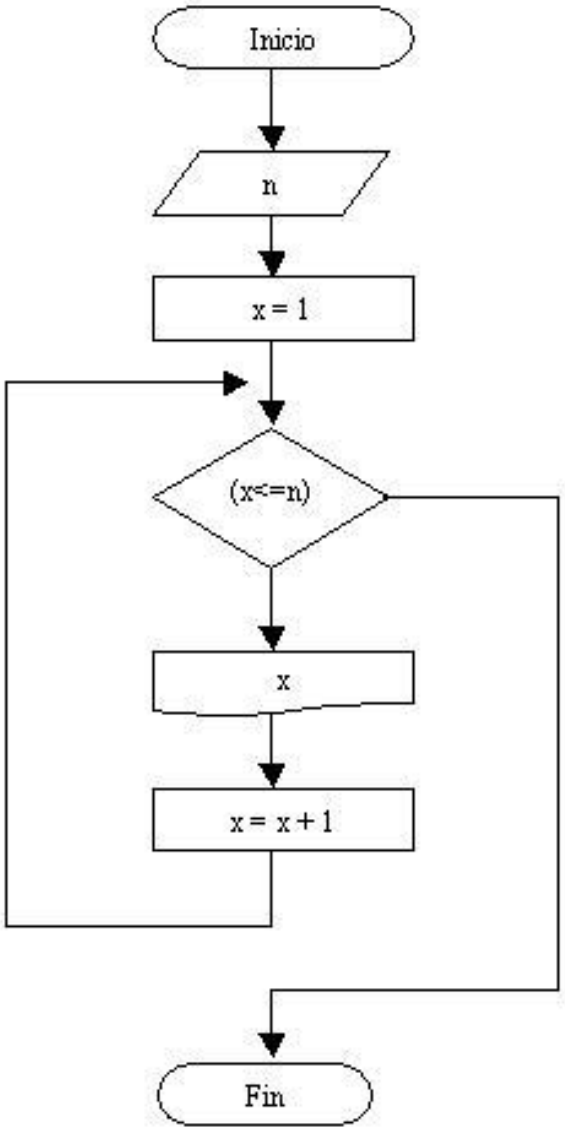
**Estructura repetitiva while**



7. Realizar un programa que imprima en pantalla los números del 1 al 100.



8. Escribir un programa que solicite la carga de un valor positivo y nos muestre desde 1 hasta el valor ingresado de uno en uno. Ejemplo: Si ingresamos 30 se debe mostrar en pantalla los números del 1 al 30.




**Declaración de métodos** (**Con esta explicación puede resolver el reto de la semana 1**)

Con la declaración de métodos divides todas las responsabilidades de la clase en pequeñas acciones.
Un método hemos visto que tiene la siguiente sintaxis:
public void [nombre del método]() {
[algoritmo]

}

Consejo: Si desea un método para devolver un valor, se puede utilizar un tipo de datos primitivo (como int, char, etc.) en lugar de void
Ejemplo:

**Métodos con parámetros.**
Un método puede tener parámetros:
public void [nombre del método]([parámetros]) {
  [algoritmo]
}
Los parámetros los podemos imaginar como variables locales al método, pero su valor se inicializa con datos que llegan cuando lo llamamos.

```
public class Main {
  static void myMethod() {
    System.out.println("I just got executed!");
  }

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

**Métodos que retornan un dato.**
Un método puede retornar un dato:

public [tipo de dato] [nombre del método]([parámetros]) {
  [algoritmo]
  return [tipo de dato]
}
Cuando un método retorna un dato en vez de indicar la palabra clave void previo al nombre del método indicamos el tipo de dato que retorna. Luego dentro del algoritmo en el momento que queremos que finalice el mismo y retorne el dato empleamos la palabra clave return con el valor respectivo.

```
public class Main {
  static int myMethod(int x) {
    return 5 + x;
  }
  public static void main(String[] args) {
    System.out.println(myMethod(3));
  }
}
```