



En breve iniciamos



Lenguaje de JAVA.

Primeros Pasos.



www.sena.edu.co

Nelson E. Rincón C.
Instructor – Líder.
ADSO.

Centrando Ideas.

Que es Programar?

Tipo de Variables?

Que son bucles?

Que son condicionales?

Que es una declaración de datos o variables?

Que es asignación de datos o variables?

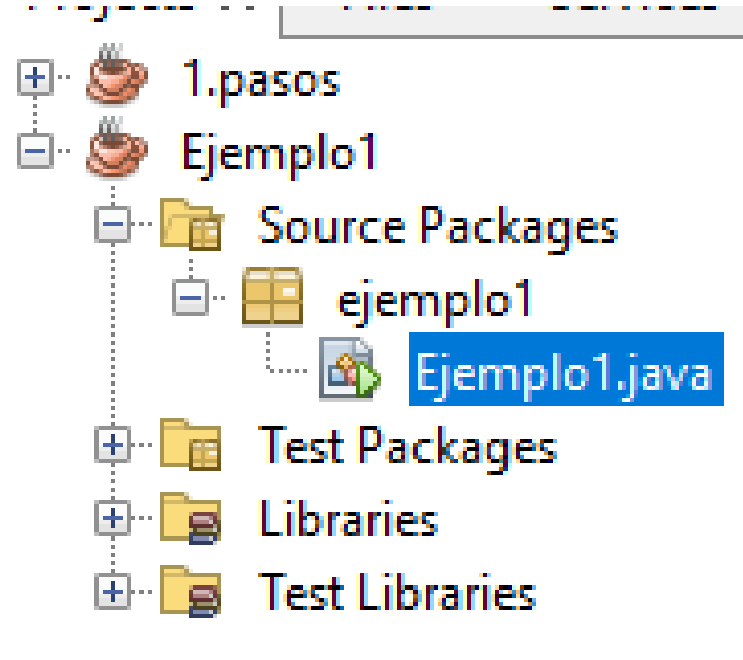
Tipos de visibilidad de variables?

Tipos de visibilidad de funciones?



Nelson E. Rincón C.
Instructor - ADSO.

Estructura de Carpetas.



Proyecto: La carpeta raíz, lleva el nombre del proyecto y contiene los archivos de configuración específicos del proyecto.

Source Packages: Aquí es donde se almacena el código fuente de tu proyecto Java. Las clases que creas se organizan en paquetes.

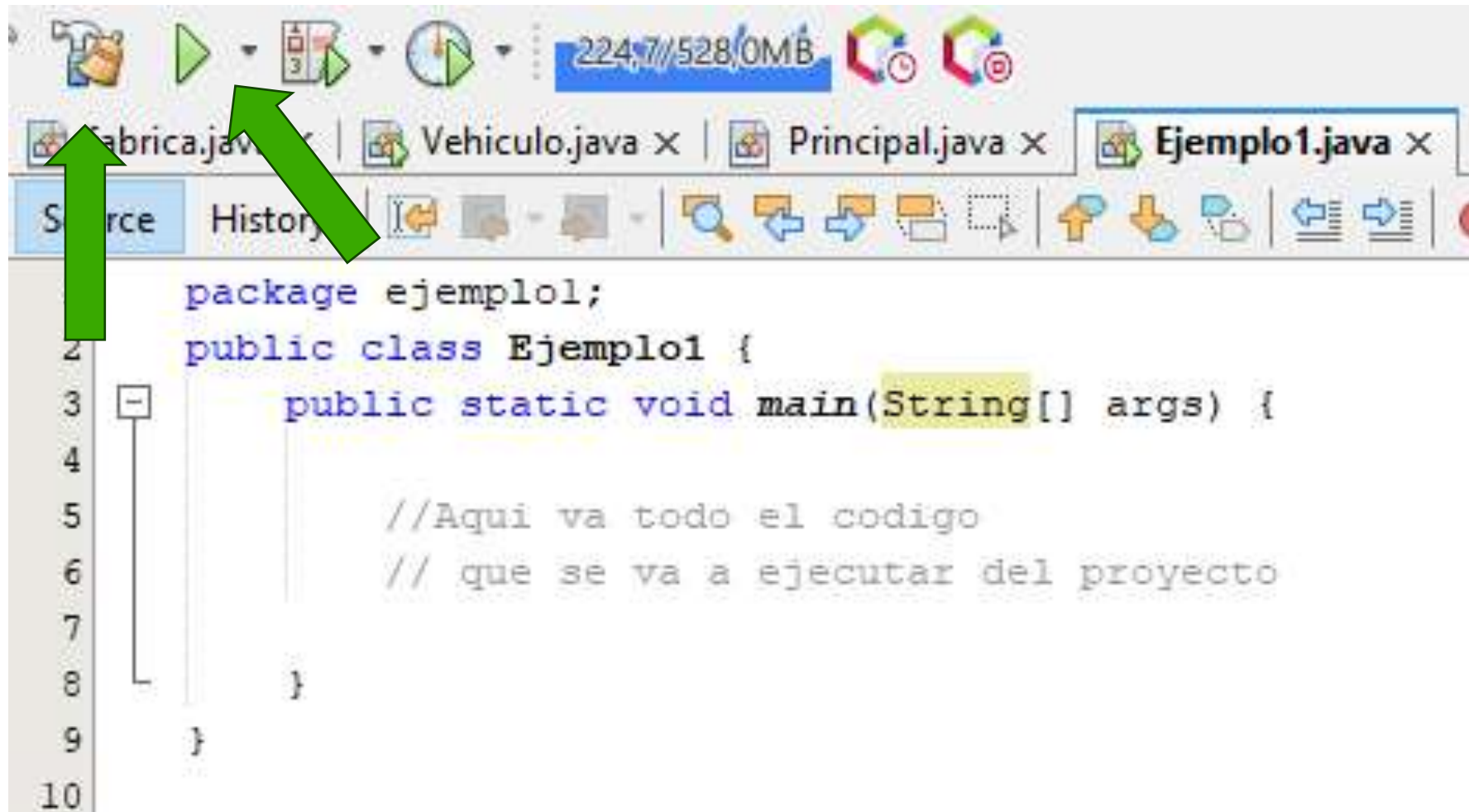
Test Packages: para pruebas unitarias, las clases de prueba generalmente se almacenan aquí. Puedes organizar tus pruebas en paquetes similares a cómo organizas tu código fuente.

Libraries: son las bibliotecas externas o JARs que tu proyecto necesita para funcionar.



Nelson E. Rincón C.
Instructor - ADSO.

Sintaxis código JAVA.



```
package ejemplo1;
public class Ejemplo1 {
    public static void main(String[] args) {

        //Aqui va todo el codigo
        // que se va a ejecutar del proyecto

    }
}
```

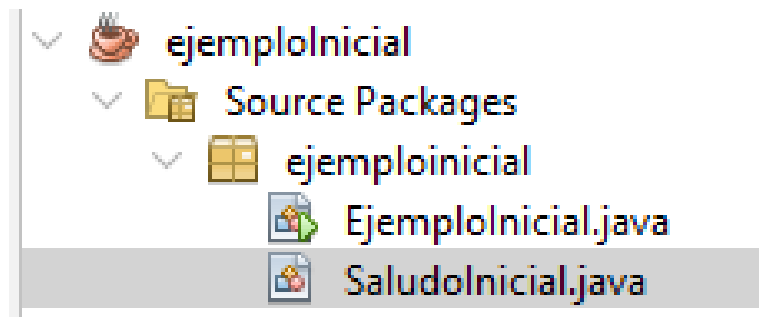
1. Importaciones de Clases o paquetes
2. Declaración de clase Principal
3. Campos o variables de la clase (Atributos)
4. Constructor: Define un constructor Principal y único
5. Métodos: son funciones, definen el comportamiento de la clase.



Nelson E. Rincón C.
Instructor - ADSO.

Sintaxis código JAVA.

```
3 package ejemploinicial;
4 //clase principal que tiene el metodo main()
5 public class EjemploInicial {
6     //Inicializacion de atributos
7     public static String nombreCliente;
8     private static int edadCliente;
9     public static String correoCliente;
10    //Metodo Principal
11    public static void main(String[] args) {
12        // TODO code application logic here
13        EjemploInicial inicio = new EjemploInicial();
14        nombreCliente = "Nelson Rincon";
15        EjemploInicial.edadCliente = 35;
16        correoCliente = "nerincon@sena.edu.co";
17        String validar = validarCliente(datoNombre: nombreCliente);
18        System.out.println(string: validar);
19        System.out.println("la edad del cliente es: " + edadCliente);
20    }
21    //metodo constructor
22    public EjemploInicial(){
23        System.out.println(string: "Bienvenido al sistema");
24    }
25    //metodo que ejecuta accion
26    private static String validarCliente(String datoNombre){
27        String aux="Cliente existe..." + datoNombre;
28        SaludoInicial saludo = new SaludoInicial(nombreCliente: edadCliente);
29        saludo.imprimirDatos();
30        return aux;
31    }
32 }
```



1. Importaciones de Clases o paquetes
2. Declaración de clase Principal
3. Campos o variables de la clase (Atributos)
4. Constructor: Define un constructor Principal y único
5. Métodos: son funciones, definen el comportamiento de la clase.



Nelson E. Rincón C.
Instructor - ADSO.

Sintaxis código JAVA.

```
5 package ejemploinicial;
6
7 public class SaludoInicial {
8     public SaludoInicial(int nombreCliente){
9         System.out.println(string: "hola estimado cliente...");
10    }
11    public void imprimirDatos(){
12        System.out.print(string: "Datos del cliente fueron impresos...");
13    }
14
15 }
```

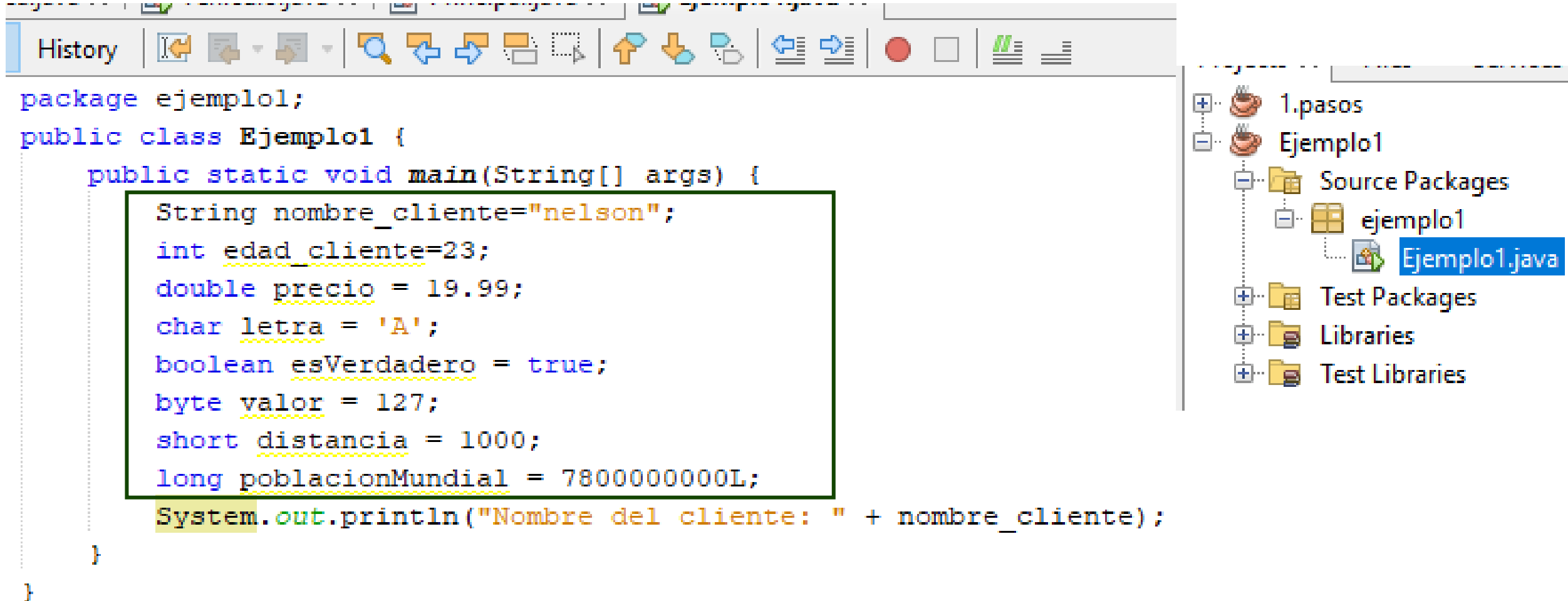
```
private static String validarCliente(String datoNombre){
    String aux="Cliente existe..." + datoNombre;
    SaludoInicial saludo = new SaludoInicial(nombreCliente: edadCliente);
    saludo.imprimirDatos();
    return aux;
}
```

1



Nelson E. Rincón C.
Instructor - ADSO.

Definición o creación de variables.



The screenshot shows an IDE window with a Java code file named `Ejemplo1.java`. The code defines a package `ejemplo1` and a public class `Ejemplo1` with a `main` method. Inside the `main` method, several variables are declared and assigned values:

```
package ejemplo1;

public class Ejemplo1 {
    public static void main(String[] args) {
        String nombre_cliente="nelson";
        int edad_cliente=23;
        double precio = 19.99;
        char letra = 'A';
        boolean esVerdadero = true;
        byte valor = 127;
        short distancia = 1000;
        long poblacionMundial = 78000000000L;
        System.out.println("Nombre del cliente: " + nombre_cliente);
    }
}
```

The project explorer on the right shows the following structure:

- 1.pasos
- Ejemplo1
 - Source Packages
 - ejemplo1
 - Ejemplo1.java
 - Test Packages
 - Libraries
 - Test Libraries

Que es un lenguaje tipado y no tipado.?



Nelson E. Rincón C.
Instructor - ADSO.

Ejemplos de Variables.

- Defina dos variables de texto, una llamada "nombre" y la otra "apellido". Asigna tu nombre y tu apellido a estas variables y luego imprime un mensaje de bienvenida que incluya el nombre y el apellido.
- Crea un programa que declare una variable entera llamada "edad". Asigna un valor numérico a esta variable y luego muestra un mensaje que diga: "Tienes X años", donde X es el valor de la variable "edad".
- Desarrolla un programa que defina una variable real llamada "precio" y asigna un valor decimal a esta variable, que represente el precio de un artículo. Luego, muestra un mensaje que diga: "El precio del artículo es \$X", donde X es el valor de la variable "precio".
- Crea un programa en que declare una variable booleana llamada "esEstudiante". Asigna un valor verdadero (Verdadero) o falso (Falso) a esta variable para indicar si el usuario es estudiante o no. Luego, muestra un mensaje que diga: "¿Es usted estudiante? [Verdadero/Falso]: X", donde X es el valor de la variable "esEstudiante".
- Escribe un programa en PSeInt que defina una variable de carácter llamada "letra". Asigna una letra del alfabeto a esta variable y luego muestra un mensaje que diga: "La letra asignada es: X", donde X es el valor de la variable "letra".



Captura de datos y Visualización.

```
1 package ejemplo1;
2
3 import java.util.Scanner;
4
5 public class Ejemplo1 {
6     public static void main(String[] args) {
7         //Definen las variables
8         String nombre_cliente="";
9         // Crear un objeto Scanner para la entrada de datos desde la consola
10        Scanner TomarDatos = new Scanner(System.in);
11        // Solicitar al usuario que ingrese su nombre
12        System.out.print("Por favor, ingrese su nombre: ");
13        // Capturar el nombre ingresado por el usuario
14        nombre_cliente = TomarDatos.nextLine();
15        // Cerrar el objeto Scanner
16        TomarDatos.close();
17        // Imprimir el nombre del cliente por consola
18        System.out.println("Hola, " + nombre_cliente + ". ¡Bienvenido!");
19    }
20 }
```

1. Nombre Clase
2. Nombre Variable
3. Signo igual
4. Palabra reservada new
5. La clase con argumentos o vacía.

Operadores.

- operadores aritméticos (+, -, *, /).
- operadores relacionales (>, <, >=, <=, ==, !=).
- operadores lógicos (y, o, no).



Condicional if - else / if – else if - else.

```
import java.util.Scanner;

public class Ejemplo1 {
    public static void main(String[] args) {
        parImpar();
    }
    public static void parImpar() {
        // Crear un objeto Scanner para la entrada de datos
        Scanner TomarDatos = new Scanner(System.in);
        // Solicitar al usuario que ingrese su nombre
        System.out.print("Por favor, ingrese un numero: ");
        // Capturar el nombre ingresado por el usuario
        int numero = TomarDatos.nextInt();
        // Cerrar el objeto Scanner
        TomarDatos.close();
        // Imprimir el nombre del cliente por consola
        if(numero%2==0) {
            System.out.println("el numero es par");
        }
        else{
            System.out.println("el numero es impar");
        }
    }
}
```

Se utiliza para ejecutar un bloque de código si se cumple una condición. Si la condición es verdadera, se ejecuta el bloque de código. Si la condición es falsa, se salta el bloque de código y ejecuta la parte negativa de la condición.



Nelson E. Rincón C.
Instructor - ADSO.

Ejercicio: Crear una Factura – Repaso.

- Escribe un programa que simule la creación de una factura. Debes utilizar todas las variables mencionadas anteriormente en un contexto relacionado con una compra.
- Asigna valores a las variables. Por ejemplo, define un nombre de cliente, un producto, la cantidad de productos, el precio unitario, indica si el cliente es estudiante o no, y asigna un tipo de cliente representado por una letra (A, B, C, etc.).
 - Calcula el subtotal multiplicando la cantidad de productos por el precio unitario.
 - Calcula el impuesto, que puede ser un 13% del subtotal si el cliente no es estudiante, o un 5% si lo es.
 - Calcula el total sumando el subtotal y el impuesto.

Muestra un mensaje que contenga la factura con los siguientes detalles:



Nelson E. Rincón C.
Instructor - ADSO.

Ejercicios condicionales.

- Escribe un programa que simule el funcionamiento de un ascensor. El usuario ingresa el piso al que desea ir, y el programa verifica si es un piso válido (usar un rango de numero). Si es válido, el ascensor se mueve al piso deseado. Si no es válido, muestra un mensaje de error.
- Desarrolla un programa que simule un ascensor que tiene un límite de peso. El usuario ingresa su peso y el piso al que desea ir. El programa debe verificar si el peso está dentro del límite permitido. Si es así, el ascensor se mueve al piso deseado. Si no, muestra un mensaje de que el ascensor está sobrecargado.
- Crea un programa que simule un sensor de temperatura ambiente. El usuario ingresa la temperatura actual y el programa verifica si la temperatura está dentro de un rango aceptable (por ejemplo, entre 18°C y 25°C). Si está dentro del rango, muestra un mensaje indicando que la temperatura es adecuada. Si no, muestra un mensaje de que la temperatura está fuera del rango deseado.



Ejercicios condicionales.

- Escribe un programa que simule un sensor de temperatura ambiente y un umbral de alerta. El usuario ingresa la temperatura actual, y el programa verifica si la temperatura está por encima del umbral de alerta. Si es así, muestra un mensaje de alerta. Si no, muestra un mensaje de que la temperatura es normal.
- Combina las dos simulaciones anteriores. El programa debe simular un ascensor que lleva al usuario a un piso deseado. Antes de moverse, el programa verifica si la temperatura ambiente está dentro del rango aceptable. Si la temperatura está dentro del rango, el ascensor se mueve al piso deseado. Si la temperatura está fuera del rango, muestra un mensaje de que la temperatura no es adecuada y no permite el movimiento del ascensor.



Condicional de Switch – case.

se utiliza para ejecutar diferentes bloques de código según el valor de una variable. Se especifican diferentes casos y el bloque de código correspondiente se ejecuta si el valor de la variable coincide con el caso.

```
public class Ejemplo1 {  
    public static void main(String[] args) {  
        Scanner OpcionValor = new Scanner(System.in);  
        System.out.print("Por favor, ingrese una opcion 1. Calcular /  
        int valoropcion= OpcionValor.nextInt();  
        switch (valoropcion){  
            case 1:  
                parImpar();  
            case 2:  
                System.out.print("Hasta luego BYE");  
            default:  
                System.out.print("Opcion Invalida.....");  
        }  
    }  
}
```



Ejercicios condicional: Switch - Case.

- En una tienda se venden tres tipos de productos: "A" (alimentos), "V" (vestimenta) y "E" (electrónicos). Cada producto tiene un precio base diferente. Escribe un programa en que solicite al cliente el tipo de producto que desea comprar (A, V o E) y la cantidad de unidades. Para calcular el costo total de la compra. Si el cliente compra productos de tipo "A", obtiene un 10% de descuento. Si compra productos de tipo "V", obtiene un 5% de descuento. Si compra productos de tipo "E", no hay descuento. Finalmente, muestra el costo total de la compra con el descuento aplicado y el costo sin descuento. Asegúrate de validar que la cantidad de unidades sea un número positivo.
- Escribe un programa como un asistente de un nutricionista, ayude a los pacientes a calcular su índice de masa corporal (IMC). El programa debe solicitar al usuario su peso y altura, y luego determinar su categoría de IMC (bajo peso, peso normal, sobrepeso u obesidad). IMC según los siguientes criterios:
 - Menor a 18.5 bajo peso, menor a 24.9 peso normal , menor a 29.9 "Sobrepeso" y mayor a 29,90 "Obesidad".



Ejercicios Condicional Switch - Case.

- Desarrolla un programa que simule un asistente virtual para un cine. El programa debe solicitar al usuario que ingrese su edad y, en función de la edad, recomendar una película adecuada: Niños (Edad menor a 7 años): Recomendará películas animadas y educativas aptas para todas las edades. Niños y Adolescentes (Edad entre 7 y 17 años): Sugerirá películas adecuadas para la familia, como animaciones, aventuras y comedias familiares. Adultos (Edad entre 18 y 30 años): Ofrecerá una variedad de géneros, como acción, drama, comedia y ciencia ficción. Personas Mayores (Edad mayor a 31 años): Recomendará películas clásicas y dramas que pueden ser de su interés.
- Crea un programa que simule un sistema de clasificación para un campeonato deportivo. El programa debe solicitar al usuario que ingrese el resultado de un partido (ganado, perdido o empatado) y, según el resultado, otorgar puntos a los equipos y mostrar la clasificación actual.
- Escribe un programa que simule un menú de restaurante. El programa debe permitir al cliente seleccionar varios platos y bebidas, calcular el costo total y aplicar un descuento según el método de pago (efectivo, tarjeta de crédito o cheque)



Bucles while

- Se ejecuta mientras se cumple una condición.

La condición se evalúa al principio de cada iteración del bucle. Si la condición es verdadera, el bucle se ejecuta. Si la condición es falsa, el bucle se detiene.

```
package pkgwhile.ejemplo;
import java.util.Scanner;
public class WhileEjemplo {
    public static void main(String[] args) {
        // TODO code application logic here
        int opcion = 0;
        Scanner dato_opcion= new Scanner( source: System.in);
        System.out.print( s: "digite una opcion: ");
        opcion= dato_opcion.nextInt();
        while (opcion>0){
            System.out.println( x: "Entro al WHILE ");
            System.out.println( x: "Hacer el BUCLE");
            System.out.println( x: "digite una opcion: ");
            opcion= dato_opcion.nextInt();
        }
    }
}
```



Ejercicios Bucles while

- Diseña un programa que simule un cajero automático. El usuario debe ingresar el monto que desea retirar. El programa debe verificar que el monto no exceda el saldo disponible en la cuenta. Si el monto es mayor que el saldo, se debe mostrar un mensaje de error. Si el monto es válido, el programa debe calcular y mostrar el saldo restante.
- Crea un programa que simule una compra en una tienda en línea. El usuario debe ingresar el precio de un producto y la cantidad que desea comprar. El programa debe calcular el costo total y mostrarlo. Luego, debe solicitar si el usuario desea agregar otro producto. El bucle "Mientras" debe permitir agregar múltiples productos y calcular el costo total de la compra.
- Desarrolla un programa que simule el funcionamiento de una impresora. El usuario debe ingresar la cantidad de páginas que desea imprimir. El programa debe verificar que la cantidad de páginas no exceda la capacidad de papel en la impresora. Si es así, debe imprimir las páginas y actualizar la cantidad de papel restante. El bucle "Mientras" debe permitir al usuario imprimir múltiples páginas hasta que no haya papel suficiente.



Ejercicios Bucles while.

- ■ Crea un programa que simule un viaje en coche. El usuario debe ingresar la distancia total del viaje y la velocidad promedio del coche. El programa debe calcular el tiempo estimado de viaje. Luego, debe preguntar si el usuario desea hacer otro viaje. El bucle "Mientras" debe permitir simular múltiples viajes y calcular el tiempo para cada uno.
- Diseña un programa que simule una cuenta regresiva. El usuario debe ingresar un número entero positivo y el programa debe comenzar a contar hacia atrás desde ese número hasta llegar a cero. Muestra cada número en la cuenta regresiva y finaliza con un mensaje de "¡Tiempo cumplido!".
- Desarrolla un juego simple de adivinanza en el que el programa genera un número aleatorio entre 1 y 100. El usuario debe intentar adivinar el número. El programa debe proporcionar pistas (mayor o menor) y permitir al usuario seguir adivinando hasta que lo adivine correctamente. Utiliza un bucle "Mientras" para controlar el flujo del juego.



Bucles repetir (do - while)

- Este bucle se ejecuta al menos una vez y luego se repite mientras se cumple una condición. La condición se evalúa al final de cada iteración del bucle. Si la condición es verdadera, el bucle se repite. Si la condición es falsa, el bucle se detiene.

```
package pkgwhile.ejemplo;
import java.util.Scanner;
public class DoWhileEjemplo {
    public static void main(String[] args) {
        // TODO code application logic here
        int opcion = 0;
        Scanner dato_opcion= new Scanner( source: System.in);

        do {
            System.out.println( x: "*** Menu de Cliente *** ");
            System.out.println( x: "1. Registrar cliente. ");
            System.out.println( x: "2. Modificar cliente. ");
            System.out.println( x: "3. Eliminar cliente. ");
            System.out.println( x: "4. Salir.... ");

            System.out.print( s: "Seleccione una opcion: ");
            opcion=dato_opcion.nextInt();
        }while (opcion>5);
    }
}
```

Ejercicios Bucles do - while

- Escribe un programa que simule un registro de bibliotecas. El programa debe permitir al usuario ingresar información sobre libros, como título, autor y número de páginas. Utiliza un bucle para solicitar la información del libro y continuar registrando libros hasta que el usuario decida detenerse. Al final, muestra la lista de libros registrados. Usar un acumulador de texto.
- Crea un programa que simule un menú de comidas rápidas. El usuario puede elegir entre diferentes opciones de menú, cada una con un precio. Utiliza un bucle para permitir al usuario realizar pedidos hasta que decida finalizar. Luego, muestra el costo total de los pedidos y un mensaje de agradecimiento.
- Escribe un programa que simule un sensor de temperatura. El usuario ingresa la temperatura actual, y el programa verifica si está dentro de un rango aceptable (por ejemplo, entre 18°C y 25°C). Si no está dentro del rango, muestra una alerta y continúa solicitando la temperatura hasta que esté dentro del rango.



Ejercicios Bucles do - while

- Desarrolla un programa que permita al usuario ingresar sus resultados de exámenes. El programa debe calcular el promedio de los resultados y determinar si el estudiante ha aprobado o reprobado. Utiliza un bucle para solicitar los resultados hasta que el usuario decida no ingresar más resultados.
- Crea un programa que simule la gestión de tareas. El usuario puede ingresar tareas con sus descripciones y fechas de vencimiento. Utiliza un bucle para permitir al usuario agregar tareas hasta que decida no agregar más. Al final, muestra la lista de tareas. Usar un acumulador de texto.
- Escribe un programa que simule un registro de ventas. El usuario puede ingresar los productos vendidos y sus precios. Utiliza un bucle para registrar ventas hasta que el usuario decida finalizar. Luego, muestra el monto total de las ventas y un resumen de los productos vendidos.



Bucles for.

- Este bucle se utiliza para iterar sobre una secuencia de valores.
- Se especifica un valor inicial, un valor final y un incremento. El bucle se ejecuta mientras el valor actual sea menor o igual al valor final.

```
package pkgwhile.ejemplo;
import java.util.Scanner;
public class FOR_Ejemplo {
    public static void main(String[] args) {
        // TODO code application logic here
        int opcion;
        Scanner dato_opcion= new Scanner( source: System.in);

        System.out.print( ":Digite un numero: ");
        opcion=dato_opcion.nextInt();

        for(int i=0; i<=opcion; i++){
            System.out.println("El conteo va en: " + Integer.toString(i));
        }
    }
}
```



Ejercicios Bucles para (for)

- Desarrolla un programa que simule el registro de vehículos en un parqueadero. El programa debe solicitar al usuario el número de vehículos que ingresarán al parqueadero. Luego, mediante un bucle, solicita la placa de cada vehículo y registra la hora de ingreso. Al finalizar, muestra un resumen de los vehículos registrados con sus respectivas placas y horas de ingreso. Usar variables para acumular información de carros
- Escribe un programa que permita a un usuario realizar operaciones de compra en una tienda virtual. Utiliza un bucle para que el usuario pueda agregar varios productos a su carrito de compras. Debes solicitar el nombre del producto y su precio, y luego mostrar el total de la compra y los productos agregados. Acumulador
- Crea un programa que funcione como una agenda de contactos. Utiliza un bucle para permitir al usuario agregar contactos a la agenda. Cada contacto debe incluir un nombre y un número de teléfono. Al final, muestra la lista de contactos registrados.



Ejercicios Bucles para (for)

- Escribe un programa que permita al usuario ingresar una lista de números. Utiliza un bucle para sumar todos los números ingresados y muestra el resultado al final. Asegúrate de validar que los números ingresados sean válidos.
- Crea un programa que simule el proceso de ventas en una tienda. Utiliza un bucle para registrar las ventas de varios productos. Debes solicitar el nombre del producto, su precio y la cantidad vendida. Al finalizar, muestra el total de ventas y los productos vendidos.





G R A C I A S

Línea de atención al ciudadano: 01 8000 910270
Línea de atención al empresario: 01 8000 910682



www.sena.edu.co