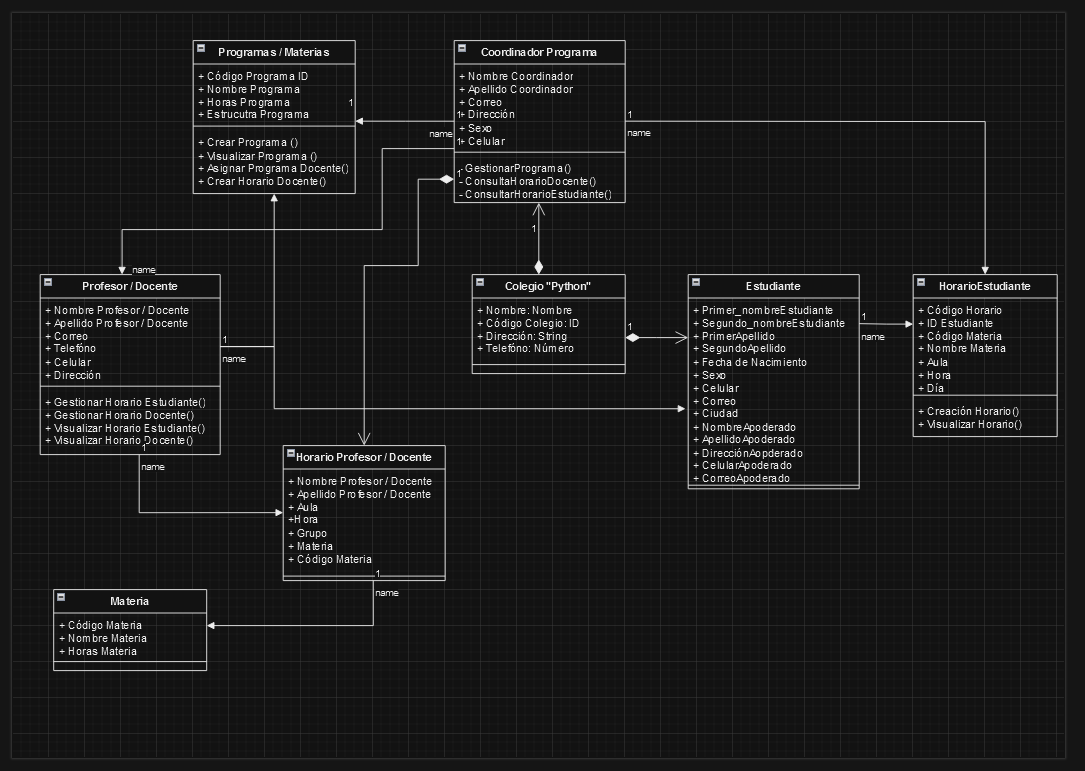
Diagrama de Clases

Programación Avanzada Python – POO

Alumno: Jeff Palacios Sosa

Curso: Full Stack Python Diurno

Esquema Diagrama de Clases – Colegio



El siguiente diagrama fue diseñado pensando en la estructura de un reciento educacional con sus diferentes áreas, las cuales podemos referenciar como “Clases” e interacciones, y entre ellos también están detalladas sus “atributos” o características.

Diagrama de Clases – Descripción

-- Diagrama de Clases Colegio "Python" --

Para comprender el diagrama, debemos partir con la Clase Principal, este seria Colegio "Python".

Esta posee sus atributos:

- Nombre

- Código Colegio

- Dirección

- Teléfono

La Clase "Estudiante" se genera por la Clase "Colegio Python", y este a su vez, tiene la Clase "HorarioEstudiante" con sus propios atributos.

La Clase "Coordinador de Programa" proviene de la Clase "Colegio Python", y cuenta con sus propios atributos.

Esta clase se relaciona con la Clase "HorarioEstudiante", esta hereda los valores del Profesor o Docente que gestiona el programa.

A sí mismo, la Clase "Programas / Materias" posee sus atributos, y se relaciona con la "Clase Coordinador Programa" heredando los Programas creados y visualizándolos.

Para la Clase "Profesor / Docente" también posee sus propios atributos y se relacionan con la clase "Coordinador Programa", "Estudiante" y "Horario Profesor / Docente".

Los métodos que maneja la clase "Profesor / Docente" permiten "Gestionar Horario Docente", "Gestionar Horario Profesor / Docente" y "Visualizar Horario".

La clase "Horario Profesor / Docente", posee sus atributos que se relacionan con la clase "Materia", heredando datos como "Código Materia".

Resumen:

El "Colegio Python" es la clase principal que permite que las demás clases se puedan ejecutar.

Las demás clases poseen atributos que detallan su forma de interactuar, y algunas poseen dependencia de la clase principal "Colegio Python".

Por ejemplo, la Clase profesor no tendría acceso a la información del "Programa de Materias" sin que este se haya creado, lo mismo ocurre con la clase "Estudiante", sino hay estudiante ingresado, no hay datos que observar desde la clase "Profesor/Docente".

La forma en cómo se presenta el "Diagrama de clases, permite que los datos y su interacción con las demás clases se realicen de manera ordenada y clara.

El "Colegio Python":

Ingresa alumnos con atributos específicos,

El estudiante tendrá un horario específico

El "Colegio Python":

Tiene un Coordinador de Programa con atributos específicos,

Este se relaciona con la clase "Programas / Materias con sus propios atributos

La clase "Profesor / Docente:

Ingresa a los profesores o docentes con sus propios atributos,

El "Profesor / Docente" posee en métodos la posibilidad de Gestionar Horario Estudiante, así como Gestionar Horario Profesor / Docente.

Esta clase se relaciona con la clase "Horario Profesor / Docente".

La Clase "Horario Profesor / Docente":

Tiene sus propios atributos, y se relaciona con la Clase "Materia"

La Clase "Materia":

Posee sus propios atributos y se relaciona con la Clase "Horario Profesor / Docente.

Conclusión

>> De los ejercicios propuestos para la confección del diagrama de clases y su codificación, destaco:

* Que, al momento de poder diseñar un diagrama, se ve relativamente fácil de dibujar, conectar y hasta relacionar, sin embargo, esta "facilidad" desaparece al momento de poder codificar cada una de las clases, atributos, métodos e instancias dentro de un código.
* También es destacable la importancia de tener un orden al momento de realizar el diagrama de clases, para que al momento de codificar la información sea clara y concisa.
* Que es esencial poder manejar con claridad los fundamentos de la programación orientada a objetos (POO) en Python con sus 4 características:
* Encapsulamiento,
* Abstracción,
* Herencia y
* Polimorfismo