

### Informe Entrega 3

Elegimos el diagrama 1 porque se ocupan dependencias acertadas para lo que queremos modelar, es decir dependencias por agregación y composición, además se hace un buen uso de la herencia. Por ejemplo la herencia utilizada para el caso de los distintos usuarios, su diseño se hace más fácil y coherente la programación creando una clase Usuario para las instancias de profesor, alumno y administrador.

Además clases como Semestre y Malla tienen una alta cohesión ya que las responsabilidades de estas clases están bien focalizadas. Para el futuro, la Clase Usuario podría actuar como un controlador, para recibir los datos personales de ingreso de los usuarios y redirigir estos datos al sistema y a las clases pertinentes.

Dadas estas características del modelo UML 1 de nuestra entrega pasada, es que decidimos codificar sobre el.

Ahora bien, descartamos el diagrama número dos ya que sus dependencias eran todas del mismo tipo y al mismo tiempo eran muchas, lo que lo hacía muy complejo de entender y al momento de programar iba a ser muy complicado abordar los métodos y relaciones.

También rechazamos este modelo ya que las clases tenían muchos métodos, por ejemplo la clase alumno ( calcularEdad() , consultarHistorial(), verAvanze(), buscarCursos() , etc) lo que no es conveniente, esto mismo se puede apreciar para otras clases como BuscadorCursos. Cuando en realidad es mejor que cada clase contenga los métodos que le correspondan y no todos los métodos en una sola clase.

Se hicieron modificaciones en el diagrama actual, en el Historial Académico se agregaron dos métodos: obtenerAvanze() y agregarSemestre(). Además deberían haber flechas que indiquen una relación entre el Administrador y las entidades que este puede crear, esto se debe a que el Administrador conoce las estructuras de estas entidades. Por último a Ramo se le agrega un identificador "int id", y las notas de un Semestre pasan de ser un int a ser una List<Double> ya que será la lista con las notas del semestre.

El código está comentado para que se entienda mejor lo que se está programando y también para cumplir con los requisitos de esta entrega