**Desarrollo taller de Programación Orientada a Objetos (POO)**

**Parte 1 – Investigación teórica**

**1.** La Programación Orientada a Objetos (POO) es un paradigma de programación (estilo o modelo de pensamiento) que define cómo resolver problemas y organizar código; que organiza el código en objetos, los cuales representan entidades del mundo real o conceptos abstractos.

Cada objeto combina datos (atributos) y comportamientos (métodos), lo que facilita modelar sistemas de forma más cercana a la realidad.

Además de que esta ofrece varias ventajas frente a la programación estructurada, permitiendo modelar los problemas del mundo real de manera más natural, presenta la modularidad (dividir un programa en partes más pequeñas, independientes y reutilizables), la facilidad para la reutilización de código y una mayor mantenibilidad de los programas.

**2.** Una clase puede definirse como el plano para la creación de objetos, definiendo cómo serán estos objetos, dejando sus atributos y métodos.

Un objeto es creado siguiendo las instrucciones de la clase (creamos un objeto concreto). Con unas características (atributos) y acciones (métodos definidos) podemos entender así la reutilización de código y la separación de funcionalidades.

**3.** El polimorfismo significa que diferentes clases pueden usar el mismo método, pero comportarse de manera diferente según cómo esté programado.

**4.** La abstracción consiste en tomar los elementos importantes de algo y ocultar lo que no lo hace. Se trata de representar lo más relevante de manera sencilla y general.

**5.** UML (Lenguaje Unificado de Modelado) es un lenguaje visual usado en la ingeniería de software para representar sistemas y orientar su desarrollo.

Está compuesto por diagramas que ayudan a mostrar los procesos y los objetos, así como las interrelaciones de un sistema. Permitiendo así explicar y detallar mejor un proyecto.

Existen múltiples tipos de diagramas, algunos de los cuales son:

* **Diagrama de clases:** Este diagrama muestra las clases de un sistema, sus atributos y métodos, así como las relaciones entre ellas.
* **Diagrama de objetos:** Similar al diagrama de clases, pero muestra objetos específicos y sus relaciones en un momento determinado.
* **Diagrama de secuencia:** Muestra la interacción entre objetos a lo largo del tiempo, cómo se envían mensajes entre sí.
* **Diagrama de actividad:** Representa el flujo de actividades o procesos en un sistema.

**Encapsulación**  
Mientras Java y C++ tienen modificadores de clases internos (público, privado, protegido), Python no tiene modificadores de acceso estrictos.

**Abstracción**  
Python usa clases abstractas con el módulo ABC, mientras que Java tiene clases abstractas e interfaces integradas, y C++ usa clases abstractas.

**Herencia**  
Python permite la herencia múltiple, Java usa interfaces para simular la herencia múltiple y en C++ este proceso es manual.

**Polimorfismo**  
En Python y Java es automático, mientras que en C++ este proceso es manual (usando funciones virtuales puras).

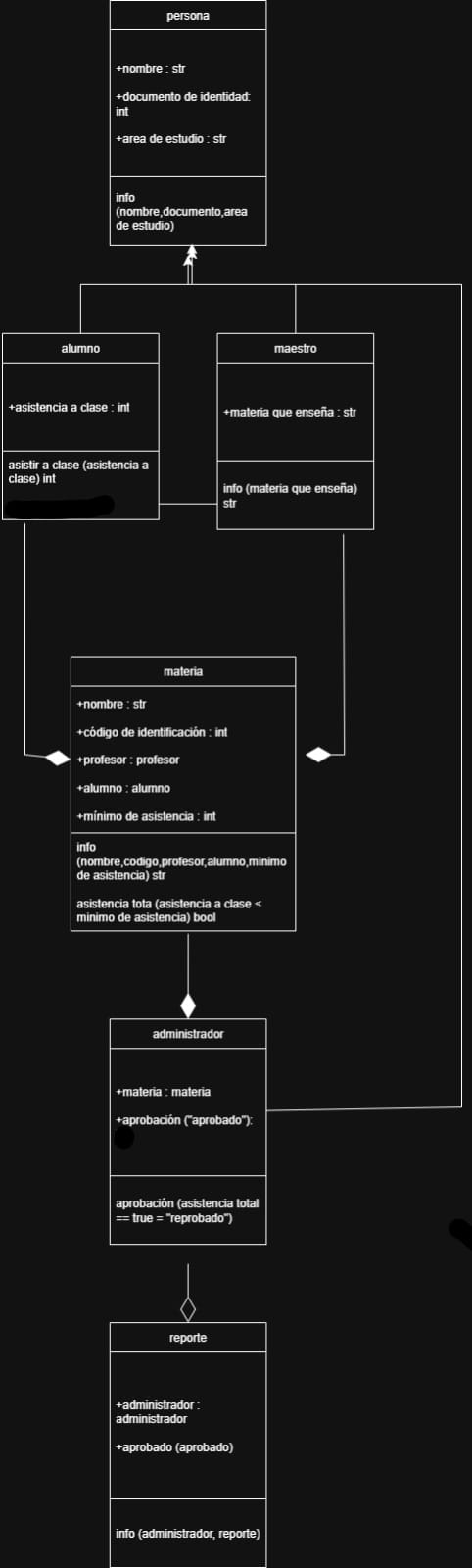
**Parte 2 – Diagramas UML**

Diagrama de clases del problema:

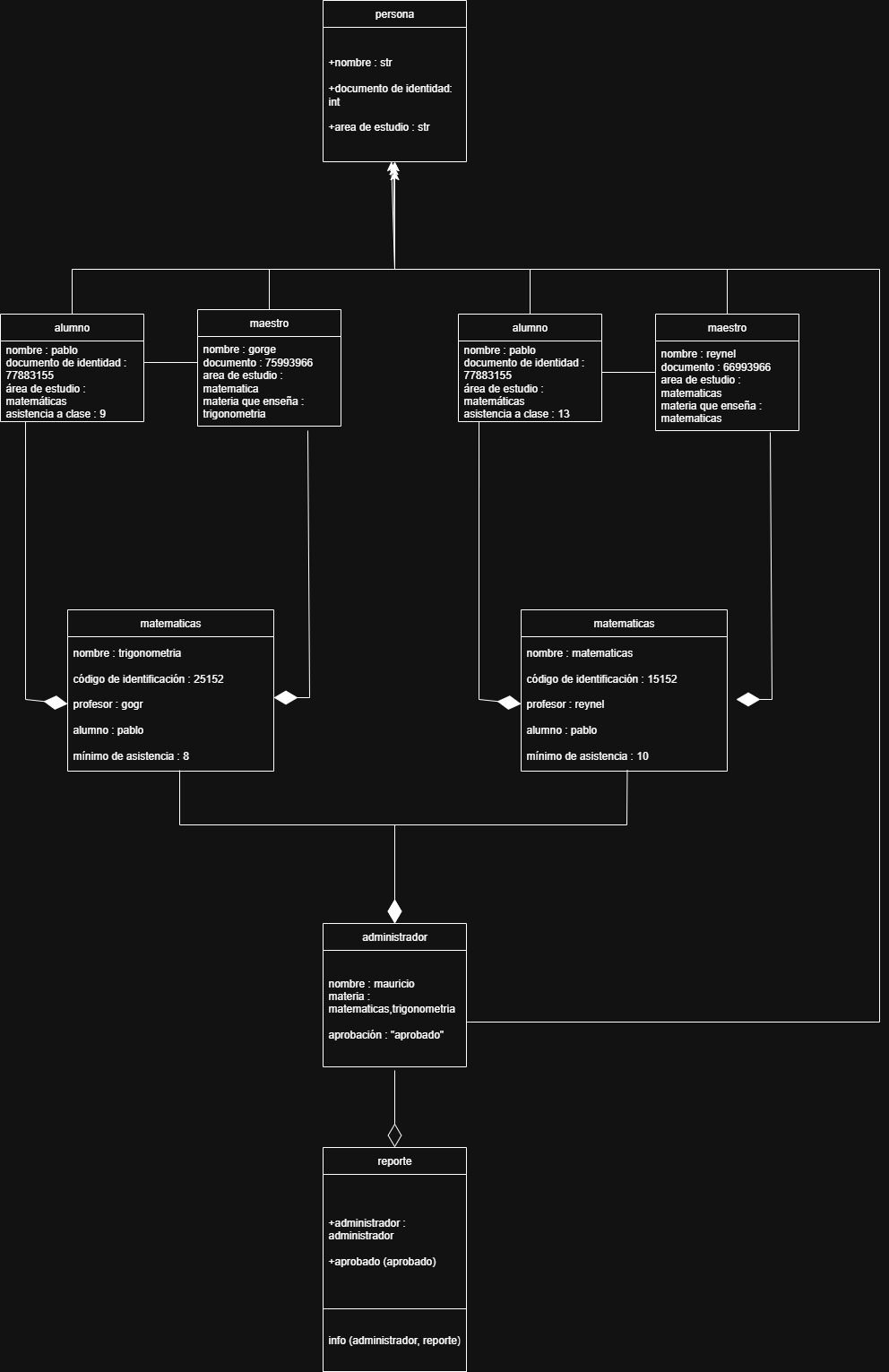
En mi mini proyecto de clases de programación, administro alumnos, maestros y materias.

* **Clase Alumno:** Atributos → nombre, documento de identidad, otros datos de estudio. Métodos → asistir a clases, registrarse, etc.
* **Persona:** La porsona as una clase padre de la Cual derivan todas las demos personas de este caso pues contiene los datos de identificaran basicos de los miembros como la son el nombre, al documento de Identidad y el area de estudio y posee los metodos info
* **Maestro:** Esta clase adamas de los atributas haredados de la close padre persona pasee al atributo de materia que enseñar y su metaodo es info
* **Materia:** Esta clase es una compocision pues depende de que halla alumna y maestro para axistin y esta a su misma vez los contiere tentenda odemos nombre de la materia, Codigo do interaccon y el minima de asistencia y sus metodos son info() y asistenen total que es un dato booleano de si la asistencia del estudiante ha sido manor que el minimi de asistencia
* **Administrador:** Esta clase además de ser una clase hoja de la case persona posee los atributos de materia de tipo materias y la oporobocon que se tama como un aprobada y sus metodos son aprobaci рога que si el dato booleano de que asistenca es menor a minino de asistencias raquerido cambie el reprobado por aprobado
* **Método:** contiene el administrador que hizo el reporte y si aprobó o no

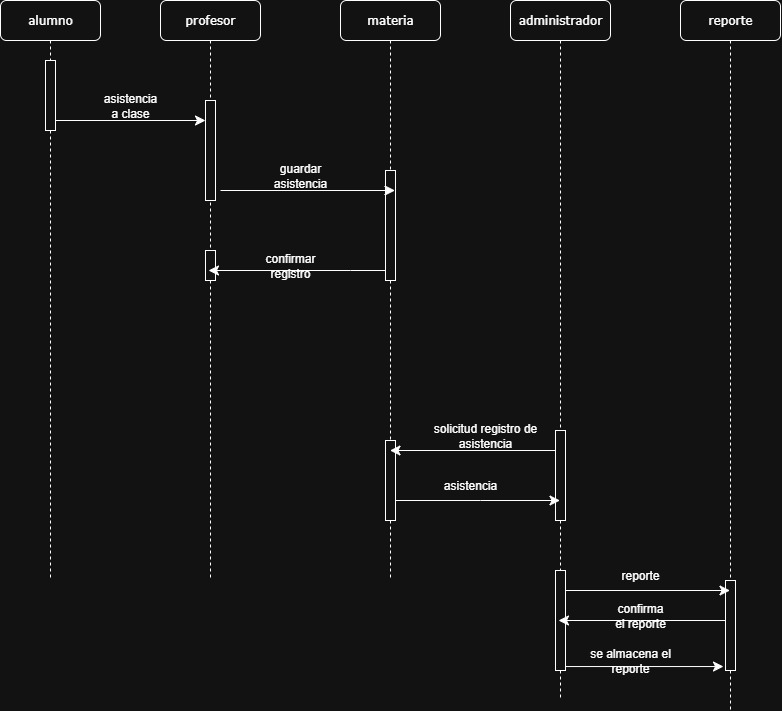
**8. realiza un diagrama de clases para el problema elegido**

****

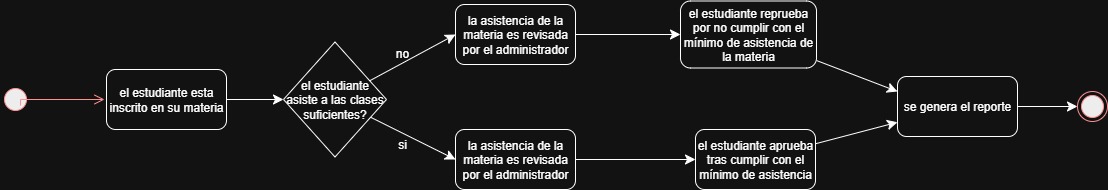
**9. Crea un diagrama de objetos con instancias concretas de tu clase**

****

**10. Realiza un diagrama de secuencia que muestre como interactúan los objetos en caso de uso**

****

**11. Realiza un diagrama de actividades con el flujo de pasos del caso de uso elegido**

****

**Parte 5 - Reflexión final**  
Los lenguajes usan la idea de definir clases y crear objetos a partir de ellas teniendo atributos y metodos, se pueden tener objetos públicos y privados. Uno de los lenguajes mas usados es python pues permite que las estructuras de datos puedan tener sus funciones y facilitarlas cuando para tener los marcados mas flexibles y la programación orientada a objetos (POO) es mas sencilla de realizar pues los tipos de variables y la herencia multiple estan simplificadas a diferente de en otros lenguajes como Java con su mayor complejidad sintáctica o c++ con su especificacidad

21. Lo que más se me complicó fue proyectar mis ideas para los diagrama uml al físico por qué ni yo mismo me terminaba de entender en ocasiones.

22El trabajo de investigación me ayudo muchisimo a entender mas fácil la poo y también me ayudo a entender más los datos y clases de forma global.

23 agregar una interfaz gráfica organizada para así mostrar de mejor forma la información en los programas.

Principio del formulario

Final del formulario