



¡Bienvenidos a la tercera clase de Algoritmos y Estructuras de Datos 1!

En esta oportunidad continuaremos con las Estructuras Condicionales pero, a diferencia de la clase anterior, trabajaremos con situaciones problemáticas que dependen de condiciones más complicadas y que, por lo tanto, implican para su resolución caminos alternativos más complejos. Específicamente, estudiaremos hoy las Estructuras Condicionales Anidadas y la instrucción Switch-Case.

Para organizar está clase te proponemos tres grandes temas:

| El primero de ellos aborda concretamente la **Estructura Condicional Anidada**.

| El segundo nos introducirá a la **Instrucción Switch-Case**. | Y finalmente aprenderemos cómo **insertar comentarios** en los programas que realicemos. A lo largo de la clase te encontrarás con ejercicios para que realices vos y también con otros ejemplos que nosotros iremos comentando y explicando para que practiquemos juntos.

¡Nos vemos!

Lavarropas y Wi-Fi, ¿para qué?

Internet de las Cosas. La empresa cordobesa Alladio presentó su lavarropas Drean Next, que incorpora conectividad Wi-Fi. Describimos su modo de funcionamiento, su integración con una aplicación móvil y las posibilidades que esta tecnología abre a futuro para la marca.

Los electrodomésticos conectados dejaron de ser una promesa del futuro: no sólo han llegado a nuestro país, sino que ya se producen localmente, como lo confirma el lanzamiento del lavarropas Drean Next, un producto íntegramente desarrollado en nuestra provincia por los ingenieros de Alladio con la colaboración de empresas tecnológicas locales.

Wi-Fi, ¿para qué?

Un sector fundamental de la Internet de las Cosas (IoT, por sus siglas en inglés) es el de los electrodomésticos conectados, que abren un nuevo abanico de posibilidades en cuanto a control remoto e integración en el hogar inteligente.

El lavarropas Drean Next se constituye en el primer electrodoméstico desarrollado en el país que integra conectividad. Se complementa con una aplicación móvil que permite su operación desde un teléfono, tablet o cualquier dispositivo conectado a Internet. Además, comunica al usuario el estado del proceso de lavado y centrifugado, brinda avisos para el mantenimiento preventivo, permite acceder a tutoriales, actualizar el software y personalizar programas, entre otras funciones.

A la hora de pensar en utilidades concretas, desde la empresa plantean diversos escenarios frecuentes en la vida actual. Por caso, los usuarios de estos equipos podrán dejar la carga lista e iniciar el proceso de lavado remotamente, para que al volver a casa la ropa esté recién lavada y lista para secar. O, frente a una demora imprevista, activar desde la app móvil un nuevo proceso de enjuague, para que las prendas no tomen mal olor. También resulta muy útil recibir en el smartphone las notificaciones de finalización del lavado.

Baggage hall A ROUGH BAGGAGE HALL TO BAGGAGE H

La inteligencia de este lavarropas no sólo radica en sus posibilidades de conectividad: además, reconoce el programa más utilizado y permite darle inicio con sólo presionar un botón. Del mismo modo, es posible modificar los programas de lavado predefinidos de fábrica para crear nuevas versiones personalizadas de ellos.

Recopilación de datos

Uno de los temas sobre los que más se discute a la hora de hablar de la Internet de las Cosas y de electrodomésticos conectados es el de la privacidad de los datos recopilados por los fabricantes. En esta línea, desde Alladio destacan que recolectarán información de uso de sus productos con el único propósito de contribuir a su proceso de mejora continua. Así, Agustín Roberi, director y gerente de Marketing de la firma, sostiene que los datos recopilados "nos permitirán contar con información fidedigna sobre patrones de uso que volcaremos en nuevos desarrollos, aprendiendo mucho más sobre cómo usa la gente nuestros lavarropas: por ejemplo, sus horarios, con qué carga, en qué programas".

El ejecutivo arriesga que "en el futuro, según la zona del país en la cual se esté utilizando, podremos indicarle al lavarropas que emplee más o menos cantidad de jabón al lavar".

Tecnología cordobesa

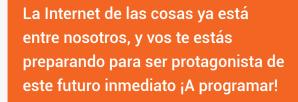
Tanto la aplicación móvil para el control remoto como la electrónica propia del lavarropas fueron desarrolladas por tecnológicas locales. La app pertenece a Geminus-Qhom, mientras que el módulo de Wi-Fi fue desarrollado por Linetec, otro emprendimiento local que hace tiempo viene fabricando las plaquetas utilizadas en los lavarropas cordobeses.

"Es todo un desafío, porque estamos entre las primeras empresas a nivel internacional en introducir esto para lo que, además, las partes vitales del producto también son de empresas cordobesas. O sea, somos pioneros con un desarrollo bien autóctono, que se ubica a la vanguardia del lavado de ropa en el mundo", indica Marcos Alladio. Esto abre perspectivas de generar un ecosistema de electrodomésticos conectados e interactuando en el hogar, tomando en cuenta que la firma produce bajo otras denominaciones comerciales tanto heladeras como hornos de microondas, extractores, anafes y hornos eléctricos.

La Voz del Interior - 11/7/2015

Baggage hall Arrivals hall Transfer T4-9

La programación está invadiendo hasta las áreas más impensadas: ¿lavarropas controlados por WiFi? Pero lo más importante, es que no se trata de una exposición futurista que se desarrolla en los países centrales, sino de productos que se fabrican en nuestra provincia y con desarrollos de hardware y software construidos por profesionales y empresas cordobesas.









Durante la última clase, aprendimos a diseñar programas que pudieran tomar decisiones mediante las estructuras condicionales. Pero hay veces que esas decisiones son más complejas o deben contemplar más opciones posibles.

Por eso en esta tercera clase nos vamos a dedicar a profundizar estos conceptos mediante las **estructuras condicionales anidadas** y otras instrucciones de salida múltiple.

ISSD

Supogamos que un amigo nos llama porque su impresora no imprime. En primer lugar, le preguntamos si está prendido el led que indica que está encendida. Si no lo estuviera, ya encontramos el problema: debe prender la impresora. Pero si el led está encendido, puede ocurrir que no tenga tinta. Si tuviese tinta, la cuestión es más compleja: deberá llamar al servicio ténico.

Si analizamos el problema anterior, notaremos que hay dos condiciones, pero no está una a **continuación** de otra, sino una **dentro** de otra: sólo verificaremos si tiene tinta en el caso de que esté encendida. En términos de diagramas de flujo, la estructura será como la de la imagen 1.



Veamos un ejemplo de su aplicación.

Ejemplo 11 ¿Positivo, negativo o nulo?

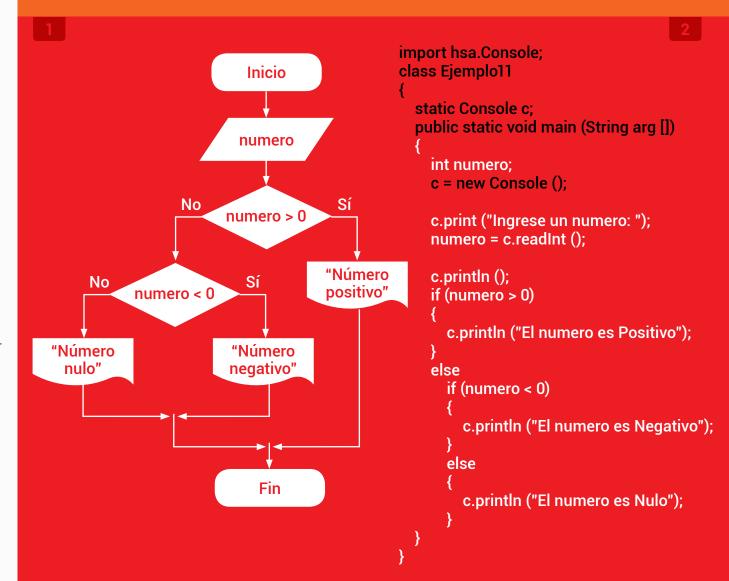
Ingresar un número y determinar si es positivo, negativo o nulo.

Pensemos un poco: ¿Cómo hacemos habitualmente para determinar si un número cualquiera es positivo, negativo o nulo? No vale mirar ya la respuesta... primero analizá el problema intuitivamente. Por ejemplo, -5, -140 y -18 son negativos. Por otra parte, 100, 3 y 92 son positivos. El único nulo es el cero ¿ya pensaste?

Para saber si un número es positivo o negativo, simplemente se lo debe comparar con el cero: si el número en cuestión es mayor a cero, será positivo y si es menor a cero será negativo. Si no cumple con ninguna de las condiciones anteriores, será nulo.

Vemos cómo quedaría el diagrama de flujo y la codificación correspondiente en la imagen 1 y 2 respectivamente.

Se dice que una estructura condicional es anidada cuando por la rama del verdadero o el falso de una estructura condicional hay otra estructura condicional.



1 1

Como podrás apreciar en la codificación, hay una estructura if dentro del else de la primera estructura if, por eso se llama estructura condicional "anidada".

Ejemplo 12 Condición de un alumno a partir de su promedio

Realizá un programa que lea por teclado tres notas de un alumno, y calcule su promedio, imprimiendo algunos de los siguientes mensajes:

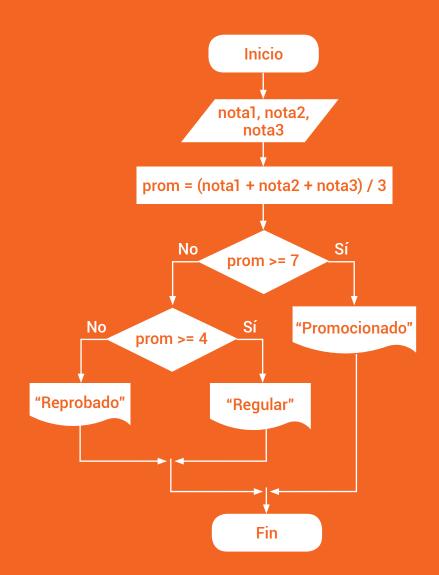
Si el promedio es mayor o igual a 7, mostrar "Alumno Promocionado".

Si el promedio es mayor o igual a 4 y además, menor a 7, mostrar "Alumno Regular".

Si el promedio es menor a 4, mostrar "Alumno Reprobado" (imagen 1).

En el ejemplo de la imagen 1, se ingresan tres valores por teclado que representan las notas de un alumno y se obtiene el promedio sumando los tres valores y dividiendo por 3 dicho resultado.

Primeramente se pregunta si el promedio es superior o igual a 7, en caso afirmativo



por la rama del Verdadero de la estructura condicional se muestra un mensaje que indica "Promocionado" (con comillas indicamos un texto que debe imprimirse en pantalla).

En el caso que la condición sea Falsa, por la rama del Falso aparece otra estructura condicional, porque todavía se tienen dos posibilidades: que el promedio del alumno sea superior o igual a cuatro, o también se puede dar el caso de que sea inferior a cuatro.

En la segunda condición, se averigua si el promedio es mayor o igual a 4; en el caso de ser Verdadero, se imprime "Regular" y en el caso contrario, se imprime "Reprobado", ya que por descarte, el promedio es menor a 4. La codificación se observa en la imagen 2.

La palabra *float* encerrada entre paréntesis, indica que se debe transformar el valor que figure al lado de ella en un valor de tipo *float*, debido a lo siguiente: cada una de las notas son de tipo entero (int), y la suma de valores enteros en Java es otro entero; además, la división de dos enteros, es también otro entero. Por lo tanto, si se ingresaran como

```
import hsa.Console;
class Ejemplo12
  static Console c:
  public static void main (String arg [])
    int nota1, nota2, nota3;
    float prom;
    c = new Console ();
    c.print ("Ingrese Primer nota: ");
    nota1 = c.readInt();
    c.print ("Ingrese Segunda nota: ");
    nota2 = c.readInt();
    c.print ("Ingrese Tercer nota: ");
    nota3 = c.readInt();
    prom = (float)(nota1 + nota2 +
nota3) / 3;
    c.println ();
    c.println ("El promedio es: " + prom);
    if (prom >= 7)
```

```
c.println ("Alumno
Promocionado");
}
else
if (prom >= 4)
{
    c.println ("Alumno Regular");
}
else
{
    c.println ("Alumno Reprobado");
}
}
```

La única instrucción nueva en este programa es la palabra (float) en la línea:

```
prom = (float)(nota1 + nota2 + nota3)/3;
```

ejemplo las notas: 7, 7 y 8 el promedio daría como resultado el valor de 7 (sin decimales) y no de 7.33, que sería lo correcto. Por este motivo, es que se coloca la palabra float delante de la suma de las notas, para transformar dicha suma en un valor de tipo float, y , al dividir un valor float con un entero, el resultado es un float, por lo tanto, ahora sí dará el valor del promedio correcto con decimales.

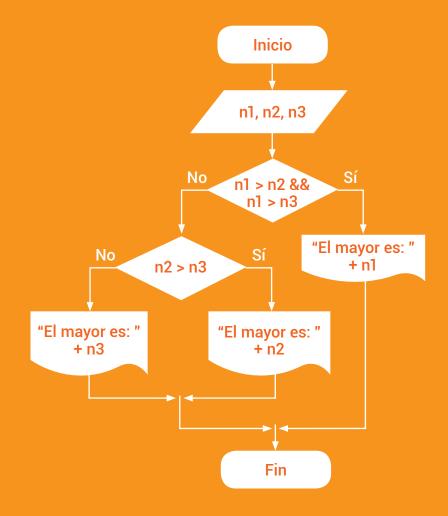
Ejemplo 13 El mayor de tres números

Ingresá tres números y determiná cuál es el mayor de ellos.

Observá el diagrama en la imagen 1 y el código en la imagen 2 de la próxima página.

Analicemos la lógica del proceso que observamos en la imagen 1.

Luego de ingresar los tres valores, comenzamos preguntando si el primer número es mayor a los otros dos, lo que se logra con una condición doble utilizando el operador lógico and (&&). La condición se lee de esta manera:



Si el valor almacenado en n1 es mayor al valor almacenado en n2, y además, el valor almacenado en la variable n1 es mayor al valor almacenado en la variable n3, entonces imprimir "El Mayor es : " n1, de lo contrario, (descartado que el valor de n1 es el mayor), se pregunta si el valor almacenado en n2 es mayor al valor almacenado en n3: en caso de ser verdadero, se muestra "el Mayor es: " n2, de lo contrario se muestra "El mayor es: " n3.

Se debe tener en cuenta que en la condición doble...

if (n1 > n2 && n1 > n3)

...se deben cumplir obligatoriamente las dos condiciones independientes para que toda la condición resulte verdadera (*True* en inglés). Esto es indicado por el signo and (&&). Si alguna de las dos no se cumple, toda la condición resultará falsa (*False* en inglés).

```
import hsa.Console;
class Ejemplo13
  static Console c:
  public static void main (String arg [])
    int n1, n2, n3;
    c = new Console ();
    c.print ("Ingrese Primer Numero: ");
    n1 = c.readInt();
    c.print ("Ingrese Segundo Numero: ");
    n2 = c.readInt();
    c.print ("Ingrese Tercer Numero: ");
    n3 = c.readInt();
    c.println ();
    if (n1 > n2 && n1 > n3)
       c.println ("El Mayor es: " + n1);
    else
```

```
if (n2 > n3)
{
            c.println ("El Mayor es: " + n2);
}
else
{
            c.println ("El Mayor es: " + n3);
}
```

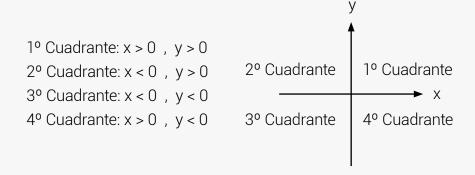
En las páginas posteriores a estos desempeños te vas a encontrar con los diagramas de flujo para que puedas auto-evaluarte. A la codificación en cambio, deberás realizarla solo porque creemos que ya estás en condiciones de hacerlo.

El ejercicio 22 es muy similar al 21 por lo que queda como tarea que desarrolles tanto el diagrama como el programa.

¡Adelante!

ISSD

Ingresá tres valores por teclado, si todos son iguales imprimí la suma del primero con el segundo y a este resultado multiplicalo por el tercero. Si no son iguales, mostrá una leyenda. Escribí un programa que pida ingresar las coordenadas de un punto en el plano, es decir, dos valores enteros x e y. Posteriormente imprimí en pantalla en qué cuadrante se ubica dicho punto.



13

ISSD

De un candidato a un empleo que ya realizó un test de capacitación, se obtuvo la siguiente información: nombre del postulante, cantidad total de preguntas que se le realizaron y la cantidad de preguntas que contestó correctamente.

Confeccioná un programa que lea los datos del postulante e informe el nivel del mismo según el porcentaje de respuestas correctas que ha obtenido, sabiendo que:

| Nivel superior: Porcentaje>=90%.

| Nivel medio: Porcentaje>=75% y <90%.

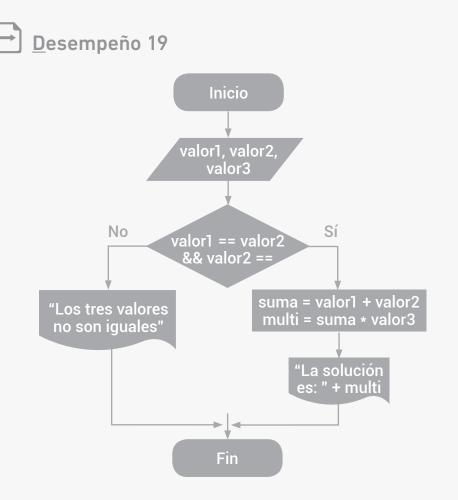
| Nivel regular: Porcentaje>=50% y <75%.

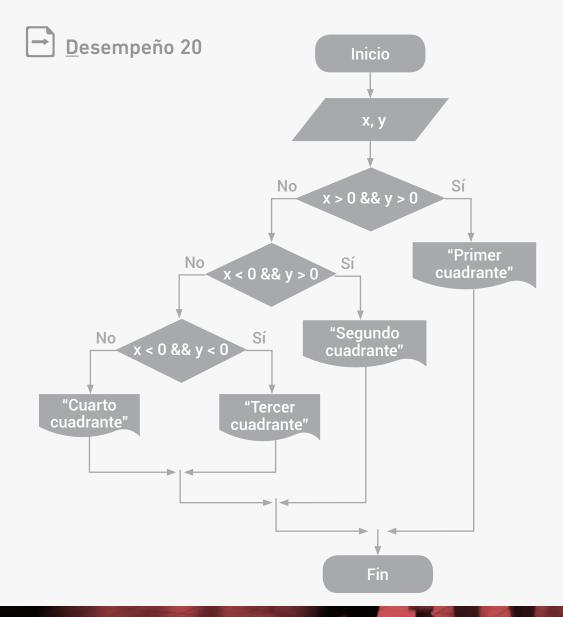
| Fuera de nivel: Porcentaje<50%.

Ingresá un valor de frecuencia e indicá a que banda corresponde, según la siguiente clasificación en telecomunicaciones:

BANDA	SUBDIVISIÓN DE	RANGO DE
	FRECUENCIA	FRECUENCIA
ULF	Frecuencia ultra baja	3 a 30 Hz
ELF	Frecuencia extremadamente baja	30 a 300 Hz
ILF	Frecuencia infra baja	300 Hz a 3 Khz
VLF	Frecuencia muy baja	3 a 30 Khz
LF	Frecuencia baja	30 a 300 Khz
MF	Frecuencia media	300 Khz a 3 Mhz
HF	Frecuencia alta	3 a 30 Mhz
VHF	Frecuencia muy alta	30 a 300 Mhz
UHF	Frecuencia ultra alta	300 Mhz a 3 Ghz
SHF	Frecuencia super alta	3 a 30 Ghz
EHF	Frecuencia extremadamente alta	30 a 300 Ghz

A continuación se muestran los diagramas de flujo que resuelven estos ejercicios, a vos te resta realizar la codificación correspondiente.

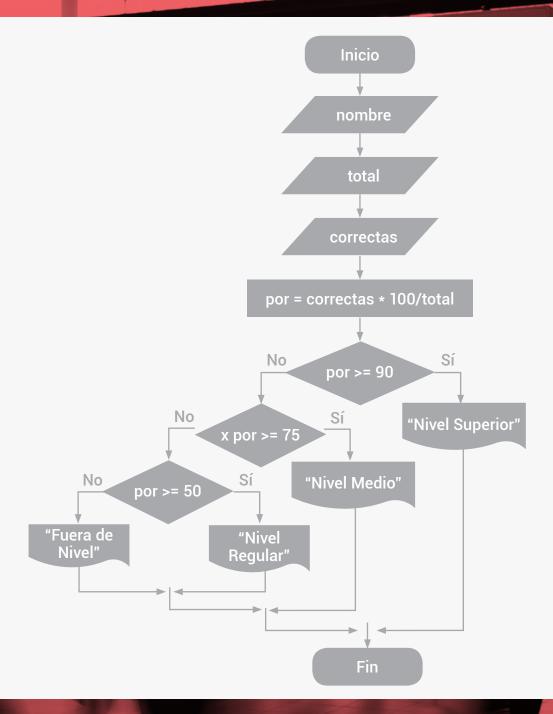




Desempeño 21

La variable nombre, debe ser definida de tipo String, y para pedir un nombre, se lo debe hacer con la instrucción readLine(), de la siguiente manera:

String nombre; c.print("Ingrese nombre"); nombre = c.readLine();



Pensando un poco, te darás cuenta de que se puede resolver con lo que sabemos hasta aquí, mediante una serie de estructuras condicionales anidadas... pero el tamaño del diagrama de flujo sería demasiado grande y difícil de seguir.

ISSD

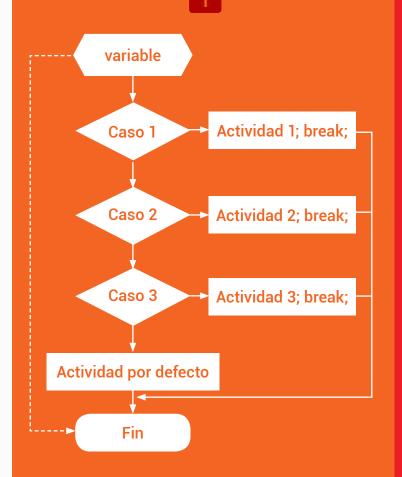
Como estas situaciones son frecuentes, existe una estructura especialmente diseñada para realizar salidas múltiples de acuerdo al valor de una determinada variable. La sentencia switch() es una instrucción de decisión múltiple, mediante la cual el compilador busca el valor contenido en una variable dentro de una lista de constantes ints (números enteros) o chars (caracteres). Cuando encuentra el valor de igualdad entre variable y constante, entonces ejecuta el grupo de instrucciones asociados a dicha constante. Si no encuentra el valor de igualdad, entonces ejecuta un grupo de instrucciones por defecto que se asocia a la instrucción default, aunque esta última es opcional.

Existen numerosas simbologias sobre la instracción switch para realizar un diagrama de flujo, pero la más utilizada es la que podemos observar en la imagen 1.

Ejemplo 14 Nombre del mes a partir de su número

Ingresá un número, y mostrá a que mes le corresponde.

Observá el diagrama y el código en la imagen 2 de la derecha y 3 en la página siguiente.





El programa comienza con la introducción del número que se almacena en la variable **mes**. Luego, empieza la estructura de la instrucción **switch**. Esta instrucción es muy fácil de seguir: en este caso, si el mes es 1 se imprime la leyenda "Corresponde a Enero", si el mes es 2, se imprime la leyenda "Corresponde a Febrero" y así sucesivamente.

Observen que al finalizar cada línea, se coloca una sentencia **break**. El uso del **break** en un **switch** es critico ya que si no se coloca, se continúan evaluando los distintos valores restantes e ingresando en cada una de las alternativas.

Al final, se puede colocar la instrucción **default**, que es opcional, indicando qué hacer si el valor no corresponde a ninguna de las alternativas anteriores.

```
import hsa.Console;
                                                 case 6: c.println ("Corresponde a
class Ejemplo14
                                             Junio");
                                                      break:
 static Console c;
                                                 case 7: c.println ("Corresponde a Julio");
 public static void main (String arg [])
                                                      break:
                                                 case 8: c.println ("Corresponde a
  int mes:
                                             Agosto");
  c = new Console ();
                                                      break:
  c.print ("Ingrese un Numero: ");
                                                 case 9: c.println ("Corresponde a
  mes = c.readInt();
                                             Septiembre");
                                                      break;
  switch (mes)
                                                 case 10: c.println ("Corresponde a
                                             Octubre");
    case 1: c.println ("Corresponde a
                                                      break:
                                                 case 11: c.println ("Corresponde a
Enero"):
        break:
                                             Noviembre");
   case 2: c.println ("Corresponde a
                                                      break:
                                                 case 12: c.println ("Corresponde a
Febrero");
        break:
                                             Diciembre"):
   case 3: c.println ("Corresponde a
                                                      break;
                                                 default: c.println ("No Corresponde a un
Marzo");
                                             mes valido");
        break:
    case 4: c.println ("Corresponde a
                                                      break;
Abril");
        break:
   case 5: c.println ("Corresponde a
Mayo");
        break:
```

Ejemplo 15 Cálculo de sueldos según la categoría

Determiná el sueldo que se le deberá abonar a un empleado, teniendo en cuenta que se le aplicará un incremento de acuerdo a la categoría a la cual pertenezca. El usuario ingresará el sueldo básico y la categoría correspondiente.

Observá el diagrama y el código en la imagen 1 de la derecha y 2 de la página siguiente.

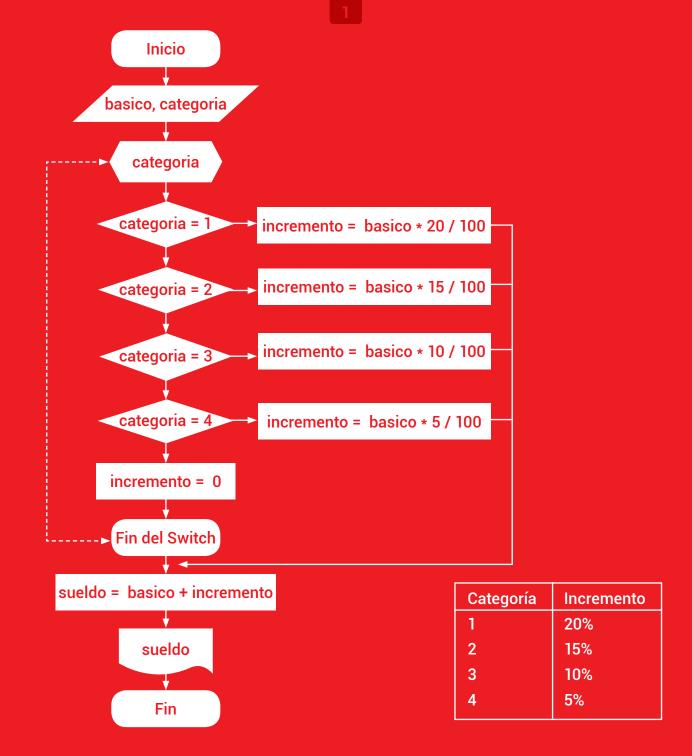


Imagen 2

En algunos casos, puede ser que una misma actividad sea común a varias alternativas, entonces se coloca cada altenativa sin la sentencia break, y a la última alternativa de una determinada actividad, se le coloca el break. Para entenderlo mejor, veamos el siguiente ejemplo.

```
import hsa.Console;
class Ejemplo15
  static Console c;
  public static void main (String arg [])
    float basico; // Sueldo Basico
    float incremento, sueldo;
    int categoria;
    c = new Console ();
    c.print ("Ingrese Sueldo Basico: ");
    basico = c.readFloat();
    c.print ("Ingrese Categoria: ");
    categoria = c.readInt ();
    switch (categoria)
      case 1: incremento = basico * 20
/100;
           break;
      case 2: incremento = basico * 10
/100;
           break:
      case 3: incremento = basico * 15
/100;
```

```
break;
    case 4: incremento = basico * 5
/100;
    break;
    default : incremento = 0; break;
    }
    sueldo = basico + incremento;
    c.println("El sueldo a cobrar es: " +
sueldo);
    }
}
```

Ejemplo 16 ¿Cuántos días tiene un mes?

Ingresá una fecha, y determiná cuantos días le corresponden al mes de esa fecha. (Se deberá imprimir: 28,29, 30 o 31). Observá el código en la imagen 1 de la próxima página.

En el caso del ejemplo 16, para las alternativas 1, 3, 5, 7, 8, 10 y 12 se le asigna a la variable **cantidad** el valor 31, ya que corresponden a los meses enero, marzo, mayo, julio, agosto, octubre y diciembre que tienen esa cantidad de días. Luego, para las alternativas 4, 6, 9 y 11, se le asigna a la variable **cantidad** el valor de 30, ya que corresponden a los meses de abril, junio, septiembre y noviembre.

¿Cómo hacemos con febrero, que puede tener 28 o 29 días según sea un año bisiesto o no? Observen detenidamente el programa: si la alternativa es 2, se pregunta si el año es bisiesto (año bisiesto es el divisible en 4, salvo que sea año secular -último de cada siglo, terminado en «00»-, en cuyo caso ha de ser divisible en 400). En caso de ser verdadero, se le asigna a **cantidad** el valor 29, de lo contrario, se le asigna 28.

Finalmente, se imprime la variable cantidad.

```
import hsa.Console;
class Ejemplo16
  static Console c:
  public static void main (String arg [])
    int dia, mes, anio, cantidad;
    c = new Console ();
    c.print ("Ingrese dia: ");
    dia = c.readInt();
    c.print ("Ingrese mes: ");
    mes = c.readInt();
    c.print ("Ingrese anio: ");
    anio = c.readInt();
    cantidad = 0;
    switch (mes)
       case 1:
       case 3:
       case 5:
       case 7:
```

case 8:

```
case 10:
       case 12: cantidad = 31; break;
      case 4:
      case 6:
       case 9:
       case 11: cantidad = 30; break;
      case 2:
            if ( ((anio%4 == 0) &&
!(anio%100 == 0))
              || (anio % 400 == 0) )
            cantidad = 29;
         else
           cantidad = 28:
         break:
   c.println();
   c.println("Al mes de esa fecha, le
coresponden: "+
         cantidad + " dias");
```



Como podemos observar en el programa anterior, se agregaron dos barras inclinadas al lado del la sentencia **float basico**. Estas dos barras inclinadas se utilizan para colocar **comentarios** en el programa.

Los comentarios no son tenidos en cuenta por el compilador, y pueden escribirse en cualquier lugar del programa. Se utilizan para hacer una determinada acotación, observación o nota, que nunca se verá durante la ejecución. Su utilidad radica en constituir una guía para el programador.

Para colocar un **comentario de varias líneas** en el programa, se utilizan los símbolos: /* y */ para abrir y cerrar un comentario respectivamente. Por ejemplo:

Esto es un comentario de tres lineas en el progama desarrollado.

*/

/*

Ingresá los sueldos de 4 empleados y calculá el promedio de ellos. Deberás imprimir una leyenda que indique si el promedio está por encima o por debajo del Nivel de Pobreza. Este valor de Nivel de Pobreza, también deberá ser ingresado por teclado.

Escribí un programa en el cual dada una lista de tres valores numéricos distintos, se calcule e informe su rango de variación (debe mostrar el menor y el mayor de ellos).

Ejemplo:

Si se introducen los números: 8,2 y 5

La impresión debería ser: Rango = [2..8]

Leé un valor entero. Mostrá el número de día que corresponde a ese valor. Suponé que 0 corresponde a domingo, debés mostrar el nombre del día que le corresponde.

Ejemplo:

ISSD

Ingrese número de día: 4

Corresponde al día: Jueves

Obtené un algoritmo que determine el Sueldo Neto a cobrar por un empleado ingresando el Sueldo Básico y su antigüedad. Tené en cuenta que al Sueldo Básico se le debe realizar un aumento por antigüedad según lo siguiente:

| Hasta 5 años un 3% | Entre 5 y 10 años un 5% | Más de 10 años un 8%

En todos los casos se practica un descuento del 10% en concepto de aportes.

Ingresá 4 números y mostrá el mayor de ellos.

| Si los tres son iguales, imprimí el cubo de ellos.

| Si dos son iguales, imprimí el cuadrado de ellos.

| Si todos son distintos, imprimí la leyenda: "Todos son distintos".

Ingresá dos ángulos de un triángulo y determiner a qué tipo de triángulo corresponde (Equilátero: 3 ángulos iguales, Isósceles: 2 ángulos iguales, Escaleno: todos los ángulos desiguales).

Ejemplo:

Ingrese un ángulo: 40 Ingrese otro ángulo: 70

Corresponde a un Triángulo Isósceles

Ingresá el precio de una prenda y la categoría a la que pertenece. Deberás imprimir el importe a pagar, teniendo en cuenta el porcentaje a incrementar, de acuerdo a su categoría:

- 0			
	Categoría	Porcentaje	
	'Á'	5%	
	'B'	8%	
	'C'	10%	
	'D'	12%	

Tipo de	Característica	Precio por día
Vehículo		
1	Sedan 4 puertas	\$ 80
2	Coupe	\$ 100
3	Pick up	\$ 120
4	4 x 4	\$ 150

Ejemplo:

27

ISSD

Ingrese Tipo de Vehiculo: 2 Ingrese cantidad de días: 5

Importe a abonar: \$ 500



Imágenes

www.pexels.com

www.pixabay.com

www.flickr.com

28

ISSD

