

## ***Retos JAVA***

**Leandro Jossue  
Ramírez Vezga**

**Ficha: 2450473**

**Análisis y desarrollo de sistemas de información**

**2022**

## Reto 1:

Teniendo en cuenta los conceptos vistos, cree la siguiente estructura:

Defina el Paquete Salud, dentro de él cree la Clase Persona y defina los atributos tipoDoc, documento, nombre, apellido, peso, estatura, edad, sexo y los métodos:

☐ pedirDatos() → que solicite los datos de la persona

☐ mostrarPersona() → imprime los datos ingresados

☐ calcularImc() → Primero se calcula

$\text{pesoActual} = \text{peso}(\text{en kg}) / (\text{estatura}(\text{en m})^2)$

Si resultado menor que 20, “el peso está por debajo de lo ideal”,

Si devuelve un número entre 20 y 25 (incluidos), “el peso es ideal” y

Si devuelve un valor mayor que 25 significa que tiene sobrepeso

☐ mayorEdad() → Determinar si es mayor de edad o no

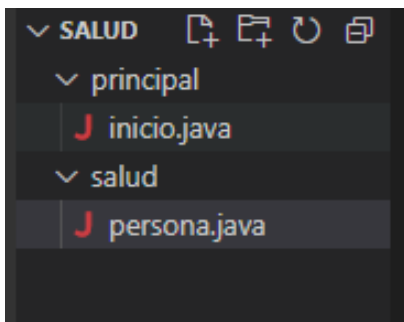
Defina otro paquete llamado Principal y dentro de él una clase

llamada Inicio

☐ Definir objetos para la clase (Considerar los modificadores de acceso necesarios para conseguirlo).

☐ Ejecutar los métodos para ver las operaciones definidas

**Paso 1:** Se crea los paquetes Salud y Principal, a su vez la clase inicio y persona.



**Paso 2:** Se instancia la clase Scanner y se declaran las variables

```
Scanner read = new Scanner(System.in);  
private String tipoDoc, nombre, apellido, sexo;  
  
private double documento, peso, estatura, edad, pesoActual;
```

**Paso 3:** Se crean los métodos constructores

```
public persona() {  
    }  
  
    public persona(String tipoDoc, String nombre, String apellido,  
String sexo, double documento, double peso,  
        double estatura, double edad, double pesoActual) {  
        this.tipoDoc = tipoDoc;  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.sexo = sexo;  
        this.documento = documento;  
        this.peso = peso;  
        this.estatura = estatura;  
        this.edad = edad;  
        this.pesoActual = pesoActual;  
    }  
}
```

**Paso 4:** Se crean los getter y setter

```
public String getTipoDoc() {  
    return tipoDoc;  
}  
  
public void setTipoDoc(String tipoDoc) {  
    this.tipoDoc = tipoDoc;  
}  
  
public String getNombre() {  
    return nombre;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
}
```

```
public String getApellido() {  
    return apellido;  
}  
  
public void setApellido(String apellido) {  
    this.apellido = apellido;  
}  
  
public String getSexo() {  
    return sexo;  
}  
  
public void setSexo(String sexo) {  
    this.sexo = sexo;  
}  
  
public double getDocumento() {  
    return documento;  
}  
  
public void setDocumento(double documento) {  
    this.documento = documento;  
}  
  
public double getPeso() {  
    return peso;  
}  
  
public void setPeso(double peso) {  
    this.peso = peso;  
}  
  
public double getEstatura() {  
    return estatura;  
}  
  
public void setEstatura(double estatura) {  
    this.estatura = estatura;  
}  
  
public double getEdad() {  
    return edad;  
}  
  
public void setEdad(double edad) {
```

```

        this.edad = edad;
    }

    public double getPesoActual() {
        return pesoActual;
    }

    public void setPesoActual(double pesoActual) {
        this.pesoActual = pesoActual;
    }
}

```

**Paso 5:** Se crea el método para pedir los datos

```

public void pedirDatos() {
    System.out.println("Ingrese el tipo de documento: ");
    setTipoDoc(read.nextLine());
    System.out.println("Ingrese el numero de documento: ");
    setDocumento(read.nextDouble());
    System.out.println("Ingrese el nombre: ");
    setNombre(read.next());
    System.out.println("Ingrese el apellido: ");
    setApellido(read.next());
    System.out.println("Ingrese el sexo: ");
    setSexo(read.next());
    System.out.println("Ingrese el peso (EN KG): ");
    setPeso(read.nextDouble());
    System.out.println("Ingrese la estatura(EN M): ");
    setEstatura(read.nextDouble());
    System.out.println("Ingrese la edad: ");
    setEdad(read.nextDouble());
}

```

**Paso 6:** Se crea el método para calcular el IMC

```

public void calcularImc() {
    pesoActual = peso / (Math.pow(estatura, 2));
    if (pesoActual < 20) {
        System.out.println("Esta por debajo de lo ideal");
    } else if (pesoActual >= 20 && pesoActual <= 25) {
        System.out.println("El peso es el ideal");
    } else if (pesoActual > 25) {
        System.out.println("Tiene sobrepeso");
    }
}

```

**Paso 7:** Se crea un método para verificar si el usuario es mayor o menor de edad

```
public void mayorEdad() {  
    if (edad >= 18) {  
        System.out.println("Es mayor de edad");  
    } else {  
        System.out.println("Es menor de edad");  
    }  
}
```

**Paso 8:** Se crea el método para mostrar los datos

```
public void mostrarPersona() {  
    System.out.println("Los datos de la persona son: \nTipo de  
documento: " + getTipoDoc() + "\nNumero de documento: "  
        + getDocumento() + "\nNombre: " + getNombre() + "\nApellido: "  
+ getApellido() + "\nSexo: " + getSexo()  
        + "\nPeso: "  
        + getPeso()  
        + "\nEstatura: " + getEstatura() + "\nEdad: " + getEdad());  
}
```

**Paso 9:** En la clase de ejecución se importa la clase persona y se crea la estructura del método main, llamando a los métodos de la clase persona

```
package principal;  
  
import salud.persona;  
  
public class inicio {  
    public static void main(String[] args) {  
        persona per = new persona();  
        System.out.println("Calcular si su peso es el ideal.");  
        per.pedirDatos();  
        per.mostrarPersona();  
        per.mayorEdad();  
        per.calcularImc();  
    }  
}
```

## Reto 2:

Modifique la estructura del método `calcularImc()`, para que este retorne en cada posible caso lo siguiente:

Si resultado menor que 20, retorne PESOBAJO

Si devuelve un número entre 20 y 25 (incluidos), PESOIDEAL

Si devuelve un valor mayor que 25 SOBREPESO

Y cambie la ejecución en la clase Inicio para que se evalúe el retorno del valor con un condicional y se muestren los mensajes correspondientes a cada caso.

**Paso 1:** Se modifica el método `calcularImc()` a un método String, y se usa un if para que el valor del peso cambie según el peso ingresado.

```
public String calcularImc(double peso, double estatura) {
    pesoActual = peso / (Math.pow(estatura, 2));

    if (pesoActual < 20) {
        cPeso = "PESOBAJO";
        return cPeso;
    } else if (pesoActual >= 20 && pesoActual <= 25) {
        cPeso = "PESOIDEAL";
        return cPeso;
    } else {
        cPeso = "SOBREPESO";
        return cPeso;
    }
}
```

**Paso 2:** Se modifica el main en la clase de ejecución para agregar un if que devuelva un mensaje dependiendo del valor retornado en el método `calcularImc()`

```
public static void main(String[] args) {
    persona per = new persona();
    System.out.println("Calcular si su peso es el ideal.");
    per.pedirDatos();
    per.mostrarPersona();
    per.mayorEdad();
    per.calcularImc(per.getPeso(), per.getEstatura());
    if (per.getcPeso() == "PESOBAJO") {
        System.out.println("Esta por debajo de lo ideal");
    } else if (per.getcPeso() == "PESOIDEAL") {
        System.out.println("El peso es el ideal");
    }
}
```

```

    } else if (per.getcPeso() == "SOBREPESO") {
        System.out.println("Tiene sobrepeso");
    }
}
}
}

```

### Reto 3:

☐ Aplicar el principio de encapsulamiento a la Clase Persona garantizando la adecuada ejecución de la misma.

☐ Crear una subclase Empleado que herede los atributos y métodos de la superclase Persona y cuente adicionalmente con:

Los atributos:

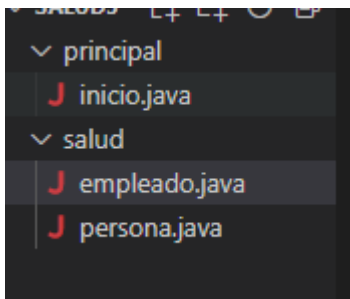
cargo, valorhora, horastrabajadas, departamento.

Los métodos:

✓ calcularHonorarios(), donde se calcule el valor de los honorarios del empleado → que es equivalente al producto del valor por hora y la cantidad de horas trabajadas menos RETEICA que corresponde al 0,966% sobre el valor total.

Imprimir el tipo y número de documento, nombres y apellidos, cargo, horas trabajadas, valor por hora y total a pagar.

**Paso 1:** Se crea la clase empleado



**Paso 2:** Se usa el extends en la clase empleado para heredar los atributos de la clase padre persona

```
public class empleado extends persona {
```

**Paso 3:** Se instancian las variables

```
private String cargo, departamento;
```



```
private double valorHora, horasTrabajadas, reteica, total,
valorTotal;
```

#### Paso 4: Se crean los constructores

```
public empleado() {
}

public empleado(String nombre, String apellido, String cargo, String
departamento, double valorHora,
double horasTrabajadas, double reteica, double total, double
valorTotal) {
    super(nombre, apellido);
    this.cargo = cargo;
    this.departamento = departamento;
    this.valorHora = valorHora;
    this.horasTrabajadas = horasTrabajadas;
    this.reteica = reteica;
    this.total = total;
    this.valorTotal = valorTotal;
}
```

#### Paso 5: Se crean los getter y setter

```
public String getCargo() {
    return cargo;
}

public void setCargo(String cargo) {
    this.cargo = cargo;
}

public String getDepartamento() {
    return departamento;
}

public void setDepartamento(String departamento) {
    this.departamento = departamento;
}

public double getValorHora() {
    return valorHora;
}

public void setValorHora(double valorHora) {
```

```

        this.valorHora = valorHora;
    }

    public double getHorasTrabajadas() {
        return horasTrabajadas;
    }

    public void setHorasTrabajadas(double horasTrabajadas) {
        this.horasTrabajadas = horasTrabajadas;
    }

    public double getReteica() {
        return reteica;
    }

    public void setReteica(double reteica) {
        this.reteica = reteica;
    }

    public double getTotal() {
        return total;
    }

    public void setTotal(double total) {
        this.total = total;
    }

    public double getValorTotal() {
        return valorTotal;
    }

    public void setValorTotal(double valorTotal) {
        this.valorTotal = valorTotal;
    }

```

**Paso 6:** Se crea el método para pedir los datos

```

public void pedirDatosemp() {
    System.out.println("Ingrese el nombre");
    setNombre(read.next());
    System.out.println("Ingrese el apellido");
    setApellido(read.next());
    System.out.println("Ingrese el cargo:");
    setCargo(read.next());
    System.out.println("Ingrese el departamento:");
}

```

```

    setDepartamento(read.next());
    System.out.println("Ingrese las horas trabajadas:");
    setHorasTrabajadas(read.nextDouble());
    System.out.println("Ingrese el valor por hora:");
    setValorHora(read.nextDouble());
}

```

**Paso 7:** Se crea el método para calcular los honorarios

```

public void calcularHonorarios() {
    total = valorHora * horasTrabajadas;
    reteica = total * 0.00966;
    valorTotal = total - reteica;
}

```

**Paso 8:** Se crea el método para mostrar los datos

```

public void mostraremp() {
    System.out.println("HONORARIOS\nEl trabajador " + getNombre() + "
" + getApellido() + " con el cargo de "
        + getCargo() + " y con " + getHorasTrabajadas() + " horas
trabajadas, con un valor cada una de "
        + getValorHora() + " pesos, y para un total a pagar de:\n " +
getValorTotal());
}

```

```

package principal;

import salud.empleado;

public class inicio {
    public static void main(String[] args) {
        empleado emp = new empleado();
        emp.pedirDatosemp();
        emp.calcularHonorarios();
        emp.mostraremp();
    }
}

```

**Paso 9:** En la clase de ejecución se llaman los métodos de la clase empleado

Portafolio:

[https://drive.google.com/drive/folders/1EzKdCFU\\_xRa9SVnyvsKQoJRyiRbmrWt?usp=sharing](https://drive.google.com/drive/folders/1EzKdCFU_xRa9SVnyvsKQoJRyiRbmrWt?usp=sharing)

Github:

<https://github.com/Ljossue/RetosPoo>