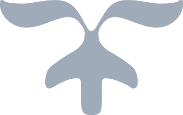


practica 3B

Facultad de ciencias de la computación

Ing. En ciencias de la computación



Irvyn xicale cabrera

201963582

# Introducción

Esta practica nos ayuda a comprender el lenguaje java además de dar inicio a los principios de la programación orientada a objetos.

# Diagramas ULM

Diagrama

Descripción generada automáticamente

# Código

Ala.java

public class ala {

    Double tama;

    // constructor

    public ala(){

    }

    // metodos

    public ala(Double tama1){

        tama = tama1;

    }

    public Double verTama(){

        return tama;

    }

}

Pata.java

public class pata {

    String color;

    // constructor

    public pata(String col){

        color = col;

    }

    // metodos

    public String verColor(){

        return color;

    }

}

Pico.java

public class pico {

    String largo;

    String color;

    // constructor

    public pico(){

    }

    // metodos

    public pico(String lar, String col){

        largo = lar;

        color = col;

    }

    public String verLargo(){

        return largo;

    }

    public String verColor(){

        return color;

    }

}

Pluma.java

public class pico {

    String largo;

    String color;

    // constructor

    public pico(){

    }

    // metodos

    public pico(String lar, String col){

        largo = lar;

        color = col;

    }

    public String verLargo(){

        return largo;

    }

    public String verColor(){

        return color;

    }

}

Gallina.java

public class gallina {

    static int numGallina=1;

    pluma plumas;

    ala ala1, ala2;

    pico pico;

    pata patas;

    Double cAla1, cAla2;

    // constructor

    public gallina(String colPluma, int numPluma,Double tamAla,Double tamAla1,String larPico,String colPico,String colPata){

        plumas = new pluma(colPluma,numPluma);

        ala1 = new ala(tamAla);

        ala2 = new ala(tamAla1);

        pico = new pico(larPico,colPico);

        patas = new pata(colPata);

    }

    // metodos

    public void verInfo() {

        System.out.println("informacion de la gallina " + numGallina);

        System.out.println("numero de plumas: " + plumas.verCantidad());

        System.out.println("color de las plumas " + plumas.verColor());

        System.out.println("el tamaño de la ala 1 es: " + ala1.verTama());

        System.out.println("el tamaño de la ala 2 es: " + ala2.verTama());

        System.out.println("el largo de su pico es: " + pico.verLargo());

        System.out.println("el color de su pico es: " + pico.verColor());

        System.out.println("el color de sus patas es: " + patas.verColor());

        System.out.println("");

        numGallina++;

    }

}

Gallinero.java

public class gallinero {

    gallina gallina1,gallina2,gallina3,gallina4,gallina5;

    // constructor

    public gallinero(){

    }

    // metodos

    public void gallinero1(){

        gallina1 = new gallina("rojo",400,50.5,50.0,"grande","negro","negro");

        gallina2 = new gallina("negro",450,52.5,52.5,"pequeño","amarillo","amarillo");

        gallina3 = new gallina("rojo",500,60.5,55.5,"grande","blanco","negro");

        gallina4 = new gallina("blanco",600,55.5,55.5,"pequeño","amarillo","amarillo");

        gallina5 = new gallina("rojo",550,40.0,40.0,"grande","negro","negro");

    }

    public void verGallinas(){

        System.out.println("");

        gallina1.verInfo();

        gallina2.verInfo();

        gallina3.verInfo();

        gallina4.verInfo();

        gallina5.verInfo();

    }

}

Ejecuta.java

public class ejecuta {

    public static void main(String[] args){

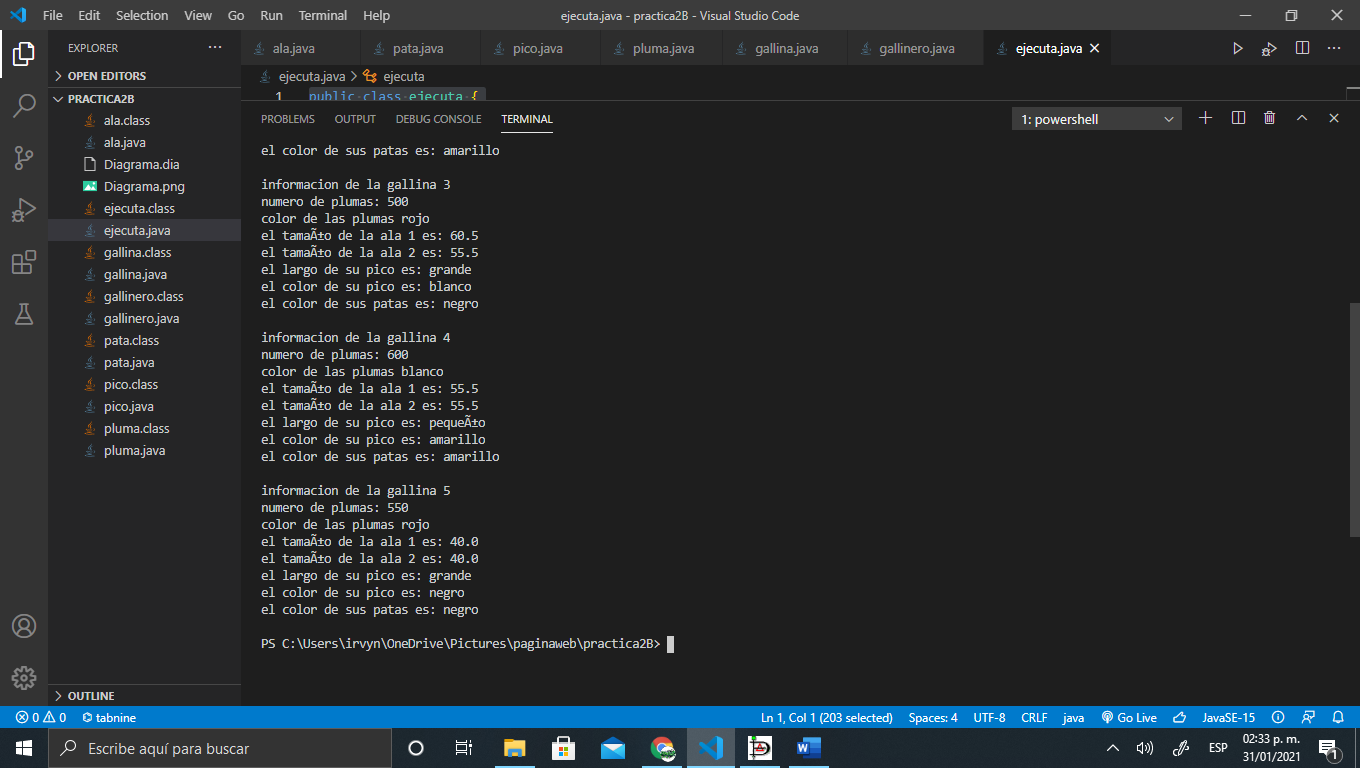
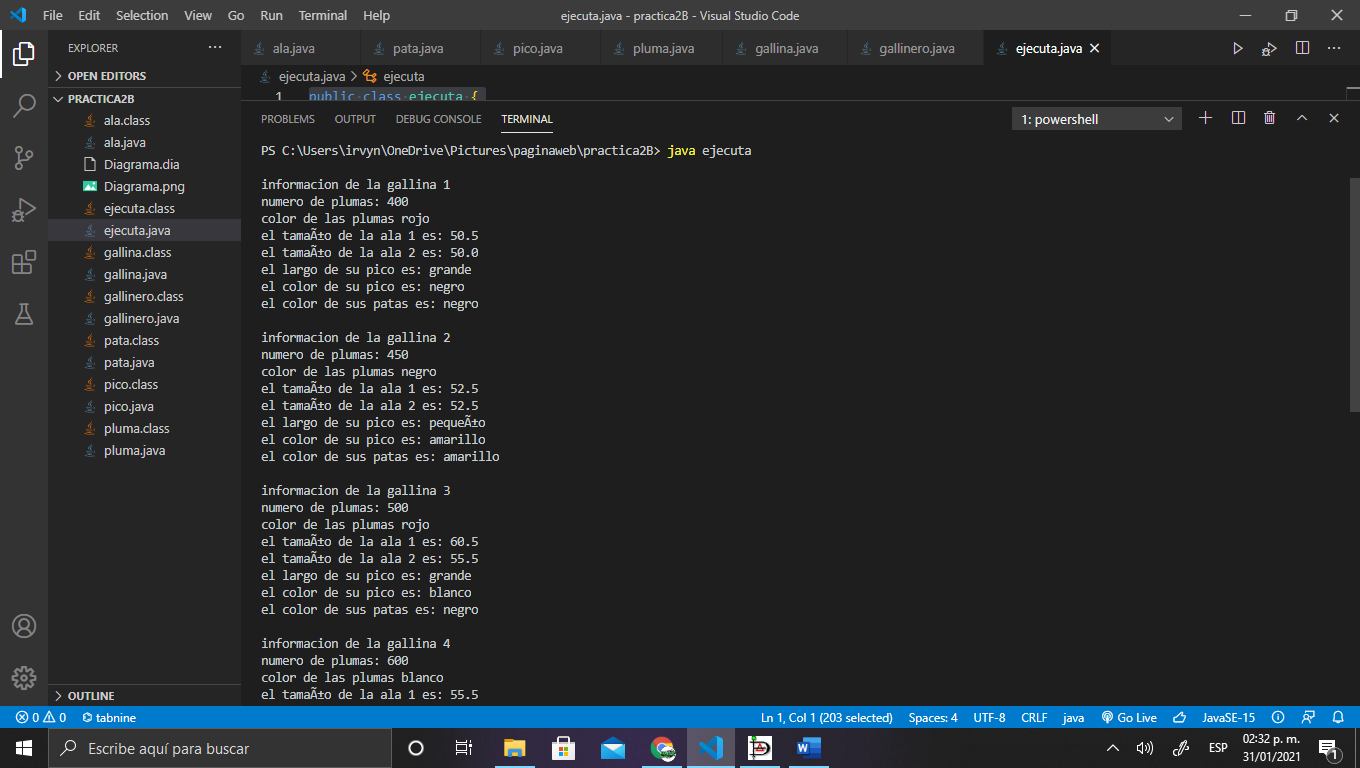
        gallinero objgallinero = new gallinero();

        objgallinero.gallinero1();

        objgallinero.verGallinas();

    }

}



# Conclusión

Esta practica nos ayudo a comprender la programación orientada a objetos

# 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 ( )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Deficiente** | **Suficiente** | **Bueno** | **Excelente** | **Calificación Obtenida** | |
| **CRITERIOS** | **1-5.9** | **6‐7.9** | **8‐9** | **9.1‐10** |
| **%** | **Puntos** |
| **CONOCIMIENTO TEORICO**  **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. |  |  |
| **EJECUCCIÓN DE LA PRACTICA**  **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 PRACTICA**  **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**

