

Córdoba, 23 de abril de 2005

Ing. Raúl Morchio
Jefe Dpto. Ing. en
Sistemas de Información
S_____ / _____ D

Ref. Actualización de contenidos Algoritmos y Estructuras de Datos

Con respecto al tema en referencia, tratado y aprobado en la reunión del Área Programación en Diciembre del año pasado, la cátedra continuó trabajando, y paso ha hacerle un breve resumen de lo actuado con posterioridad a ella.

- **Reunión plenaria de cátedra AED** en fecha 7 marzo 2005. El director de cátedra presentó una planificación de contenidos de la asignatura en Orientación Objetos, con lenguaje Java para la implementación práctica. Se debatió en profundidad, se le hicieron modificaciones substanciales, y se consensó una distribución de temas en cuatro unidades temáticas. Asimismo se combinó un Plan de Implementación que incluía la conformación de un Equipo de Tareas para la elaboración del material de clases (Apunte de la Cátedra) y material para el dictado de Talleres de Actualización de Docentes.
- **Equipo de tareas.** A la fecha se tienen terminadas las dos primeras unidades, que de acuerdo a la distribución de temas consensado en la reunión corresponden a 8(ocho) semanas de clase, o sea un poco mas de la mitad del programa a desarrollar. Son aproximadamente 130(ciento treinta) paginas conteniendo tópicos teóricos, vasta ejemplificación en código Java y ejercicios propuestos.
- **Curso 1k7 (Contracurso).** Se llevan 3(tres) semanas de dictado de clases. Desde el inicio se abordó la problemática de la programación en la Orientación Objetos usándose lenguaje Java. Puntualmente al inicio de cada semana se distribuyó el apunte completo correspondiente a la semana, y se lo usó en clase. Los Ings. Jorge Tymoschuk y Analía Guzmán, integrantes del Equipo de Tareas, se hallan a cargo del dictado Teórico/Practico en este curso. La recepción por parte de los alumnos es **muy buena**, y está contribuyendo a difundir la factibilidad, o mejor, realidad, del cambio.
- **Talleres de capacitación.** Se comenzarán a dictar en la primera quincena de Mayo. En su dictado están comprometidos los siguientes docentes:
 - Unidad I** – Introducción, Propiedades y Características de la POO, Ing. Analía Guzman.
 - Unidad II** – Modularidad, Ing. Jorge Tymoschuk.Tenemos el compromiso del Ing. Valerio Frittelli para dictar alguna de las restantes unidades. Estas son:
 - Unidad III - Estructuras en memoria dinámica.
 - Unidad IV – Archivos
- **Talleres de capacitación.** Justificación, Impacto de la actualización de contenidos. Desarrollo estos temas en hoja anexa.

Sin más, aprovecho la oportunidad de saludarle con mi consideración mas distinguida.

Ing. Jorge P. Tymoschuk
Dir. de cátedra

C.C. Secr. Académica

Justificación del cambio propuesto/en curso

El arte de la programación no es una disciplina que parezca estar en su estado final. Ya en el curso de nuestra experiencia profesional hemos asistido a un par de cambios de sus Paradigmas.

Comenzamos en lo que podríamos llamar de Programación Lineal, con sus Go To, Go To Depending, Alter; No había funciones ni procedimientos, el main era un solo bloque, si no se controlaba de cerca al programador todo el sistema podía acabar siendo un único, gigantesco e ingobernable programa.

Luego apareció la Programación Estructurada. Prohibidos los Go To. Se comienza a hablar de modularización, estructuración. Se hace énfasis en un programa principal que invoca a módulos (Funciones, procedimientos) los que a su vez pueden invocar a otros módulos, lo que permite que un programa se organice atendiendo a un nivel de detalle mínimo en su primer nivel, máximo en el final. Sin embargo, no hay un criterio definido de cómo se distribuye la responsabilidad en los distintos niveles, ni cuantos niveles ni tampoco cual es el mejor nivel de complejidad de estos módulos, etc. En realidad, el principal motivo del nacimiento de la Programación Estructurada no es atender a una optimización en la forma de programar, sino atender las necesidades de un nuevo concepto en los Sistemas Operativos de aquella época: Apareció e concepto de memoria virtual, soportada en disco, y había que reducir la paginación al mínimo. La programación en módulos automáticamente paginados (page in, page out) era mucho mejor que dejar que el programador definiera que parte del programa debía permanecer residente, que parte podía ser desalojado. En realidad, con ese enfoque, ha sido mucha suerte que la Programación Estructurada realmente mejorara y mucho la forma de hacerlo anteriormente.

La Programación Orientada a Objetos es la primera en centrar su atención en el Hombre. (Podríamos decir que la tecnología había avanzado lo suficiente como para preocuparse por su dueño). Por primera vez se cae en la cuenta que la programación la hacen seres humanos, no las máquinas. Y siendo el Hombre un ser natural, por que no tornar la programación también en algo natural? Y entonces en algo mas fácil de aprender, desarrollar, utilizar en todas sus etapas, etc. Esta es la idea fuerza que subyace en este paradigma.

Que es lo primero que se puede hacer para "naturalizar" la programación? Que usa mucho el ser humanos en su vida cotidiana? Los Conceptos. Un concepto incluye atributos y comportamiento de algo. Los Conceptos funcionan como cápsulas de conocimiento acumulado: Una comunicación humana que referencia conceptos necesita de un mínimo adicional, apenas lo necesario para puntualizar una mínima variante de comportamiento, definir un parámetro, y listo, la comunicación se estableció en tiempo record. Estamos suponiendo aquí que ambos interlocutores dominaban los conceptos involucrados. Si no es así, pensemos en el tipo de comunicación que se puede establecer entre un especialista y un profano. Comunicación difícil, demorada. Y entre dos profanos? Bueno, hablarán del tiempo, pero claro, se supone que de eso ambos saben bastante ...

La Programación centrada en Conceptos es la base de la Programación Orientada a Objetos. Los conceptos se implementan en Clases, con características (Atributos) y Comportamiento (Métodos). Los métodos de una clase se ocupan de la gestión de todos los atributos esa clase y de comunicarse con el mundo externo. Se establece una muy clara división de responsabilidades, cosa muy deseable en un mundo complejo.

Los conceptos son cosas bien definidas en la mente humana, las clases deben serlo en la POO. Se encapsula lo que sea necesario, al nivel mas adecuado, se deja público lo que precisamos que se vea. (Cuanto menos, mejor). Es natural que no todo lo que usamos sea de igual acceso para todos. Es razonable que todo el mundo tenga acceso a mi teléfono, menos a mi celular, mucho menos a mi cuenta bancaria. Todo esto y mas, no tenía sentido en el paradigma anterior. Tampoco había nada que enfatizara lo de una programación orientada a generar componentes utilizables en otros sitios, otras clases. Como no existía la idea de agrupar características y comportamiento en clases, mucho menos tenía sentido pensar en estructuras de clases (y en herencia) y era difícil implementar con nombres comunes, genéricos acciones que respondían a un mismo nombre (Concepto de Polimorfismo) ...

Bueno, pero **si lo que hay que justificar es porque propiciamos un arranque directo en la POO** en vez de, como estamos haciendo hasta ahora:

Comenzar con Programación Estructurada y luego, paso a paso, incorporamos los nuevos conceptos de programación en alumnos que ya no son "tabla rasa", (Concepto adquirido en la Maestría en Docencia Universitaria)... Esto es proceder con sentido común, pareciera, no?

Sostengo que no hay nada mas errado que la idea anterior y lo vamos a justificar.

La POO no es más difícil que la estructurada, y no requiere de ella como paso previo. Más bien es lo contrario, genera dificultades en un alumno que ya (Quizás dificultosamente) aprendió a programar. Y más aún a un docente muy acostumbrado y conforme con la forma que lo estaba haciendo. Si hilamos un poco mas fino, **la POO sigue siendo estructurada**, solo que incorpora un criterio claro de cómo distribuir responsabilidades, (modularizar) conjuntamente con herramientas que hacen esto posible.

En el esquema actual, enseñamos a diseñar y programar algoritmos en el "viejo esquema". Lo hacemos en una variedad de temas, datos simples, estructurados, luego estructuras como arrays, pilas, colas, listas, árboles, archivos ... En segundo año introducimos POO, decimos bueno, ahora se programa así, y rápidamente vamos a temas nuevos. O sea que lo que fue aprendido en la forma anterior, en forma teórico/practica, así queda.

En el esquema que estamos implementando, haremos las cosas de una vez, usando una forma de programar actual, herramientas actuales y un lenguaje actual. Algún contenido que no llegue a ser dado en AED, lo podemos transferir a PPR, y verlo allí como "retomada" del paradigma, antes de encarar temas nuevos. Los contenidos quedan mejor secuenciados y toda la disciplina mucho mas coherente.

En realidad, el hecho de tener AED y PPR como actualmente, expresa ahora una imagen que ha quedado atrás en el tiempo. Hemos estado actualizando permanentemente contenidos en PPR, en AED nos limitamos a modificaciones menores. Ya hace varios años que estoy propugnando esta modificación, y he enfrentado resistencia bastante seria. Solo últimamente, cuando ya es muy evidente que el cambio es necesario, cuando la evolución en el uso del software de programación por parte de nuestros profesionales lo hace obvio, he obtenido buen consenso.

Si hacemos un repaso de la bibliografía, vemos también una evolución, que por supuesto acompañó la demanda. Los autores publican libros que esperan vender. En un principio, en los libros de Algoritmos, la POO era un pequeño apéndice, luego pasó

a ser la segunda parte del libro, finalmente ya es "Estructuras de Datos y Algoritmos en Java" (Y por consiguiente en POO). Para mi lo importante es que sea en Orientación Objetos, lo de hacerla en Java también me ayuda, y de paso satisfago a algunos docentes Jóvenes de la cátedra, que así la prefieren.

Impacto

Por impacto entiendo las repercusiones que este cambio genera en la comunidad. Entiendo que por lo ya expuesto está bastante explicito que tipo de impacto puede esperarse, así que en este tópico puedo ser mas breve.

- **A nivel profesores.** Una minoría de docentes necesitan ser entrenados en esta nueva forma de organizar la programación, de modularizar. Para ello están previstos una serie de Talleres, a comenzar en la primera quincena de mayo, donde se trataran tanto los aspectos del cambio de metodología como de lenguaje.
- **A nivel Alumnos.** Se espera una recepción positiva. Así ya está ocurriendo con el curso 1k7, como ya hemos dicho.
- **A nivel asignaturas.** El impacto está esencialmente limitado al par de asignaturas de programación de las que soy titular. Asimismo he invitado a participar al Ing. Freddy Dias, quien se mostró interesado en participar y bien sorprendido. Entendí que en su cátedra de Análisis de Sistemas la mayoría de los cursos siguen el modelo de Orientación Objetos.
- **A nivel regularización/promoción de prácticos.** Pensamos seguir aplicando el esquema actual (Prácticos, 2 parciales, recuperatorio, trabajo de investigación) en la nueva metodología y lenguaje.
- **Aprobación de la asignatura.** Para los alumnos que la han regularizado en el esquema anterior, mantendremos también la evaluación anterior, por lo menos hasta mediados de 2007, a menos que recibamos otras directivas. Para los alumnos del actual año académico, será de acuerdo a los contenidos establecidos en el Programa/Cronograma de la asignatura.

Ing. Jorge P. Tymoschuk
Dir. de Cátedra.