**JInstituto Tecnológico de Costa Rica**

**Curso IC-2101 Programación Orientada a Objetos**

**Oscar Víquez Acuña**

**Descripción del primer proyecto programado**

**Valor: 20%**

**Administra-Horarios para cursos IC-SSC**

**Objetivos**

* Aprender el lenguaje Java y familiarizarse con la Programación Orientada a Objetos.
* Aplicar conceptos de diseño de aplicaciones utilizando la teoría de Orientación a Objetos.
* Generar una herramienta que canalice todos los conceptos estudiados del tema.

**Descripción**

La coordinación de la Carrera de Ingeniería en Computación, con sede en San Carlos, se encargará de la confección de los horarios de clase de los cursos de la carrera que se imparten. Este proceso lejos de ser manual, debe ser automático de manera que se diseñe e implemente un software que permita automatizar esta gestión. El principal objetivo para la coordinación, es la generación automática de los horarios a partir de determinadas informaciones como:

* Las asignaturas de las cuales consta cada semestre.
* Las asignaturas a impartir por cada profesor.
* Las aulas disponibles para impartir las clases.

De cada profesor interesa conocer su nombre, cédula y horario, así como el departamento/unidad al que pertenece. Cada departamento tiene un nombre. Las asignaturas tienen un nombre y un número de créditos, que indica el número de horas de dedicación a la semana (1 crédito equivale a 3 horas). Deben hacer la diferenciación de cuáles recursos o componentes forman parte de la carrera de forma exclusiva (cursos de carrera vs cursos de servicio de otras escuelas) y cuáles no.

Hay dos tipos de asignaturas: las que son teóricas, de las que debe recogerse la página Web donde se publican los apuntes, y las que son prácticas, que deben tener asociado el material de apoyo y el sistema operativo a emplear. Igualmente hay dos tipos de aulas: para clases de teoría y para clases de laboratorio (asignaturas prácticas). En general, las aulas tienen un nombre, número de aula, ubicación y capacidad. De las aulas de teoría interesa conocer si tiene aire acondicionado y equipo multimedia, y de las de prácticas se quiere saber de cuánto y cual equipo técnico disponen. Los posibles semestres son ocho, de primero al octavo, pero interesa que el sistema sea abierto para poder evolucionar a situaciones con un número de semestres mayor o menor. Así mismo se cuenta con un número de asignaciones o cursos por semestre que de igual forma podría ser menor o mayor. Hay que tener en cuenta que una asignatura puede formar parte de cualquier semestre aunque exista estipulada para uno en particular en el programa de la carrera. Los días de clase van de lunes a viernes, y las horas posibles son de 7:00 a 11:30 y de 12:30 a 16:00. Los horarios nocturnos son opcionales. Las asignaturas se imparten en bloques de 50 minutos contemplando además las pausas entre asignaciones de 5 minutos.

Los profesores deben introducir en el sistema la información de las asignaturas que imparten. También deben especificar en cuales días desean tener clases de acuerdo con sus cursos asignados. Este dato es utilizado por la coordinación para generar automáticamente los horarios. Esta opción de generación de horarios automática para un semestre cualquiera (se debe seleccionar el semestre que se desea) debe realizarse tomando en cuenta los cursos del semestre, los horarios sugeridos por el profesor del curso y las aulas disponibles, de manera que se genere un horario sin choques de aulas ni cursos ni profesor, en el mismo período (semestre). No obstante, puede darse el caso de que la generación automática no obtenga una solución completa al problema de los horarios y sea preciso que se retoque manualmente la asignación de horas de clase a las asignaturas en las diferentes aulas. De esto, y de la introducción del resto de informaciones necesarias, también se encarga la coordinación.

El sistema también debe permitir la consulta de horarios por parte de los alumnos para que puedan ver los detalles asociados a los cursos en un semestre (bloque) particular, a saber, tipo curso, aulas, características en general, etc; en fin no deben quedar detalles sin mostrar porque no tendría sentido almacenar información que no se puede ver. Si el horario no ha sido generado previamente por el coordinador para un semestre en específico, esta opción mostrará “Horario no disponible”.

Deberá implementarse un login que permita verificar si es un estudiante, profesor o coordinador quien accede al sistema.

**Requerimientos de programación orientada a objetos**

* Es importante recordar utilizar adecuadamente los modificadores de acceso public, private, protected, etc de manera efectiva y eficiente.
* Los métodos set y get deben ser desarrollados dentro de cada clase cuando correspondan.
* Deberá generarse un javadoc del código final de la aplicación.
* El diseño de las clases para la solución del proyecto será propuesta por cada grupo de trabajo, sin embargo, dicha implementación deberá incluir al menos un caso de herencia y una clase abstracta o un interface (en este punto se debe justificar en la documentación externa el porqué de la escogencia entre clase abstracta o interface).
* Sea que prefieran utilizar una clase abstracta o un interface, deberán definir al menos tres métodos abstractos, estos deberán ser implementados en las clases que extiendan de esta y se deben crear siguiendo un sentido lógico en cuanto a su creación.
* Las clases generadas deberán encontrarse clasificadas en paquetes que permitan diferenciar la funcionalidad de estas dentro de la solución.

**Documentación**

Portada.

Descripción del problema.

Solución del problema (la última solución, indique cuales son las estructuras utilizadas, lógica de cómo se trabajó para realizar el programa, como fue el recorrido utilizado para resolver cada una de las consultas y operaciones).

Diagrama de clases para los objetos definidos dentro de la solución.

Análisis de Resultados (Resultados finales, indique que partes están completas, cuales defectuosos, y cuales no se realizaron y el porqué).

Conclusiones y recomendaciones (con respecto al proyecto, lenguaje, tiempo para la realización, la explicación en clase, horas de consulta etc.)

Literatura citada (mínimo de debe incluir dos con su respectivo resumen).

Se deberá también documentar internamente el código siguiendo las convenciones de desarrollo para Java.

**Aspectos administrativos**

* La tarea será desarrollada en grupos de trabajo de máximo 2 personas. Cualquier acción de plagio será sancionada con un **0** en la nota y la carta al expediente respectiva a todos los implicados.
* Se realizará un control de versiones vía el servicio Web denominado GitHub los días 18 y 25 de septiembre y 2 de octubre de 2013. Deben brindar acceso al profesor por medio del correo [oviqueztec@gmail.com](mailto:oviqueztec@gmail.com). La evaluación final del trabajo tendrá un rubro de calificación referente al correcto uso del control de versiones.
* Los informes de avance, documentación externa y la interna pueden ser redactados en inglés y esto tendrá un plus de 5 puntos extra en la nota final en función a la calidad de los escritos en el idioma inglés.
* La fecha de entrega será el miércoles 9 de octubre de 2013 a las 23:55.
* La entrega será mediante la plataforma Tec-Digital.
* Cualquier archivo infectado con virus no será revisado. Se recomienda que se empiece a trabajar lo más antes posible.