



# **“ANÁLISIS Y DISEÑO DE SISTEMAS I”**

TRABAJO PRÁCTICO N° 7

## **Diagramas de Clases**

Prof. Lic. Paszco Gustavo Ariel

Año: 2022

---

a) OBJETIVOS GENERALES

- Que el alumno comprenda las características y alcances del uso de diagramas de clases dentro del Lenguaje Unificado de Modelado.
- Que el alumno aprenda los fundamentos de la programación orientada a objetos, así como la introducción de estos conceptos en el desarrollo de software y los objetivos buscados por los métodos de diseño orientado a objetos.

b) OBJETIVOS ESPECIFICOS

- Utilizar el lenguaje PHP como herramienta para la programación orientada a objetos.
- Conocer con detalle sus características relativas a la POO, y como herramienta para implementar interfaces gráficas de usuario.
- Desarrollar capacidades de programación e implementación de problemas, adecuadas a los requerimientos de las prácticas.

## EJERCICIO Nº 1

```
import javax.swing.*;
public class variables {
    private int lado;
    private double base;
    private double altura;
    private double area;
    private double radio;
    private double pi=3.1416;

    public void cuadrado(){
        lado=Integer.parseInt(JOptionPane.showInputDialog("Dame el lado"));
        area=lado*lado;
        JOptionPane.showMessageDialog(null,"El área del Cuadrado es "+ area);
    }

    public void triangulo(){

        base=Double.parseDouble(JOptionPane.showInputDialog("Dame la base"));
        altura=Double.parseDouble(JOptionPane.showInputDialog("Dame la altura"));
        area=(base*altura)/2;
        JOptionPane.showMessageDialog(null,"El área del Triángulo es "+area);
    }

    public void Circulo(){
        radio=Double.parseDouble(JOptionPane.showInputDialog("¿Qué radio tienes?"));
        area=pi*Math.pow(radio,2);
        JOptionPane.showMessageDialog(null, "El área es "+ area);
    }

    public void rectangulo(){
        base=Double.parseDouble(JOptionPane.showInputDialog("Dame la base"));
        altura=Double.parseDouble(JOptionPane.showInputDialog("Dame la altura"));
        area=(base*altura);
        JOptionPane.showMessageDialog(null,"El área del ectángulo es "+area);
    }
}
```

Dibujar el diagrama

## EJERCICIO Nº 2

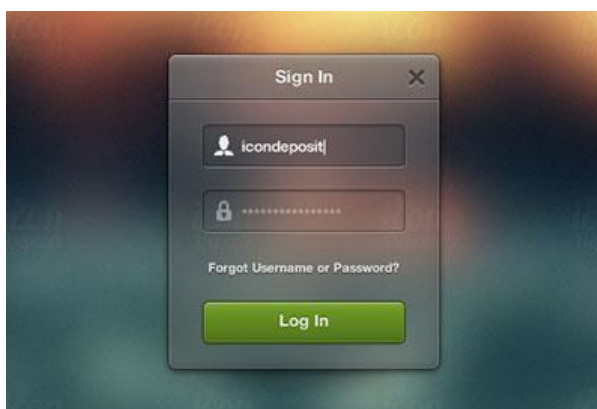
Un zoológico necesita una aplicación informática para llevar su organización respecto a las especies que posee, los empleados (cuidadores y guías), y los distintos itinerarios de visita que ofrece. La información está estructurada de la siguiente manera:

- Especies: de las especies interesa saber el nombre en español, el nombre científico y una descripción general. Hay que tener en cuenta que una especie puede vivir en diferentes hábitats naturales y que un hábitat puede ser ocupado por diferentes especies. Las especies se encuentran en distintas zonas del parque de manera que cada especie está en una zona y en una zona hay varias especies.
- Hábitats: los diferentes hábitats naturales vienen definidos por el nombre, el clima y el tipo de vegetación predominantes, así como el continente o continentes en los que se encuentran.

- Zonas: las zonas del parque en las que se encuentran las distintas especies vienen definidas por el nombre y la extensión que ocupan.
- Itinerarios: los itinerarios discurren por distintas zonas del parque. La información de interés para los itinerarios es: código de itinerario, la duración del recorrido, la longitud del itinerario, el máximo número de visitantes autorizado y el número de distintas especies que visita. Hay que tener en cuenta que un itinerario recorre distintas zonas del parque y que una zona puede ser recorrida por diferentes itinerarios.
- Guías: los guías del parque vienen definidos por el nombre, dirección, teléfono y fecha en la que comenzaron a trabajar en el zoo. Interesa saber qué guías llevan qué itinerarios, teniendo en cuenta que un guía puede llevar varios itinerarios y que un itinerario puede ser asignado a diferentes guías en diferentes horas, siendo éstas un dato de interés.
- Cuidadores: los cuidadores vienen definidos por el nombre, dirección, teléfono y fecha de ingreso en el parque. Hay que tener en cuenta que un cuidador puede estar a cargo de varias especies y que una especie puede ser atendida por varios cuidadores, siendo de interés la fecha en la que un cuidador se hace cargo de una especie.

### EJERCICIO Nº 3

Dado el formulario siguiente, realizar un diagrama de clases que dibuje el contenido: barra de título, campos y etiquetas, botones, íconos, marco, funciones, entre otros.



### EJERCICIO Nº 4

El Ministerio de Defensa desea diseñar una Base de Datos para llevar un cierto control de los soldados que realizan el servicio militar. Los datos significativos a tener en cuenta son:

- Un soldado se define por su código de soldado (único), su nombre y apellidos, y su graduación.
- Existen varios cuarteles, cada uno se define por su código de cuartel, nombre y ubicación.
- Hay que tener en cuenta que existen diferentes Cuerpos del Ejército (Infantería, Artillería, Armada, etc), y cada uno se define por un código de Cuerpo y denominación.
- Los soldados están agrupados en compañías, siendo significativa para cada una de éstas, el número de compañía y la actividad principal que realiza.
- Se desea controlar los servicios que realizan los soldados (guardias, imaginarias, cuarteros, ...), y se definen por el código de servicio y descripción.

Consideraciones de diseño:

- Un soldado pertenece a un único cuerpo y a una única compañía, durante todo el servicio militar. A una compañía pueden pertenecer soldados de diferentes cuerpos, no habiendo relación directa entre compañías y cuerpos.
- Los soldados de una misma compañía pueden estar destinados en diferentes cuarteles, es decir, una compañía puede estar ubicada en varios cuarteles, y en un cuartel puede haber varias compañías. Eso sí, un soldado sólo está en un cuartel.

Un soldado realiza varios servicios a lo largo de la milicia. Un mismo servicio puede ser realizado por más de un soldado (con independencia de la compañía), siendo significativa la fecha de realización.

## **EJERCICIO Nº 5**

La Policía quiere crear una base de datos sobre la seguridad en algunas entidades bancarias. Para ello tiene en cuenta:

- Que cada entidad bancaria se caracteriza por un código y por el domicilio de su Central.
- Que cada entidad bancaria tiene más de una sucursal que también se caracteriza por un código y por el domicilio, así como por el número de empleados de dicha sucursal.
- Que cada sucursal contrata, según el día, algunos vigilantes jurados, que se caracterizan por un código y su edad. Un vigilante puede ser contratado por diferentes sucursales (incluso de diferentes entidades), en distintas fechas y es un dato de interés dicha fecha, así como si se ha contratado con arma o no.

- Por otra parte, se quiere controlar a las personas que han sido detenidas por atracar las sucursales de dichas entidades. Estas personas se definen por una clave (código) y su nombre completo.
- Algunas de estas personas están integradas en algunas bandas organizadas y por ello se desea saber a qué banda pertenecen, sin ser de interés si la banda ha participado en el delito o no. Dichas bandas se definen por un número de banda y por el número de miembros.
- Así mismo, es interesante saber en qué fecha ha atracado cada persona una sucursal. Evidentemente, una persona puede atracar varias sucursales en diferentes fechas, así como que una sucursal puede ser atracada por varias personas.
- Igualmente, se quiere saber qué Juez ha estado encargado del caso, sabiendo que un individuo, por diferentes delitos, puede ser juzgado por diferentes jueces. Es de interés saber, en cada delito, si la persona detenida ha sido condenada o no y de haberlo sido, cuánto tiempo pasará en la cárcel. Un Juez se caracteriza por una clave interna del juzgado, su nombre y los años de servicio.

NOTA: En ningún caso interesa saber si un vigilante ha participado en la detención de un atracador.

## EJERCICIO Nº 6

```
<?php
class Vehiculo{
    protected $gasolina = 100;
}

class Nissan extends Vehiculo{
    public function avanzar(){
        echo "Ahora el Nissan tiene ".$this->gasolina -= 20;
    }
}

class Suzuki extends Vehiculo{
    public function avanzar(){
        echo "Ahora el Suzuki tiene ".$this->gasolina -= 8;
    }
}

class Honda extends Vehiculo{
    public function avanzar(){
        echo "Ahora el Honda tiene ".$this->gasolina -= 15;
    }
}

class Usuario {
    private $vehiculo;
    public function __construct($objeto){
        $this->Vehiculo = $objeto;
    }

    public function avanzarVehiculo(){
        $this->Vehiculo->avanzar();
    }
}
?>
```

Dibujar el diagrama de clases

- *Diagramar usando herramientas CASE*

Fecha Entrega: 15/19/22

Prof. Lic. Gustavo Ariel Paszco