

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Facultad de Ciencias Químicas e Ingeniería

Programas de Ingeniero en Computación e Ingeniero en Software y Tecnologías Emergentes

INFORMACIÓN DE LA MATERIA

Nombre de la materia y clave: Lenguajes de Programación Orientado a Objetos (40006).
Grupo y periodo: 341 (2022-2)
Profesor: Manuel Castañón Puga.

INFORMACIÓN DE LA ACTIVIDAD

Nombre de la actividad: Actividad de taller 3.1.1: Modelado de la relación de herencia entre clases.
Lugar y fecha: A 13 de octubre de 2022 en el Edificio 6E, Salón 204.
Carácter de la actividad: Individual
Participante(es): Diego Andrés González Beltrán

REPORTE DE ACTIVIDADES

Objetivo de la actividad:

En esta actividad se tiene como objetivo el modelado de las relaciones entre clases utilizando distintas relaciones como PartOf e IsA. En esta práctica se continuó el programa sobre el ecosistema añadiendo conceptos como estos.

Primeramente añadí nuevas relaciones como IsA, es decir, generalización de las clases Planta y Animal para heredar atributos y los respectivos setters y getters a dos clases nuevas, Arbol y Dog, heredando de Planta y otro de Animal respectivamente.

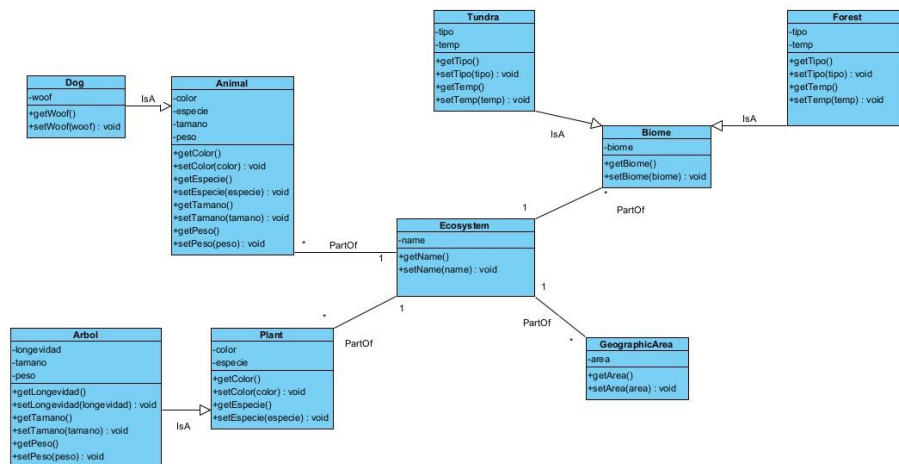


Figura 1. Diagrama de clases

Sin embargo, al ser clases nuevas heredando de otra (como se puede ver en la figura 2), se requieren de nuevos atributos específicos con sentido a la clase, por ejemplo en Arbol se añadieron longevidad, tamaño y peso. Y para probarlo se utilizó Secuoya, 80 (m), y 2000 (toneladas).

```
public class Arbol extends Plant{
    private int longevidad;
    private Integer tamano, peso;

    public void setLongevidad(int longevidad) {
        this.longevidad = longevidad;
    }

    public int getLongevidad() {
        return longevidad;
    }
}
```

Figura 2. Arbol extends Plant (herencia)

Además, se agregaron dos nuevas clases, tundra y forest. Tundra acepta los tipos, artico y alpino. Y forest, templado y boreal. Y ambos tienen temperaturas.

Y finalmente se puede observar un resultado de prueba con valores insertados desde el método main.

```
[Running] cd "d:\Users\diego\Documents\Programming_Projects\LP00\T2.2.1_ModeladoRelacionesClases\Ecosystem\" && javac Ecosystem.java && java Ecosystem
Acuatico
Ballena
Azul
130000 kg
24 m
Hemisferio sur
Oceano
Blanco
Daisy

Color: Cafe y blanco
Especie: Beagle
Tamano: 110
Peso: 30
Woof!

Color: Cafe rojizo
Especie: Secuoya
Longevidad: 1800 anios

Tundra tipo: artico
Temperatura: -32.4 Celsius

Bosque tipo: tropical
Temperatura: 36.2 Celsius
```

Figura 3. Prueba del código

Enlace del repositorio de GitHub:

https://github.com/DiegoAndresGlez/T3.1.1_ModeladoRelacionesClases

RESUMEN/REFLEXIÓN/CONCLUSIÓN

En esta actividad se utilizaron conceptos de relaciones entre clases para la continuación del diseño del diagrama de clases y actualizando el código según al nuevo diagrama.

Aprendí a utilizar y traducir las relaciones que existen entre clases desde su diseño para actualizar el código de Ecosistema.

Concluyó que saber sobre las relaciones entre clases va ser de gran ayuda para el entendimiento de la programación orientada a objetos, ya que con buen diseños de clases y de sus relaciones el código será de mejor eficacia y fácil de entender. Además de entender lo que es la generalización ya que es importante en POO.

Doy fe de que toda la información dada es completa y correcta.

Diego Andres Gonzalez Beltran

