



GUIA 1. PROGRAMACION ORIENTADA POR OBJETOS

Análisis Multivariado y Funcional

Guía 1 - Actividad 1 y actividad 2

Presentado por:

Cascavita Rodriguez Camilo Andres

Docente: Acosta Salas Luz Amparo,

17 de Febrero del 2024, Bogotá D.C.

1. Tabla de Contenidos

1. TABLA DE CONTENIDOS
2. INTRODUCCIÓN
3. DESARROLLO DE ACTIVIDAD 1
4. DESARROLLO DE ACTIVIDAD 2
5. REFERENCIAS

2. Introducción

Los Diagramas de Clases en UML (Unified Modeling Language) son una herramienta fundamental en el análisis y diseño de sistemas orientados a objetos. Estos diagramas permiten visualizar la estructura estática de un sistema, mostrando las clases que lo componen, sus atributos, métodos, relaciones y las interacciones entre ellas.

3. Desarrollo de las actividades

Actividad 1. Diagramas de clases en UML

Productos

Informe que evidencie lo siguiente:

- Desarrollo de un diagrama de clases UML para la «Serie de Ejercicios de Modelado No. 1» **(individual)**.
- El código fuente en Java de las clases que se generan del diagrama de clases de la «Serie de Ejercicios de Modelado No. 1» almacenado en el repositorio individual GitHub, GitLab o BitBucket del estudiante **(individual)**.
- Solución al «Cuestionario en línea No. 1» **(individual)**.
- Informe técnico que incluya el diagrama de clases y la dirección donde el tutor pueda encontrar las clases generadas **(individual)**.

Desarrollo

1. La clase Fraccionario: debemos poder almacenar el numerador y el denominador del Fraccionario.

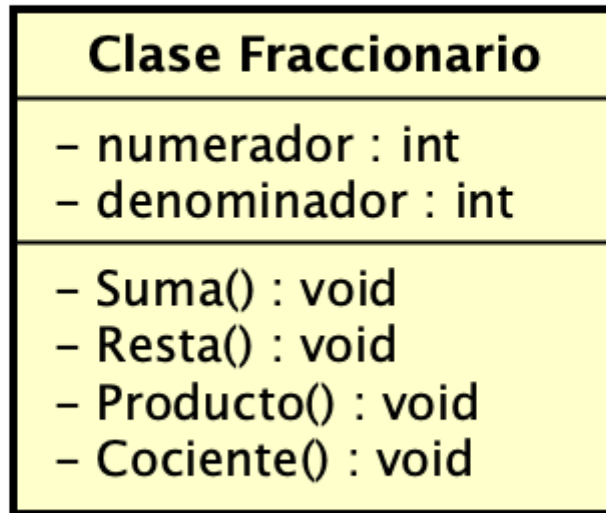


Ilustración 1: Diagrama UML: clase fraccionario. (Elaboración propia asta uml)

```
public class Fraccionario {  
    private int numerador;  
    private int denominador;  
}
```

2. Un reloj es una clase que almacena la hora, los minutos y los segundos.

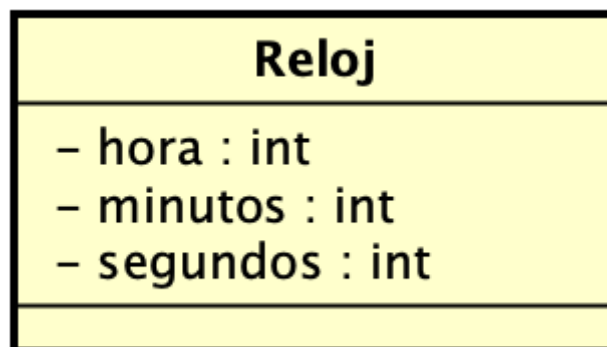


Ilustración 2: Diagrama UML: clase fraccionario. (Elaboración propia asta uml)

```
public class Reloj {  
    private int horas;  
    private int minutos;  
    private int segundos;  
}
```

3. La clase Fecha permite almacenar año, mes y día.

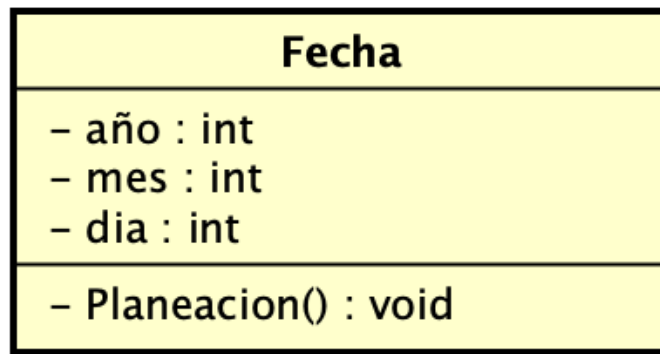


Ilustración 3: Diagrama UML: clase fraccionario. (Elaboración propia asta uml)

```
public class Fecha {
    private int año;
    private int mes;
    private int dia;
```

4. Un proyectil es una clase que permite trabajar la velocidad a la que es disparado el elemento, así como el ángulo con el que sale disparado.

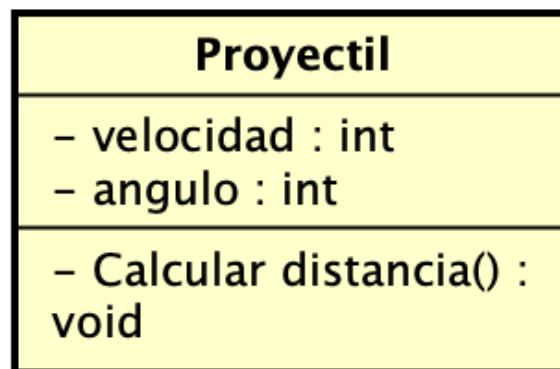


Ilustración 4: Diagrama UML: clase fraccionario. (Elaboración propia asta uml)

```
public class Proyectil {
    private double velocidad;
    private double angulo;
```

5. Una tienda (de la que guardamos el nombre, la dirección y el teléfono) ofrece 4 productos. De cada uno de estos productos queremos almacenar un código numérico,

el nombre, el tipo del producto (que puede ser LACTEO, CÁRNICO, FRUTA o ENLATADO), la fecha de expiración, el nombre del fabricante, la cantidad en inventario y el precio unitario.

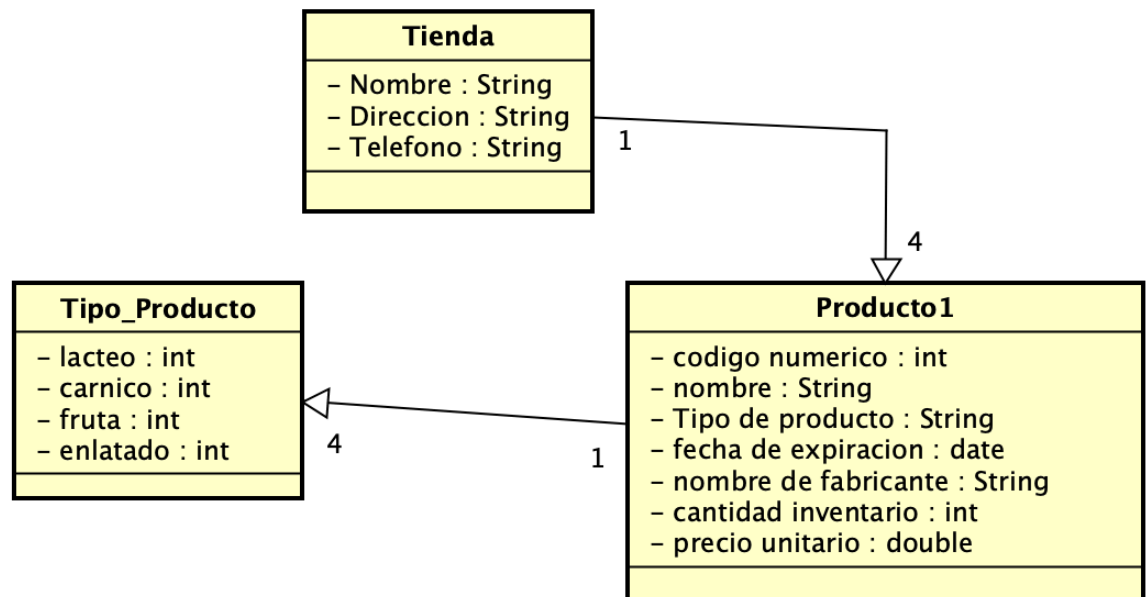


Ilustración 5: Diagrama UML: clase fraccionario. (Elaboración propia asta uml)

```

import java.util.Date;

public class Producto1 {
    private int codigo;
    private String nombre;
    private TipoProducto tipo;
    private Date fechaExpiracion;
    private String fabricante;
    private int cantidadInventario;
    private double precioUnitario;
  }

```

```
public enum TipoProducto {  
    LACTEO,  
    CARNICO,  
    FRUTA,  
    ENLATADO,  
}
```

```
public class Tienda{  
  
    private String nombreTienda;  
    private String direccion;  
    private String telefono;
```

6. Una Universidad (de la que nos interesa su nombre, nombre del rector y nombre de la ciudad donde funciona) guarda información de sus facultades. Una facultad tiene un nombre y un código numérico. Un profesor es una persona de la que tenemos que guardar también la profesión, la nacionalidad, y el sueldo. Un profesor pertenece a una sola facultad, pero una facultad tiene muchos profesores. Hay también carreras, y de una carrera necesitamos guardar el nombre, el número de créditos totales, el número de semestres que dura la carrera y el nivel de la carrera (PREGRADO o POSTGRADO). Una carrera pertenece a una sola facultad, pero una facultad tiene cero o más carreras. Los estudiantes son personas de las que tenemos que almacenar la siguiente información adicional: el colegio del cual se graduó del bachillerato y la fecha en que ingresó a la universidad. Un estudiante pertenece a 1 o más carreras, y claro, una carrera posee muchos estudiantes. Hay que guardar información de los cursos, y de cada curso hay que almacenar el código (un número), el nombre del curso, el número de créditos, el número del salón donde se imparte y el nombre del edificio donde se dicta. Un curso tiene 1 o más profesores que lo dictan, y un profesor puede dictar muchos cursos también. En un curso pueden estar inscritos muchos estudiantes y un estudiante puede estar inscrito en muchos cursos. Los cursos pertenecen a una sola carrera, y la carrera tiene muchos cursos. Una persona tiene cédula, nombre, fecha de nacimiento y lugar de nacimiento (ciudad, departamento, país).

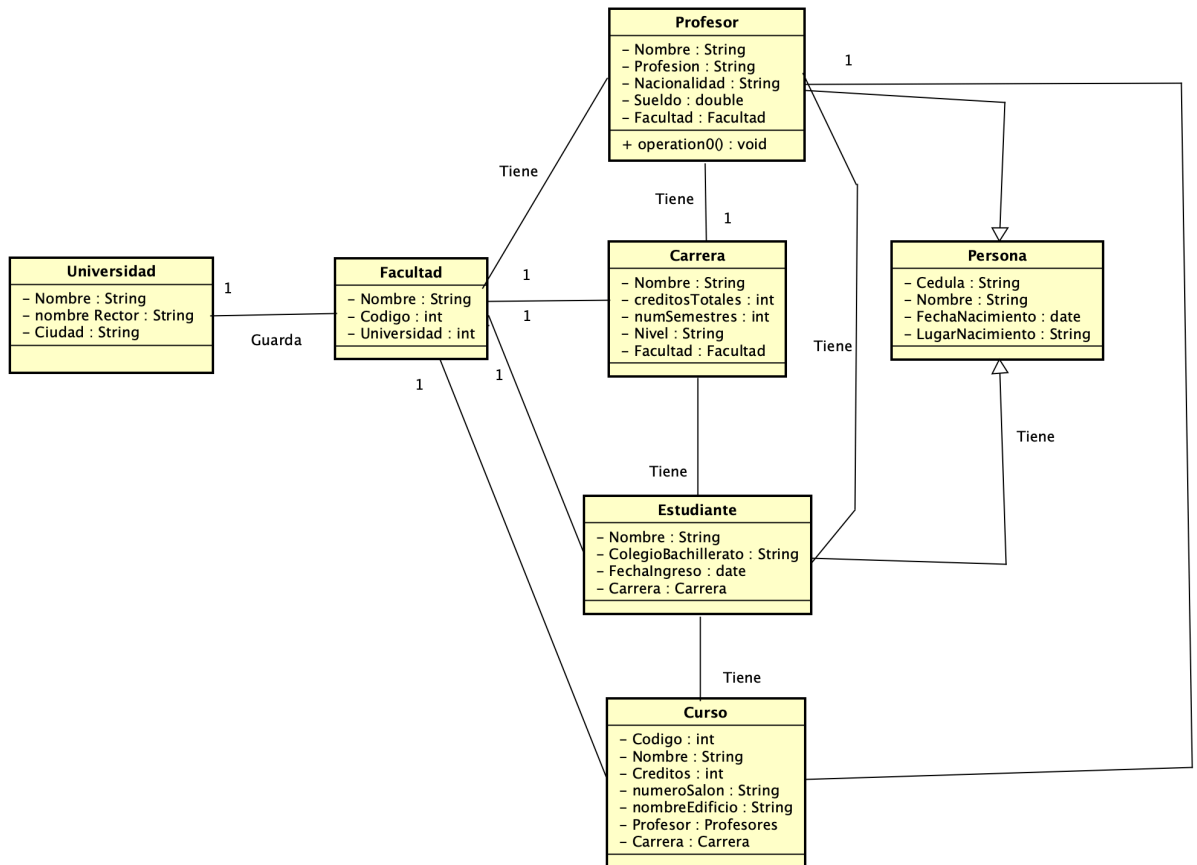


Ilustración 5: Diagrama UML: clase fraccionario. (Elaboración propia asta uml)

```

class Universidad {
    private String nombre;
    private String nombreRector;
    private String ciudad;
}
  
```

```

class Facultad {
    private String nombre;
    private int codigo;
    private Universidad universidad;
}
  
```



```
class Profesor {  
    private String nombre;  
    private String profesion;  
    private String nacionalidad;  
    private double sueldo;  
    private Facultad facultad;
```

```
class Carrera {  
    private String nombre;  
    private int creditosTotales;  
    private int numSemestres;  
    private String nivel;  
    private Facultad facultad;
```

```
class Estudiante {  
    private String nombre;  
    private String colegioBachillerato;  
    private Date fechaIngreso;  
    private Carrera[] carreras;
```

```
class Curso {  
    private int codigo;  
    private String nombre;  
    private int creditos;  
    private String numeroSalon;  
    private String nombreEdificio;  
    private Profesor[] profesores;  
    private Carrera carrera;
```

```
class Persona {  
    private String cedula;  
    private String nombre;  
    private Date fechaNacimiento;  
    private String lugarNacimiento;
```

Actividad 1. Cuestionario:

Cuestionario en línea No. 1

Fecha de entrega 19 de feb en 23:59 Puntos 4 Preguntas 4
Disponibile 13 de feb en 0:00 - 19 de feb en 23:59
Límite de tiempo 30 minutos Intentos permitidos Ilimitados

Volver a realizar el examen

Historial de intentos

	Intento	Hora	Puntaje
MÁS RECIENTE	Intento 1	10 minutos	3 de 4

Los resultados del examen están protegidos en este caso y se pueden ver una sola vez justo después de la entrega.

! Las respuestas correctas están ocultas.

Puntaje para este intento: 3 de 4

4. Actividad 2. Programación Orientada por Objetos básica en Java

Link repositorio:

<https://github.com/camilocascavita/Guia-1-actividad-1-y-2.git>

5. Referencias

Deitel, P., Deitel, H., Romero Elizondo, A. V., & Fuenlabrada Velázquez, S. (2016). Cómo programar en Java (10a edición). Capítulos 8, 9 y 10. México, México: Pearson Educación. Recuperado de <https://bit.ly/39yCOVd>Links to an external site.

Joyanes Aguilar, L., & Zahonero Martínez, I. (2014). Programación en C, C++, Java y UML (2a ed.). Capítulo 13. México, México: McGraw-Hill / Interamericana. Recuperado de <https://bit.ly/3qhp6Mi>Links to an external site.

Villalobos S., J. A., & Casallas, R. (2006). Fundamentos de programación: aprendizaje activo basado en casos: un enfoque moderno usando Java, UML, Objetos y Eclipse. Capítulos 1, 2, 3 y 4. Bogotá, Colombia: Pearson Educación. Recuperado de <https://bit.ly/36aHRc2>