Instituto Tecnológico de Costa Rica

Ingeniería en Computación

"I Proyecto - UNISCH"

Programación Orientada a Objetos

Fabricio Salazar Espinoza	**fabri10.se@gmail.com**	**8857-6838**
Steven Rojas Ramírez	**stvrjs.k12@gmail.com**	**8765-2545**

Sede San Carlos

Índice

1. Descripción del Problema

Para el siguiente proyecto se debe realizar un sistema que permita llevar a cabo la confección de los horarios de clase de los cursos de la carrera de Ingeniería en Computación, de manera automatizada. Este proceso se debe realizar tomando como referencia los siguientes puntos: asignaturas de las cuales consta cada semestre, asignaturas a impartir por cada profesor, y las aulas disponibles para impartir las clases.

Se cuenta con una especificación del proyecto, donde se mencionan los requisitos que debe cumplir, por lo tanto se debe realizar una interpretación de la información, con el objetivo de elaborar un diagrama de clases, que permita visualizar el diseño que tendrá la aplicación.

Posteriormente, se llevará a cabo la etapa de implementación o codificación, lo cual se realizará con base al diagrama de clases desarrollado anteriormente.

Finalmente, se realizarán constantes revisiones del proyecto a lo largo del proceso, a través de un control de versiones llamado Github.

2. Solución del Problema

Para solucionar este problema se inició visualizando los escenarios como se conocen en la vida real, como vemos los procesos que son necesarios en la universidad. Luego creando el diagrama de clases, para tener una idea más clara de cómo estarían organizadas las estructuras y como estarían relacionadas entre ellas.

El proyecto está básicamente desarrollado tomando en cuenta 4 estructuras que se identificaron como principales, las cuales se alimentan entre ellas y contienen las subestructuras. Estas estructuras son:

- Una lista de aulas, la cual contiene todas las aulas de la institución.

Esta es una estructura principal e independiente, ya que las aulas no se crean cada vez que uno curso, sino que "siempre están disponibles" para cada semestre y para los cursos desde el momento en que se crean.

- Lista de escuelas, la cual contiene todas las escuelas de la institución.

Esta es otra estructura principal e independiente, ya que no se necesita que exista ninguna de las demás estructuras principales, para que esta exista. En ella se contienen las listas de profesores que imparten los cursos de su respectiva escuela y los cursos de la misma, los cuales pueden ser impartidos por profesores de su respectiva escuela.

- Lista de semestres, la cual contiene cada semestre que va ocurriendo a lo largo de la vida del proceso de educación de la institución.

Esta es otra estructura principal e independiente, ya que se puede crear sin que existan las demás listas. En ella se encapsulan los cursos que se estarán dando en cada semestre. Aquí cada semestre tendrá su horario y su aula asignada.

- Lista de usuarios, la cual contiene a las personas que trabajan, estudian o coordinan en la institución.

Esta lista, otra independiente y principal, la cual almacena todos los usuario de la institución, para luego identificarlos a la hora de ingresar en el sistema. De esta lista se toman los profesores, los cuales son asignados luego a las escuelas.

Resolver el choque de horarios:

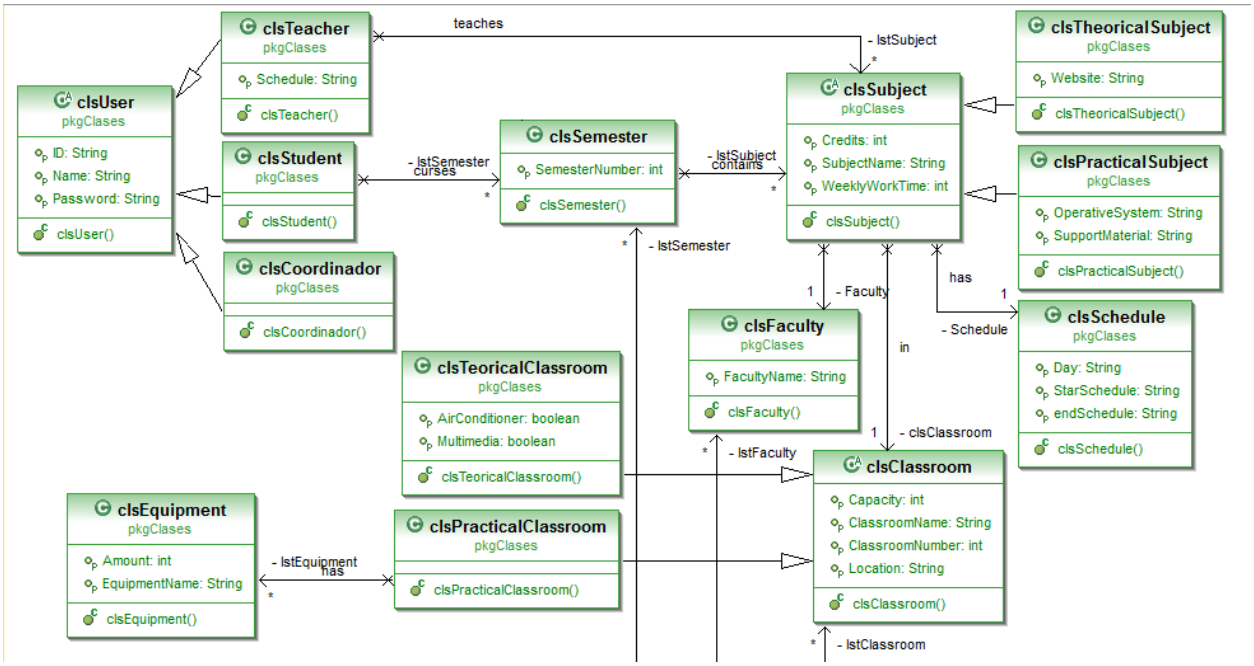
Para resolver esto lo que se hizo fue:

- Recorrer la lista de cursos del profesor, validar que el curso sea del semestre indicado, si lo es, entonces revisa el horario del curso actual con respecto al horario nuevo, si genera conflictos entonces tira una excepción, sino continua con el siguiente curso. Esto lo que quiere primero es revisar todos los cursos del profesor, para ver que no genere conflictos con el horario del profesor.
- Luego de evaluar todos los cursos del profesor, evalua que no genere conflictos con el horario del semestre, para esto recurre los cursos del semestre revisando que cada

curso no tenga conflictos con el horario del nuevo horario del curso, a setear. Si genera conflictos entonces tira una excepción

- Finalmente se toma el curso y se setea el nuevo horario.

3. Diagrama de Clases



4. Análisis de Resultados

Tabla de resultados

	Implementado	Observaciones
Las asignaturas de las cuales consta cada semestre.	✓	Funcionando al 100%
Las asignaturas a impartir por cada profesor.	✓	Funcionando al 100%
Las aulas disponibles para impartir las clases.	✓	Funcionando al 100%

Observaciones:

Se había pedido que el coordinador pudiera modificar los horarios de los cursos, para generar conflictos y que el sistema le informara. El sistema evalúa los horarios e informa si hay choques en el semestre o si el horario del profesor tiene choques, sin embargo, en la interfaz no se implementó algo para cambiarlo, por lo tanto habría que cambiarlo en el "PrjUNISCH.java" el cual es el que hace las inserciones en la parte llamada: "Subject-Schedule insertions".

5. Conclusiones y Recomendaciones

Quizás podría ser importante explicar de mejor manera el trabajo a realizar en la especificación, ya que en este proyecto en particular, quedó un poco abierto y dejaba muchas dudas, y algunos compañeros que no consultaron, hicieron trabajo de más que luego no les sirvió para nada (la creación de inserts reflejados en la interfaz, para insertar los datos).

6. Literatura Citada

StackOverflow (2010). How to split a String in Java. Consultado el 1 de octubre de 2013 en http://stackoverflow.com/questions/3481828/how-to-split-a-string-in-java.

StackOverflow (2010). In java, What do the return values of Comparable.compareTo mean?. Consultado el 2 de octubre de 2013 en http://stackoverflow.com/questions/3767090/in-java-what-do-the-return-values-of-comparable-compareto-mean.