



---

# PRACTICA 5

---

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

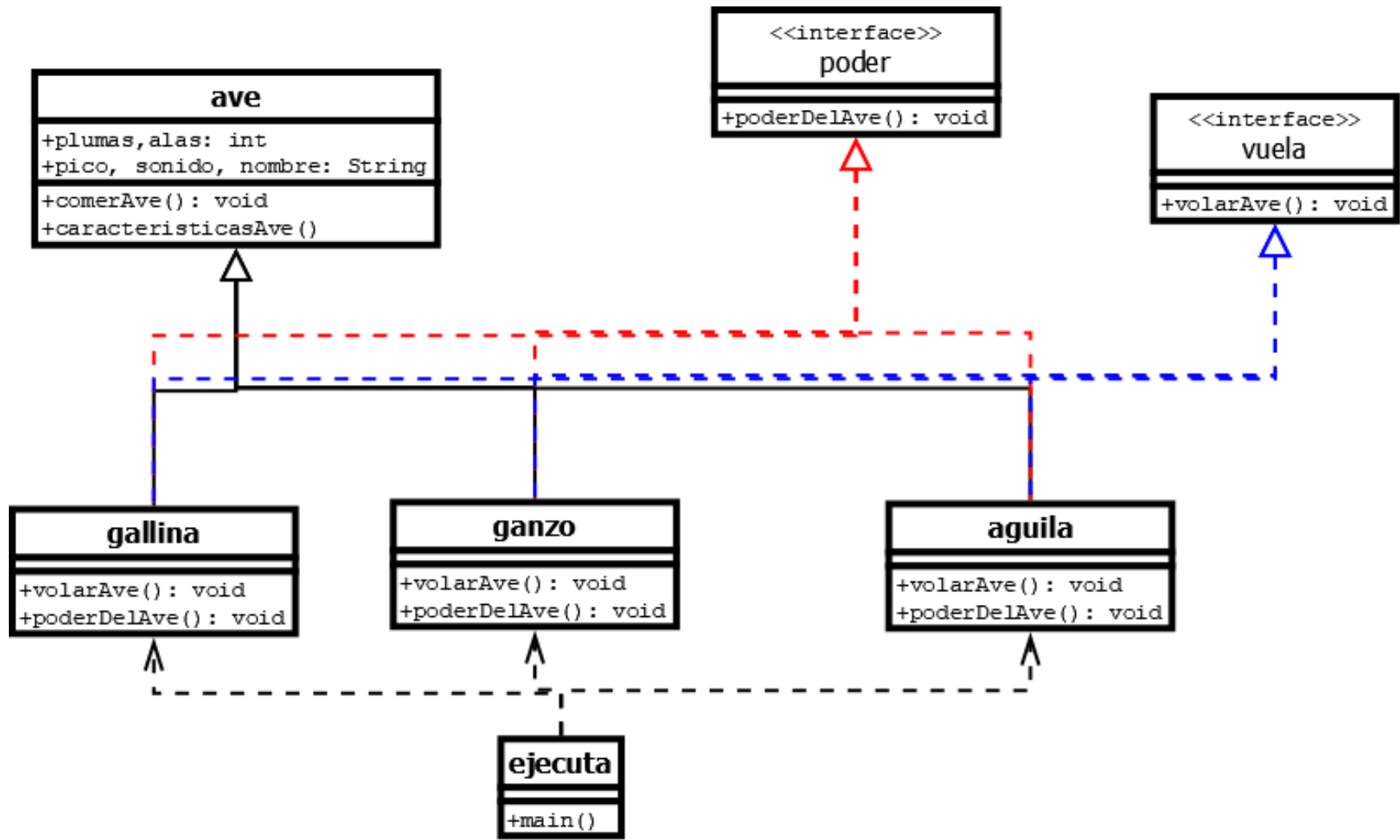


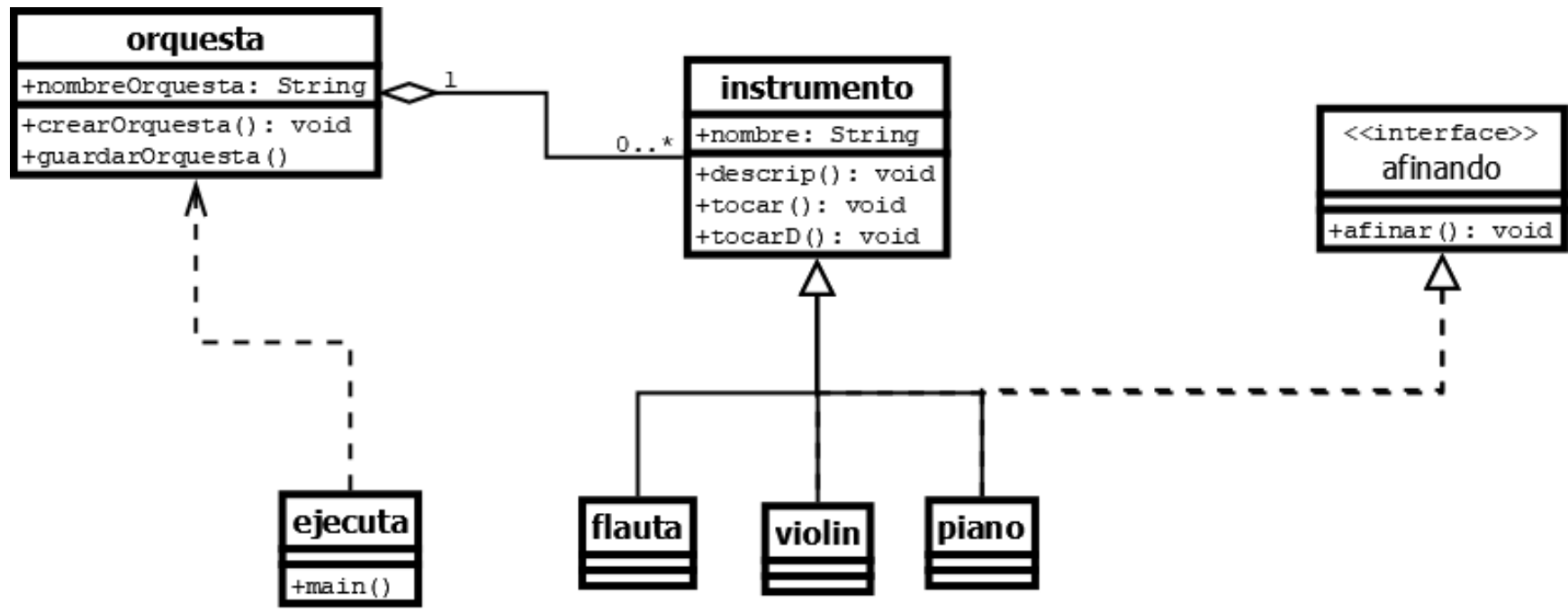
IRVYN XICALE CABRERA  
201963582

## Introducción

Esta práctica nos ayuda a comprender el uso de las interfases y su utilidad en la programación orientada a objetos.

## Diagramas ULM





## Código

Ave.java

```

public class ave {
    int plumas;
    int alas;
    int numAve;
    String pico;
    String sonido;
    String nombre;
    static int cont=0;
}
  
```

```

public ave(int plumas,int alas, String pico, String sonido, String nombre)
{
    this.plumas=plumas;
    this.alas=alas;
    this.pico=pico;
    this.sonido=sonido;
    this.nombre=nombre;
    cont++;
    // numAve=cont;
}
public ave()
{ }
public void comerAve()
{
    System.out.println("");
    System.out.println("Toda ave come semillas ");
}
public void caracteristicasAve()
{
    System.out.println("Ave "+cont);
    System.out.println("Nombre:"+nombre);
    System.out.println("Tam del pico:"+pico);
    System.out.println("Tipo de sonido:"+sonido);
    System.out.println("");
}
}

```

Águila.java

```
public class aguila extends ave implements vuela {
    public aguila(int plumas, int alas, String pico, String sonido, String nombre)
    {
        super(plumas, alas, pico, sonido, nombre);
        super.comerAve();
        super.caracteristicasAve();
    }
    public void volarAve()
    {
        System.out.println("El aguila " + super.nombre + " esta volando");
    }
    public void poderDelAve(){
        System.out.println("El aguila " + super.nombre + " puede detener el tiempo y espacio a voluntad");
        System.out.println("mejor no la molestamos \n");
    }
}
```

Gallina.java

```
public class gallina extends ave implements vuela {
    public gallina(int plumas, int alas, String pico, String sonido, String nombre)
    {
        super(plumas, alas, pico, sonido, nombre);
        super.comerAve();
        super.caracteristicasAve();
    }
    public void volarAve()
    {
        System.out.println("La gallina " + super.nombre + " esta volando");
    }
}
```

```

    }
    public void poderDelAve(){
        System.out.println("la gallina " + super.nombre + " tiene superfuerza y des-
troza todo a su paso >:( \n");
    }
}

```

Ganzo.java

```

public class ganzo extends ave implements vuela {
    public ganzo(int plumas, int alas, String pico, String sonido, String nombre)
    {
        super(plumas, alas, pico, sonido, nombre);
        super.comerAve();
        super.caracteristicasAve();
    }
    public void volarAve()
    {
        System.out.println("El ganzo " + super.nombre + " esta volando");
    }
    public void poderDelAve(){
        System.out.println("El ganzo " + super.nombre + " tiene supervelocidad por que tiene los te-
nis del rayo mcqueen \n");
    }
}

```

Poder.java

```

public interface poder {
    abstract public void poderDelAve();
}

```



```
}
```

Vuela.java

```
public interface vuela {  
    abstract public void volarAve();  
}
```

Ejecuta.java

```
public class ejecuta {  
    public static void main(String[] args){  
        gallina gallina1 = new gallina(650,2,"Mediano","ko koro kooooo","Maria");  
        gallina1.volarAve();  
        gallina1.poderDelAve();  
  
        ganzo ganzo1 = new ganzo(720,2,"grande","cuak cuak cuak cuak cuak","pedro");  
        ganzo1.volarAve();  
        ganzo1.poderDelAve();  
  
        aguila aguila1 = new aguila(720,2,"grande","kaaaaa ","mariano");  
        aguila1.volarAve();  
        aguila1.poderDelAve();  
    }  
}
```

```
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica5\aves> java ejecuta
```

```
Toda ave come semillas
```

```
Ave 1
```

```
Nombre: Maria
```

```
Tam del pico: Mediano
```

```
Tipo de sonido: ko koro kooooo
```

```
La gallina Maria esta volando
```

```
la gallina Maria tiene superfuerza y destroza todo a su paso >:(
```

```
Toda ave come semillas
```

```
Ave 2
```

```
Nombre: pedro
```

```
Tam del pico: grande
```

```
Tipo de sonido: cuak cuak cuak cuak cuak
```

```
El ganzo pedro esta volando
```

```
El ganzo pedro tiene supervelocidad por que tiene los tenis del rayo mcqueen
```

```
Toda ave come semillas
```

```
Ave 3
```

```
Nombre: mariano
```

```
Tam del pico: grande
```

```
Tipo de sonido: kaaaaa
```

```
El aguila mariano esta volando
```

```
El aguila mariano puede detener el tiempo y espacio a voluntad  
mejor no la molestamos
```

```
PS C:\Users\irvyn\OneDrive\Pictures\paginaweb\practica5\aves> |
```

## ORQUESTA

Afinando.java

```
public interface afinando {
```

```
    public void afinar();  
}
```

Instrumento.java

```
public class instrumento{  
    String nombre;  
  
    public instrumento(String nombre){  
        this.nombre = nombre;  
    }  
    public void descrip(){  
        System.out.println("nombre: " + nombre);  
    }  
    public void tocar(){  
        System.out.println("suena muy bien");  
    }  
    public void tocarD(){  
        System.out.println("suena muy desafinado");  
    }  
}
```

Flauta.java

```
public class flauta extends instrumento{  
    public flauta(String nombre) {  
        super(nombre);  
    }  
}
```

```
        System.out.println("el flautista " + nombre);
    }
}
```

Piano.java

```
public class piano extends instrumento implements afinando{
    public piano(String nombre){
        super(nombre);
        System.out.println("el pianista " + nombre);
    }
    public void afinar(){
        System.out.println("se esta afinando el piano de " + super.nombre);
        System.out.println("ya esta afinando");
    }
}
```

Violín.java

```
public class violin extends instrumento implements afinando {
    public violin(String nombre) {
        super(nombre);
        System.out.println("el violinista " + nombre);
    }
    public void afinar(){
        System.out.println("se esta afinando el instrumento de " + super.nombre);
        System.out.println("ya esta afinando");
    }
}
```

```
}  
}
```

Orquesta.java

```
public class orquesta {  
    String nombreOrquesta;  
    public orquesta(String nombre) {  
        nombreOrquesta = nombre;  
    }  
    public void CrearOrquesta(){  
        piano piano = new piano("jorge");  
        piano.tocarD();  
        piano.afinar();  
        piano.tocar();  
        System.out.println("");  
  
        flauta flauta= new flauta("mario");  
        flauta.tocar();  
        System.out.println("");  
  
        violin violin= new violin("pablo");  
        violin.tocarD();  
        violin.afinar();  
        violin.tocar();  
        System.out.println("");  
    }  
  
    public void guardarOrquesta(){
```

```
        System.out.println("se a guardado la orquesta " + nombreOrquesta);  
    }  
}
```

Ejecuta.java

```
public class ejecuta {  
    public static void main(String[] args){  
        orquesta objeto1 = new orquesta("los norteros");  
        objeto1.CrearOrquesta();  
        objeto1.guardarOrquesta();  
    }  
}
```

## 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 ( ☒ ) COLABORATIVA ( ☐ )

CRITERIOS	Deficiente	Suficiente	Bueno	Excelente	Calificación Obtenida	
	1-5.9	6-7.9	8-9	9.1-10	%	Puntos
<b>CONOCIMIENTO TEÓRICO</b> 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.		
<b>EJECUCIÓN DE LA PRÁCTICA</b> 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).		

<b>SOLUCIÓN DE LA PRÁCTICA</b>  <b>40%</b>	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.		
<b>ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO</b>  <b>10%</b>	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		
<b>Total</b>						

Firma del Alumn@

Vo. Bo. Docente

