

TEMA 2. APLICACIÓN DE LAS ESTRUCTURAS DE CONTROL

INDICE

| | |
|---|----|
| Indice..... | 1 |
| Estructuras Condicionales O Alternativas. Pseudocodigo..... | 2 |
| Ejercicios De Estructuras Condicionales O Alternativas. Pseudocodigo..... | 2 |
| Tablas De Decisión | 3 |
| Estructuras Condicionales O Alternativas. Java | 5 |
| Ejercicios De Estructuras Condicionales O Alternativas. Java | 5 |
| Estructuras Selectivas. Pseudocodigo | 7 |
| Ejercicios De Estructuras Selectivas. Pseudocodigo | 7 |
| Estructuras Selectivas. Java | 8 |
| Ejercicios De Estructuras Selectivas. Java | 9 |
| Estructuras Repetitivas O Iterativas. | 9 |
| Estructuras Repetitivas De 0 A N Veces. Pseudocodigo | 10 |
| Estructuras Repetitivas De 1 A N Veces. Pseudocodigo | 10 |
| Estructuras Repetitivas De N Veces. Pseudocodigo | 10 |
| Ejercicios De Estructuras Repetitivas. Pseudocodigo | 10 |
| Estructuras Repetitivas De 0 A N Veces. Java..... | 16 |
| Estructuras Repetitivas De 1 A N Veces. Java..... | 16 |
| Estructuras Repetitivas De N Veces. Java..... | 16 |
| Ejercicios De Estructuras Repetitivas. Java | 17 |

ESTRUCTURAS CONDICIONALES O ALTERNATIVAS. PSEUDOCODIGO

Las estructuras condicionales o alternativas nos permiten seleccionar el código que se ejecuta en función de una **condición**. Si la condición es verdadera (**true**) ejecutará un código y, opcionalmente, si es falsa (**false**) ejecutará otro código distinto.

Su representación mediante pseudocódigo si no se hace nada cuando la condición es false es

| |
|--|
| SI condición ENTONCES Código si se cumple la condición FIN-SI |
|--|

Su representación mediante pseudocódigo si tenemos que hacer algo cuando la condición es false es

| |
|--|
| SI condición ENTONCES Código si se cumple la condición SINO Código si NO se cumple la condición FIN-SI |
|--|

EJERCICIOS DE ESTRUCTURAS CONDICIONALES O ALTERNATIVAS. PSEUDOCODIGO

1. Realiza el pseudocódigo para el programa VALORABS que lee el valor de un número entero por teclado y muestra su valor absoluto por pantalla.
2. Realiza el pseudocódigo para el programa PARIMPAR que lee el valor de un número entero por teclado y muestra por pantalla si es par o impar.
3. Realiza el pseudocódigo para el programa CALCUIVA que lee una cantidad en euros por teclado y muestra por pantalla su IVA sabiendo que si la cantidad es menor que 20000 el IVA es el 7% y sino del 16%.
4. Realiza el pseudocódigo para el programa NXN que lee un número entero por teclado y calcula el cuadrado de ese número. Si el cuadrado es mayor que 100 se le restara 100 y se mostrara por pantalla. Si no es mayor que 100, se mostrara lo que le falta para llegar a 100.
5. Realiza el pseudocódigo para el programa MAYORMENOR que lee dos números enteros por teclado a y b y muestra por pantalla si el primero es mayor que el segundo, si el segundo es mayor que el primero o si son iguales.
6. Realiza el pseudocódigo para el programa RETESUEL que lee el sueldo de un trabajador por teclado y muestra por pantalla la retención que se le aplica sabiendo que si

```
suelo < 1000.00 retencion 10%
suelo 1000.00 retencion 12%
suelo > 1000.00 retencion 14%
```
7. Realiza el pseudocódigo para el programa RETESUE2 que lee el sueldo de un trabajador por teclado y muestra por pantalla la retención que se le aplica sabiendo que si

```
suelo < 1000.00 retencion 10%
suelo 1000.00 retencion 12%
suelo < 2000.00 retencion 14%
suelo 2000.00 retencion 16%
suelo > 2000.00 retencion 18%
```

8. Realiza el pseudocódigo para el programa SEGUNDOI que lee una hora con sus minutos y segundos por teclado, la incrementa un segundo y la muestra por pantalla. Por ejemplo, si recibe 54h 55m 59s devolverá 54h 56m 0s.

9. Realiza el pseudocódigo para el programa TABNOTAS que lee la nota de un alumno por teclado y muestra por pantalla la calificación que le corresponde sabiendo que si

| NOTA | CALIFICACION |
|------------|--------------|
| >=0 y <3 | MD |
| >=3 y <5 | INS |
| >=5 y <6 | SUF |
| >=6 y <7 | BIEN |
| >=7 y <9 | NOT |
| >=9 y <=10 | SOBRE |

10. Realiza el pseudocódigo para el programa ARTICULOS que pide un código de artículo, la cantidad comprada de ese artículo y el precio unitario de ese artículo por pantalla, calcula el descuento en función de la siguiente tabla y muestra los datos anteriores junto con el descuento y el precio total por pantalla.

Tabla

| CANTIDAD | DESCUENTO |
|-------------|-----------|
| >100 | 40% |
| >25 y <=100 | 20% |
| >10 y <=25 | 10% |
| <=10 | 0% |

11. Realiza el pseudocódigo para el programa MENUIF que pide dos números enteros por pantalla y muestra un menú para elegir la operación que se desea realizar usando IF.

1. Sumar
2. Restar
3. Multiplicar
4. Dividir
5. Resto (%)
0. Salir

12. Realiza el pseudocódigo para el programa NOTAVALI que lee la nota de un alumno por teclado y muestra por pantalla si es válida o no, sabiendo que es válida si está entre 0 y 10 ambos inclusive.

13. Realiza el pseudocódigo para el programa BISIESTO que pide un año por teclado y muestra por pantalla si es o no bisiesto. Un año es bisiesto si es múltiplo de 4 pero no lo es de 100 excepto si también es múltiplo de 400.

14. Realiza el pseudocódigo para el programa EC2GRADO que calcula las soluciones de una ecuación de 2º grado. Si las raíces son complejas muestra el mensaje de error "Raíces Complejas".

15. Realiza el pseudocódigo para el programa MAYOR3N que lee tres números enteros por teclado y muestra por pantalla cual o cuales son los mayores.

16. Realiza el pseudocódigo para el programa UNDECEN que pide un número entero entre 0 y 999 por teclado y muestra por pantalla sus unidades, decenas, y centenas.

17. Realiza el pseudocódigo para el programa DECEMIL que pide un número entero entre 0 y 99999 por teclado y muestra por pantalla sus unidades, decenas, centenas, unidades de millar, y decenas de millar.

TABLAS DE DECISIÓN

En determinadas situaciones donde tenemos que combinar varias condiciones es necesario usar tablas de decisión para determinar cuál o cuáles son las condiciones que tenemos que utilizar.

Las tablas de decisión nos facilitan la construcción de condiciones complejas.

Una tabla de decisión consta de todos los valores posibles de las condiciones y del resultado de cada una de las combinaciones de esas condiciones. Es muy importante comprobar que estén todos los valores posibles.

Una vez completada la tabla de decisión debemos estudiarla para determinar cuáles son las condiciones que debemos usar para resolver el problema.

Por ejemplo, podemos resolver el ejercicio BISIESTO usando una tabla de decisión para determinar de una manera más eficiente las condiciones que debemos usar.

La tabla de decisión para el ejercicio BISIESTO sería

| Múltiplo de 4 | Múltiplo de 100 | Múltiplo de 400 | Resultado |
|---------------|-----------------|-----------------|-----------|
| V | V | V | V |
| V | V | F | F |
| V | F | V | V |
| V | F | F | V |
| F | V | V | F |
| F | V | F | F |
| F | F | V | F |
| F | F | F | F |

A la hora de rellenar esta tabla nos damos cuenta de que hay opciones que nunca se van a dar. Por ejemplo, si un año no es múltiplo de 4 no puede ser múltiplo de 100, y si no es múltiplo de 100 no puede ser de 400. Estas opciones las he marcado en color azul aunque en realidad no deberían aparecer en la tabla.

Una vez completada la tabla marcamos los resultados que queremos controlar. Podemos controlar las condiciones que producen un resultado Verdadero o las que producen un resultado Falso.

Si queremos controlar las condiciones que producen un resultado **Verdadero**, las marcamos de color rojo y las estudiamos para deducir las condiciones que debemos usar para resolver el problema. Después de realizar el estudio llegamos a la conclusión de que la condición que debemos usar es:

SI (((AÑO MOD 4 == 0) Y (AÑO MOD 100 != 0))) O
((AÑO MOD 400 == 0)) ENTONCES

Si por el contrario queremos controlar las condiciones que producen un resultado **Falso**, las marcamos rellenando las celdas de color verde y las estudiamos para deducir las condiciones que debemos usar para resolver el problema. Después de realizar el estudio llegamos a la conclusión de que la condición que debemos usar es:

SI ((AÑO MOD 4 != 0) O
((AÑO MOD 4 == 0) Y (AÑO MOD 100 == 0) Y (AÑO MOD 400 != 0)) ENTONCES

En la mayoría de los casos los problemas se pueden resolver de manera más sencilla mediante el uso de condiciones sencillas anidadas en vez de usando condiciones compuestas. Sin embargo en otros casos sólo existe la posibilidad de resolver el problema mediante el uso de condiciones compuestas. En estos casos se recomienda el uso de tablas de decisión.

18. Realiza el pseudocódigo para el programa BISIESTO1C que pide un año por teclado y muestra por pantalla si es o no bisiesto. Un año es bisiesto si es múltiplo de 4 pero no lo es de 100 excepto si también es múltiplo de 400. Para la solución se usa una única condición.

ESTRUCTURAS CONDICIONALES O ALTERNATIVAS. JAVA

Las estructuras condicionales o alternativas nos permiten seleccionar el código que se ejecuta en función de una **condición**. Si la condición es verdadera (**true**) ejecutará un código y, opcionalmente, si es falsa (**false**) ejecutará otro código distinto.

Su representación en Java se realiza mediante la siguiente estructura si no se hace nada cuando la condición es false es

```
if condición {  
    Código si se cumple la condición  
}
```

Su representación mediante Java si tenemos que hacer algo cuando la condición es false es

```
if condición {  
    Código si se cumple la condición  
}  
else {  
    Código si NO se cumple la condición  
}
```

Su representación mediante Java si tenemos que usar condiciones anidadas es

```
if condición {  
    Código si se cumple condición  
}  
else if condición2 {  
    Código si se cumple condición2  
}  
else {  
    Código si se no se cumple ninguna condición  
}
```

EJERCICIOS DE ESTRUCTURAS CONDICIONALES O ALTERNATIVAS. JAVA

19. Realiza la clase Java Valorabs que lee el valor de un número real por teclado y muestra su valor absoluto por pantalla.

20. Realiza la clase Java Parimpar que lee el valor de un número entero por teclado y muestra por pantalla si es par o impar.

21. Realiza la clase Java Calcuiva que lee una cantidad en euros por teclado y muestra por pantalla su IVA sabiendo que si la cantidad es menor que 20000 el IVA es el 7% y sino del 16%.

22. Realiza la clase Java Nxn que lee un número por teclado y calcula el cuadrado de ese número. Si el cuadrado es mayor que 100 se le restará 100 y se mostrará por pantalla. Si no es mayor que 100, se mostrará lo que le falta para llegar a 100.

23. Realiza la clase Java Mayormenor que lee dos números por teclado a y b y muestra por pantalla si el primero es mayor que el segundo, si el segundo es mayor que el primero o si son iguales.

24. Realiza la clase Java Retesuel que lee el sueldo de un trabajador por teclado y muestra por pantalla la retención que se le aplica sabiendo que si
- ```
sueldo < 1000.00 retencion 10%
sueldo 1000.00 retencion 12%
sueldo > 1000.00 retencion 14%
```
25. Realiza la clase Java Retesue2 que lee el sueldo de un trabajador por teclado y muestra por pantalla la retención que se le aplica sabiendo que si
- ```
sueldo < 1000.00 retencion 10%
sueldo 1000.00 retencion 12%
sueldo < 2000.00 retencion 14%
sueldo 2000.00 retencion 16%
sueldo > 2000.00 retencion 18%
```
26. Realiza la clase Java Segundoi que lee una hora con sus minutos y segundos por teclado, la incrementa un segundo y la muestra por pantalla. Por ejemplo, si recibe 54h 55m 59s devolverá 54h 56m 0s.
27. Realiza la clase Java Tabnotas que lee la nota de un alumno por teclado y muestra por pantalla la calificación que le corresponde sabiendo que si
- | NOTA | CALIFICACION |
|------------|--------------|
| >=0 y <3 | MD |
| >=3 y <5 | INS |
| >=5 y <6 | SUF |
| >=6 y <7 | BIEN |
| >=7 y <9 | NOT |
| >=9 y <=10 | SOBRE |
28. Realiza la clase Java Articulos que pide un código de artículo, la cantidad comprada de ese artículo y el precio unitario de ese artículo por pantalla, calcula el descuento en función de la siguiente tabla y muestra los datos anteriores junto con el descuento y el precio total por pantalla. Tabla
- | CANTIDAD | DESCUENTO |
|-------------|-----------|
| >100 | 40% |
| >25 y <=100 | 20% |
| >10 y <=25 | 10% |
| <=10 | 0% |
29. Realiza la clase Java Menuif que muestra un menú y, en función de la opción, pide dos números enteros por pantalla y calcula una operación.
- ```
1. Sumar
2. Restar
3. Multiplicar
4. Dividir
5. Resto (%)
0. Salir
```
30. Realiza la clase Java Notavali que lee la nota de un alumno por teclado y muestra por pantalla si es válida o no, sabiendo que es válida si está entre 0 y 10.
31. Realiza la clase Java Bisiesto que pide un año por teclado y muestra por pantalla si es o no bisiesto. Un año es bisiesto si es múltiplo de 4 pero no lo es de 100 excepto si también es múltiplo de 400.
32. Realiza la clase Java Ec2grado que calcula las soluciones de una ecuación de 2º grado. Si las raíces son complejas muestra el mensaje de error "Raíces Complejas". Para calcular la raíz cuadrada hay que usar el método **Math.sqrt(determinante)**.

33. Realiza la clase Java Mayor3n que lee tres números por teclado y muestra por pantalla cual o cuales son los mayores.
34. Realiza la clase Java Undecen que pide un número entero entre 0 y 999 por teclado y muestra por pantalla sus unidades, decenas, y centenas.
35. Realiza la clase Java Decemil que pide un número entero entre 0 y 99999 por teclado y muestra por pantalla sus unidades, decenas, centenas, unidades de millar, y decenas de millar.

## ESTRUCTURAS SELECTIVAS. PSEUDOCODIGO

En determinadas ocasiones, por ejemplo en el caso de un menú, todas las condiciones dependen del valor de una única variable. En esos casos especiales podemos usar estructuras selectivas que simplifican las instrucciones a escribir.

Su representación mediante pseudocódigo es

```

SELECCIONA variable
 CASO valor1:
 Código si variable vale valor1
 CASO valorn:
 Código si variable vale valorn
 PORDEFECTO:
 Código si variable tiene otro valor
FIN-SELECCIONA

```

La opción PORDEFECTO es opcional y sólo se usa cuando queremos controlar lo que pasa cuando se introduce un valor desconocido.

## EJERCICIOS DE ESTRUCTURAS SELECTIVAS. PSEUDOCODIGO

36. Realiza el pseudocódigo para el programa MENCASE que muestra un menú y, en función de la opción, muestra un mensaje indicando la opción seleccionada usando una estructura selectiva.
1. Opción 1
  2. Opción 2
  3. Opción 3
  4. Opción 4
  5. Opción 5
  0. Salir
37. Realiza el pseudocódigo para el programa MENUDIAS que pide un carácter por pantalla y muestra el día de la semana que le corresponde según la siguiente tabla
- |      |           |       |
|------|-----------|-------|
| L    | Lunes     |       |
| M    | Martes    |       |
| X    | Miércoles |       |
| J    | Jueves    |       |
| V    | Viernes   |       |
| S    | Sábado    |       |
| D    | Domingo   |       |
| Otro | Carácter  | ERROR |
38. Realiza el pseudocódigo para el programa MENCMESES que pide un String con el nombre del mes por pantalla y muestra el número de mes que le corresponde.
39. Realiza el pseudocódigo para el programa EXPENDEDOR que pide un saldo en euros al cliente, muestra un menú donde elegir café (0,43 €), refrescos (1,11 €), o agua (0,36 €) y, si el saldo es mayor o igual que el

importe del producto saca el mensaje “Su producto. Gracias” y le devuelve el cambio indicando el número de monedas de cada tipo de las que se compone el cambio (2€, 1€, 50cts, 20cts, 10cts, 5cts, 2cts, 1cts). Si el saldo no es mayor o igual que el importe del producto muestra el mensaje “Saldo insuficiente” y devuelve el saldo. Si se introduce una opción incorrecta mostrar el mensaje “Opción incorrecta” y devuelve el saldo.

40. Realiza el pseudocódigo para el programa ROMANOS que pide un número entero entre 1 y 99 y lo muestra en números romanos. Si se introduce un número incorrecto muestra el mensaje “Número incorrecto”.

## ESTRUCTURAS SELECTIVAS. JAVA

En determinadas ocasiones, por ejemplo en el caso de un menú, todas las condiciones dependen del valor de una única variable. En esos casos especiales podemos usar estructuras selectivas que simplifican las instrucciones a escribir.

En Java la sentencia que se usa es **switch case** que sirve para datos de tipo **int**, **char**, y **String**.

Su representación mediante Java es

```
switch (variable) {
 case valor1:
 Código si variable vale valor1
 break;
 case valorn:
 Código si variable vale valorn
 break;
 default:
 Código si variable tiene otro valor
}
```

La opción **default** es opcional y sólo se usa cuando queremos controlar lo que pasa cuando se introduce un valor desconocido.

La sentencia **break** permite finalizar la ejecución de una sentencia de manera incondicional. En este caso permite salir del switch tras finalizar la ejecución del código relativo al valor introducido. Si no se incluye la sentencia break, al finalizar el código de una opción se continua ejecutando el de la siguientes opciones hasta que se llega al final de la sentencia switch.

Si varios valores de la variable (por ejemplo valor1, valor2 y valor3) comparten el mismo código se puede simplificar la sintaxis de la siguiente manera

```
switch (variable) {
 case valor1:
 case valor2:
 case valor3:
 Código si variable vale valor1, valor2, o valor3
 break;
 case valorn:
 Código si variable vale valorn
 break;
 default:
 Código si variable tiene otro valor
}
```



## EJERCICIOS DE ESTRUCTURAS SELECTIVAS. JAVA

41. Realiza la clase Java Menucase que muestra un menú y, en función de la opción, muestra un mensaje indicando la opción seleccionada usando una estructura selectiva.

1. Opción 1
2. Opción 2
3. Opción 3
4. Opción 4
5. Opción 5
0. Salir

42. Realiza la clase Java Menudias que pide un carácter por pantalla y muestra el día de la semana que le corresponde según la siguiente tabla

|               |           |       |
|---------------|-----------|-------|
| L             | Lunes     |       |
| M             | Martes    |       |
| X             | Miércoles |       |
| J             | Jueves    |       |
| V             | Viernes   |       |
| S             | Sábado    |       |
| D             | Domingo   |       |
| Otro Carácter |           | ERROR |

43. Realiza la clase Java Menumeses que pide un String con el nombre del mes por pantalla y muestra el número de mes que le corresponde.

44. Realiza la clase Java Menunotas que pide la nota de un alumno y muestra su calificación siguiendo la tabla del ejercicio Tabnotas usando una estructura selectiva.

45. Realiza la clase Java Expendedor que pide un saldo en euros al cliente, muestra un menú donde elegir café (0,43 €), refrescos (1,11 €), o agua (0,36 €) y, si el saldo es mayor o igual que el importe del producto saca el mensaje “Su producto. Gracias” y le devuelve el cambio indicando el número de monedas de cada tipo de las que se compone el cambio (2€, 1€, 50cts, 20cts, 10cts, 5cts, 2cts, 1cts). Si el saldo no es mayor o igual que el importe del producto muestra el mensaje “Saldo insuficiente” y devuelve el saldo. Si se introduce una opción incorrecta mostrar el mensaje “Opción incorrecta” y devuelve el saldo.

46. Realiza la clase Java Romanos que pide un número entero entre 1 y 99 y lo muestra en números romanos. Si se introduce un número incorrecto muestra el mensaje “Número incorrecto”.

## ESTRUCTURAS REPETITIVAS O ITERATIVAS.

Las estructuras repetitivas (también llamadas iterativas) permiten repetir un determinado código en función del valor de una condición.

Toda estructura repetitiva consta de las siguientes partes:

**Inicialización.** Consiste en dar un valor inicial a las variables que forman parte de la estructura repetitiva. Se aconseja realizar el proceso de inicialización de las variables justo antes de la estructura repetitiva.

**Condición de terminación.** Toda estructura repetitiva tiene una condición de terminación que permite salir de la ejecución de la misma. Si no se cumple nunca la condición de terminación el programa entra en un bucle infinito.

**Actualización de valores.** Toda estructura repetitiva debe incluir un proceso de actualización de valores que permita que se deje de cumplir la condición de terminación. Se aconseja realizar el proceso de actualización de valores justo antes de que se vaya a comprobar la condición.

**Código** a ejecutar si se cumple la condición. Es el núcleo de la estructura repetitiva.

En función del número de repeticiones que permiten nos encontramos con estructuras repetitivas de 0 a n veces, de 1 a n veces, y de n veces.

### ESTRUCTURAS REPETITIVAS DE 0 a N VECES. PSEUDOCODIGO

En las estructuras repetitivas de 0 a n veces la condición se comprueba antes de comenzar a ejecutar el código. De tal modo que si la primera vez no se cumple la condición el código no se ejecuta.

Su representación mediante pseudocódigo es

|                                                                                        |
|----------------------------------------------------------------------------------------|
| <b>MIENTRAS</b> (condición)<br>Código si se cumple la condición<br><b>FIN-MIENTRAS</b> |
|----------------------------------------------------------------------------------------|

### ESTRUCTURAS REPETITIVAS DE 1 a N VECES. PSEUDOCODIGO

En las estructuras repetitivas de 1 a n veces la condición se comprueba después de ejecutar el código. De tal modo que el código siempre se ejecuta por lo menos una vez. Para distinguir las repetitivas de 0 a N de las de 1 a N se pone un ‘;’ después de la condición en las de 1 a n.

Su representación mediante pseudocódigo es

|                                                                                  |
|----------------------------------------------------------------------------------|
| <b>HACER</b><br>Código si se cumple la condición<br><b>MIENTRAS</b> (condición); |
|----------------------------------------------------------------------------------|

### ESTRUCTURAS REPETITIVAS DE N VECES. PSEUDOCODIGO

En las estructuras repetitivas de n veces el código se ejecuta un número determinado de veces. Ese número de veces es conocido a la hora de ejecutar el código. Las estructuras repetitivas de n veces son una versión compacta de las estructuras repetitivas de 0 a n veces.

Su representación mediante pseudocódigo es

|                                                                                                             |
|-------------------------------------------------------------------------------------------------------------|
| <b>PARA</b> (inicialización;condición;actualización)<br>Código si se cumple la condición<br><b>FIN-PARA</b> |
|-------------------------------------------------------------------------------------------------------------|

### EJERCICIOS DE ESTRUCTURAS REPETITIVAS. PSEUDOCODIGO

47. Realiza el pseudocódigo para el programa NUM100 que muestra por pantalla los 100 primeros números enteros.
48. Realiza el pseudocódigo para el programa SACANUMW que pide un número entero por teclado y muestra los números desde 1 hasta él inclusive. Con MIENTRAS.
49. Realiza el pseudocódigo para el programa SACAPARW que pide un número entero por teclado y muestra los números pares desde 0 hasta él inclusive. Con MIENTRAS.

50. Realiza el pseudocódigo para el programa SACANUDW que pide un número entero por teclado y muestra los números desde él inclusive hasta 1. Con MIENTRAS.
51. Realiza el pseudocódigo para el programa SACAPADW que pide un número entero por teclado y muestra los números pares desde él inclusive hasta 0. Con MIENTRAS.
52. Realiza el pseudocódigo para el programa SUMANUMW que pide un numero entero por teclado y muestra la suma de todos los números desde 1 hasta el inclusive. Con MIENTRAS.
53. Realiza el pseudocódigo para el programa SUMANUMF que pide un numero entero por teclado y muestra la suma de todos los números desde 1 hasta el inclusive. Con PARA.
54. Realiza el pseudocódigo para el programa SUMANUMD que modifica SUMANUMW para que si el número entero que se pide por teclado es negativo siga pidiendo números hasta que se introduzca un número positivo (con HACER MIENTRAS).
55. Realiza el pseudocódigo para el programa SUMAPARW pide un numero entero por teclado y muestra la suma de todos los números pares desde 0 hasta n. Con MIENTRAS.
56. Realiza el pseudocódigo para el programa SUMAPARF pide un numero entero por teclado y muestra la suma de todos los números pares desde 0 hasta n. Con PARA.
57. Realiza el pseudocódigo para el programa SUMAPADW pide un numero entero por teclado y muestra la suma de todos los números pares desde n hasta 0 (en orden descendente). Con MIENTRAS.
58. Realiza el pseudocódigo para el programa SUMAPADF pide un numero entero por teclado y muestra la suma de todos los números pares desde n hasta 0 (en orden descendente). Con PARA.
59. Realiza el pseudocódigo para el programa FACTORIA que pide un número por teclado y calcula su factorial.
60. Realiza el pseudocódigo para el programa FIBONACI que pide un número n por teclado y calcula la serie de fibonacci para los n primeros números. La serie de Fibonacci tiene las siguientes características:
- ```
F(0)=0;  
F(1)=1;  
Para n>=2  
F(n)=F(n-1) + F(n-2);
```
61. Realiza el pseudocódigo para el programa MAXIMINI que calcula el valor máximo y mínimo de unos números introducidos por teclado. La introducción de números cesará cuando se introduzca un número negativo que no será tenido en cuenta.
62. Realiza el pseudocódigo para el programa TABLAMUL que pide un numero por teclado y muestra su tabla de multiplicar por pantalla con el formato siguiente:
- ```
Numero: n n x 1 = r
 n x 2 = r
```
63. Realiza el pseudocódigo para el programa TABLAMUW que modifica TABLAMUL para que el proceso se repita mientras que el número del que se quiere calcular su tabla sea distinto de 0.
64. Realiza el pseudocódigo para el programa NUMPRIMO que lee un número por teclado y muestra por pantalla si el número es o no primo.
65. Realiza el pseudocódigo para el programa NUMPRIMOIMPAR que modifica NUMPRIMO para que sólo se compruebe si el número es divisible por números impares.

66. Realiza el pseudocódigo para el programa MULTN100 que pide un número entero por teclado y muestra por pantalla sus 100 primeros múltiplos.
67. Realiza el pseudocódigo para el programa PRIMO100 que muestra por pantalla los 100 primeros números primos.
68. Realiza el pseudocódigo para el programa BUSCANUM que guarda un número entero entre 1 y 99 en memoria y pide números hasta que se acierta el número por parte del usuario. El programa muestra si el número introducido es mayor o menor que el almacenado en memoria así como el número de intentos que lleva el usuario.
69. Realiza el pseudocódigo para el programa CIFRAS que pide un número entero por teclado y muestra por pantalla sus cifras una por una.
70. Realiza el pseudocódigo para el programa CADENACIFRAS que pide un número entero por teclado, calcula sus cifras una a una y muestra por pantalla una cadena de caracteres con las cifras calculadas.
71. Realiza el pseudocódigo para el programa MEGABYTE que pide un tamaño expresado en bytes por teclado, y muestra por pantalla el número de Mbytes, Kbytes, y Bytes equivalente.
72. Realiza el pseudocódigo para el programa CIFRANUM que pide cifras (números enteros de 0-9) hasta que se introduzca un número negativo por teclado y muestra por pantalla el número resultante.
73. Realiza el pseudocódigo para el programa NMULTI que pide un número entero por teclado, después pide el número de múltiplos que se desean calcular de ese número, y muestra por pantalla esos N primeros múltiplos.
74. Realiza el pseudocódigo para el programa FACTONEG que pide números por teclado hasta que se introduzca el 0. Por cada número, si el número es positivo calcula su factorial y, si es negativo, primero lo cambia de signo y después calcula su factorial.
75. Realiza el pseudocódigo para el programa PRIMOCUB que pide números enteros por teclado hasta que se introduce un número negativo. Con cada número, si el número es impar comprueba si es primo o no y muestra un mensaje por pantalla de "Es Primo" o "No es Primo"; y si es par, calcula su cubo y muestra el resultado por pantalla.
76. Realiza el pseudocódigo para el programa RANGONUM que pide dos números enteros por teclado y muestra los números enteros que hay desde el primero hasta el segundo ambos inclusive.
77. Realiza el pseudocódigo para el programa RANGOSUM que pide dos números enteros por teclado y calcula la suma de los números enteros que hay desde el primero hasta el segundo ambos inclusive.
78. Realiza el pseudocódigo para el programa N10PARES que pide un número entero por teclado y muestra por pantalla los 10 siguientes números pares.
79. Realiza el pseudocódigo para el programa SUELNETO que lee sueldos brutos anuales de trabajadores hasta que se lee un sueldo con valor negativo y calcula el sueldo neto de esos sueldos brutos teniendo en cuenta que, por cada 150.00 que se supere 1000.00, se debe descontar un 1%, con un máximo del 56%. Por ejemplo, si el sueldo bruto es de 1330.00 el sueldo neto será un 3% menor.
80. Realiza el pseudocódigo para el programa RAIZNEW2 que calcula la raíz cuadrada de un número por el método de Newton. Este método consiste en calcular términos hasta que el valor absoluto de la

diferencia entre el término actual y el anterior sea menor que una cantidad de Error determinada (en nuestro caso  $\text{ERROR}=0.00000001$ ). Inicialmente  $R_{\text{anterior}}$  es la mitad del número. Para el cálculo de la raíz se utiliza la formula

$$R_{\text{actual}} = ((\text{Numero}/R_{\text{anterior}}) + (R_{\text{anterior}})) / 2$$

81. Realiza el pseudocódigo para el programa RAIZNEWT que modifica RAIZNEW2 para que el error en lugar de ser fijo se pida por pantalla, y para que  $R_{\text{anterior}}$  inicialmente sea 1.

82. Realiza el pseudocódigo para el programa RAIZNEWN que pide números por teclado hasta que se introduce cero o un número negativo. Por cada número, calcula su raíz cuadrada mediante el método de Newton. Este método consiste en calcular términos hasta que el valor absoluto de la diferencia entre el término actual y el anterior sea menor que una cantidad de Error determinada (en nuestro caso  $\text{ERROR}=0.00000001$ ). Inicialmente  $R_{\text{anterior}}$  es la mitad del número. Para el cálculo de la raíz se utiliza la formula

$$R_{\text{actual}} = ((\text{Numero}/R_{\text{anterior}}) + (R_{\text{anterior}})) / 2$$

83. Realiza el pseudocódigo para el programa HIPOTENU que lee el valor de los dos catetos de un triángulo rectángulo y, aplicando el teorema de Pitágoras, calcula el valor de la hipotenusa. Para calcular la raíz cuadrada se utiliza el método de Newton.

84. Realiza el pseudocódigo para el programa EC2GRAIZ que calcula las soluciones de una ecuación de 2º grado. Si las raíces son complejas las muestra de la forma  $\text{real}+i\text{imag}$   $\text{real}-i\text{imag}$ . Para calcular la raíz cuadrada se utiliza el método de Newton.

85. Realiza el pseudocódigo para el programa PITAGORA que calcula los valores de X, Y y Z que cumplen el Teorema de Pitágoras hasta un valor de Z de 50, teniendo en cuenta que el valor de Y es mayor que el de X, y que Los valores cumplen la expresión

$$Z^2 = X^2 + Y^2$$

86. Realiza el pseudocódigo para el programa TRIAFIWA que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
N=4
1
2 2
3 3 3
4 4 4 4
```

87. Realiza el pseudocódigo para el programa TRIAFIFA que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

```
N=4
1
2 2
3 3 3
4 4 4 4
```

88. Realiza el pseudocódigo para el programa TRIAFIWD que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

```
N=4
4 4 4 4
3 3 3
2 2
1
```

89. Realiza el pseudocódigo para el programa TRIAFIFD que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

```
N=4
4 4 4 4
3 3 3
2 2
1
```

90. Realiza el pseudocódigo para el programa TRIANUWA que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
N=4
1
1 2
1 2 3
1 2 3 4
```

91. Realiza el pseudocódigo para el programa TRIANUFA que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
N=4
1
1 2
1 2 3
1 2 3 4
```

92. Realiza el pseudocódigo para el programa TRIANUWD que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
N=4
1 2 3 4
1 2 3
1 2
1
```

93. Realiza el pseudocódigo para el programa TRIANUFD que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
N=4
1 2 3 4
1 2 3
1 2
1
```

94. Realiza el pseudocódigo para el programa TRIACOWA que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

N=5

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

N=4

```
1 2 3 4
2 3 4
3 4
4
```

95. Realiza el pseudocódigo para el programa TRIACOFA que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

N=5

```
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5
```

N=4

```
1 2 3 4
2 3 4
3 4
4
```

96. Realiza el pseudocódigo para el programa TRIACOWD que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

N=5

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

N=4

```
4 3 2 1
3 2 1
2 1
1
```

97. Realiza el pseudocódigo para el programa TRIACOFD que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

N=5

```
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
```

N=4

```
4 3 2 1
3 2 1
2 1
1
```

98. Realiza el pseudocódigo para el programa TRIANFLO que muestra por pantalla el triángulo de FLOYD para un número de filas dado.

Para numFilas=5

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

99. Realiza el pseudocódigo para el programa CUADRANU que lee un número por teclado y escribe en pantalla el siguiente cuadrado numérico

N=5

```
1 2 3 4 5
2 3 4 5 0
3 4 5 0 0
4 5 0 0 0
5 0 0 0 0
```

N=4

```
1 2 3 4
2 3 4 0
3 4 0 0
4 0 0 0
```

100. Realiza el pseudocódigo para el programa CUADRANI que lee un número por teclado y escribe en pantalla el siguiente cuadrado numérico

|           |         |
|-----------|---------|
| N=5       | N=4     |
| 0 0 0 0 5 | 0 0 0 4 |
| 0 0 0 4 5 | 0 0 3 4 |
| 0 0 3 4 5 | 0 2 3 4 |
| 0 2 3 4 5 | 1 2 3 4 |
| 1 2 3 4 5 |         |

101. Realiza el pseudocódigo para el programa RECTANUM que lee un número por teclado y escribe en pantalla el rectángulo siguiente:

|                          |                          |
|--------------------------|--------------------------|
| <u>Para el número 7:</u> | <u>Para el número 4:</u> |
| 1 2 3 4 5 6 7            | 1 2 3 4                  |
| 0 2 3 4 5 6 0            | 0 2 3 0                  |
| 0 0 3 4 5 0 0            | 0 0 0 0                  |
| 0 0 0 4 0 0 0            | 0 2 3 0                  |
| 0 0 0 0 0 0 0            | 1 2 3 4                  |
| 0 0 0 4 0 0 0            |                          |
| 0 0 3 4 5 0 0            |                          |
| 0 2 3 4 5 6 0            |                          |
| 1 2 3 4 5 6 7            |                          |

## ESTRUCTURAS REPETITIVAS DE 0 a N VECES. JAVA

En las estructuras repetitivas de 0 a n veces la condición se comprueba antes de comenzar a ejecutar el código. De tal modo que si la primera vez no se cumple la condición el código no se ejecuta.

Su representación en Java es

```
while (condición) {
 Código si se cumple la condición
}
```

## ESTRUCTURAS REPETITIVAS DE 1 a N VECES. JAVA

En las estructuras repetitivas de 1 a n veces la condición se comprueba después de ejecutar el código. De tal modo que el código siempre se ejecuta por lo menos una vez. Para distinguir las repetitivas de 0 a N de las de 1 a N se pone un ';' después de la condición en las de 1 a n.

Su representación en Java es

```
do {
 Código si se cumple la condición
}while (condición);
```

## ESTRUCTURAS REPETITIVAS DE N VECES. JAVA

En las estructuras repetitivas de n veces el código se ejecuta un número determinado de veces. Ese número de veces es conocido a la hora de ejecutar el código.

Su representación mediante pseudocódigo es

```
for(inicialización;condición;actualización){
 Código si se cumple la condición
}
```



## EJERCICIOS DE ESTRUCTURAS REPETITIVAS. JAVA

102. Realiza la clase Java Num100 que muestra por pantalla los 100 primeros números enteros.
103. Realiza la clase Java Sacanumw que pide un número entero por teclado y muestra los números desde 1 hasta él inclusive. Con While.
104. Realiza la clase Java Sacaparw que pide un número entero por teclado y muestra los números pares desde 0 hasta él inclusive. Con While.
105. Realiza la clase Java Sacanudw que pide un número entero por teclado y muestra los números desde él inclusive hasta 1. Con While.
106. Realiza la clase Java Sacapadw que pide un número entero por teclado y muestra los números pares desde él inclusive hasta 0. Con While.
107. Realiza la clase Java Sumanumw que pide un numero entero por teclado y muestra la suma de todos los números hasta el inclusive. Con While.
108. Realiza la clase Java Sumanumf que pide un numero entero por teclado y muestra la suma de todos los números hasta el inclusive. Con For.
109. Realiza la clase Java Sumanumd que modifica Sumanumw para que si el número entero que se pide por teclado es negativo siga pidiendo números hasta que se introduzca un número positivo (con Do While).
110. Realiza la clase Java Sumaparw pide un numero entero por teclado y muestra la suma de todos los números pares desde 0 hasta n. Con While.
111. Realiza la clase Java Sumaparf pide un numero entero por teclado y muestra la suma de todos los números pares desde 0 hasta n. Con For.
112. Realiza la clase Java Sumapadw pide un numero entero por teclado y muestra la suma de todos los números pares desde n hasta 0 (en orden descendente). Con While.
113. Realiza la clase Java Sumapadf pide un numero entero por teclado y muestra la suma de todos los números pares desde n hasta 0 (en orden descendente). Con For.
114. Realiza la clase Java Factoria que pide un número por teclado y calcula su factorial.
115. Realiza la clase Java Fibonaci que pide un número n por teclado y calcula la serie de fibonacci para los n primeros números. La serie de Fibonacci tiene las siguientes características:  
 $F(0)=0;$   
 $F(1)=1;$   
Para  $n \geq 2$   
 $F(n)=F(n-1) + F(n-2);$
116. Realiza la clase Java Maximini que calcula el valor máximo y mínimo de unos números introducidos por teclado. La introducción de números cesará cuando se introduzca un numero negativo que no será tenido en cuenta.
117. Realiza la clase Java Tablamul que pide un numero por teclado y muestra su tabla de multiplicar por pantalla con el formato siguiente:  
Numero:    i                    i x 1 = j  
                                     i x 2 = j

118. Realiza la clase Java Tablamuw que modifica TABLAMUL para que el proceso se repita mientras que el número del que se quiere calcular su tabla sea distinto de 0.
119. Realiza la clase Java Numprimo que lee un número por teclado y muestra por pantalla si el número es o no primo.
120. Realiza la clase Java Numprimoimpar que modifica Numprimo para que sólo se compruebe si el número es divisible por números impares.
121. Realiza la clase Java Multn100 que pide un número entero por teclado y muestra por pantalla sus 100 primeros múltiplos.
122. Realiza la clase Java Primo100 que muestra por pantalla los 100 primeros números primos.
123. Realiza la clase Java Buscanum que guarda un número en memoria y pide números hasta que se acierta el número por parte del usuario. El programa muestra si el número introducido es mayor o menor que el almacenado en memoria así como el número de intentos que lleva el usuario.
124. Realiza la clase Java Cifras que pide un número entero por teclado y muestra por pantalla sus cifras una por una.
125. Realiza la clase Java Megabyte que pide un tamaño expresado en bytes por teclado, y muestra por pantalla el número de Mbytes, Kbytes, y Bytes equivalente.
126. Realiza la clase Java Cifranum que pide cifras (números enteros de 0-9) hasta que se introduzca un número negativo por teclado y muestra por pantalla el número resultante.
127. Realiza la clase Java Nmulti que pide un número entero por teclado, después pide el número de múltiplos que se desean calcular de ese número, y muestra por pantalla esos N primeros múltiplos.
128. Realiza la clase Java Factoneg que pide números por teclado hasta que se introduzca el 0. Por cada número, si el número es positivo calcula su factorial y, si es negativo, primero lo cambia de signo y después calcula su factorial.
129. Realiza la clase Java Primocub que pide números enteros por teclado hasta que se introduce un número negativo. Con cada número, si el número es impar comprueba si es primo o no y muestra un mensaje por pantalla de "Es Primo" o de "No es Primo"; y si es impar, calcula su cubo y muestra el resultado por pantalla.
130. Realiza la clase Java Rangonum que pide dos números enteros por teclado y muestra los números enteros que hay desde el primero hasta el segundo ambos inclusive.
131. Realiza la clase Java Rangosum que pide dos números enteros por teclado y calcula la suma de los números enteros que hay desde el primero hasta el segundo ambos inclusive.
132. Realiza la clase Java N10pares que pide un número entero por teclado y muestra por pantalla los 10 siguientes números pares.
133. Realiza la clase Java Suelneto que lee sueldos brutos anuales de trabajadores hasta que se lee un sueldo con valor negativo y calcula el sueldo neto de esos sueldos brutos teniendo en cuenta que, por cada 150.00 que se supere 1000.00, se debe descontar un 1%, con un máximo del 56%. Por ejemplo, si el sueldo bruto es de 1330.00 el sueldo neto será un 3% menor.

134. Realiza la clase Java Raiznew2 que calcula la raíz cuadrada de un número por el método de Newton. Este método consiste en calcular términos hasta que el valor absoluto de la diferencia entre el término actual y el anterior sea menor que una cantidad de Error determinada (en nuestro caso  $\text{ERROR}=0.00000001$ ). Inicialmente Ranterior es la mitad del número. Para el calculo de la raíz se utiliza la formula

$$\text{Ractual} = ((\text{Numero}/\text{Ranterior}) + (\text{Ranterior})) / 2$$

135. Realiza la clase Java Raiznewt que modifica Raiznew2 para que el error en lugar de ser fijo se pida por pantalla, y para que Ranterior inicialmente sea 1.

136. Realiza la clase Java Raiznewn que pide números por teclado hasta que se introduce cero o un número negativo. Por cada número, calcula su raíz cuadrada mediante el método de Newton. Este método consiste en calcular términos hasta que el valor absoluto de la diferencia entre el término actual y el anterior sea menor que una cantidad de Error determinada (en nuestro caso  $\text{ERROR}=0.00000001$ ). Inicialmente Ranterior es la mitad del número. Para el calculo de la raíz se utiliza la formula

$$\text{Ractual} = ((\text{Numero}/\text{Ranterior}) + (\text{Ranterior})) / 2$$

137. Realiza la clase Java Hipotenu que lee el valor de los dos catetos de un triángulo rectángulo y, aplicando el teorema de Pitágoras, calcula el valor de la hipotenusa. Para calcular la raíz cuadrada se utiliza el método de Newton.

138. Realiza la clase Java Ec2graiz que calcula las soluciones de una ecuación de 2º grado. Si las raíces son complejas las muestra de la forma  $\text{real} + i\text{imag}$   $\text{real} - i\text{imag}$ . Para calcular la raíz cuadrada se utiliza el método de Newton.

139. Realiza la clase Java Pitagora que calcula los valores de X, Y y Z que cumplen el Teorema de Pitágoras hasta un valor de Z de 50, teniendo en cuenta que el valor de Y es mayor que el de X, y que Los valores cumplen la expresión

$$Z^2 = X^2 + Y^2$$

140. Realiza la clase Java Triafiwa que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

|           |         |
|-----------|---------|
| N=5       | N=4     |
| 1         | 1       |
| 2 2       | 2 2     |
| 3 3 3     | 3 3 3   |
| 4 4 4 4   | 4 4 4 4 |
| 5 5 5 5 5 |         |

141. Realiza la clase Java Triaififa que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

|           |         |
|-----------|---------|
| N=5       | N=4     |
| 1         | 1       |
| 2 2       | 2 2     |
| 3 3 3     | 3 3 3   |
| 4 4 4 4   | 4 4 4 4 |
| 5 5 5 5 5 |         |

142. Realiza la clase Java Triafiwd que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

```
N=4
4 4 4 4
3 3 3
2 2
1
```

143. Realiza la clase Java Triafifd que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
5 5 5 5 5
4 4 4 4
3 3 3
2 2
1
```

```
N=4
4 4 4 4
3 3 3
2 2
1
```

144. Realiza la clase Java Trianuwa que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
N=4
1
1 2
1 2 3
1 2 3 4
```

145. Realiza la clase Java Trianufa que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

```
N=4
1
1 2
1 2 3
1 2 3 4
```

146. Realiza la clase Java Trianuwd que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```
N=5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
N=4
1 2 3 4
1 2 3
1 2
1
```

147. Realiza la clase Java Trianufd que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```
N=5
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

```
N=4
1 2 3 4
1 2 3
1 2
1
```

148. Realiza la clase Java Triacowa que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```

N=5
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5

```

```

N=4
1 2 3 4
2 3 4
3 4
4

```

149. Realiza la clase Java Triacofa que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```

N=5
1 2 3 4 5
2 3 4 5
3 4 5
4 5
5

```

```

N=4
1 2 3 4
2 3 4
3 4
4

```

150. Realiza la clase Java Triacowd que pide un número n por teclado y muestra la siguiente salida en pantalla. Con While.

```

N=5
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1

```

```

N=4
4 3 2 1
3 2 1
2 1
1

```

151. Realiza la clase Java Triacofd que pide un número n por teclado y muestra la siguiente salida en pantalla. Con For.

```

N=5
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1

```

```

N=4
4 3 2 1
3 2 1
2 1
1

```

152. Realiza la clase Java Trianflo que muestra por pantalla el triángulo de FLOYD para un numero de filas dado.

```

Para numFilas=5
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15

```

153. Realiza la clase Java Cuadranu que lee un número por teclado y escribe en pantalla el siguiente cuadrado numérico

```

N=5
1 2 3 4 5
2 3 4 5 0
3 4 5 0 0
4 5 0 0 0
5 0 0 0 0

```

```

N=4
1 2 3 4
2 3 4 0
3 4 0 0
4 0 0 0

```

154. Realiza la clase Java Cuadrani que lee un número por teclado y escribe en pantalla el siguiente cuadrado numérico

N=5

```
0 0 0 0 5
0 0 0 4 5
0 0 3 4 5
0 2 3 4 5
1 2 3 4 5
```

N=4

```
0 0 0 4
0 0 3 4
0 2 3 4
1 2 3 4
```

155. Realiza la clase Java Rectanum que lee un número por teclado y escribe en pantalla el rectángulo siguiente:

Para el número 7:

```
1 2 3 4 5 6 7
0 2 3 4 5 6 0
0 0 3 4 5 0 0
0 0 0 4 0 0 0
0 0 0 0 0 0 0
0 0 0 4 0 0 0
0 0 3 4 5 0 0
0 2 3 4 5 6 0
1 2 3 4 5 6 7
```

Para el número 4:

```
1 2 3 4
0 2 3 0
0 0 0 0
0 2 3 0
1 2 3 4
```