# Primer Entregable del curos de IOS

## Aclaraciones y respuestas de la primera parte del examen

* No todos los gets y sets están en el UML, solo los que por alguna razón adquieren una importancia como para mostrarlos. Por ejemplo en el caso de un alumno que “puede” tener un contacto, esto implica que contacto será opcional, que Alumno tendrá dos constructores, uno con contacto y otro sin contacto y al mismo tiempo que debe haber un setContacto(contacto:String) para poder agregar el contacto más tarde si el alumno no lo provee en el momento de la inscripción.
* Parte D pregunta 1: Se agregan al UML la clase abstracta Profesor que contiene todos los atributos en común de todos los profesores y las clases concretas ProfesorTitular y ProfesorAdjunto con sus atributos respectivos (especialidad y capacidaHorasConsulta)y los gets y sets asociados.
* Parte E Pregunta 1: Se agregan los atributos de cupoMaximoDeAlumnos, una listaDeAlumnos y un profesorTitular y otro ProfesorAdjuntos, estos dos últimos como opcionales, ya que pueden no estar definidos a la hora de creación del curso. Se crean dos constructores, uno incluyendo los profesores y otro asignando ***nil*** a ambos profesores.

## Adicional Destacado, Aclaraciones

* En el enunciado no aclara si la relación entre cursos y guías es *uno a muchos* **o** *muchos a muchos*. En la resolución yo interpreté que la relación es muchos a muchos, porque me pareció que respondía a una realidad cotidiana (por ejemplo el apunte de UML puede ser común al curso de IOS y al curso de Android).
* El supuesto anterior junto con el comentario de la falta de recursos de la Fotocopiadora (avisado por el Arquitecto) implican que la búsqueda tiene que ser rápida (por eso implemento el diccionario en Fotocopiadora) y que no se deben repetir Guías. Esto lleva a manejar la integridad referencial desde la clase Manager y a mantener normalizado el modelo.
* Si por el contrario la Fotocopiadora no tuviese restricciones de almacenamiento y la relación entre cursos y Guías fuese de 1:\*, se podría haber creado un objeto compuesto por el código de Curso y la Guía, por ejemplo llamemos a este objeto GuiaDeCurso en cual se podría haber almacenado en el diccionario de la fotocopiadora como [codigoGuia:GuiaDeCurso]. Esto simplificaría la integridad referencial entre objetos, pero en una relación \*:\* implicaría duplicidad de Guías en la fotocopiadora.