INFORME TAREA 1 PARADIGMAS DE LA PROGRAMACIÓN

Integrantes: Esteban Leyton, Oscar Tobar

Universidad Andrés Bello®

Conectar • Innovar • Liderar

INTRODUCCIÓN

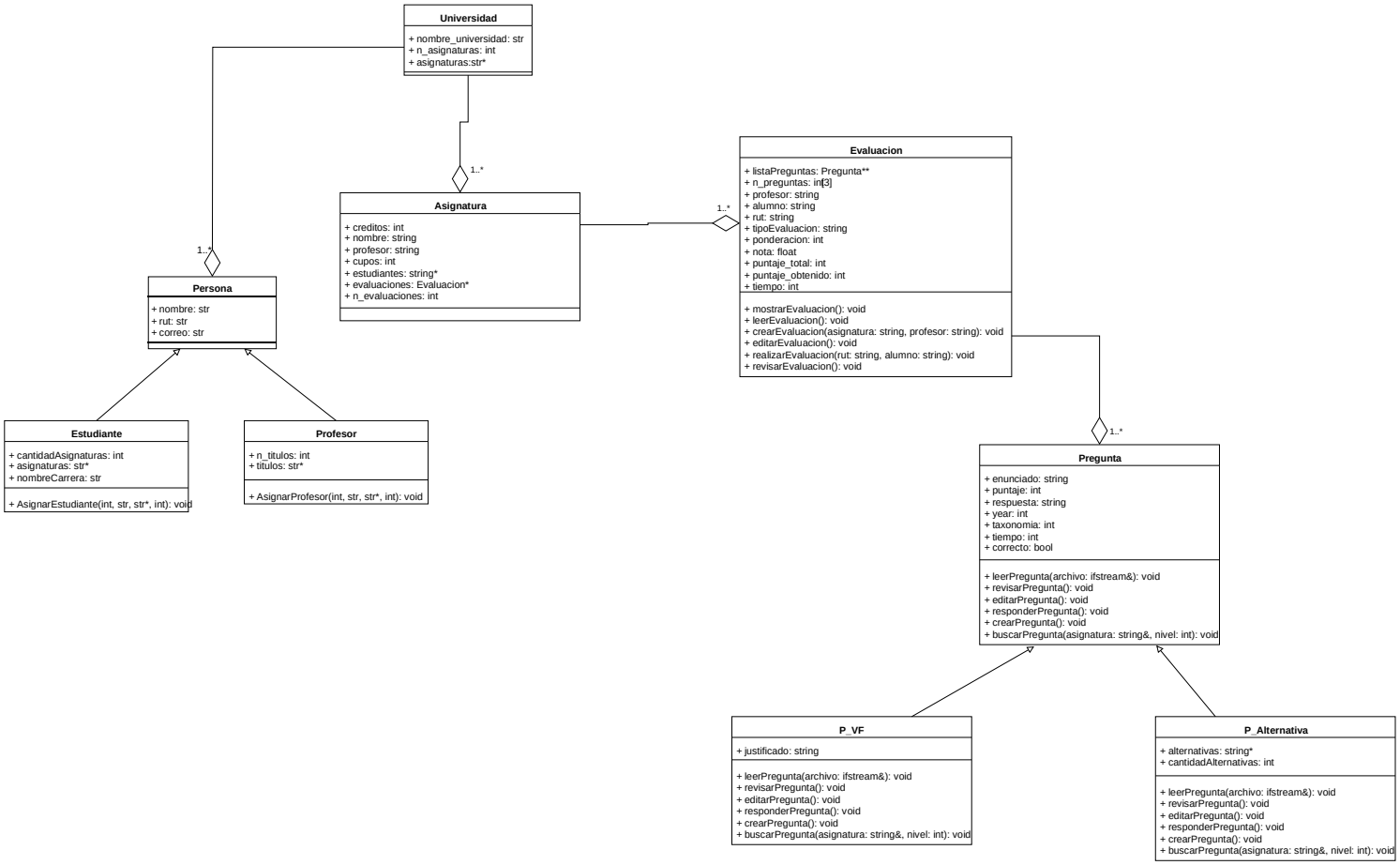
El objetivo principal de este trabajo es dar una herramienta para tanto profesores como alumnos puedan acceder a las evaluaciones de las asignaturas, los alumnos puedan realizarlas y los profesores puedan crear y editas las mismas. En este informe se explicará el como se llego al resultado, el diseño del UML y porque de cada clase y como es la relación.

Como objetivo principal se enfocó en la creación y edición de evaluaciones, mediante tres tipos de pregunta que se pueda crear a una elección aleatoria de una base de datos o que el profesor la cree de 0.

Como objetivos secundarios se propuso la implementación de los alumnos, que estos puedan realizar una evaluación, quede registrada y que luego un profesor pueda revisar esta y entregue la nota obtenida.

DESCRIPCIÓN DE LA SOLUCIÓN

A continuación, se presenta el UML diseñado para solucionar el problema dado:



Ahora se describirá porque la implementación de cada clase:

- **Universidad:** esta clase tiene como objetivo indicar el nombre de la institución, indicar cuantas, y que asignaturas existen, esto se realiza al iniciar el programa, revisando que carpetas existen, guardando sus nombres y contándolas.
- **Asignatura:** Aunque es una clase que se implemento tanto en el UML como en el código, no es algo que se aproveche. Pero el objetivo iba de tener un registro de cuantos créditos vale la asignatura, la cantidad de cupos, información del profesor y de los alumnos inscritos.
- **Persona:** se busca generalizar algunos atributos que puedan tener en común los estudiantes como profesores, como el nombre, rut y correo.
- **Profesor:** en esta clase se guarda información del profesor, como sus títulos. Mientras que a nivel de código esto permite poder crear evaluaciones en alguna asignatura, aquí falto poder implementar que los profesores tengan asignada las asignaturas y solo permita crear o editar evaluaciones cuando son los docentes de estas.
- **Estudiante:** Aquí se busco que se guardara las asignaturas que el alumno esta registrado, para a posterior poder realizar una evaluación y si esta calificada, ver la nota.
- **Evaluación:** Esta clase es de las más completas, ya que guarda la información del profesor que la creo, la lista de preguntas, cuantas preguntas de cada tipo hay, el tiempo estimado que debería poder realizarse, que tipo de evaluación es y su ponderación en la asignatura.
- **Pregunta:** Esta es la clase que es la base del otro tipo de pregunta ya que posee el enunciado, un booleano que indica si la respuesta esta correcta o no, cuantos puntos vale la pregunta y el año que se implementó la pregunta, se toma como el tipo de pregunta corta, ya que puede guardar la respuesta del estudiante.
- **P_VF:** Son las preguntas de verdadero y falso, como atributo extra se tiene la justificación de la respuesta en caso de que la respuesta otorgada sea falso.
- **P_Alternativa:** Las preguntas de alternativa posee la cantidad de alternativas que posee y un arreglo de string con las alternativas.

CONCLUSIÓN

Este trabajo tomo mas tiempo del esperado, la implementación de varias clases y que estas funcionen en armonía es laborioso, por otro lado, el objetivo principal se pudo realizar sin problemas, los objetivos secundarios quedaron por implementar, están los métodos, pero por falta de pruebas no quedaron disponibles en el producto final. Este trabajo ayudo a saber cómo implementar polimorfismo ya que, para poder hacer una lista de preguntas, se implementa los punteros dobles que permite almacenar la clase padre y mutar a alguna clase hija. También en como estructurar archivos de texto para luego la lectura del programa se haga de forma correcta.