



## PRÁCTICA DE LABORATORIO / TALLER Nro. 4

Carrera Computación

A. DATOS INFORMATIVOS		
<b>Asignatura:</b> Teoría de la Programación	<b>Ciclo / Semestre:</b> Primero	<b>Paralelo:</b> A
<b>Docente:</b> José Oswaldo Guamán Quinche	<b>Período Académico:</b> Septiembre 2024 – Marzo 2025	
<b>Integrantes:</b> Cael Alejandro Soto Castillo		

B. INFORMACIÓN GENERAL	
<b>Unidad:</b> 3. Estructuras algorítmicas de control	
<b>Tema:</b> Resolución de problemas a través de programas utilizando las estructuras algorítmicas repetitivas	
<b>Fecha:</b> Loja, 30 de noviembre de 2024	<b>Nro. horas:</b> 6
<b>Objetivos:</b> <ul style="list-style-type: none"><li>● Usar el proceso o metodología de resolución de problemas a través de programas.</li><li>● Obtener la representación de la resolución del problema a través de un algoritmo y/o diagrama.</li><li>● Implementar estructuras algorítmicas repetitivas y subprogramas en el lenguaje de programación C.</li><li>● Presentar los resultados obtenidos del programa</li></ul>	
<b>Corresponde al resultado de aprendizaje:</b>	
Aplica las estructuras de programación en la resolución de problemas básicos, bajo los principios de solidaridad, transparencia, responsabilidad y honestidad	
<b>Recursos y/o materiales:</b> <ul style="list-style-type: none"><li>● Computador.</li><li>● Compilador gcc</li><li>● Visual Studio Code (IDE)</li><li>● Lenguaje de programación C</li><li>● Material bibliográfico o recurso indicado en el EVA.</li></ul>	

C. DESARROLLO
<b>Instrucciones:</b>
<ol style="list-style-type: none"><li>1. Para cada ejercicio(s) planteado(s) realizar lo siguiente:<ol style="list-style-type: none"><li>1. Análisis y obtención de requerimiento(s) .</li><li>2. Elabore un diseño (idearse gráfico) de las tareas que debería realizar tomando en cuenta el análisis y requerimiento(s) obtenido(s); Construir Diagrama de Flujo.</li><li>3. OPCIONAL: Construir el algoritmo principal y de las funciones relevantes en base al diseño o idea de como resolver esa tarea. Adicionalmente realizar las pruebas de escritorio o validación respectivas de los algoritmos de las tareas específicas. (Si lo hace ya modular solo de las funciones principales, si lo hace todo en un solo algoritmo la parte del bucle)</li><li>4. Traducir el diseño y algoritmos a lenguaje de programación C considerando mensajes para interacción con el usuario.</li><li>5. Ejecutar el programa y verificar el funcionamiento correcto del mismo.</li></ol></li><li>2. Desarrollar el informe demostrando el proceso de desarrollado</li><li>3. Resolver las preguntas planteadas.</li><li>4. Establecer sus respectivas conclusiones (tomar en cuenta los objetivos planteados)</li></ol>



### Equipo de trabajo de máximo de 1 persona

#### Ejercicios:

- Una compañía de autos usados paga \$2,500.00 de sueldo a sus empleados por mes, Además agrega pagos extras a sus sueldos como: una comisión de \$250.00 por cada automóvil vendido cuyo valor de venta supere el \$10,000.00 y el 5% de utilidad del valor total de ventas. La compañía necesita emitir un informe desglosado por empleado donde indique el número total de autos vendidos, el valor total de los autos que vendió, el monto total que se le debería pagar al final del mes, indicando: el sueldo mensual, la comisión por cada automóvil y la utilidad por el valor total de la venta. También debe permitir ingresar la venta de los automóviles hasta que se desee y luego emitir el respectivo informe.
- El costo de las llamadas telefónicas internacionales depende de la zona geográfica en la que se encuentre el país destino y del número de minutos hablados. En la siguiente tabla se presenta el costo del minuto por hora. Cada zona se le ha asociado con una clave (Use tipos de datos ENUMERADOS PARA LA ZONA).

Zona	Clave	Precio Minuto (USD)
AMÉRICA DEL NORTE	12	2.75
AMÉRICA CENTRAL	15	1.89
AMÉRICA DEL SUR	18	1.60
EUROPA	19	3.5
ASIA	23	4.5
ÁFRICA	25	3.1
OCEANÍA	29	3
RESTO DEL MUNDO	31	6

- La empresa municipal de agua potable de Loja desea cobrar y calcular mensualmente el valor exacto de consumo del agua potable de cada medidor que pertenece a un contribuyente, de acuerdo a la ordenanza vigente, la cual establece la planilla de acuerdo a los siguientes rubros:

A) SERVICIO DE AGUA POTABLE: Para obtener el rubro ríjase a la siguiente tabla:

Consumo en m <sup>3</sup>	Valor a cobrar
0 hasta 15	\$ 3.00
15 hasta 25	\$ 0.10 por cada m <sup>3</sup> adicional a partir de la base y rango anterior.
25 hasta 40	\$ 0.20 por cada m <sup>3</sup> adicional a partir de la base y rango anterior..
40 hasta 60	\$ 0.30 por cada m <sup>3</sup> adicional a partir de la base y rango anterior.
60 en adelante	\$ 0.35 por cada m <sup>3</sup> adicional a partir de la base y rango anterior.

B) IMPUESTO DE ALCANTARILLADO: 35% DEL VALOR DEL RUBRO DEL SERVICIO DE AGUA POTABLE.

C) TASA POR RECOLECCIÓN DE BASURA: 0.75 DÓLARES;

D) TASA POR COSTO DE PROCESAMIENTO DE DATOS: 0.50 DÓLARES

La ordenanza también permite descuentos para los medidores que estén registrados al contribuyente de la tercera edad o posean algún tipo discapacidad, para lo cual sólo se los puede aplicar sobre el rubro del SERVICIO DE AGUA POTABLE, en base a las siguientes condiciones.

(a) Si pertenece a la tercera edad y su consumo esta entre 0 hasta 15 m<sup>3</sup> (rango base) existe un descuento del 50%, en caso de exceder sólo se realizará el descuento del 30% sobre el rango base.

(b) Si pertenece a un discapacitado, se tomará como descuento su porcentaje de discapacidad sólo sobre el rango base.

- Reconozca, genere mediante teclado, y resuelva la siguiente serie

$$(1/1)^2 + (1/3)^4 - (2/5)^6 - (3/7)^8 + (5/9)^{10} + (8/11)^{12} \dots$$

**Resultados:**

**1) EJERCICIO 1:**

- **Requerimientos del problema:**

- Datos de entrada:
- Sueldo base mensual por empleado: \$2,500.00.
- Comisión adicional de \$250.00 por cada automóvil vendido que tenga un precio mayor a \$10,000.00.
- Utilidad adicional de 5% del valor total de ventas realizadas por el empleado.
- Poder ingresar datos de varios automóviles hasta que el usuario decida detenerse.

Salida: el programa nos pide elaborar una factura que tenga:

- Nro. total de autos vendidos.
- Valor total de los autos vendidos.
- Comisión total de autos vendidos.
- Utilidad de las ventas.
- Sueldo total de fin de mes.

- **Pseudocódigo:**

INICIO:

Definir constantes (estas no cambiarán en ningún momento en la ejecución):

final float Sueldo\_Base = 2500.00

final float Comision\_Auto = 250.00

final float Porcentaje\_Utilidad = 0.05

Definir variable boolean continuar = verdadero

**Mientras principal (continuar = verdadero):**

Iniciar y definir variables dentro del bucle:

Int autosVendidos = 0

float totalVentas = 0.0

float comisiones = 0.0

boolean Registrar ventas = true

Imprimir: "Ingrese el nombre del trabajador:" → Guardar en String nombre.

**Bucle anidado 1: Mientras(Registrar\_ventas)**

Imprimir: "Ingrese el valor del automóvil vendido" → Guardar en valorVenta.

Actualizar totalVentas sumando valorVenta.

Incrementar autosVendidos en 1 por interacción.

**Si(if) valorVenta > 10000:**

comisiones = comisiones + Comision\_auto.

cerrar if

Imprimir: "¿Desea registrar otra venta?" → Guardar en: char respuesta.  
Repetir registro de venta

Crear un comprobador de respuesta (s/n)

Mientras anidado 2: respuesta diferente de s && n

Imprimir: Respuesta incorrecta

Volver a preguntar hasta que el usuario ingrese correctamente

cerrar mientras anidado 2

Si(if) respuesta = 'n':

Cambiar registrarVentas a falso.

cerrar if

Fin Mientras anidado 1.

Cálculos:

float utilidad = totalVentas \* Porcentaje\_Utilidad.

float sueldo total = Sueldo\_Base + comisiones + utilidad.

Mostrar informe final del trabajador "x":

"Total de autos vendidos: ", autosVendidos.

"Valor total de las ventas: ", totalVentas.

"Sueldo base: ", Sueldo\_Base.

"Comisión total por autos vendidos: ", comisiones.

"Utilidad total por ventas: ", utilidad.

"Sueldo total a pagar: ", sueldoTotal.

Solicitar y leer: ¿Desea registrar otro trabajador?

guardar la respuesta en char resp\_trabajador

Mientras anidado 3

seguir las lógicas anteriores de verificación (s/n)

Fin mientras anidado 3

Si la respuesta es n, entonces:

continuar = false (cerrar el bucle principal)

Fin mientras principal

FIN.

## • Código en Java:

```
src > controller > ejercicios > AutosComp.java > AutosComp > ejecutar()
1 package controller.ejercicios;
2
3 import java.util.Scanner;
4
5 public class AutosComp {
6     /*
7      * Una compañía de autos usados paga $2,500.00 de sueldo a sus empleados por
8      * mes, Además agrega pagos extras a sus sueldos como:
9      * una comisión de $250.00 por cada automóvil vendido cuyo valor de venta supere
10     * el $10,000.00 y el 5% de utilidad del valor total de ventas.
11     * La compañía necesita emitir un informe desglosado por empleado donde indique
12     * el número total de autos vendidos, el valor total de los
13     * autos que vendió, el monto total que se le debería pagar al final del mes,
14     * indicando: el sueldo mensual, la comisión por cada automóvil
15     * y la utilidad por el valor total de la venta. También debe permitir ingresar
16     * la venta de los automóviles hasta que se desee y
17     * luego emitir el respectivo informe.
18     */
19
20     // Por Cael Soto
21
22     // Definir las constantes, estas no se modificaran por eso se usa la palabra final, esto es opcional pero es ético hacerlo en un ámbito
23     // más académico
24     final float Sueldo_Base = 2500.0f;
25     final float Comision_Auto = 250.0f;
26     final float Porcentaje_Utilidad = 0.05f;
27
28     public void ejecutar() {
29         Scanner sc = new Scanner(System.in); // llamamos al scanner al programa
30         boolean continuar = true; //variable booleana que determina la ejecución del proceso principal, da la green flag en proceder con el código
31         System.out.println("\nRegistro de ventas de automoviles:"); //Dar la bienvenida al programa
32
33         while (continuar) {
34             System.out.print(s:"Ingrese el nombre del trabajador: "); //pedir ingresar el nombre del empleado para brindarle su informe
35             String nombre = sc.nextLine(); // se usa nextLine() para capturar el nombre completo
36
37             //Variables para registrar ventas y comisiones
38             int autosVendidos = 0;
39             float totalVentas = 0.0f;
40             float comisiones = 0.0f;
41
42             //Bucle para registrar ventas hasta que el usuario decida parar
43             boolean registrarVentas = true; //boolean que determina hasta cuando parar las acciones de pedir valor de un nuevo auto y de registrar
44             //otra venta
45             while (registrarVentas) { //bucle destinado a calcular los valores como: nro de autos vendidos, total de ventas, comisiones, además,
46                 //también sirve para repetir si desee agregar otra venta
47                 System.out.print(s:"Ingrese el valor del automovil vendido: ");
48                 float valorVenta = sc.nextFloat();
49                 totalVentas += valorVenta;
50                 autosVendidos++;
51
52                 // Si el valor de la venta supera los $10,000, se aplica la comisión dispuesta por el problema
53                 if (valorVenta > 10000.0f) {
54                     comisiones += Comision_Auto; //guardar en cada interacción el valor de la comisión de los autos en comisiones
55                 }
56
57                 // Preguntar si desea registrar otra venta
58                 System.out.print(s:"¿Desea registrar otra venta? (s/n): ");
59                 char respuesta = sc.next().toLowerCase().charAt(index:0); //leer variable char llamada respuesta, esta sirve para registrar s/n
60                 while (respuesta != 's' && respuesta != 'n') { //bucle anidado que sirve en caso de escribir mal las respuestas s/n
61                     System.out.println(x:"Respuesta incorrecta. Por favor ingrese 's' para registrar otra venta o 'n' para finalizar.");
62                     System.out.print(s:"¿Desea registrar otra venta? (s/n): ");
63                     respuesta = sc.next().toLowerCase().charAt(index:0);
64                 }
65
66                 if (respuesta == 'n') {
67                     registrarVentas = false; //el usuario no quiere registrar otra venta
68                 }
69             }
70
71             //Calcular utilidad y sueldo total
72             float utilidad = totalVentas * Porcentaje_Utilidad;
73             float sueldoTotal = Sueldo_Base + comisiones + utilidad;
74
75             //Presentar el informe final para este trabajador
76             System.out.println("\nINFORME DE VENTAS DEL TRABAJADOR: " + nombre);
77             System.out.println(x:"-----"); //darle una mejor interfaz al informe
78             System.out.println("Total de autos vendidos: " + autosVendidos);
79             System.out.println("Valor total de las ventas: $" + totalVentas);
80             System.out.println("Sueldo base: $" + Sueldo_Base);
81             System.out.println("Comisión total por autos vendidos: $" + comisiones);
82             System.out.println("Utilidad total por ventas: $" + utilidad);
83             System.out.println("Sueldo total a pagar: $" + sueldoTotal);
84
85             // Preguntar si desea registrar otro trabajador, y repetir el proceso desde el inicio del programa
86             System.out.print(s:"¿Desea registrar otro trabajador? (s/n): ");
87             char resp_trabajador = sc.next().toLowerCase().charAt(index:0); //guardar la respuesta s/n
88             while (resp_trabajador != 's' && resp_trabajador != 'n') { //bucle para verificar la respuesta y pedir que ingrese correctamente
89                 System.out.println(x:"Respuesta incorrecta. Ingrese 's' para registrar otro trabajador o 'n' para finalizar.");
90                 System.out.print(s:"¿Desea registrar otro trabajador? (s/n): ");
91                 resp_trabajador = sc.next().toLowerCase().charAt(index:0);
92             }
93
94             if (resp_trabajador == 'n') {
95                 continuar = false; //Si no hay más trabajadores cerrar el programa
96             }
97
98             // Finalmente, hay que consumir el salto de línea pendiente del Scanner, pues de caso contrario al momento de agregar otro vendedor,
99             //ocurre un error y sobrepone el scanner de valor del automóvil
100             sc.nextLine(); //Para ello se usa sc.nextLine, lo que sirve para que no haya problemas con nextLine() después de next()
101         }
102     }
103 }
```



- Validación de funcionamiento:

```
Registro de ventas de automoviles:
Ingrese el nombre del trabajador: Pedro
Ingrese el valor del automovil vendido: 14537.34
¿Desea registrar otra venta? (s/n): s
Ingrese el valor del automovil vendido: 1524
¿Desea registrar otra venta? (s/n): h
Respuesta incorrecta. Por favor ingrese 's' para registrar otra venta o 'n' para finalizar.
¿Desea registrar otra venta? (s/n): n

INFORME DE VENTAS DEL TRABAJADOR: Pedro
-----
Total de autos vendidos: 2
Valor total de las ventas: $16061.34
Sueldo base: $2500.0
Comisión total por autos vendidos: $250.0
Utilidad total por ventas: $803.067
Sueldo total a pagar: $3553.067

¿Desea registrar otro trabajador? (s/n): t
Respuesta incorrecta. Ingrese 's' para registrar otro trabajador o 'n' para finalizar.
¿Desea registrar otro trabajador? (s/n): s
Ingrese el nombre del trabajador: Pepe
Ingrese el valor del automovil vendido: 12346
¿Desea registrar otra venta? (s/n): n

INFORME DE VENTAS DEL TRABAJADOR: Pepe
-----
Total de autos vendidos: 1
Valor total de las ventas: $12346.0
Sueldo base: $2500.0
Comisión total por autos vendidos: $250.0
Utilidad total por ventas: $617.3
Sueldo total a pagar: $3367.3

¿Desea registrar otro trabajador? (s/n): n
```

## 2) EJERCICIO 2:

- Requerimientos:

Datos de Entrada:

- La zona geográfica se debe identificar mediante una clave única.
- Cada zona tiene un costo por minuto en USD.

Se debe ingresar:

- La clave de la zona.
- El número de minutos hablados.

Datos de salida:

- Calcular el costo total de la llamada en función de la zona y los minutos hablados.
- Solicitar los datos de la llamada (clave y minutos).
- Mostrar un desglose del cálculo.
- Usar un tipo de datos enumerado (enum) para representar las zonas.

- Pseudocódigo:

INICIO:

importar Utilidades

Utilidades:

//validar la clave de la zona, si no corresponde, mostrar en la ejecución la  
invalidación

**Método booleano validarClaveZona (int clave):**

retorna clave == 12 o clave == 15 o clave == 18 o clave == 19 o clave == 23 o  
clave == 25 o clave == 29 o clave == 31 al ser llamada

**Fin del método validarClaveZona**

**Generar lista enum llamada Zona:**

AMERICA\_DEL\_NORTE(12, 2.75)

AMERICA\_CENTRAL(15, 1.89)

AMERICA\_DEL\_SUR(18, 1.60)

EUROPA(19, 3.50)

ASIA(23, 4.50)

AFRICA(25, 3.10)

OCEANIA(29, 3.00)

RESTO\_DEL\_MUNDO(31, 6.00)

**Definir variables:**

int clave;

float precioPorMinuto;

**Crear un constructor y asignar valores:**

Zona(int clave, float precioPorMinuto)

clave = clave

precioPorMinuto = precioPorMinuto

- **Método ejecutar:**

Imprimir zona y precio de cada una utilizando el enum Zona:

"AMERICA\_DEL\_NORTE - Clave: 12 - Precio por minuto: \$2.75"

"AMERICA\_CENTRAL - Clave: 15 - Precio por minuto: \$1.89"

"AMERICA\_DEL\_SUR - Clave: 18 - Precio por minuto: \$1.60"

"EUROPA - Clave: 19 - Precio por minuto: \$3.50"

"ASIA - Clave: 23 - Precio por minuto: \$4.50"

"AFRICA - Clave: 25 - Precio por minuto: \$3.10"

"OCEANIA - Clave: 29 - Precio por minuto: \$3.00"

"RESTO\_DEL\_MUNDO - Clave: 31 - Precio por minuto: \$6.00"

Imprimir: "Ingrese la clave de la zona geográfica:"

Leer claveZona

Validar claveZona (utilizando el método de validación de Utilidades):

**Si** claveZona no es válida:

Imprimir: "Error..."

Continuar con la siguiente iteración del bucle principal

**cerrar if**

crear variable float precioMinuto = 0.0f

Obtener precioMinuto utilizando el enum Zona:

cada case corresponde a la numeración del enum zona, asignar el precio de la zona a precioMinuto

Crear switch:

switch (claveZona):

case 12:

precioPorMinuto = 2.75

break

case 15:

precioPorMinuto = 1.89

break

case 18:

precioPorMinuto = 1.60

break

case 19:

precioPorMinuto = 3.50

break

case 23:

precioPorMinuto = 4.50

break

case 25:

precioPorMinuto = 3.10

break

case 29:

precioPorMinuto = 3.00

break

case 31:

precioPorMinuto = 6.00

break

Calcular costoTotal = precioMinuto \* minutos

Imprimir: "El costo de la llamada es: \$" + costoTotal

**Fin del método ejecutar**

FIN



## • Código Java:

```

1 package controller.ejercicios;
2
3 import controller.util.Utilidades; //importar la clase Utilidades para usar el método de validación (opcional)
4
5 import java.util.Scanner;
6
7 //Por Cael Soto
8
9 public class Llamadas { //creamos la clase principal Llamadas
10
11     public enum Zona { //declaramos un enum llamado Zona para representar las zonas geográficas por clave y por precio
12         AMERICA_DEL_NORTE(clave:12, precioPorMinuto:2.75f), //zona: américa del norte con clave 12 y precio por minuto 2.75
13         AMERICA_CENTRAL(clave:15, precioPorMinuto:1.89f), //zona: américa central con clave 15 y precio por minuto 1.89
14         AMERICA_DEL_SUR(clave:18, precioPorMinuto:1.60f), //zona: américa del sur con clave 18 y precio por minuto 1.60
15         EUROPA(clave:19, precioPorMinuto:3.5f), //zona: europa con clave 19 y precio por minuto 3.50
16         ASIA(clave:23, precioPorMinuto:4.5f), //zona: asia con clave 23 y precio por minuto 4.50
17         AFRICA(clave:25, precioPorMinuto:3.1f), //zona: áfrica con clave 25 y precio por minuto 3.10
18         OCEANIA(clave:29, precioPorMinuto:3.0f), //zona: oceania con clave 29 y precio por minuto 3.00
19         RESTO_DEL_MUNDO(clave:31, precioPorMinuto:6.0f); //zona: resto del mundo con clave 31 y precio por minuto 6.00
20
21         //Las constantes del enum como AMERICA_DEL_NORTE, EUROPA, etc., tienen asociados dos valores: clave, que es el identificador numérico de
22         //la zona, y precioPorMinuto que es el costo por minuto para llamar a esa zona.
23
24         int clave; //definir una variable para almacenar la clave de la zona
25         float precioPorMinuto; //definir una variable para almacenar el precio por minuto de la zona
26
27         //this.clave y this.precioPorMinuto son las variables de instancia del enum, o sea, hace referencia a la instancia actual del enum
28         Zona(int clave, float precioPorMinuto) { //crear un constructor del enum que inicializa clave y precioPorMinuto
29             this.clave = clave; //asigna el valor de clave al atributo de la zona
30             this.precioPorMinuto = precioPorMinuto; //asigna el valor del precio por minuto al atributo de la zona
31         }
32     }
33
34     public void ejecutar() { //método principal que ejecutará el main
35         Scanner scanner = new Scanner(System.in); //se llama al Scanner para leer la entrada del usuario
36         System.out.println(x:"\nCosto de llamadas telefónicas Internacionales:");
37
38         Zona[] zonas = Zona.values(); //ahora se debe convertir el enum Zona en un arreglo para poder mostrar la lista enum
39         System.out.println(x:"A continuación, se mostraran las zonas geograficas disponibles:"); //mostrar las zonas disponibles
40         for (int i = 0; i < zonas.length; i++) { //bucle for para poder interactuar sobre las zonas, zonas.length nos da el número total de zonas
41             //definidas en el enum
42             Zona zona = zonas[i]; //obtenemos la zona actual en cada iteración, que corresponde a una constante del enum Zona
43
44             //se muestra la información de cada zona llamando a cada elemento del enum, zona.name: el nombre: zona.clave al zona y zona.precioPor
45             //minuto al precio por minuto de dicha zona
46             System.out.println(zona.name() + " - Clave: " + zona.clave + " - Precio por minuto: $" + zona.precioPorMinuto);
47         }
48
49         System.out.println(x:"\ningrese la clave de la zona geografica: "); //solicita al usuario la clave de la zona
50         int claveZona = scanner.nextInt(); //lee la clave ingresada por el usuario
51
52         if (!Utilidades.validarClaveZona(claveZona)) { //verifica si la clave ingresada es válida usando el método de Utilidades
53             System.out.println(x:"Querido usuario, la clave de zona es inválida"); //muestra un mensaje si la clave es inválida
54             return; //termina la ejecución del método si la clave es inválida
55         }
56
57         System.out.println(x:"\ningrese el numero de minutos de la llamada: "); //solicita al usuario los minutos de la llamada
58         int minutos = 0; //creamos la variable minutos
59
60         //validamos que el número de minutos ingresado sea un valor entero positivo con .hasNextInt para saber si lo que el usuario ingresa es
61         //un número entero
62         if (scanner.hasNextInt()) {
63             minutos = scanner.nextInt(); //si se valida, entonces el usuario asigna el valor ingresado a minutos
64         } else {
65             System.out.println(x:"Querido usuario, por favor ingrese un numero entero para los minutos"); //si no se ingresó un número válido
66             //mostramos error
67             return; //termina la ejecución del método si no es un número entero
68         }
69
70         if (minutos <= 0) { //opcionalmente también crearemos un verificador por si los minutos son negativos o cero
71             System.out.println(x:"Querido usuario, el numero de minutos debe ser mayor que cero");
72             return; //termina la ejecución si los minutos no son validos
73         }
74
75         float precioMinuto = 0.0f;
76
77         switch (claveZona) { //usa un switch para determinar el precio por minuto según la clave de la zona
78             case 12: //si la clave es 12
79                 precioMinuto = 2.75f; //asigna el precio de américa del norte
80                 break; //finaliza este caso
81             case 15: //si la clave es 15
82                 precioMinuto = 1.89f; //asigna el precio de américa central
83                 break; //finaliza este caso
84             case 18: //si la clave es 18
85                 precioMinuto = 1.60f; //asigna el precio de américa del sur
86                 break; //finaliza este caso
87             case 19: //si la clave es 19
88                 precioMinuto = 3.5f; //asigna el precio de europa
89                 break; //finaliza este caso
90             case 23: //si la clave es 23
91                 precioMinuto = 4.5f; //asigna el precio de asia
92                 break; //finaliza este caso
93             case 25: //si la clave es 25
94                 precioMinuto = 3.1f; //asigna el precio de áfrica
95                 break; //finaliza este caso
96             case 29: //si la clave es 29
97                 precioMinuto = 3.0f; //asigna el precio de oceania
98                 break; //finaliza este caso
99             case 31: //si la clave es 31
100                 precioMinuto = 6.0f; //asigna el precio del resto del mundo
101                 break; //finaliza este caso
102         }
103
104         float costoTotal = precioMinuto * minutos; //calcula el costo total de la llamada multiplicando el precio por minuto por los minutos
105         System.out.println("El costo de la llamada es: $" + costoTotal); //muestra el costo total de la llamada al usuario
106     }
107 }
108

```

### Apartado de utilidades:

```
77 //Opcional: método para validar que la clave de zona esté en el rango permitido
78 public static boolean validarClaveZona(int clave) {
79     //claves válidas de zona:
80     return clave == 12 || clave == 15 || clave == 18 || clave == 19 || clave == 23 || clave == 25 || clave == 29 || clave == 31;
81 }
82 }
```

- Validación:

Costo de llamadas telefónicas Internacionales:

A continuación, se mostraran las zonas geograficas disponibles:

AMERICA\_DEL\_NORTE - Clave: 12 - Precio por minuto: \$2.75

AMERICA\_CENTRAL - Clave: 15 - Precio por minuto: \$1.89

AMERICA\_DEL\_SUR - Clave: 18 - Precio por minuto: \$1.6

EUROPA - Clave: 19 - Precio por minuto: \$3.5

ASIA - Clave: 23 - Precio por minuto: \$4.5

AFRICA - Clave: 25 - Precio por minuto: \$3.1

OCEANIA - Clave: 29 - Precio por minuto: \$3.0

RESTO\_DEL\_MUNDO - Clave: 31 - Precio por minuto: \$6.0

Ingrese la clave de la zona geografica:

19

Ingrese el numero de minutos de la llamada:

rt

Querido usuario, por favor ingrese un numero entero para los minutos

Costo de llamadas telefónicas Internacionales:

A continuación, se mostraran las zonas geograficas disponibles:

AMERICA\_DEL\_NORTE - Clave: 12 - Precio por minuto: \$2.75

AMERICA\_CENTRAL - Clave: 15 - Precio por minuto: \$1.89

AMERICA\_DEL\_SUR - Clave: 18 - Precio por minuto: \$1.6

EUROPA - Clave: 19 - Precio por minuto: \$3.5

ASIA - Clave: 23 - Precio por minuto: \$4.5

AFRICA - Clave: 25 - Precio por minuto: \$3.1

OCEANIA - Clave: 29 - Precio por minuto: \$3.0

RESTO\_DEL\_MUNDO - Clave: 31 - Precio por minuto: \$6.0

Ingrese la clave de la zona geografica:

45

Querido usuario, la clave de zona es invalida

Costo de llamadas telefónicas Internacionales:

A continuación, se mostraran las zonas geograficas disponibles:

AMERICA\_DEL\_NORTE - Clave: 12 - Precio por minuto: \$2.75

AMERICA\_CENTRAL - Clave: 15 - Precio por minuto: \$1.89

AMERICA\_DEL\_SUR - Clave: 18 - Precio por minuto: \$1.6

EUROPA - Clave: 19 - Precio por minuto: \$3.5

ASIA - Clave: 23 - Precio por minuto: \$4.5

AFRICA - Clave: 25 - Precio por minuto: \$3.1

OCEANIA - Clave: 29 - Precio por minuto: \$3.0

RESTO\_DEL\_MUNDO - Clave: 31 - Precio por minuto: \$6.0

Ingrese la clave de la zona geografica:

29

Ingrese el numero de minutos de la llamada:

45

El costo de la llamada es: \$135.0

### 3) EJERCICIO 3:

- Requerimientos del programa:

Datos de entrada:

- Cantidad de agua consumida en m<sup>3</sup>.
- Si el contribuyente pertenece a la tercera edad y su consumo está entre 0 hasta 15 m<sup>3</sup> se le brinda un descuento del 50%, en caso de exceder sólo se realizará el descuento del 30%.
- Si es discapacitado su porcentaje se tomará como descuento su porcentaje de discapacidad

Cálculos necesarios:

- Servicio de Agua Potable: Según la tabla de consumo.
- Impuesto de Alcantarillado: 35% del valor del rubro de servicio de agua potable.

- Tasa de Recolección de Basura: \$0.75.
- Tasa de Procesamiento de Datos: \$0.50.
- Descuento por tercera edad o discapacidad: Aplicable solo al rango base del servicio de agua potable.

Datos de salida:

- Importe total desglosado:
- Servicio de Agua Potable.
- Impuesto de Alcantarillado.
- Tasas adicionales.
- Total a pagar después de aplicar los descuentos.

- **Pseudocódigo:**

INICIO:

Declarar las siguientes variables:

- double consumo (para almacenar el consumo de agua).
- char respuesta (para almacenar la respuesta del usuario si es o no de la tercera edad o tiene discapacidad).
- double costoAgua (para almacenar el costo calculado del agua potable).
- double impuestoAlcantarillado (para almacenar el impuesto de alcantarillado calculado).
- double totalFactura (para almacenar el total a pagar calculado).

Declarar constantes ():

- final double tasaBasura (tasa fija de recolección de basura).
- final double tasaProcesamiento (tasa fija por procesamiento de datos).

Solicitar y leer: Ingrese el consumo de agua en m3

**Validar la respuesta sobre si el usuario tiene descuento:**

**Mientras(while)** (respuesta no es "s" ni "n")

Imprimir "¿El usuario es de la tercera edad? (s/n):"

Leer respuesta

**cerrar bucle**

Imprimir "¿El usuario tiene discapacidad? (s/n)"

Leer respuestaDiscapacidad

**Aplicar descuento por tercera edad si es necesario:**

**Si(if)** respuesta es "s", entonces:

aplicarDescuento = 0.50

**Sino(else):**

aplicarDescuento = 0.00

**Fin(if)**

**Aplicar descuento por discapacidad si es necesario:**

**Si** (respuestaDiscapacidad es "s") Entonces:

Imprimir "Ingrese el porcentaje de discapacidad"

Leer porcentajeDiscapacidad

descuentoDiscapacidad = costoServicio \* porcentajeDiscapacidad / 100

costoServicio = costoServicio - descuentoDiscapacidad

**Fin(if)**

costoServicio = Llamar a CalcularCosto

impuesto = costoServicio \* 0.35

totalFactura = costoServicio + impuesto + tasaBasura + tasaProcesamiento

**Mostrar la factura:**

Mostrar "Factura Detallada:"

Mostrar "Costo del servicio: \$" + Redondear(costoServicio, 2)

Mostrar "Impuesto: \$" + Redondear(impuesto, 2)

Mostrar "Tasa adicional: \$" + tasaBasura

Mostrar "Otro cargo adicional: \$" + tasaProcesamiento

Mostrar "Total a Pagar: \$" + Redondear(totalFactura, 2)

**Calcular el costo del agua potable:**

Función CalcularCosto(consumo)

**Si** (consumo <= límite1) Entonces:

costoServicio = precioBase

**SinoSi** (consumo <= límite2) Entonces:

costoServicio = precioBase + (consumo - límite1) \* tarifaAdicional1

**SinoSi** (consumo <= límite3) Entonces:

costoServicio = precioBase + (límite2 - límite1) \* tarifaAdicional1 + (consumo - límite2) \* tarifaAdicional2

**Sino** costoServicio = precioBase + (límite2 - límite1) \* tarifaAdicional1 + (límite3 - límite2) \* tarifaAdicional2 + (consumo - límite3) \* tarifaAdicional3

**FinSi**

Retornar costoServicio

FIN

- Código Java:

```
src > controller > ejercicios > CalculadoraFactura.java > CalculadoraFactura > ejecutar()
1 package controller.ejercicios;
2
3 import java.util.Scanner;
4
5 public class CalculadoraFactura {
6     /*3. La empresa municipal de agua potable de Loja desea cobrar y calcular mensualmente el valor exacto de consumo del agua potable de
7     cada medidor que pertenece a un contribuyente, de acuerdo a la ordenanza vigente, la cual establece la planilla de acuerdo a los
8     siguientes rubros:
9     A) SERVICIO DE AGUA POTABLE: Para obtener el rubro rijase a la siguiente tabla
10    B) IMPUESTO DE ALCANTARILLADO: 35% DEL VALOR DEL RUBRO DEL SERVICIO DE AGUA POTABLE.
11    C) TASA POR RECOLECCIÓN DE BASURA: 0.75 DÓLARES;
12    D) TASA POR COSTO DE PROCESAMIENTO DE DATOS: 0.50 DÓLARES
13
14    La ordenanza también permite descuentos para los medidores que estén registrados al contribuyente de la tercera edad o posean algún
15    tipo discapacidad, para lo cual sólo se los puede aplicar sobre el rubro del SERVICIO DE AGUA POTABLE, en base a las siguientes
16    condiciones.
17    (a) Si pertenece a la tercera edad y su consumo esta entre 0 hasta 15 m³ (rango base) existe un descuento del 50%, en caso de exceder
18    sólo se realizará el descuento del 30% sobre el rango base.
19    (b) Si pertenece a un discapacitado, se tomará como descuento su porcentaje de discapacidad sólo sobre el rango base.*/
20
21    //Por Cael Soto
22
23    public void ejecutar() {
24        Scanner sc = new Scanner(System.in); //llamar al scanner para registrar via teclado lo necesario
25        // Definir variables y constantes
26        final double tasaBasura = 0.75;
27        final double tasaProcesamiento = 0.50;
28        int nroMedidores;
29        double consumo;
30        char respuesta;
31        double costoAgua;
32        double impAlcantarillado;
33        double totalFactura;
34
35        System.out.println(x:"\nCalcular la factura de consumo de agua potable:"); //dar la bienvenida
36
37        while (true) { //validamos número de medidores a través de un bucle while
38            System.out.println(x:"Ingrese el numero de medidores:");
39
40            //a continuación se muestran 2 casos if, el principal que valida que sea un número entero, y el segundo que valida que sea
41            //positivo
42            if (sc.hasNextInt()) { //entero
43                nroMedidores = sc.nextInt();
44
45                if (nroMedidores > 0) { //positivo
46                    break; //salir del bucle, o sea continuar la ejecución del programa si el consumo es válido
47                } else {
48                    System.out.println(x:"Querido usuario el consumo de agua debe ser un numero positivo. Vuelva a intentarlo");
49                }
50            } else {
51                System.out.println(x:"Error: Debe ingresar un numero entero. Vuelva a intentarlo");
52                sc.next();
53            }
54            //usamos esto para eliminar el valor incorrecto que quedó en el buffer para evitar errores en la siguiente entrada
55        }
56
57        //bucle para calcular valores de cada medidor, usamos for porque ya sabemos el nro de medidores
58        for (int i = 1; i <= nroMedidores; i++) {
59            System.out.println("\nMedidor nro. " + i); //mostrar el medidor del cual se piden los datos
60
61            while (true) { //validar consumo de agua y que se ingrese correctamente
62                System.out.println(x:"Ingrese el consumo de agua en m3 (debe ser un número positivo):");
63
64                //al igual que la validación anterior del nro de medidores, usamos casos if para validar el nro de consumo (en double)
65                //y que este numero sea positivo
66                if (sc.hasNextDouble()) { //Nro en double
67                    consumo = sc.nextDouble();
68
69                    if (consumo > 0) { //positivo
70                        break; //salir del bucle, o sea continuar la ejecución del programa si el consumo es válido
71                    } else {
72                        System.out.println(x:"Querido usuario el consumo de agua debe ser un numero positivo. Vuelva a intentarlo");
73                    }
74                } else {
75                    System.out.println(x:"Error: Debe ingresar un numero valido para el consumo. Vuelva a intentarlo");
76                    sc.next();
77                }
78                //usamos esto para eliminar el valor incorrecto que quedó en el buffer para evitar errores en la siguiente entrada
79            }
80        }
81    }
}
```

```

82
83 // Validar si el usuario tiene tercera edad o discapacidad a través de un bucle while
84 while (true) {
85     System.out.println(x:"¿El usuario es de la tercera edad? (s/n):");
86     respuesta = sc.next().trim().toLowerCase().charAt(index:0); //guardar respuesta como char
87
88     if (respuesta == 's' || respuesta == 'n') { // validar que se escribió ya sea s o n
89         break; //continuar con la ejecución si la respuesta es válida
90     }
91     } else {
92         System.out.println(x:"Queridísimo usuario, debe ingresar 's' para sí o 'n' para no. Vuelva a intentarlo");
93     }
94 }
95
96 System.out.println(x:"¿El usuario tiene discapacidad? (s/n):"); // Preguntar si el usuario tiene discapacidad
97 char tieneDiscapacidad = sc.next().trim().toLowerCase().charAt(index:0); //guardar en char
98
99 costoAgua = calcularCostoAgua(consumo); //calcular el costo base del agua
100
101 //aplicar descuento por tercera edad
102 if (respuesta == 's') { //si el usuario tiene tercera edad
103     if (consumo <= 15) { //si el consumo está en el rango base de 0 a 15 m³
104         costoAgua *= 0.5; //aplicar el descuento del 50% si el consumo está entre 0 y 15 m³
105     } else { //si el consumo excede los 15 m³
106         double costoBase = calcularCostoAgua(consumo:15); //obtener el costo de los primeros 15 m³
107         double exceso = consumo - 15; //calcular la cantidad de m³ que excede los 15 m³
108         costoAgua = costoBase * 0.7 + calcularCostoAgua(exceso); //aplicar el descuento del 30% solo sobre los primeros 15 m³
109     }
110 }
111
112 //aplicar descuento por discapacidad
113 if (tieneDiscapacidad == 's') { //si el usuario tiene discapacidad
114     System.out.println(x:"Ingrese el porcentaje de discapacidad (por ejemplo, 30 para referirte al 30%):");
115     double porcentajeDiscapacidad = sc.nextDouble(); //solicitar el porcentaje de discapacidad
116     double descuentoDiscapacidad = calcularCostoAgua(consumo:15) * (porcentajeDiscapacidad / 100); //calcular descuento sobre
117     //los 15 m³
118     costoAgua -= descuentoDiscapacidad; //reducir el costo base por el descuento de discapacidad
119 }
120
121 //realizar los cálculos solicitados por la factura
122 impAlcantarillado = costoAgua * 0.35; // Impuesto de alcantarillado (35% del costo del agua)
123 totalFactura = costoAgua + impAlcantarillado + tasaBasura + tasaProcesamiento; // Total de la factura
124
125 //presentar la factura con los datos redondeados
126 System.out.printf(format:"\nSU FACTURA DE AGUA POTABLE PARA EL MEDIDOR #%d ES LA SIGUIENTE:\n", i);
127 System.out.println(x:"-----");
128 System.out.printf(format:"Servicio de Agua Potable: $%.2f\n", costoAgua);
129 System.out.printf(format:"Impuesto de Alcantarillado: $%.2f\n", impAlcantarillado);
130 System.out.printf(format:"Tasa por Recolección de Basura: $%.2f\n", tasaBasura);
131 System.out.printf(format:"Tasa por Procesamiento de Datos: $%.2f\n", tasaProcesamiento);
132 System.out.printf(format:"Total a pagar: $%.2f\n", totalFactura);
133 }
134
135 //método para calcular el costo del agua en base a la tabla proporcionada por el problema
136 private double calcularCostoAgua(double consumo) {
137     double costo; //definir variable costo para aplicar la tabla
138
139     if (consumo <= 15) { //casos en base a la tabla, que determinan el costo en base al consumo de agua
140         costo = 3.00; // 0-15 m³ tiene un costo fijo de 3.00
141     } else if (consumo <= 25) {
142         costo = 3.00 + (consumo - 15) * 0.10; // el consumo que exceda los 15 m³ se cobra a $0.10 por m³
143     } else if (consumo <= 40) {
144         costo = 3.00 + (10 * 0.10) + (consumo - 25) * 0.20; // los siguientes 10 m³ a $0.20 por m³
145     } else if (consumo <= 60) {
146         costo = 3.00 + (10 * 0.10) + (15 * 0.20) + (consumo - 40) * 0.30; // los siguientes 15 m³ a $0.30 por m³
147     } else {
148         costo = 3.00 + (10 * 0.10) + (15 * 0.20) + (20 * 0.30) + (consumo - 60) * 0.35; // el exceso a 60 m³ se cobra a $0.35 por m³
149     }
150
151     return costo; //devolver costo al método principal
152 }
153
154 }

```

## • Ejecución:

```

Ingrese el numero de medidores:
2
Medidor nro. 1
Ingrese el consumo de agua en m3 (debe ser un número positivo):
34
Querido usuario el consumo de agua debe ser un número positivo. Vuelva a intentarlo
Ingrese el consumo de agua en m3 (debe ser un número positivo):
15
Error: Debe Ingresar un numero valido para el consumo. Vuelva a intentarlo
Ingrese el consumo de agua en m3 (debe ser un número positivo):
30
¿El usuario es de la tercera edad? (s/n):
n
¿El usuario tiene discapacidad? (s/n):
n
SU FACTURA DE AGUA POTABLE PARA EL MEDIDOR #1 ES LA SIGUIENTE:
-----
Servicio de Agua Potable: $5.10
Impuesto de Alcantarillado: $1.78
Tasa por Recolección de Basura: $0.75
Tasa por Procesamiento de Datos: $0.50
Total a pagar: $8.14
Medidor nro. 2
Ingrese el consumo de agua en m3 (debe ser un número positivo):
45
¿El usuario es de la tercera edad? (s/n):
n
¿El usuario tiene discapacidad? (s/n):
s
Ingrese el porcentaje de discapacidad (por ejemplo, 30 para referirte al 30%):
90
SU FACTURA DE AGUA POTABLE PARA EL MEDIDOR #2 ES LA SIGUIENTE:
-----
Servicio de Agua Potable: $5.80
Impuesto de Alcantarillado: $2.03
Tasa por Recolección de Basura: $0.75
Tasa por Procesamiento de Datos: $0.50
Total a pagar: $9.08

```

#### 4) EJERCICIO 4:

- **Requerimientos del programa:**

Datos de entrada:

Reconocer la siguiente serie en base a lógica de cada elemento:

$(1 / 1)^2 + (1 / 3)^4 - (2 / 5)^6 - (3 / 7)^8 + (5 / 9)^{10} + (8 / 11)^{12} \dots$

La lógica es:

numerador: Fibonacci

denominador: número impar

signo: +---+--- etc

potencia: suma +2

El usuario debe ingresar el numero de términos que quiera para generar la serie

Datos de salida:

Generar la secuencia en base la cantidad de términos requeridos por el usuario

Generar la secuencia en base a la lógica de cada elemento

- **Pseudocódigo:**

INICIO

**Funciones auxiliares:**

- Crear una función auxiliar que genere impares para el denominador de la serie, este se rige en base al parámetro: índice

**Función generar\_impar(int indice):**

Retornar  $2 * \text{indice} - 1$

Esta función calcula un número impar dado un índice, utilizando la fórmula:

$2 \cdot \text{índice} - 1$

Ejemplo: Para índice = 1, impar =  $2 \cdot 1 - 1 = 1$

- Función auxiliar que genera la secuencia Fibonacci, parámetro: índice  
Este auxiliar calcula el número de Fibonacci correspondiente al índice. (la secuencia es: 1,1,2,3,5,...).

**Función int generar\_fibonacci(int indice):**

**Si** (índice == 1 o índice == 2) Entonces: Retornar 1

Definir a = 1, b = 1

**Para**  $i = 2$  ;  $i$  menor a indice hacer:

Definir  $\text{int aux} = a$

$a = b$

$b = \text{aux} + b$

**Fin Para**

Retornar  $b$

**Fin Si**

**Inicio del programa:** Método ejecutar:

**Solicitar el número de términos de la serie y convertirlo en un entero.**

Imprimir "Ingrese el número de términos de la serie:". Leer número de términos de la serie → Guardar en la variable String numero.

Convertir numero (String) a entero → Guardar en  $\text{int num\_serie}$ .

**Si**  $\text{num\_serie}$  menor o igual a 0, imprimir "Ingrese un número mayor a 0" y finalizar.

**Si**  $\text{num\_serie}$  mayor a 0, seguir con la ejecución:

Definir variables:

Definir variable String serie = "S = ".

$\text{float suma} = 0.0$ : Acumula el resultado de la serie

$\text{Int signo} = 1$ : Controla si el término es positivo (+1) o negativo (-1)

$\text{Int potencia} = 2$ : Indica la potencia de cada término

$\text{Int cont} = 0$ : Contador para alternar el signo después de dos términos consecutivos.

**Cálculo de la serie:**

**Para(for)**  $\text{int } i = 1$ ;  $i$  menor o igual a  $\text{num\_serie}$ , calcular:

- Llamar a  $\text{generar\_fibonacci}(i)$  → Guardar en  $\text{int fibonacci}$
- Llamar a  $\text{generar\_impar}(i)$  → Guardar en  $\text{int impar}$
- Definir  $\text{double fraccion} = \text{fibonacci} / \text{impar}$
- Definir  $\text{double fraccion\_potenciada} = \text{fraccion}^{\text{potencia}}$

**Si(if):** signo es positivo:

Agregar a serie: "+ ( $\text{fibonacci} / \text{impar}$ )<sup>potencia</sup> "

Guardar suma en  $\text{suma}(+=)$ :  $\text{fracción\_potenciada}$

**Caso contrario(else):** signo es negativo

Agregar a serie: "- ( $\text{fibonacci} / \text{impar}$ )<sup>potencia</sup> "

Guardar resta en  $\text{suma}(=)$ :  $\text{fracción\_potenciada}$



cerrar if

Por cada interacción:

contador irá sumando 1 por interacción (i++)

potencia almacenará 2 en 2 en cada interacción (potencia += 2)

Si: contador es igual a 2, entonces:

Cambiar signo a -signo (alternar signo)

reiniciar contador a 0

cerrar if

Incrementar potencia en 2 (potencia += 2)

cerrar for

Imprimir la serie es: presentar: +serie

Imprimir la suma total es: presentar: +suma

FIN

- Código Java:

```
1 package controller.ejercicios;
2
3 import java.util.Scanner;
4
5 //Por Cael Soto
6 //Serie: (1 / 1)^2 + (1 / 3)^4 - (2 / 5)^6 - (3 / 7)^8 + (5 / 9)^10 + (8 / 11)^12 ...
7
8 public class Serie2 {
9
10     //método para generar el número Fibonacci
11     private int generar_fibonacci(int indice) {
12         if (indice == 1 || indice == 2) return 1; //los dos primeros números de Fibonacci son 1 segun la cadena
13         int a = 1, b = 1;
14         for (int i = 2; i < indice; i++) {
15             int aux = a;
16             a = b;
17             b = aux + b;
18         }
19         return b; //retorna el Fibonacci en el índice dado
20     }
21
22     //método para generar los números impares
23     private int generar_impar(int indice) {
24         return 2 * indice - 1; // Genera números impares 1, 3, 5, 7, ... etc
25     }
26
27     //método principal que ejecuta el cálculo de la serie
28     public void ejecutar() {
29         Scanner sc = new Scanner(System.in); // Scanner para leer la entrada
30         System.out.println(x:"\nPrograma para generar la serie de tipo: (1 / 1)^2 + (1 / 3)^4 - (2 / 5)^6 - (3 / 7)^8 + (5 / 9)^10 + (8 / 11)^12 ...");
31         System.out.print(s:"Ingrese el número de términos que desea generar: ");
32         String numero = sc.nextLine();
33
34         //se debe convertir la entrada a entero usando un método de validación y transformación
35         int num_serie = Integer.parseInt(numero);
36
37         //if general que verifica si el número de términos es válido
38         if (num_serie > 0) {
39             String serie = "S = "; //es lo que inicia la cadena de la serie
40             double suma = 0.0; //variable que se utilizará para la suma total de la serie
41             int signo = 1; //variable que determina el signo, además, el signo de inicio es positivo
42             int potencia = 2; //la potencia de la secuencia comienza en 2
```

```

42 int potencia = 2; //la potencia de la secuencia comienza en 2
43 int cont = 0; //contador para alternar el signo
44
45 // Bucle para generar los términos de la serie
46 for (int i = 1; i <= num_serie; i++) {
47     int fibonacci = generar_fibonacci(i); // Obtener el número Fibonacci correspondiente
48     int impar = generar_impar(i); // Obtener el número impar correspondiente
49     double fraccion = (double) fibonacci / (double) impar; //el numerador de la fracción debe ser el Fibonacci correspondiente
50     double fraccion_potenciada = Math.pow(fraccion, potencia); //elevar la fracción a la potencia
51
52     //realizar la secuencia correspondiente segun el caso
53     if (signo == 1) {
54         serie += "(" + fibonacci + "/" + impar + ")"^" + potencia + " ";
55         suma += fraccion_potenciada;
56     } else {
57         serie += "(" + fibonacci + "/" + impar + ")"^" + potencia + " ";
58         suma -= fraccion_potenciada;
59     }
60
61     cont++; //incrementar el contador en cada interacción
62     potencia += 2; //incrementar la potencia por 2 en cada iteración
63
64     //cuando el contador llegue a 2, se debe alternar el signo y reiniciar el contador para repetir el proceso
65     if (cont == 2) {
66         signo = -signo; //alternar el signo
67         cont = 0; //reiniciar el contador
68     }
69 }
70
71 //mostrar la serie y el resultado final
72 System.out.println("La serie es: " + serie);
73 System.out.println("El resultado es: " + suma);
74 } else {
75     System.out.println("Ingrese un número mayor a 0."); //en caso de que el usuario ponga un numero negativo o 0 a la cadena
76 }
77 }
78 }

```

- Validación:

Seleccione el ejercicio:

1. Autos usados / Informe del sueldo
2. Costo de las llamadas telefónicas internacionales
3. Cobro de Agua potable
4. Serie
5. Terminar programa

4

Programa para generar la serie de tipo:  $(1/1)^2 + (1/3)^4 - (2/5)^6 - (3/7)^8 + (5/9)^{10} + (8/11)^{12}$  ?

Ingrese el número de términos que desea generar: 6

La serie es:  $S = (1/1)^2 + (1/3)^4 - (2/5)^6 - (3/7)^8 + (5/9)^{10} + (8/11)^{12}$

El resultado es: 1.031808462100441

PS C:\Users\CAEL\Desktop\practicas\_programacion\ProyectoJava\PRIMERO>

Programa para generar la serie de tipo:  $(1/1)^2 + (1/3)^4 - (2/5)^6 - (3/7)^8 + (5/9)^{10} + (8/11)^{12}$  ?

Ingrese el número de términos que desea generar: -6

Ingrese un número mayor a 0.

Programa para generar la serie de tipo:  $(1/1)^2 + (1/3)^4 - (2/5)^6 - (3/7)^8 + (5/9)^{10} + (8/11)^{12}$  ?

Ingrese el número de términos que desea generar: 0

Ingrese un número mayor a 0.

## 5) VIDEO DEMOSTRATIVO:

Link:

<https://drive.google.com/file/d/1sZG5yPJlqwKFzqScVINup7vIG9eqkuSS/view?usp=sharing>

### Preguntas de ensayo:

#### 1. Las estructuras de control repetitivas me permiten realizar saltos en un programa (Si o No)?.

No, no están diseñadas para realizar saltos en un programa.

#### 2. Desarrolle un ejemplo de la estructura condicional MIENTRAS con CONTADORES en Pseudo-código y en

### lenguaje de programación C?

Pseudo-código: Contador del 0 al 7

INICIO

Int contador  $\leftarrow$  0 //iniciar variable del contador

Mientras (contador  $\leq$  7) Hacer:

Mostrar "El contador es: ", contador

contador  $\leftarrow$  contador + 1 //incrementar el contador por interacción hasta cumplir la condición

Fin Mientras

FIN

Código en JAVA:

```
src > controller > ejercicios > J Mientras.java > Mientras
1 package controller.ejercicios;
2
3 //Por Cael Soto
4
5 public class Mientras {
6     public static void main(String[] args) {
7         int contador = 0;
8
9         //Bucle mientras
10        while (contador <= 7) {
11            System.out.println("El contador es: " + contador);
12            contador++;
13        }
14    }
15 }
```

PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\CAEL\Desktop\practicas_programacion\ProyectoJava\PRIMERO> & 'C:\jdk\bin\java.exe' '-cp' 'C:\Users\CAEL\Desktop\practicas_programacion\ProyectoJava\PRIMERO\bin' 'controller.ejercicios.Mientras'
El contador es: 0
El contador es: 1
El contador es: 2
El contador es: 3
El contador es: 4
El contador es: 5
El contador es: 6
El contador es: 7
PS C:\Users\CAEL\Desktop\practicas_programacion\ProyectoJava\PRIMERO>
```

3. Desarrolle un ejemplo de la estructura condicional HACER ... MIENTRAS (REPETIR .. MIENTRAS) con BANDERAS en Pseudo-código y en lenguaje de programación C?

Pseudo-código:

INICIO

leer variables:

boolean bandera  $\leftarrow$  Falso

Int intentos  $\leftarrow$  1

Repetir(do)

Mostrar "Intento número: ", intentos

Si intentos = 6 Entonces

bandera  $\leftarrow$  Verdadero

Fin Si

intentos  $\leftarrow$  intentos + 1 //contador

Mientras/while (bandera = Falso)

FIN

Código en JAVA:

```
src > controller > ejercicios > J Hacermientras.java > Hacermientras
1 package controller.ejercicios;
2
3 //Por Cael Soto
4
5 public class Hacermientras {
6     public static void main(String[] args) {
7         boolean bandera = false;
8         int intentos = 1; //de este valor inician los intentos
9
10        //Bucle HACER...MIENTRAS
11        do {
12            System.out.println("Intento número: " + (intentos)); //mostrar el número de intento por interacción
13
14            if (intentos == 6) { //cambiar la bandera después de x intentos que se quiera poner, en este caso 3
15                bandera = true;
16            }
17
18            intentos++; //incrementar el contador por interacción
19        } while (!bandera); //continuar mientras la bandera sea falsa
20    }
21 }
22 }
```

PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\CAEL\Desktop\practicas_programacion\ProyectoJava\PRIMERO> & 'C:\jdk\bin\java.exe' '-cp' 'C:\Users\CAEL\Desktop\practicas_programacion\ProyectoJava\PRIMERO\bin' 'controller.ejercicios.Hacermientras'
Intento número: 1
Intento número: 2
Intento número: 3
Intento número: 4
Intento número: 5
Intento número: 6
PS C:\Users\CAEL\Desktop\practicas_programacion\ProyectoJava\PRIMERO>
```

4. Conteste verdadero o falso a lo siguiente. La estructura repetitiva HACER ... MIENTRAS (REPETIR .. MIENTRAS) permite iterar el contenido de su bloque de código ninguna o muchas veces.

Verdadero

5. Defina que es una función o subprograma? (0.25)

Una función o subprograma es un bloque de código que se define con un nombre único y puede aceptar parámetros de entrada y devolver un valor. Estas funciones son reutilizables y realizan una tarea específica, además de que ayudan a estructurar el programa en módulos más pequeños y manejables.

6. Por defecto en una función en C el paso de parámetros son por valor para los tipos datos simples. (Verdadero o Falso).

Verdadero

7. Desarrolle un ejemplo pequeño en lenguaje de programación en C donde demuestre la utilización de funciones con paso de parámetros por valor? (0.5)

```
C funcionvalor.c > main()
1 #include <stdio.h>
2
3 //Por Cael Soto
4
5 void duplicar(int w) {
6     w = w * 2;
7     printf("valor dentro de la función: %d\n", w); //en teoría es 13
8 }
9
10 int main() {
11     int num = 13;
12     printf("valor antes de la función: %d\n", num);
13     duplicar(num); //en teoría debe ser 26
14
15     printf("valor después de la función: %d\n", num);
16     return 0; //en teoría debe seguir como 13
17 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\CAEL\Desktop\practicas\_programacion> bash

CAEL@ACER /cygdrive/c/Users/CAEL/Desktop/practicas\_programacion  
\$ ./funcionvalor  
valor antes de la función: 13  
valor dentro de la función: 26  
valor después de la función: 13  
CAEL@ACER /cygdrive/c/Users/CAEL/Desktop/practicas\_programacion  
\$

8. Desarrolle un ejemplo pequeño en lenguaje de programación en C donde demuestre la utilización de funciones con paso de parámetros por referencia? (0.5)

```
C funcionrefe.c > main()
1 #include <stdio.h>
2
3 //Por: Cael Soto
4
5 void cuadriplicar(int *v) { //casi mismo ejem con referencia
6     *v = *v * 4;
7 }
8
9 int main() {
10     int num = 2;
11     printf("El valor antes de la función es: %d\n", num);
12     cuadriplicar(&num); //en teoría 2
13
14     printf("El valor después de la función es: %d\n", num);
15     return 0; //en teoría 8
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\CAEL\Desktop\practicas\_programacion> bash

CAEL@ACER /cygdrive/c/Users/CAEL/Desktop/practicas\_programacion  
\$ ./funcionrefe  
El valor antes de la función es: 2  
El valor después de la función es: 8  
CAEL@ACER /cygdrive/c/Users/CAEL/Desktop/practicas\_programacion  
\$

9. Indique el(los) error(s), muestre la corrección del mismo(s) si existe(n) **y/o depure** el siguiente bloque de código en lenguaje de programación C [Omita como error la inclusión de librerías stdio.h o stdlib.h]: (1.0 puntos)  
La idea de la porción de código, es Imprimir los días de la semana hasta el valor que se encuentra en día (no eliminar break) y los días pares se considera que es un gran día para aprender Teoría de la Programación.

**Código:**

```
enum semana_laboral {LUNES=1, MARTES=2, MIERCOLES=3, JUEVES=4, VIERNES=5};
int dia = 1
scanf("%d", &dia);
for (int i = LUNES; i < VIERNES; i++){
```

```
printf ("El día de la semana es: %d\n", i);
if ((i % 2) != 0){
    printf ("Hoy es un gran día para Teoría de la Programación\n", dia);
}
if (dia == i){
    break;
}
}
```

#### Errores del código:

- Falta el punto y coma (;) al declarar la variable `int dia = 1`.
- La condición del `for` no incluye el día VIERNES.
- El formato en `printf` "Hoy es un gran día ..." presenta un parámetro extra (`dia`) que no es necesario.
- El valor de `%` en la condición para determinar días pares está incorrecto.
- **Opcionalmente / no tomar en cuenta:** Querido Ing, a nivel de labor de un programador, el código debe ser accesible para todos los usuarios, por lo que lo correcto sería agregar una forma de parar el programa y advertir del error que esté cometiendo un usuario si escribe un número fuera del rango sugerido. Algo que también no se encontrará en el código brindado para ser arreglado es un indicativo al usuario de lo que literalmente está haciendo (en que programa está, qué está ejecutando o que debe hacer), por ende, se necesitaría un `printf` que de la instrucción que el usuario debe hacer (ingresar un número del 1 al 5 para que "algo ocurra". Este apartado no sería en sí un error de código sino un error del programador, sin embargo, al no ser parte del código brindado se añadirá como comentario `/**`.

#### Ejecución del programa de la corrección del código: (0.5 puntos)

```
C correcproblema.c > main()
1  #include <stdio.h>
2
3  //Por Cael Soto
4
5  enum semana_laboral {LUNES=1, MARTES=2, MIERCOLES=3, JUEVES=4, VIERNES=5};
6
7  int main() {
8      int dia = 1;
9      scanf("%d", &dia);
10     //printf("Ingrese un numero del día límite (1-5): ");
11
12     //Opcional/Adicional: Agregar una verificación de rango 1-5:
13     //if (dia < LUNES || dia > VIERNES) {
14     //printf("Error: El número ingresado debe estar entre 1 y 5.\n");
15     //return 1;
16     //}
17
18     for (int i = LUNES; i <= VIERNES; i++) {
19         printf("El día de la semana es: %d\n", i);
20         if ((i % 2) == 0) {
21             printf("Hoy es un gran día para Teoría de la Programación\n");
22         }
23         if (dia == i) {
24             break;
25         }
26     }
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
CAEL@ACER /cydrive/c/Users/CAEL/Desktop/practicas_programacion
$ ./correcproblema
5
El día de la semana es: 1
El día de la semana es: 2
Hoy es un gran día para Teoría de la Programación
El día de la semana es: 3
El día de la semana es: 4
Hoy es un gran día para Teoría de la Programación
El día de la semana es: 5
```

#### Conclusiones:

- El uso de un proceso estructurado para resolver problemas a través de programas permite a los programadores (de cualquier nivel) abordar situaciones complejas de manera lógica y eficiente. Si el programador decide seguir una metodología clara, se está asegurando que cada paso en la solución del problema que busca resolver esté bien planificado y justificado, lo que mejora la precisión y efectividad del programa final.



- Obtención de la representación de la resolución del problema a través de un algoritmo y/o diagrama: Representar la solución a un problema mediante algoritmos y diagramas de flujo es una herramienta fundamental en la programación, pues estos métodos permiten visualizar la lógica del programa antes de escribir el código, lo que facilita la identificación de posibles errores, optimiza la implementación del programa y permite tener una solución organizada y comprensible frente al problema a resolver.
- Las estructuras repetitivas, como los bucles, y el uso de subprogramas son elementos clave en la programación de C y Java, ya que permiten simplificar tareas repetitivas y organizar mejor el código. En C, las estructuras repetitivas permiten controlar el flujo de ejecución mediante condiciones, mientras que en Java, estas estructuras son igualmente esenciales, pero con la ventaja de que se combinan con la orientación a objetos, lo que facilita aún más la organización y modularización del código.
- Finalmente, la presentación de los resultados obtenidos de cada programa es crucial para verificar la efectividad de la solución implementada, esto no solo permite confirmar que el programa funciona correctamente, sino también facilita la identificación de errores o áreas de mejora. A través de la elaboración del taller se pudo comprobar la efectividad de cada programa, cada vez más complejo debido a la necesidad de tapar baches que interfieran con su correcta ejecución y correcto manejo por parte del usuario, verificar estos resultados fue de gran vitalidad para elaborar este taller pues permitía mejorar el código y aprender mientras se corrigen los errores hasta encontrar el resultado esperado.

#### D. RÚBRICA DE EVALUACIÓN

**Nota:** En caso de no cumplir con alguno de los parámetros establecidos se calificará la nota igual a 0  
Si se encuentra copia con algún compañero o prácticas realizadas de otros años, o bajados del internet o usando Inteligencia artificial, se aplicará el reglamento de deshonestidad estudiantil y se calificará sobre 0.  
No se aceptará trabajos atrasados ni incompletos, se calificará sobre 0.  
Todo acerca de deshonestidad académica que no diga este documento.  
Todas las secciones de carácter obligatorias se deben cumplir, caso contrario se asignará una nota igual a cero

<b>Informe de trabajo:</b> <ul style="list-style-type: none"> <li>• Contenido: pertinente y concreto; (obligatorio)</li> <li>• Estructura y organización: Elementos vinculados y estructurados coherentemente. (obligatorio)</li> <li>• Originalidad y creatividad: trabajo inédito, presentación de nuevas ideas.</li> </ul>	2 pts
<b>Resolución de Ejercicios:</b> <ul style="list-style-type: none"> <li>• Proceso de resolución de ejercicios (algoritmos y programas): con originalidad y creatividad (obligatorio)</li> <li>• Pruebas de escritorio y ejecución de programas (obligatorio)</li> </ul>	6 pts
<b>Conclusiones:</b> <ul style="list-style-type: none"> <li>• Redacción (obligatorio)</li> <li>• Originalidad y creatividad: conclusiones inéditas en base a su experiencia y objetivos planteados. (obligatorio)</li> </ul>	1 pto
<b>Video</b> <ul style="list-style-type: none"> <li>• Se debe entregar un video con la demostración y explicando el código de su solución (obligatorio)</li> </ul>	1 pto
<b>Total</b> (Ponderado en calificación final 60%)	10 pts

#### E. FIRMAS DE RESPONSABILIDAD DE LO ACTUADO

Estudiante(s):	Firma
Cael Alejandro Soto Castillo	