



PRACTICA 9

Facultad de ciencias de la computación
Ing. En ciencias de la computación



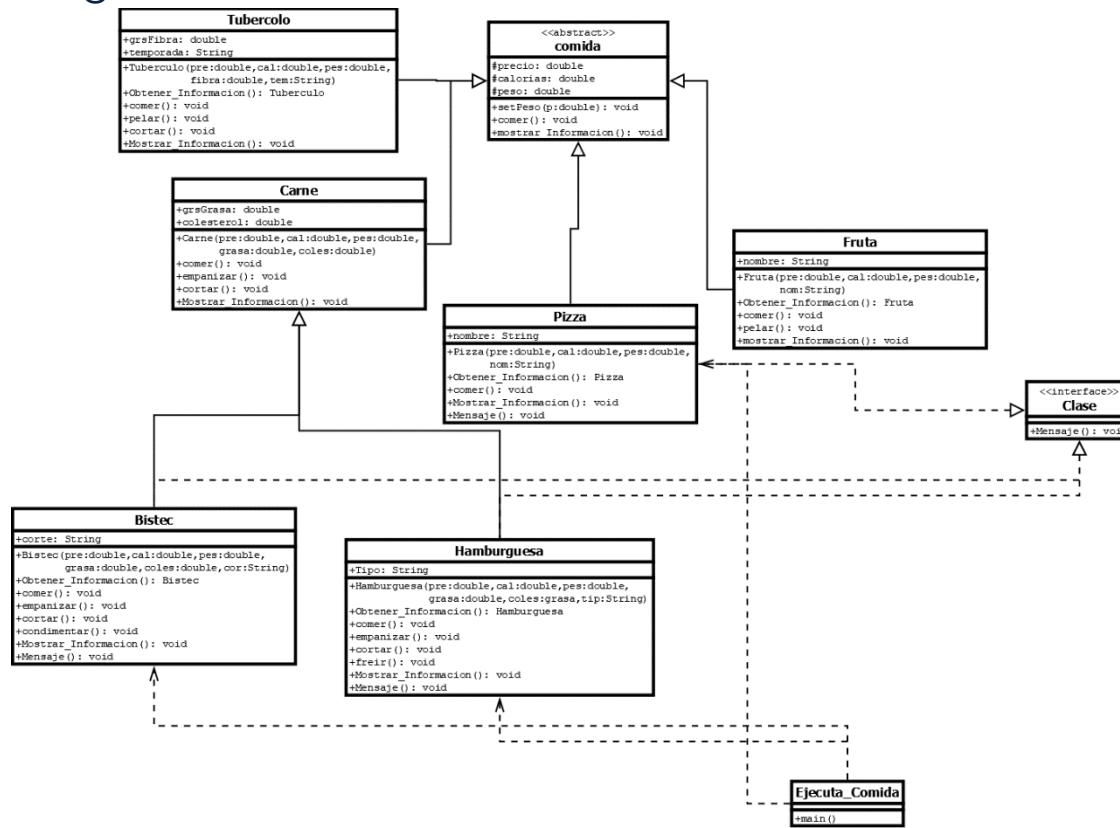
IRVYN XICALE CABRERA
201963582

Introducción

El objetivo de esta practica es implementar lo aprendido en el curso con las herencias clases abstractas e interfaces.

Comida

Diagrama ULM



Código

Comida.java

```
public abstract class Comida
{
    protected double precio;
    protected double calorias;
    protected double peso;
    //
    public Comida()
    {
    }
    public void SetPeso(double p)
    {
        peso=p;
    }
    public abstract void comer();

    public void Mostrar_Informacion()
    {
        System.out.println();
        System.out.println("Precio: "+precio);
        System.out.println("Calorias: "+calorias);
        System.out.println("Peso: "+peso);
    }
}
```

NoSaludable.java

```
public interface NoSaludable {  
    abstract public void Mensaje();  
}
```

Tuberculo.java

```
import java.io.*;  
import java.util.*;  
public class Tuberculo extends Comida  
{  
    double grsFibra;  
    String temporada;  
    public Tuberculo()  
    {  
    }  
    public Tuberculo(double pre, double cal, double pes, double fibra, String tem)  
    {  
        super.precio=pre;  
        super.calorias=cal;  
        super.peso=pes;  
        grsFibra=fibra;  
        temporada=tem;  
    }  
    public Tuberculo Obtener_Informacion()  
    {  
        Scanner Leer=new Scanner(System.in);  
  
        System.out.println("Dame temporada del tuberculo: ");
```

```
temporada=Leer.nextLine();
System.out.println("Dame precio: ");
super.precio=Leer.nextDouble();
System.out.println("Dame calorías: ");
super.calorias=Leer.nextDouble();
System.out.println("Dame peso: ");
super.peso=Leer.nextDouble();
System.out.println("Dame los gramos de fibra: ");
grsFibra=Leer.nextDouble();

Tuberculo objeto2=new Tuberculo(precio,calorias,peso,grsFibra,temporada);

return objeto2;
}
public void comer()
{
System.out.println("Comiendo tuberculo");
}
public void pelar()
{
System.out.println("Pelando tuberculo");
}
public void cortar()
{
System.out.println("Cortando tuberculo");
}
public void Mostrar_Informacion()
{
System.out.println();
System.out.println("Informacion del tuberculo");
super.Mostrar_Informacion();
}
```

```
        System.out.println("Gramos de fibra: "+grsFibra);  
        System.out.println("Temporada: "+temporada);  
    }  
}
```

Fruta.java

```
import java.util.*;  
public class Fruta extends Comida  
{  
    String nombre;  
    public Fruta()  
    {  
    }  
    public Fruta(double pre, double cal, double pes,String nom)  
    {  
        super.precio=pre;  
        super.calorias=cal;  
        super.peso=pes;  
        nombre=nom;  
    }  
    public Fruta Obtener_Informacion()  
    {  
        //METODO  
        Scanner Leer=new Scanner(System.in);  
  
        System.out.println("Dame nombre: ");  
        nombre=Leer.nextLine();  
        System.out.println("Dame precio: ");
```

```
super.precio=Leer.nextDouble();
System.out.println("Dame calorías: ");
super.calorias=Leer.nextDouble();
System.out.println("Dame peso: ");
super.peso=Leer.nextDouble();

Fruta objeto=new Fruta(precio,calorias,peso,nombre);

return objeto;
}
public void comer()
{
System.out.println("Comiendo fruta");
}
public void pelar()
{
System.out.println("Pelando fruta");
}
public void Mostrar_Informacion()
{
System.out.println();
System.out.println("Información de la fruta");
super.Mostrar_Informacion();
System.out.println("Nombre de la fruta: "+nombre);
}
}
```

Pizza.java

```
import java.util.*;
public class Pizza extends Comida implements NoSaludable{
    String nombre;
    public Pizza()
    {
    }
    public Pizza(double pre, double cal, double pes,String nom)
    {
        super.precio=pre;
        super.calorias=cal;
        super.peso=pes;
        nombre=nom;
    }
    public Pizza Obtener_Informacion()
    {
        //METODO
        Scanner Leer=new Scanner(System.in);

        System.out.println("Dame nombre: ");
        nombre=Leer.nextLine();
        System.out.println("Dame precio: ");
        super.precio=Leer.nextDouble();
        System.out.println("Dame calorias: ");
        super.calorias=Leer.nextDouble();
        System.out.println("Dame peso: ");
        super.peso=Leer.nextDouble();

        Pizza objeto=new Pizza(precio,calorias,peso,nombre);
    }
}
```



```

    return objeto;
}
public void comer()
{
    System.out.println("Comiendo Pizza");
}
public void Mostrar_Informacion()
{
    System.out.println();
    System.out.println("Informacion de la Pizza");
    super.Mostrar_Informacion();
    System.out.println("Nombre de la Pizza: "+nombre);
}

public void Mensaje(){
    System.out.println("la pizza puede dañar tu salud");
}
}

```

Carne.java

```

import java.io.*;
import java.util.*;
public abstract class Carne extends Comida
{
    double grsGrasa;
    double colesterol;
    public Carne()
    {

```

```

    }
    public Carne(double pre, double cal, double pes, double grasa, double coles)
    {
        super.precio=pre;
        super.calorias=cal;
        super.peso=pes;
        grsGrasa=grasa;
        colesterol=coles;
    }
    abstract public void comer();
    abstract public void empanizar();
    public void cortar()
    {
        System.out.println("Cortando carne");
    }
    public void Mostrar_Informacion()
    {
        System.out.println();
        super.Mostrar_Informacion();
        System.out.println("Gramos de grasa: "+grsGrasa);
        System.out.println("Colesterol: "+colesterol);
    }
}

```

Bistec.java

```

import java.io.*;
import java.util.*;
public class Bistec extends Carne implements NoSaludable

```

```

{
    String corte;
    public Bistec()
    {
    }
    public Bistec(double pre, double cal, double pes, double grasa, double coles, String cor)
    {
        super.precio=pre;
        super.calorias=cal;
        super.peso=pes;
        super.grsGrasa=grasa;
        super.colesterol=coles;
        corte=cor;
    }
    public Bistec Obtener_Informacion()
    {
        Scanner Leer=new Scanner(System.in);

        System.out.println("Dame corte: ");
        corte=Leer.nextLine();
        System.out.println("Dame precio: ");
        super.precio=Leer.nextDouble();
        System.out.println("Dame calorias: ");
        super.calorias=Leer.nextDouble();
        System.out.println("Dame peso: ");
        super.peso=Leer.nextDouble();
        System.out.println("Dame los gramos de grasa: ");
        super.grsGrasa=Leer.nextDouble();
        System.out.println("Dame cantidad de colesterol: ");
        super.colesterol=Leer.nextDouble();
    }
}

```

```

Bistec objeto5=new Bistec(precio,calorias,peso,grsGrasa,colesterol,corte);

return objeto5;
}
public void comer()
{
System.out.println("Comiendo bistec");
}
public void empanizar()
{
System.out.println("Empanizando bistec");
}
public void cortar()
{
System.out.println("Cortando carne");
}
public void condimentar()
{
System.out.println("Condimentando carne");
}
public void Mostrar_Informacion()
{
System.out.println();
System.out.println("Informacion de Bistec");
super.Mostrar_Informacion();
System.out.println("Corte de bistec: "+corte);
}
public void Mensaje(){
    System.out.println("el bistec contiene mucha grasa");
}
}

```

Hamburguesa.java

```
import java.io.*;
import java.util.*;
public class Hamburguesa extends Carne implements NoSaludable
{
    String tipo;
    public Hamburguesa()
    {
    }
    public Hamburguesa(double pre, double cal, double pes, double grasa, double coles, String tip)
    {
        super.precio=pre;
        super.calorias=cal;
        super.peso=pes;
        super.grsGrasa=grasa;
        super.colesterol=coles;
        tipo=tip;
    }
    public Hamburguesa Obtener_Informacion()
    {
        Scanner Leer=new Scanner(System.in);

        System.out.println("Dame tipo: ");
        tipo=Leer.nextLine();
        System.out.println("Dame precio: ");
        super.precio=Leer.nextDouble();
        System.out.println("Dame calorias: ");
```

```
super.calorias=Leer.nextDouble();
System.out.println("Dame peso: ");
super.peso=Leer.nextDouble();
System.out.println("Dame los gramos de grasa: ");
super.grsGrasa=Leer.nextDouble();
System.out.println("Dame cantidad de colesterol: ");
super.colesterol=Leer.nextDouble();

Hamburguesa objeto4=new Hamburguesa(precio,calorias,peso,grsGrasa,colesterol,tipo);

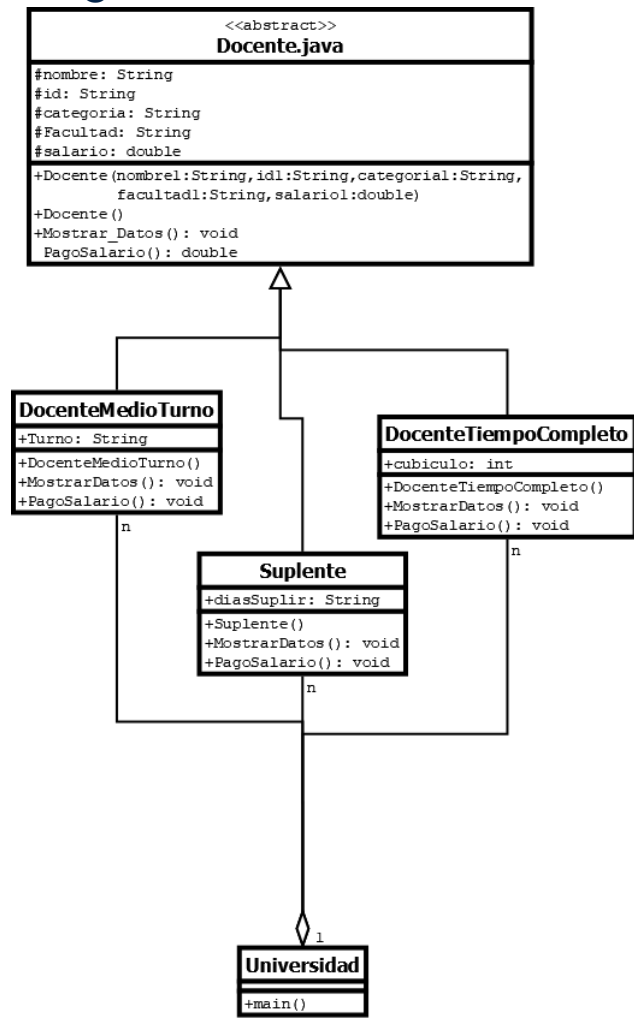
return objeto4;
}
public void comer()
{
System.out.println("Comiendo hamburguesa");
}
public void empanizar()
{
System.out.println("Empanizando carne para hamburguesa");
}
public void cortar()
{
System.out.println("Cortando carne");
}
public void freir()
{
System.out.println("Friendo carne");
}
public void Mostrar_Informacion()
{
System.out.println();
}
```

```
System.out.println("Informacion de hamburguesa");
super.Mostrar_Informacion();
System.out.println("tipo de hamburguesa: "+tipo);
}

public void Mensaje(){
    System.out.println("las hamburguesas son poco saludables");
}
}
```

Docentes

Diagrama ULM



Código

Docente.java

```
/**
 * @(#)Docente.java
 *
 *
 * @author
 * @version 1.00 2015/10/7
 */

import java.util.Scanner;

abstract public class Docente
{

    protected String Nombre;
    protected String id;
    protected String Categoria;
    protected String Facultad;
    protected double salario;

    public Docente(String Nombre1,String id1,String Categoria1,String Facultad1, double salario1)
    {
        Nombre=Nombre1;
        id=id1;
        Categoria=Categoria1;
    }
}
```

```

        Facultad=Facultad1;
        salario=salario1;
    }

    public Docente()
    {
        String Nombre1,id1,Categoria1,Facultad1;
        double salario1;
        Scanner leer=new Scanner(System.in);
        System.out.println("Ingresar Nombre");
        Nombre1=leer.next();
        System.out.println("Ingresa No_ trabajador");
        id1=leer.next();
        System.out.println("Ingresar Categoria");
        Categoria1=leer.next();
        System.out.println("Ingresa Facultad");
        Facultad1 =leer.next();
        System.out.println("Ingresar Salario");
        salario1=leer.nextDouble();
        Nombre=Nombre1;
        id=id1;
        Categoria=Categoria1;
        Facultad=Facultad1;
        salario=salario1;
    }

    public void Mostrar_Datos()
    {

        System.out.println("Nombre"+Nombre);
    }

```

```

        System.out.println("Numero de Trabajador"+id);
        System.out.println("Categoria"+Categoria);
        System.out.println("Facultad"+Facultad);
        System.out.println("Salario"+ salario);

    }

    abstract public double PagoSalario();

}

```

Suplente.java

```

import java.util.*;
public class Suplente extends Docente{
    int diasSuplir;

    Scanner leer=new Scanner(System.in);

    public Suplente()
    {
        super();
        System.out.println("Ingresar el numero de dias que va a suplir el docente");
        diasSuplir=leer.nextInt();
    }

    public void Mostrar_Datos()
    {

```

```

    super.Mostrar_Datos();
    System.out.println("Dias a suplir "+ diasSuplir);

}

public double PagoSalario()
{
    double total=0.0;
    int dias=0;
    System.out.println (" Dame los dias trabajados 1-7");
    dias=leer.nextInt();
    return total=(super.salarior*dias);

}
}

```

DocenteMedioTurno.java

```

/**
 * @(#)DocenteMedioTiempo.java
 *
 *
 * @author
 * @version 1.00 2015/10/9
 */
import java.util.Scanner;

```

```

public class DocenteMedioTiempo extends Docente {

    String Turno;

    Scanner leer=new Scanner(System.in);

    public DocenteMedioTiempo()
    {
        super();
        System.out.println("Ingresar el turno del Catedratico");
        Turno=leer.next();
    }

    public void Mostrar_Datos()
    {

        super.Mostrar_Datos();
        System.out.println("Turno"+Turno);

    }

    public double PagoSalario()
    {

        double total=0.0;
        int dias=0;
        System.out.println (" Dame los dias trabajados 1-7");
        dias=leer.nextInt();
    }
}

```

```
        return total=(super.salario*dias);  
    }  
}
```

DocenteTiempoCompleto.java

```
/**  
 * @(#)DocenteTiempoCompleto.java  
 *  
 *  
 * @author  
 * @version 1.00 2015/10/7  
 */  
import java.util.Scanner;  
  
public class DocenteTiempoCompleto extends Docente  
{  
    int cubiculo;  
  
    Scanner leer=new Scanner(System.in);  
  
    public DocenteTiempoCompleto()  
    {  
  
        super();  
        Scanner leer=new Scanner(System.in);  
        System.out.println("Ingresar No de cubiculo del Catedratico");  
        cubiculo=leer.nextInt();  
    }  
}
```

```

    }
    public void Mostrar_Datos()
    {

        super.Mostrar_Datos();
        System.out.println("Cubiculo"+cubiculo);

    }

    public double PagoSalario()
    {

        double total=0.0;
        int dias=0;
        System.out.println (" Dame los dias trabajados 1-15");
        dias=leer.nextInt();
        return total=(super.salario*dias);

    }

}

```

Universidad.java

```

import java.util.Scanner;

public class Universidad
{

```

```

public static void main(String a[])
{

    int op,i;
    double total=0;
    Docente maestros[]=new Docente[2];
    Scanner leer=new Scanner(System.in);
    for(i=0;i<2;i++){

        System.out.println("Escoja el tipo de docente: \n1.-Tiempo completo\n2.-Medio tiempo\n3.-Suplente");
        op=leer.nextInt();
        switch (op)
        {
            case 1: maestros[i]=new DocenteTiempoCompleto();
                    maestros[i].Mostrar_Datos();
                    System.out.println("Pago Total"+maestros[i].PagoSalario());
                    break;

            case 2: maestros[i]=new DocenteMedioTiempo();
                    maestros[i].Mostrar_Datos();
                    System.out.println("Pago Total"+maestros[i].PagoSalario());
                    break;

            case 3: maestros[i]=new Suplente();
                    maestros[i].Mostrar_Datos();
                    System.out.println("Pago Total"+maestros[i].PagoSalario());
                    break;

            default: System.out.println("opcion invalida");
        }
    }
    for(i=0;i<2;i++){
        total+=maestros[i].PagoSalario();
    }
}

```



```

    }
    System.out.println("Pago Total"+total);

}
}

```

```

PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica9Polimorfismo\Docente> java Universidad
Escoja el tipo de docente:
1.-Tiempo completo
2.-Medio tiempo
3.-Suplente
1
Ingresar Nombre
juan
Ingresar No_ trabajador
1
Ingresar Categoria
matematicas
Ingresar Facultad
medicina
Ingresar Salario
100
Ingresar No de cubiculo del Catedratico
251
Nombrejuan
Numero de Trabajador1
Categoriamatematicas
Facultadmedicina
Salario100.0
Cubiculo251
Dame los dias trabajados 1-15
10
Pago Total1000.0

```

```
1.-Tiempo completo
2.-Medio tiempo
3.-Suplente
3
Ingresar Nombre
pepe
Ingresar No_ trabajador
2
Ingresar Categoria
fisica
Ingresar Facultad
ciencias
Ingresar Salario
200
Ingresar el numero de dias que va a suplir el docente
5
Nombrepepe
Numero de Trabajador2
Categoriafisica
Facultadciencias
Salario200.0
Dias a suplir 5
  Dame los dias trabajados 1-7
5
Pago Total1000.0
  Dame los dias trabajados 1-15
1
  Dame los dias trabajados 1-7
2
Pago Total500.0
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica9Polimorfismo\Docente> |
```

Conclusión

Esta practica nos ayudo a reforzar los conocimiento previamente vistos de herencia, interfaces, y clases abstractas.

Hoja de firmas

RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS

MATRICULA: 201963582 FECHA: 24/01/2021

NOMBRE: Xicale Cabrera Irvyn No. PRACTICA: 1 INDIVIDUAL (x) COLABORATIVA ()

CRITERIOS	Deficiente	Suficiente	Bueno	Excelente	Calificación Obtenida	
	1-5.9	6-7.9	8-9	9.1-10	%	Puntos
CONOCIMIENTO TEÓRICO 20%	Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases.	Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto.	Conocimiento claro de los fundamentos teóricos POO pero requiere mejorar el modelado de las clases, sus métodos y sus relaciones entre clases.	Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase.		
EJECUCIÓN DE LA PRÁCTICA 30%	No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java.	Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java	Realiza la práctica pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación(sintaxis y semántica)	Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica).		

SOLUCIÓN DE LA PRÁCTICA 40%	No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación.	Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación	Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje.	Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación.		
ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO 10%	No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo.	Posee una actitud propositiva para un aprendizaje autónomo, participa pero no le gusta trabajar en equipo.	Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo pero requiere mejorar su participación y portaciones de forma profunda.	Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo		
Total						

Firma del Alumn@

Vo. Bo. Docente

