

Laboratorio II

Principios y practicas de desarrollo de software orientado a objetos

Docente:

Dario Alejandro Riaño Velandia

Estudiantes:

Isabella Callejas Mandon - 2202030

Geiner Duvan Guevara Vargas - 2201840

Grupo: A1

Subgrupo:G5

Universidad
Industrial de
Santander



Universidad Industrial de Santander
Facultad de Fisicomecánicas
Escuela de Ingeniería de Sistemas
Bucaramanga, Septiembre del 2023

Introducción

En el contexto del desarrollo de software y la programación orientada a objetos se desarrollan cuatro paquetes que demuestran la aplicación de conceptos como la abstracción y la encapsulación en una variedad de clases. El primero de estos paquetes se centra en una máquina de café, mientras que los otros tres exploran proyectos que ilustran cómo las clases interactúan entre sí utilizando un enfoque basado en contenedores. Estos proyectos incluyen un escenario en el que un automóvil muestra mapas, un sistema universitario que organiza facultades y un laboratorio especializado en el análisis de muestras.

Índice

| | |
|-------------------------|----------|
| 1. Ejercicios | 3 |
| 1.1. Cafetera | 3 |
| 1.2. Carro | 5 |
| 1.3. Escuela | 6 |
| 1.4. Muestras | 7 |
| 2. Conclusiones | 7 |

1. Ejercicios

1.1. Cafetera

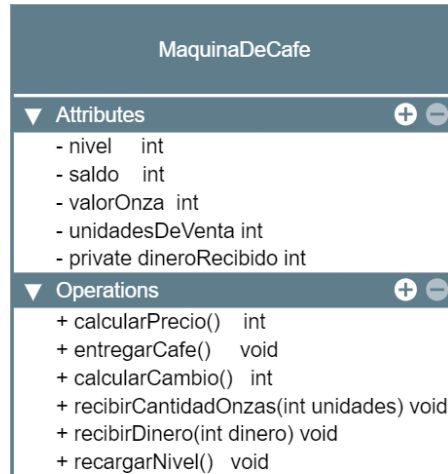


Figura 1: UML de la clase de Máquina de Café

Como el código es pequeño:

```
1 public class MaquinaDeCafe
2 {
3     //Asumimos que es cafe instantaneo
4     private int nivel; //Nivel de cafe que tiene la
        maquina
5     private int saldo; //Saldo de la maquina para vueltos
6     private int valorOnza;
7     private int unidadesDeVenta;
8     private int dineroRecibido;
9
10    MaquinaDeCafe() {
11        nivel=100;
12        saldo=20000;
13        valorOnza=100;
14    }
15
16    int cacularPrecio() {
17        int precio = unidadesDeVenta*valorOnza;
18        if (precio > dineroRecibido){
19            return dineroRecibido = 0; //No le alcanza
                para comprar
        }
```

```

20     }
21     else {
22         return precio;
23     }
24 }
25
26 void entregarCafe(){
27     if (calcularCambio() != 0){
28         nivel -= unidadesDeVenta;
29         saldo += calcularPrecio();
30         unidadesDeVenta = 0;
31         dineroRecibido = 0;
32     }
33 }
34
35 int calcularCambio(){
36     int cambio = dineroRecibido - calcularPrecio();
37     if (cambio > saldo){
38         return dineroRecibido = 0;
39     }
40     else {
41         return cambio;
42     }
43 }
44
45 void recibirCantidadOnzas(int unidades){
46     unidadesDeVenta = unidades;
47 }
48
49 void recibirDinero(int dinero){
50     dineroRecibido = dinero;
51 }
52
53 void recargarNivel(){
54     if (nivel <= 100){
55         nivel = 100;
56     } else {
57         System.out.print("Ya la cafetera esta en su
58             limite");
59     }
60 }

```

1.2. Carro

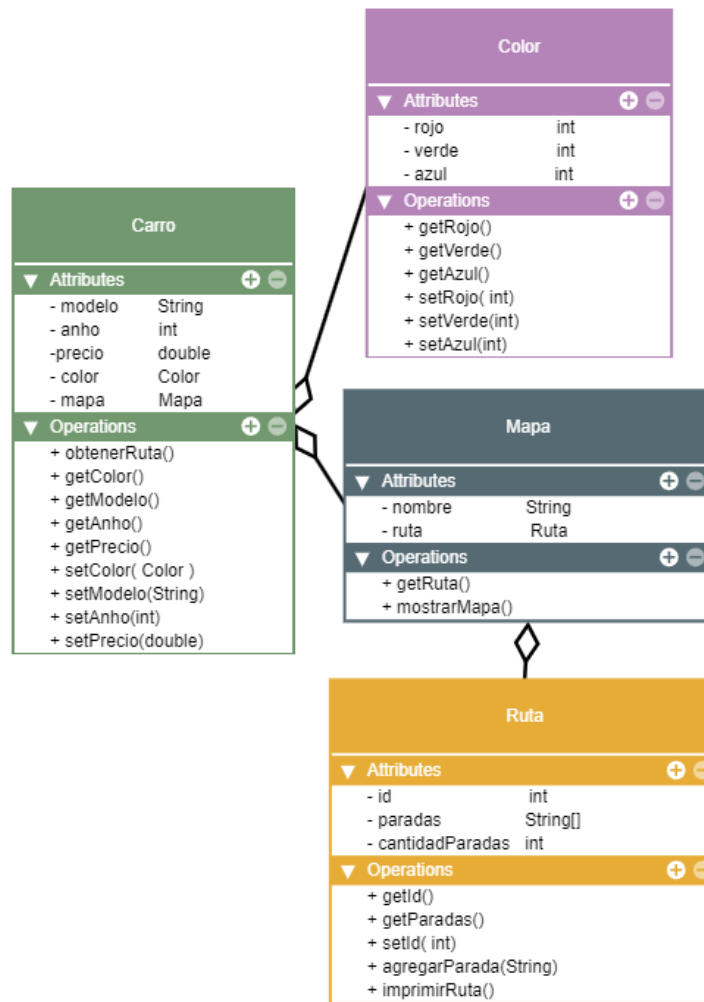


Figura 2: UML del taller de relaciones Carro

Ingresa al siguiente link para ver el código en lenguaje java [[Código del paquete carro](#)]

1.3. Escuela

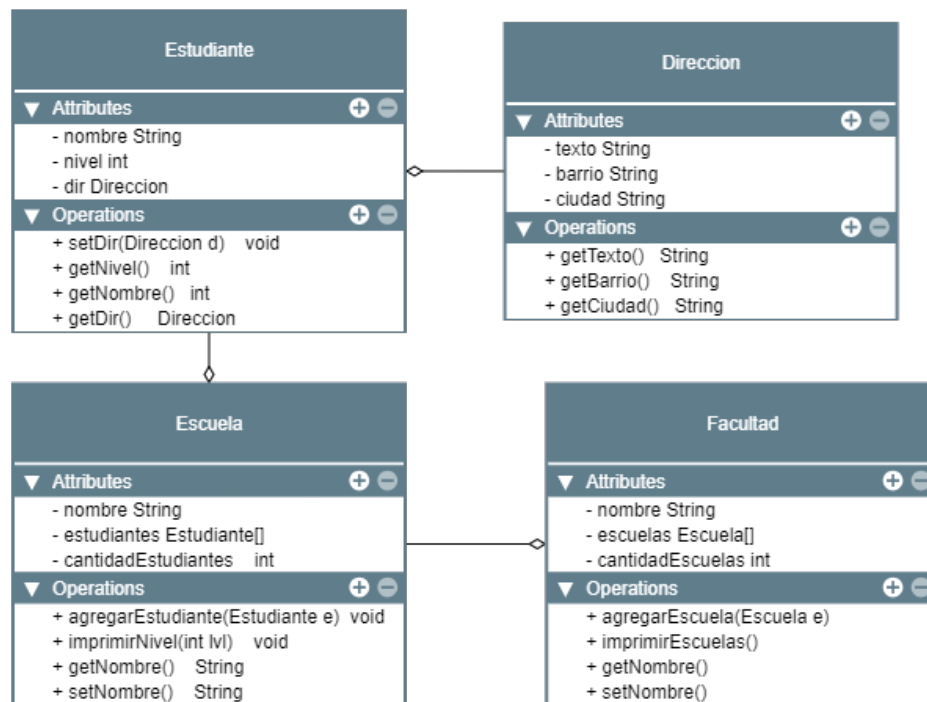


Figura 3: UML del taller de relaciones escuelas, facultades y estudiantes

[[Código del paquete de escuelas.](#)]

1.4. Muestras

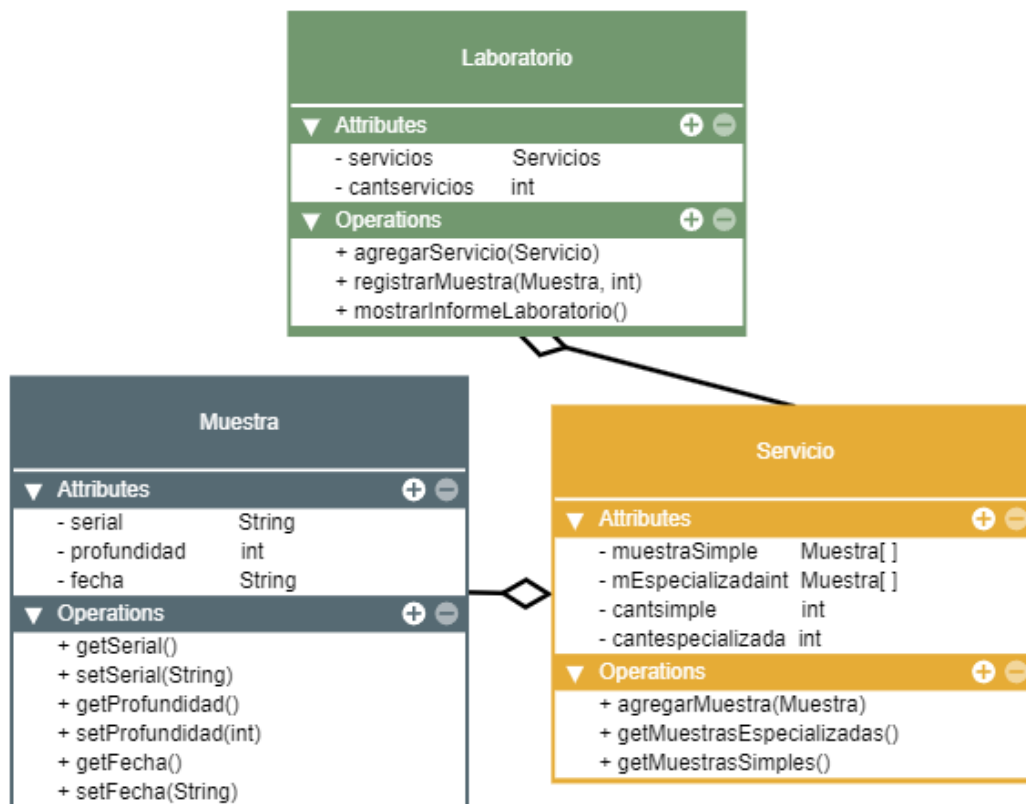


Figura 4: UML del taller2: Laboratorios

Ingresa al siguiente link para ver el código en lenguaje java [[Código del paquete de muestras](#)]

2. Conclusiones

En conclusión, se ejemplifican con la aplicación de conceptos fundamentales de programación, como la abstracción y la encapsulación. Estos ejercicios abarcan diversos contextos, desde una máquina de café simple hasta proyectos más complejos con interacciones entre clases en un enfoque basado en contenedores. En todos estos, se ha subrayado el papel crucial de la abstracción y la encapsulación en la creación de software robusto y adaptable para diversas situaciones del mundo real. Esto destaca la versatilidad y utilidad de estos conceptos en la programación orientada a objetos.