GUÍA DE TRABAJO # 3
ESTRUCTURAS CONDICIONALES

1. IDENTIFICACIÓN DE LA ASIGNATURA

Asignatura:	Lógica de Programación y Laboratorio						
Código:	000506001			**Área:**	Ciencias Básicas Tecnología		
Plan de Estudios:	2019			**Semestre:**	2		
Créditos: 5	**TPS:** 6	**TIS:** 9		**TPT:** 96	**TIT:** 144		
Trabajo Independiente:				**Trabajo Presencial:**			
Trabajo Teórico	X	Trabajo Práctico	X	Trabajo Teórico		Trabajo Práctico	X

2. TEMA Y COMPETENCIA:

Contenido Temático:	Condicional simple Condicional compuesto (o doble) Condicionales anidados Condicional múltiple
Competencia por Desarrollar:	Implementa estructuras condicionales simples, dobles y anidadas a través de su identificación previa.
Resultado de aprendizaje:	Diseña soluciones a problemas que requieren la toma de decisiones, expresadas como condicionales en un algoritmo o programa.

3. RECURSOS REQUERIDOS

- Equipo de cómputo con conexión a Internet
- Procesador de textos como Word o Notepad++
- NetBeans u otro IDE para Java

4. MARCO TEÓRICO

Cuando estamos diseñando la solución a un problema es común que nos encontremos con casos en los que se debe seguir un camino, u otro, de acuerdo con una situación determinada. Aquellas situaciones que generan bifurcaciones en la secuencia de pasos de un algoritmo se conocen como **estructuras condicionales** y se usan para tomar decisiones.

De manera general, una estructura condicional está definida sobre una expresión lógica, que al ser verdadera o falsa permite ejecutar un conjunto de instrucciones u otro, para luego continuar con el flujo normal de las instrucciones. Dicho de otra manera, las estructuras secuenciales nos permiten preguntar o evaluar cómo está el entorno que nos rodea para actuar de acuerdo con la respuesta obtenida.

Veamos un ejemplo de un algoritmo en lenguaje natural: considere el problema de ir de la casa a la tienda del barrio para comprar algunas cosas. Una posible solución para este problema es la siguiente:

1. Se debe hacer una lista de las cosas a comprar
2. Se hace un cálculo aproximado del dinero a gastar
3. Asegurarse de llevar el dinero de acuerdo con el cálculo que se realizó antes
4. Salir de la casa y caminar hasta la tienda
5. Pedir los productos con base en la lista de compras
6. Pagar los productos
7. Empacar los productos
8. Volver a la casa caminando con los productos comprados

Tómese su tiempo y analice esta solución:

- ¿Será que la solución es la misma, por ejemplo, si estuviera lloviendo?
- ¿Qué pasa si es necesario que la solución considere ambos escenarios, cuando está lloviendo y cuando no está lloviendo?

En casos como este se debe hacer uso de alguna estructura de decisión que nos permita evaluar ciertas cosas en el algoritmo. Por ejemplo, para mejorar la solución anterior se puede tomar la decisión de llevar sombrilla o no, con base en si está lloviendo o no, así:

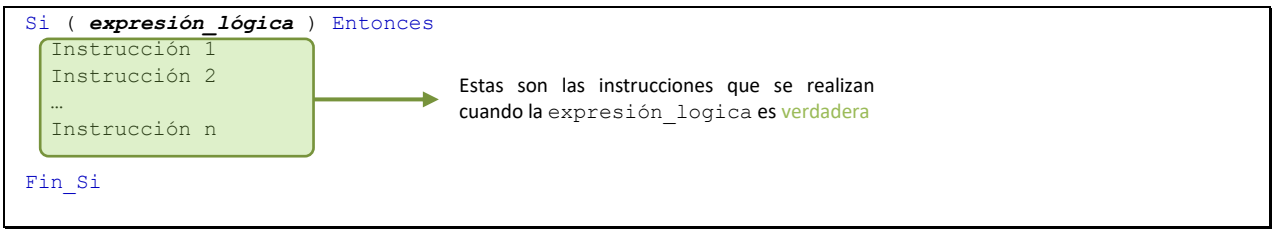
1. Se debe hacer una lista de las cosas a comprar
2. Se hace un cálculo aproximado del dinero a gastar
3. Asegurarse de llevar el dinero de acuerdo con el cálculo que se realizó antes
4. Si ***está lloviendo*** entonces:
   - 4.1 Buscar una sombrilla
   - 4.2 Abrir la sombrilla antes de salir de casa para no mojarse al salir
5. Salir de la casa y caminar hasta la tienda
6. Pedir los productos con base en la lista de compras
7. Pagar los productos
8. Empacar los productos
9. Volver a la casa caminando con los productos comprados

Ahora, analizando esa solución vemos que hay una bifurcación en el paso 4, la cual se debe a la condición de si está lloviendo. Esta condición me permite tomar un curso alterno, específicamente cuando la condición que se evalúa es verdadera. A continuación, veremos los diferentes tipos de condicionales que se pueden usar en un algoritmo y algunos ejemplos.

4.1 Condicional Simple

El condicional simple es una estructura de decisión sencilla cuyo funcionamiento consiste en evaluar una ***expresión booleana***. Si la expresión booleana es verdadera, el algoritmo ejecuta las instrucciones definidas dentro del bloque condicional, pero si la condición es falsa, el algoritmo continúa con la ejecución saltándose el bloque condicional.

La forma general de un condicional simple es la siguiente:



Note que el condicional, identificado por las palabras clave **`Si`** **`(…)`** **`Entonces`**, define el inicio de un bloque de código que contine el conjunto de instrucciones (resaltadas en verde) que se ejecutarán cuando la **`expresión_lógica`** sea verdadera. Nuevamente, cuando la **`expresión_lógica`** es falsa, el algoritmo se salta el bloque del condicional.

Ejemplo sobre condicional simple

Se debe realizar un algoritmo que pida a una persona el nombre y la edad, si la persona tiene menos de 18 años se debe indicar que esa persona es menor de edad.

Analizamos la definición del problema vemos que existe una oración del tipo Si ***pasa esto***, entonces ***haga esto***. Este tipo de oración nos indica que debemos usar una estructura condicional y tiene dos partes. En primer lugar, está la condición, la cual se identifica en la oración por la frase Si ***pasa esto*** […]. En segundo lugar, están las acciones que se deben ejecutar cuando la condición es verdadera, las cuales se enmarcan por la frase ***haga esto*** […].

Analicemos el problema, ¿cuál es la condición? Y ¿cuál o cuáles son las acciones que se deben seguir cuando esa condición sea verdadera? Las respuestas a estas preguntas son las que definen la estructura condicional simple que debemos usar:

- ***Condición:*** debemos verificar si la edad es menor a 18, esto escrito como una expresión booleana es: `edad < 18`
- ***Acción:*** mostrar un mensaje en la pantalla indicando que la persona es menor de edad, esto lo hacemos con la instrucción `Escribir`.

Una posible solución al problema es la siguiente.

```
//Declaro las variables que se requieren
Cadena: nombre
Entero: edad

//Se piden los datos de la persona y se almacenan en las variables respectivas
Escribir("Ingrese el nombre de la persona: ")
Leer(nombre)

Escribir("Ingrese la edad de la persona: ")
Leer(edad)

//Usamos el condicional simple para verificar si es menor de edad y mostrar el mensaje respectivo
Si (edad < 18) Entonces
  Escribir(nombre, " aún eres un menor de edad!")
Fin_Si
```

Note que en la condición verifica si el valor almacenado en la variable `edad` es menor que `18`, lo que indica que la persona es menor de edad. Así, cuando esa expresión booleana sea verdadera, se ejecutará la instrucción que está dentro del condicional. Por el contrario, cuando la expresión es falsa el algoritmo omite o "salta" las instrucciones que están en el condicional.

Otro ejemplo sobre condicional simple

Se requiere de un algoritmo que, dado un número, muestre el valor absoluto de ese número.

Para solucionar este problema debemos recordar que el valor absoluto de un número `x` es el valor positivo de ese número. Por ejemplo, el valor absoluto de 5 es 5, mientras que el valor absoluto de -3 es 3. Así, el algoritmo que calcula el valor absoluto de un número debe evaluar si el número es negativo y en caso de serlo cambiar el signo de dicho número a positivo. Lo anterior define cuál es la condición que debemos usar en el algoritmo y las acciones que debemos tomar cuando esa condición es verdadera:

- ***Condición:*** verificar si el número es negativo
- ***Acción:*** cambiar el signo del número multiplicando este por -1

Una posible solución al problema es la siguiente.

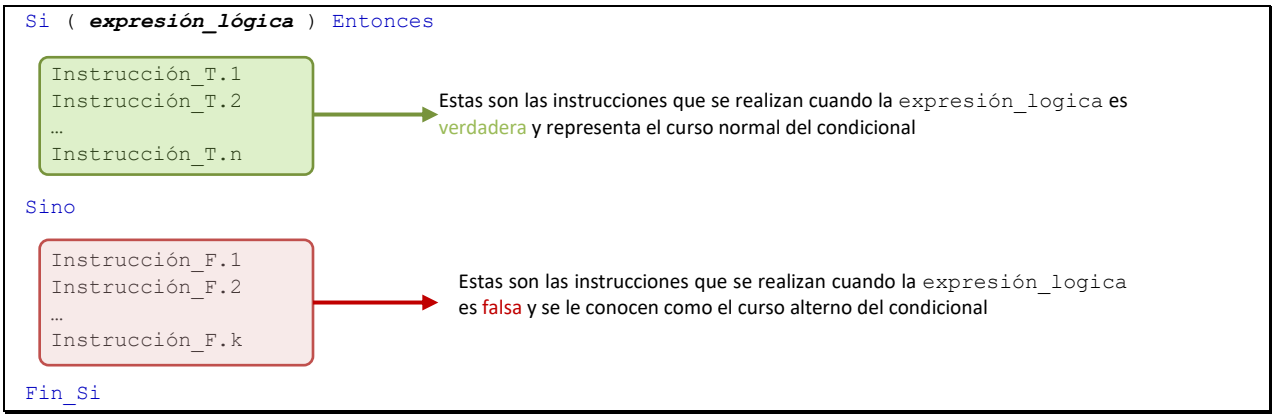
```
//Declaro una variables para almacenar el número
Real: x

//Pido el número al usuario
Escribir("Ingrese el número del que desea conocer su valor absoluto: ")
Leer(x)

//Usamos un condicional para determinar si el número es negativo y cambiar su signo
Si (x < 0) Entonces
  x = (-1)*x
Fin_Si

Escribir("El valor absoluto del número ingresado es", x)
```

Note que la condición evalúa si el número `x` es menor que cero, lo que significa que `x` es negativo, en caso de serlo (cuando la condición es verdadera) se cambia el signo de `x` a positivo, pero si la condición es falsa (lo que indica que el número no es negativo), el algoritmo salta el bloque condicional y se muestra el valor de `x` tal como se ingresó.

4.2 Condicional Compuesto (o doble)

Un condicional doble nos permite tomar dos caminos en función del valor de la condición. Un camino se abre cuando la condición es verdadera y el otro cuando la condición es falsa. Es decir, el condicional doble nos permite indicar qué hacer en caso de que la expresión lógica que estamos evaluando sea verdadera y qué hacer cuando esta sea falsa.

La forma general de un condicional doble es la siguiente:



Ejemplo sobre condicional doble

Se requiere de un algoritmo que determine si una persona puede entrar o no a una discoteca en función de su edad.

Cuando analizamos este problema identificamos que el algoritmo debe considerar dos alternativas: una cuando la persona es menor de edad y la otra cuando la persona es mayor de edad. Estas dos alternativas son las que determinan el uso de un condicional doble:

- ***Condición:*** sabemos que sólo a aquellos que son mayores edad se les permite el ingreso a las discotecas, con base en esto la condición debe determinar si una persona es mayor de edad o no, con base en su edad. La expresión lógica que puede usarse en este caso es la siguiente: `edad < 18`.
- ***Acciones:*** con base en el enunciado hay dos acciones. Cuando la condición sea verdadera debemos mostrar un mensaje indicando que la persona no puede entrar a la discoteca por ser menor de edad y cuando la condición sea falsa debemos mostrar un mensaje que indique que la persona si puede entrar a la discoteca.

Una posible solución a este problema es la siguiente.

```
//Declaro las variables que se requieren
Entero: edad

//Se pide la edad de la persona
Escribir("Ingresa tu edad: ")
Leer(edad)

//Se evalúa si la persona es menor de edad o no y se muestran los mensajes respectivos
Si (edad < 18) Entonces
  Escribir("NO puedes ingresar a la discoteca, aún eres menor de edad")
Sino
  Escribir("SI puedes ingresar a la discoteca! Disfruta de la fiesta!")
Fin_Si
```

Note que cuando una persona tiene menos de 18 años la condición se evalúa verdadera y en ese caso se harán las instrucciones del fujo normal del condicional. Ahora, cuando la persona tiene 18 años o más, la condición se evalúa como falsa y, por tanto, se harán las instrucciones del alterno del condicional.

Otro ejemplo sobre condicional doble

Se requiere de una aplicación que, con base en la nota final de una asignatura, le indique al estudiante si este ganó o perdió dicha asignatura.

Cuando analizamos el problema encontramos que se plantean dos posibles caminos: uno cuando el estudiante gana la asignatura y el otro cuando la pierde. La pregunta que nos debemos hacer ahora es, ¿cuál es la condición que determina que una materia se haya ganado o perdido? La respuesta a esta pregunta es la que define la condición a utilizar.

- ***Condición:*** sabemos que una asignatura se gana cuando su nota final es mayor o igual a 3.0, así la expresión lógica del condicional debe ser algo como: `notaFinal >= 3.0`
- ***Acciones:*** hay dos acciones, si `notaFinal >= 3.0` , entonces se muestra un mensaje indicando que el estudiante ganó la asignatura, en caso contrario, se debe mostrar un mensaje indicando que el estudiante perdió la asignatura.

Una posible solución a este problema es la siguiente.

```
//Declaro las variables que se requieren
Real: notaFinal

//Se pide al estudiante la notaFinal de la asignatura
Escribir("Ingresa la nota final de la asignatura: ")
Leer(notaFinal)

//Se evalúa si el estudiante ganó o perdió la asignatura
Si (notaFinal >= 3.0) Entonces
  Escribir("Felicitaciones, has ganado la materia!")
Sino
  Escribir("Lo siento, pero has perdido la materia")
Fin_Si
```

¿Aún no está claro?, veamos otro ejemplo!

Otro ejemplo sobre condicional doble

En el almacén de calzado "***mis piecitos***" están interesados en aumentar las ventas de zapatos para hombre. Para ello, se ha planteado una campaña que consiste en hacer un descuento del 40% a los compradores hombres y descuento del 10% a las mujeres. Haga un algoritmo que con base en el valor de la compra y el género del cliente muestre el valor de la compra, el valor del descuento y el total a pagar.

Antes de continuar, deténgase y analice el problema, ¿identifica cuál es la condición y cuáles son las acciones por tomar? Veámoslo entonces:

- ***Condición:*** de acuerdo con el enunciado, el descuento se hará con base en el género del cliente. Esto nos indica que para determinar el valor del descuento debemos evaluar si el cliente es de género masculino u otro.
- ***Acciones:*** hay dos acciones, si el cliente es de género masculino, debemos calcular un descuento del 40% sobre el valor de la compra, pero si es de otro género debemos calcular sólo un descuento del 10%.

Con base en esto, una posible solución para el problema es la siguiente.

```
//Declaro las variables que se requieren
Entero: genero //1. Masculino, 2. Femenino, 3. Otro.
Real: valorCompra, valorDescuento, totalAPagar

//Se piden los datos de entrada
Escribir("Ingrese el valor de la compra: ")
Leer(valorCompra)

Escribir("Seleccione el género del cliente:\n 1. Masculino\n 2. Femenino\n 3. Otro")
Leer(genero)

//Con base en el género del cliente se calcula el descuento
Si (genero == 1) Entonces
  descuento = valorCompra * 0.40
Sino
  descuento = valorCompra * 0.10
Fin_Si

//Calcula el valor a pagar
totalAPagar = valorCompra - descuento

//Muestra los datos solicitados en la pantalla
```

```
Escribir("Valor de la compra: ", valorCompra)
Escribir("Valor del descuento: ", descuento)
Escribir("-------------------------------------")
Escribir("Total a pagar: ", totalAPagar)
```

Note que en esta solución el género se codificó como un número entero, tal que el uno representa al género masculino, el dos al género femenino y el tres a los otros géneros. Esta es una opción de diseño común que bien puede ser reemplazada por el uso de cadenas o caracteres.

4.3 CONDICIONALES ANIDADOS

Existen muchos problemas en los que una decisión (o condición) depende de una decisión o condición que se ha evaluado previamente. En ese caso es cuando se debe hacer uso de los condicionales anidados. Ahora, se les llama anidados porque son condicionales que en uno de sus bloques de código contiene uno más condicionales. Suena como extraño, ¿no?, pero bueno, veamos un ejemplo para aclarar el concepto. Considere el siguiente problema.

Ejemplo sobre condicional anidado

El centro de mecánica automotriz "***Mi cacharrito***" está promocionado sus nuevas llantas "***Michell***". Para ello, ha establecido un sistema de descuento así: si una persona compra una o dos llantas, su precio individual será de $350.000. Si compra entre tres y cinco llantas su precio individual será de $320.000, pero si compra más de cinco llantas, el precio de cada una será de $290.000. Haga un algoritmo que muestre el valor que debe pagar una persona por las llantas que comprará.

Al leer el problema nos percatamos de que existen tres posibilidades (opciones o caminos):

- Si se compran 1 o 2 llantas, su valor individual será de $350.000
- Si se compran entre 3 y 5 llantas, su valor será de $320.000
- Si se compran más de 5 llantas su valor será de $290.000

Note que estas opciones son mutuamente excluyentes; es decir, una persona pagará uno y sólo uno de estos valores por cada llanta: $350.000, $320.000 o $290.000. Cuando se presenta un caso como este requerimos usar un condicional anidado, que una de sus posibles versiones puede ser la siguiente:

- ***Condición 1:*** se debe verificar si se compraron una o dos llantas, en caso de que esto sea verdadero, cada llanta se cobrará a $350.000. En caso de que esa primera condición sea falsa, es decir, en caso de que no haya comprado 1 o 2 llantas, aparece la segunda condición.
- ***Condición 2:*** se debe verificar si el número de llantas que se compraron está entre 3 y 5. En caso de que esta condición sea verdadera, cada llanta se factura a un valor de $320.000. Pero si esta condición es falsa, la única opción probable (por la lógica del problema) es que el usuario compre más de 5 llantas, en cuyo caso se facturarán a $290.000, cada una.

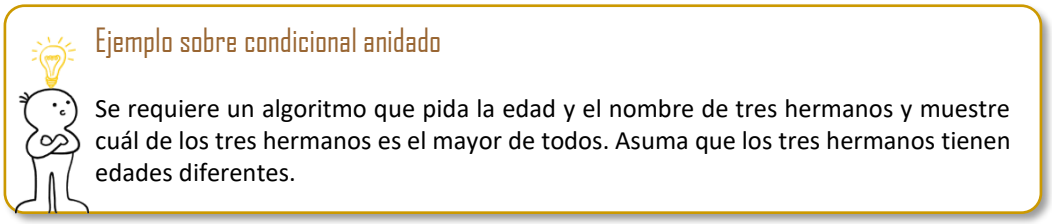
Con base en esto, una posible solución para el problema es la siguiente.

```
//Declaro las variables que se requieren
Entero: llantasCompradas
Real: valorAPagar

//Se piden el número de llantas a comprar
Escribir("Ingrese el número de llantas que se van a comprar: ")
Leer(llantasCompradas)
```

```
//Se verifica si se comprarán 1 o 2 llantas
Si (llantasCompradas <= 2) Entonces
  valorAPagar = llantasCompradas * 350000
Sino
  //Se verifica si se comprarán entre 3 y 5 llantas
  Si (llantasCompradas >= 3 AND llantasCompradas <= 5) Entonces
    valorAPagar = llantasCompradas * 320000
  Sino
    valorAPagar = llantasCompradas * 290000
  Fin_Si
Fin_Si

//Se muestra en pantalla l valor a pagar por el usuario
Escribir("El costo total de las llantas es de $", valorAPagar)
```

Note algo particular, en el último `sino` no es necesario verificar `Si(llantasCompradas > 5)` puesto que por descarte se sabe que si no son menos de dos o menos de cinco, es que porque son más de cinco. También es importante resaltar que el segundo condicional puede ser reemplazado por `Si(llantasCompradas <= 5)`. Esto es posible porque la primera condición asume todos los valores menores o iguales a 2; por tanto, cundo en la segunda condición se evalúa que el valor sea menor o igual a 5, implícitamente se están considerando también los valores que son mayores a 2, por estar en el sino de esa primera condición.

¿Aún no está claro?, veamos otro ejemplo.



Al analizar este problema vemos que hay tres posibilidades: el primero de los hermanos ingresados es el mayor, el segundo es el mayor o el tercero es el mayor. De ahí que se requiera un condicional anidado. Una posible forma de hacerlo es a partir de las siguientes condiciones:

- ***Condición 1:*** se debe verificar si el primero de los hermanos es mayor que el segundo y a la vez es mayor que el tercero.
- ***Condición 2:*** Si el primero no es el mayor, se debe verificar si el segundo es mayor que el primero y a la vez si es mayor que el tercero. Por defecto, si el primero no es el mayor o el segundo no es el mayor, entonces será el tercero.

```
//Declaro las variables que se requieren
Entero: edad1, edad2, edad3
Texto: nombre1, nombre2, nombre3

//Se piden los datos de los tres hermanos
Escribir("Ingrese la edad y el nombre de uno de los hermanos")
Leer(edad1, nombre1)

Escribir("Ingrese la edad y el nombre de otro de los hermanos")
Leer(edad2, nombre2)

Escribir("Ingrese la edad y el nombre del último de los hermanos")
Leer(edad3, nombre3)

//Se verifica si el primero es el mayor
Si (edad1 > edad2 AND edad1 > edad3) Entonces
```

```
  Escribir("El mayor de los hermanos es ", nombre1)
Sino
  //Se verifica si el segundo es el mayor
  Si (edad2 > edad1 AND edad2 > edad3) Entonces
    Escribir("El mayor de los hermanos es ", nombre2)
  Sino
    Escribir("El mayor de los hermanos es ", nombre3)
  Fin_Si
Fin_Si
```

Nuevamente, como las tres opciones son mutuamente excluyentes, solo necesitamos dos condicionales (anidados) para dar respuesta a los tres caminos que son: el primero es el mayor, el segundo es el mayor o el tercero es el mayor, considerando que sólo hay un mayor.

Para Recordar:

- Los ***condicionales simples*** se usan cuando hay sólo un curso de acciones a tomar cuando la condición es verdadera. Por otro lado, el ***condicional doble*** se usa cuando hay dos cursos de acciones: uno cuándo la condición es verdadera y otro cuando la condición es falsa. Finalmente, los condicionales anidados se usan cuando hay más de dos posibles caminos al tomar una decisión.

4.4 Condicionales Múltiples

En este tipo de condicional se evalúa una variable, denominada ***selectora***, que puede tomar diversos valores predefinidos. Según el valor que la variable selectora tome, se ejecutan las acciones correspondientes a ese valor. Este tipo de estructura no existe en todos los lenguajes de programación puesto que puede ser reemplazada por el uso de condicionales anidados.

La forma general de un condicional múltiple es la siguiente:

```
Según_sea ( variable_selectora ) Haga

  Caso 1:
    Instrucción_1.1
    Instrucción_1.2
    …
    break                 → Estas instrucciones se ejecutan cuando la variable_selectora sea igual a 1.

  Caso 2:
    Instrucción_2.1
    Instrucción_2.2
    …
    break                 → Estas instrucciones se ejecutan cuando la variable_selectora sea igual a 2.

  ...

  Caso n:
    Instrucción_n.1
    Instrucción_n.2
    …
    break                 → Estas instrucciones se ejecutan cuando la variable_selectora sea igual a n.


  Por_defecto:
    Instrucción_x.1
    Instrucción_x.2
    …
    break                 → Estas instrucciones se ejecutan cuando la variable_selectora es diferente a
                            cualquiera de los valores anteriores
Fin_Segun_sea
```

Se debe tener presente que la `variable_selectora` debe ser una variable de tipo **Entero** o **Carácter**, aunque algunos lenguajes también aceptan cadenas de texto. Igualmente, se debe considerar que el último bloque de instrucciones (Por_defecto) no es obligatorio y sólo se ejecuta cuando la `variable_selectora` tiene un valor que no se ha definido en ninguno de los casos.

Veamos un ejemplo de uso de esta estructura.

Ejemplo sobre condicional múltiple

Se requiere un algoritmo que pida dos números al usuario: `n1` y `n2`. El algoritmo debe calcular y mostrar el resultado de:

- `100^n2`, cuando `n1` sea 1,
- `100*n2`, cuando `n1` sea 2,
- `100/n2`, cuando `n1` sea 3 y `n2<100` o `n2/100` cuando `n1` sea 3 y `n2>=100`
- `0` en otro caso

Con base en este problema y usando un condicional múltiple, una posible solución es la siguiente:

```
//Declaro las variables que se requieren
Entero: n1, n2
Real: resultado

//Se piden los datos de entrada
Escribir("Ingrese un valor entero para n1: ")
Leer(n1)

Escribir("Ingrese un valor entero para n2: ")
Leer(n2)

//Usamos un condicional múltiple
Según_sea (n1) Haga

  Caso 1:
    resultado = 100^n2
    break

  Caso 2:
    resultado = 100*n2
    break

  Caso 3:
    Si (n2 < 100) Entonces
      resultado  = 100/n2
    Sino
      resultado = n2/100
    Fin_Si
    break

  Por_defecto:
    resultado = 0

Fin_Segun_sea

Escribir("El resultado de la operación realizada es: ", resultado)
```

Extiende tu conocimiento …

Si quieres ver un vídeo explicativo, te sugiero que sigas este enlace: https://www.youtube.com/watch?v=DQnUE_IbVG0

5. PROCEDIMIENTO

"***Grandes Nano-Partículas***" es una pequeña empresa que por las ganancias obtenidas en el primer semestre del año ha considerado dar una bonificación a sus empleados con base en algunas condiciones muy particulares. De acuerdo con el gerente de la empresa, el cálculo del valor de la bonificación se hará con base en: la edad, sexo (1. Hombre, 2 mujer), el nivel de su cargo (A. Alto, B. Bajo, C. Medio), la zona geográfica en la que trabaja el empleado (1. Europa, 2 Asia, 3. América del Norte, 4. América Central, 5. América del sur, 6. África) y su salario en dólares. Las reglas que se han definido para la bonificación son las siguientes:

- Si el empleado es una mujer, con una edad entre 35 y 45 años, de las zonas de Europa o América, con un salario inferior a US 2500; su bonificación será del 25% del salario.
- Si el empleado es un hombre con nivel de cargo medio o bajo y es de las zonas de Asia o África; la bonificación que recibirá es del 20% de su salario.
- Si es una mujer de nivel de cargo alto de América Central, su bonificación será del 18% del salario
- Si es un hombre entre 40 y 55 años, con un salario por encima de US 3000, su bonificación será del 16% del salario.
- Si el empleado no cumple con alguna de las condiciones antes planteadas su bonificación será del 15% del salario.

Con base en este enunciado:

- Defina el cliente, el usuario, la lista de requerimientos, las reglas de negocio y las entidades del mundo del problema que intervienen en la solución
- Defina los requerimientos en su formato extendido
- Describa los atributos y métodos que deben tener las clases que representan las entidades del mundo del problema
- Con base en la descripción anterior, bosqueje el diagrama de clases de la solución
- Implemente la solución en seudocódigo o algún lenguaje de programación orientado por objetos

6. RÚBRICA DE EVALUACIÓN

A continuación, se describen los elementos considerados en la evaluación y su rango de valoración.

Resultado de aprendizaje	Indicador de logro	Rango de valoración
Diseño de la solución al problema planteado que incluye la especificación del problema, especificación de requerimientos, diagrama de clases e implementación	**Especificación del problema:** identifica de manera correcta el cliente, el usuario, la lista preliminar de requerimientos y los elementos del mundo que intervienen en la solución del problema	0.0 –0.5
	Especificación de requerimientos: describe correctamente cada funcionalidad del sistema e identifica las entradas y salidas específicas de estas	0.0 – 1.0
	Diagrama de clases: hay una correspondencia entre las clases y las entidades del mundo del problema identificadas. Además, se	0.0 – 0.5

	evidencia la identificación de los métodos que le darán solución al problema con base en los requerimientos	
	Implementación: se entrega un seudocódigo o código en un lenguaje de programación que da solución al problema y es coherente con la especificación de requerimientos y el diagrama de clases	0.0 – 3.0

7. BIBLIOGRAFÍA

- Jorge Villalobos, Rubby Casallas, "Fundamentos de Programación –Aprendizaje Activo Basado en Casos", Universidad de los Andes.
  Disponible en: http://cupi2.uniandes.edu.co/images/APO1/fundamentos-de-programacion.pdf
- Jorge Martínez, "Fundamentos de Programación en Java", Universidad Complutense de Madrid.
  Disponible en: https://docs.google.com/file/d/0Byy7aUl9u4fBdnZjVnZOampmTjA/edit
- UskoKruM2010, "Tutorial Java - 14. ARREGLOS (ARRAYS) (VECTORES) | UskoKruM2010". Tutorial en YouTube.
  Disponible en: https://www.youtube.com/watch?v=es-SDwx-Hfc
- UskoKruM2010, "Tutorial Java - 31. Arreglos De Objetos En JAVA". Tutorial en YouTube.
  Disponible en: https://www.youtube.com/watch?v=JMzq2eN4HDI

Elaborado Por:	Carlos Andres Mera Banguero
Versión:	1.0
Fecha	Septiembre de 2022
Aprobado Por:	Comité Curricular Departamento de Sistemas de Información Acta 02 del 7 de febrero de 2023