

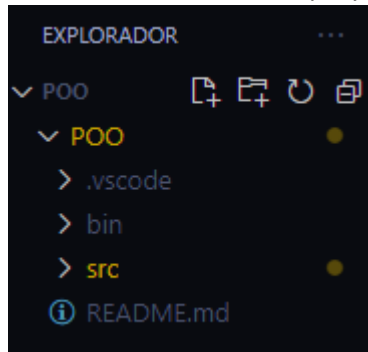
NOMBRE: JHOANN SEBASTIAN ZAMUDIO MARULANDA

FICHA: ADSO 2687351

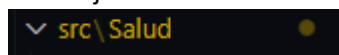
TEMA: EVIDENCIA_03_TALLER_Implementación_POO_Java

//RETO_1

1. Inicialmente creamos un proyecto llamado POO



2. Dentro de la carpeta “POO” en el apartado donde dice -src- creamos el paquete llamado “Salud.java”



3. Importamos la clase “Scanner” para que pueda realizar la captura de datos que ingrese el usuario, seguido a esto creamos una clase llamada “Persona”

```
4. package Salud;  
5. import java.util.Scanner;  
6.  
7. public class Persona {  
8. }
```

4. Seguidamente, definimos los atributos de la clase y los encapsulamos que es la visibilidad de los atributos

```
private String tipoDoc;  
private int documento;  
private String nombre;  
private String apellido;  
private double peso;  
private double estatura;  
private int edad;  
private String sexo;  
private double pesoActual;
```

5. Luego creamos los métodos, que el primero sería pedirDatos(); aquí pediremos los datos del usuario y dentro del método se imprimirá los datos a ingresar y los datos capturados se almacenarán en las variables ya establecidas. Se debe instanciar la clase Scanner el cual el nombre será "Lector" para poder capturar los datos.

```
public void pedirDatos(){
    Scanner Lector = new Scanner(System.in);

    System.out.println("Ingrese su tipo de documento");
    tipoDoc = Lector.next();

    System.out.println("Ingrese numero de documento");
    Documento = Lector.nextInt();

    System.out.println("Ingrese su nombre");
    Nombre = Lector.next();

    System.out.println("Ingrese su apellido");
    Apellido = Lector.next();

    System.out.println("Ingrese su peso en kg");
    peso = Lector.nextDouble();

    System.out.println("Ingrese su estatura en cm");
    Estatura = Lector.nextDouble();

    System.out.println("Ingrese su edad");
    Edad = Lector.nextInt();

    System.out.println("Ingrese su sexo (F o M)");
    Sexo = Lector.next();

    Lector.close();
}
```

6. Luego creamos los métodos, que el segundo sería mostrarPersona(); concatenamos las impresiones junto a las variables donde el Scanner almacena los datos que ingreso el usuario

```
public void mostrarPersona(){
    System.out.println("Su tipo de documento es "+tipoDoc);
    System.out.println("Su numero de documento es "+documento);
    System.out.println("su nombre es "+nombre);
    System.out.println("su apellido es "+apellido);
    System.out.println("su peso es "+peso);
    System.out.println("su estatura es "+estatura);
}
```

```

        System.out.println("su edades "+edad);
        System.out.println("su sexo es "+sexo);
    }

```

7. Luego creamos los métodos, que el tercero seria calcularImc(); dentro de ese método haremos una operación dentro de una variable llamada “pesoActual”, aquí el peso que ingrese el usuario será dividido a la potencia al cuadrado de estatura.

haremos una condicional if anidada donde si el valor de pesoActual es menor a 20, que imprima un mensaje que diga “el peso esta por debajo de lo ideal”, o si el pesoActual es mayor a 20 y menor que 25 que diga “esta en el peso ideal”, o si el pesoActual es mayor a 20 que diga “esta en sobrepeso”.

```

public void calcularImc(){
    pesoActual = peso/(estatura*estatura);

    if(pesoActual<20){
        System.out.println("Su IMC es "+pesoActual);
        System.out.println("El peso esta por debajo de lo ideal");
    }else if(pesoActual>20 && pesoActual<25){
        System.out.println("Su IMC es "+pesoActual);
        System.out.println("El peso es ideal");
    }else if(pesoActual>25){
        System.out.println("Su IMC es "+pesoActual);
        System.out.println("Esta en sobrepeso");
    }
}

```

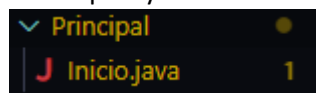
8. Luego creamos los métodos, que el cuarto seria mayorEdad(); Seguida, dentro de método creamos un condicional if donde si edad era mayor a 18 entonces imprime “Eres mayor de edad” si no, pues imprime “Eres menor de edad”

```

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

```

9. Para ejecutar lo anterior debemos invocar los métodos, creamos otro paquete llamado “Principal” y dentro de la misma una clase llamada Inicio



10. definimos un método principal dentro de la clase Inicio, el método principal es “main”

```
public class Inicio {  
    public static void main(String[] args) {  
  
        }  
    }  
}
```

11. dentro del metodo principal main, instanciamos la clase persona , para asi poder invocarlos y observarlos

```
Persona Ver = new Persona();
```

12. Seguidamente invocamos los métodos con la clase Ver seguido de un punto y el método ya definido

```
Ver.pedirDatos();  
Ver.mostrarPersona();  
Ver.calcularImc();  
Ver.mayorEdad();
```

13. Seguidamente, ejecutamos el código:

```
Ingrese su tipo de documento  
CC  
Ingrese numero de documento  
1013580753  
Ingrese su nombre  
Jhoann  
Ingrese su apellido  
Zamudio  
Ingrese su peso en KL  
69  
Ingrese su estatura  
1.83  
Exception in thread "main" java.util.InputMismatchException  
Tipo documento: CC  
Nº Documento: 1013580753  
Nombre: Jhoan  
Apellido: Zamudio  
Peso: 69.0  
Estatura: 1.83  
Edad: 18  
SExo: M  
Eres es mayor de edad  
Su IMC es 20.603780345785182  
El peso es ideal  
PS C:\Users\Joan Se\Desktop\P00> █
```

//Reto 2

1. los pasos serán los mismos que el reto 1 solo que cambiando el método calcularImc(); cambiamos el void por double que será el retorno del método, le colocamos parámetros la método, ósea lo que quiere retornar que en este caso será peso y estatura. creamos una nueva variable llamada imc y la inicializamos en cero, seguido creamos una variable pesoActual y en ella la declaramos para hacer una operación, después se crea un condicional anidado donde si pesoActual es menor a 20 que imprima "esta por debajo de lo ideal", si no, si el pesoActual es mayor a 20 y menor que 25 que imprima "el peso es el ideal", si no, si el pesoActual es mayor a 25, que imprima "esta en sobrepeso"

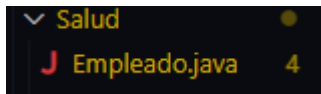
```
2. public double calcularImc(double peso, double estatura ){
3.     double imc=0;
4.     pesoActual =peso/(estatura*estatura);
5.     if(pesoActual<20){
6.         System.out.println("Su IMC es "+pesoActual);
7.         System.out.println("El peso esta por debajo de lo ideal");
8.     }else if(pesoActual>20 && pesoActual<25){
9.         System.out.println("Su IMC es "+pesoActual);
10.        System.out.println("El peso es el ideal");
11.    }else if(pesoActual>25){
12.        System.out.println("Su IMC es "+pesoActual);
13.        System.out.println("Esta en sobrepeso");
14.    }
15.    return imc;
16. }
```

2. para ejecutarlo, debemos invocar el método en el paquete llamado principal y dentro de la clase Inicio, creamos el objeto que en este caso será ver2, instanciamos el objeto de Scanner para capturar los datos seguido imprimos unos datos a almacenar, invocamos y limpiamos el buffer del scanner.

```
Persona2 ver2=new Persona2();
    double peso, estatura;
    Scanner Lector= new Scanner(System.in);
    System.out.println("Ingrese su peso en KL");
    peso=Lector.nextDouble();
    System.out.println("Ingrese su estatura");
    estatura=Lector.nextDouble();
    ver2.pedirDatos();
    ver2.mostrarPersona();
    ver2.mayorEdad();
    ver2.calcularImc( peso, estatura); //aquí esta el invoke del ver2
    Lector.close();*/
```

//RETO 3

1. Creamos dentro del mismo paquete una clase llamada Empleado



2. Crearemos la subclase de la superclase Persona, aquí se usa la palabra reservada extends

```
3. package Salud;  
4. public class Empleado extends Persona {  
5. }
```

3. En la superclase Persona se debe encapsular los atributos de la misma

```
4. public class Persona {  
5.     //ATRIBUTOS  
6.     private String tipoDoc;  
7.     private int documento;  
8.     private String nombre;  
9.     private String apellido;  
10.    private double peso;  
11.    private double estatura;  
12.    private int edad;  
13.    private String sexo;  
14.    private double pesoActual;
```

4. Seguidamente, creamos un constructor vacío

```
public Persona(){  
  
}
```

5. Seguidamente, se cambia el método constructor

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

```
    this.sexo = sexo;  
    this.pesoActual = pesoActual;  
}
```

6. Retornamos los métodos asesores que definimos previamente en los atributos:

```
public String getTipoDoc() {  
    return tipoDoc;  
}  
public void setTipoDoc(String tipoDoc) {  
    this.tipoDoc = tipoDoc;  
}  
public int getDocumento() {  
    return documento;  
}  
public void setDocumento(int documento) {  
    this.documento = documento;  
}  
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 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 int getEdad() {  
    return edad;  
}  
public void setEdad(int edad) {
```

```

        this.edad = edad;
    }
    public String getSexo() {
        return sexo;
    }
    public void setSexo(String sexo) {
        this.sexo = sexo;
    }
    public double getPesoActual() {
        return pesoActual;
    }
    public void setPesoActual(double pesoActual) {
        this.pesoActual = pesoActual;
    }
}

```

6. Volvemos a la subclase Empleado y creamos 3 atributos nuevos de manera encapsulada

```

private String cargo;
private int valorhora;
private int horastrabajadas;
private String departamento;

```

7. Creamos un constructor para inicializar el objeto y establecer propiedades con valores predeterminados, invocamos los atributos de la superclase con la palabra this

```

public Empleado(String tipoDoc, int documento, String nombre, String
apellido, double peso, double estatura,
    int edad, String sexo, double pesoActual) {
    super(tipoDoc, documento, nombre, apellido, peso, estatura, edad,
sexo, pesoActual);
    this.cargo = cargo;
    this.valorhora = valorhora;
    this.horastrabajadas = horastrabajadas;
    this.departamento = departamento;
}

```

8. Seguidamente, creamos los métodos asesores de la subclase Empleado, aplicando los get y set, que quieren decir, get: obtener, set: modificar o asignar un valor.

```

public String getCargo() {
    return cargo;
}

public void setCargo(String cargo) {

```



```

        this.cargo = cargo;
    }
    public int getValorhora() {
        return valorhora;
    }
    public void setValorhora(int valorhora) {
        this.valorhora = valorhora;
    }
    public int getHorastrabajadas() {
        return horastrabajadas;
    }
    public void setHorastrabajadas(int horastrabajadas) {
        this.horastrabajadas = horastrabajadas;
    }
    public String getDepartamento() {
        return departamento;
    }
    public void setDepartamento(String departamento) {
        this.departamento = departamento;
    }
}

```

9. Creamos el método para mostrar los datos ingresados concatenando las variables de esta manera, aquí estamos llamando con get de la superclase Persona ya que la subclase Empleado es heredera y puede acceder a los mismo

```

public void mostrarPersona(){
    System.out.println("Su nombre es: " +getNombre()+ " \nSu apellido es
" +getApellido()+ " \nSu tipo de documento es: "
    +getTipoDoc()+ "\n Su documento es: "+getDocumento()+ " \nSu edad es
: "+getEdad()+ " \nSu estatura es " +getEstatura()+
    " \nSu peso es "+getPeso()+"\n Su sexo es : "+getSexo()+"\n su valor
hora es de: "+valorhora+" \nSus horas trabajadas son: "
    +horastrabajadas+"\nSu departamento es: "+departamento);
}

```

10. Seguidamente creamos un nuevo método el cual se llamara calcularHonorarios(); dentro de la misma se definira las variables, se hará operaciones como lo son multiplicar el valor de hora X hora trabajada y después se le resta la reteica y al finalizar imprime sus ganancias

```

public void calcularHonorarios(){
    double calcular1=0;
    double calcular2=0;
    calcular1=(valorhora*horastrabajadas);
    calcular2 = calcular1-0.966;
}

```

```

        System.out.println("Sus ganancias son de: " + calcular2);
    }

```

11. Ingresamos a la clase Inicio, instanciamos el objeto Scanner para la captura de datos, creamos nuevos atributos de la subclase

```

Scanner lector= new Scanner(System.in);

String cargo;
int valorhora;
int horastrabajadas;
String departamento;

```

12. Solicitamos los datos y allí lo que el usuario digite se almacenara en las variables

```

System.out.println("Escriba su cargo ");
cargo=lector.next();
System.out.println("Escriba el valor de la hora ");
valorhora=lector.nextInt();
System.out.println("Escriba sus horas trabajadas ");
horastrabajadas=lector.nextInt();
System.out.println("Escriba su departamento ");
departamento=lector.next();

```

13. Seguidamente creamos un objeto de la superclase Persona y se llamara X, después invocamos el método para pedir los datos.

```

Persona X =new Persona();
X.pedirDatos();

```

14. instanciamos la subclase Empleado y en el mismo los get de la superclase persona con el objeto creado de esa clase

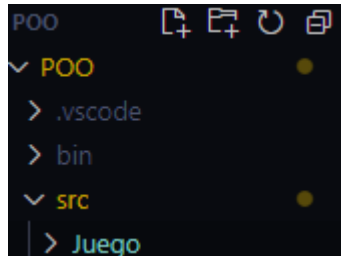
```

Empleado empleado= new
Empleado(X.getTipoDoc(),X.getDocumento(),X.getNombre(),X.getApellido()
,X.getPeso(),X.getEstatura(),X.getEdad(),X.getSexo(),X.getPesoActua
l());

```

//RETO 5

1. Creamos un paquete llamado "juego":



2. Dentro del paquete, creamos una nueva clase llamada PiedraPapelTijera y creamos el implements de juego

```
package Juego;

import jaca.util.Random;
import java.util.Scanner;

public class PiedraPapelTijera implements Juego{
```

3. Creamos la interfaz de juego, esencialmente, una interfaz no tiene métodos concretos

```
package Juego;

public interface Juego {
    void iniciar();
    void jugar();
    void finalizar();
}
```

4. volvemos a la clase PiedraPapelTijera y creamos los atributos con encapsulación private

```
private int opcion;
private String jugador;
private int jugada;
```

5. Seguidamente, instanciamos el objeto Scanner que es el que va a capturar los datos para almacenarlas en las variables, creamos un método llamado iniciar() donde imprimiremos una solicitud de datos

```
Scanner lector= new Scanner (System.in);

public void iniciar() {

    System.out.println("Ingrese el nombre del jugador");
    jugador=lector.next();
```

```
}
```

6. Seguidamente, creamos un nuevo método el cual se llamara jugar() dentro de la misma aparecerá un mensaje que le muestra al usuario que elija una de las 3 opciones y después lo que el usuario digite se almacenara en la variable opción, se instancia el objeto ramdomico con nombre aleatorio que vaya desde 1 a 3, se define asi porque son 3 opciones:

```
public void jugar() {  
    System.out.println(jugador + "Ingrese la opcion que desee para  
jugar");  
    System.out.println("1. Piedra");  
    System.out.println("2. Papel");  
    System.out.println("3. Tijera");  
  
    Opcion=Lector.nextInt();  
    Random aleatorio= new Random();  
    jugada=aleatorio.nextInt(3);  
}
```

4. Seguidamente, creamos un nuevo método (el ultimo) que se llamara finalizar(); aquí dentro del mismo ira una condicional if anidada, el cual tiene comparacion si opcion es igual a 1 continúe con la siguiente condición y si jugada es igual a 1 muestre que salió papel y perdio y si no salió tijera y gano, si opcion es igual a 2 continúe con la siguiente condición y si jugada es igual a 1 muestre que salió tijera y perdió y si no salió piedra y gano, si opcion es igual a 3 continúe con la siguiente condición y si jugada es igual a 1 muestre que salió piedra y perdió y si no salió piedra y gano

```
public void finalizar() {  
    if (opcion==1) {  
        if (jugada==1) {  
            System.out.println("Señor@ " + jugador + ", selecciono: " +  
opcion + " y salio PAPEL, perdido ");  
        } else {  
  
            System.out.println("Señor@ " + jugador + ", selecciono: " +  
opcion + " y salio TIJERA, gano");  
        }  
    }  
}
```

```

    }
    } else if (opcion==2) {
        if (jugada==1) {
            System.out.println("Señor@ " + jugador + ", selecciono: " +
opcion + " y salio TIJERA, perdido ");
        } else {

            System.out.println("Señor@ " + jugador + ", selecciono: " +
opcion + " y salio PIEDRA, gana ");
        }
    } else if (opcion==3) {
        if (jugada==1) {
            System.out.println("Señor@ " + jugador + ", selecciono: " +
opcion + " y salio PIEDRA, perdido");
        } else {

            System.out.println("Señor@ " + jugador + ", selecciono: " +
opcion + " y salio, PIEDRA, gana ");
        }
    }
}
}
}

```

5. Seguidamente creamos otra clase dentro del mismo paquete llamado juego, creamos una clase llamada Ejecucion, donde definiremos el main principal que sera le método, seguido instanciamos un objeto de la clase PiedraPapelTijera e invocamos los 3 metodos que se crearon en el paquete Juego

```

package Juego;
public class Ejecucion {
    public static void main(String[] args) {
        PiedraPapelTijera juego1=new PiedraPapelTijera();
        juego1.iniciar();
        juego1.jugar();
        juego1.finalizar();
    }
}

```

6. Ejecucion:

```

Ingrese el nombre del jugador
Jhoan
JhoanIngrese la opcion que desee para jugar
1. Piedra
2. Papel
3. Tijera
1
Señor@ Jhoan, selecciono: 1 y salio TIJERA, gana

```

Enlace Portafolio: https://soysena-my.sharepoint.com/personal/jszamudio35_soy_sena_edu_co/_layouts/15/onedrive.aspx?ga=1&id=%2Fpersonal%2Fjszamudio35%5Fsoy%5Fsena%5Fedu%5Fco%2FDocuments%2FADSO%5F2684351%2F4%5FEVIDENCIAS%5FDE%5FAPRENDIZAJE%2FTRIMESTRE%5FI%2FFUNDAMENTOS%5F%26%5FPOO%5FJAVA%5FTATIANA%5FCABRERA&view=0

Enlace GitHub: [https://github.com/JoanZamudio/FUNDAMENTOS -_POO_JAVA_TATIANA_CABRERA.git](https://github.com/JoanZamudio/FUNDAMENTOS_-_POO_JAVA_TATIANA_CABRERA.git)