

Hacia una integración y articulación de contenidos en Carreras de Ciencias de la Computación.

Experiencia desarrollada en el último año de la carrera de Analista en Computación de la Universidad Nacional de Río Cuarto.

*Marcelo Uva, Paola Martelloto, Marcela Daniele,
Fabio Zorzán, Ariel Arsaute, Mariana Frutos*

*Universidad Nacional de Río Cuarto, Facultad de Ciencias Exactas, Fco-Qcas y Naturales,
Dpto. de Computación, Ruta 36 Km 601 CP X5804BYA, Río Cuarto, Córdoba, Argentina,
{ uva , paola , marcela , fforzan , aarsaute , mfrutos }@dc.exa.unrc.edu.ar*

Resumen

La enseñanza de las Ciencias de la Computación no es ajena a las problemáticas que se manifiestan en otras disciplinas. La integración y articulación de contenidos se ha planteado desde un principio como un elemento fundamental para la formación del profesional informático. Desde el año 2006 hasta la fecha, el cuerpo docente autor de este artículo, ha realizado una serie de trabajos con el objetivo de mejorar los procesos de enseñanza-aprendizaje dentro de la carrera de Analista en Computación dictada en el Departamento de Computación, perteneciente a la Facultad de Ciencias Exactas, Físico-Químicas y Naturales de la Universidad Nacional de Río Cuarto.

En este trabajo se presenta una experiencia orientada a articular los contenidos curriculares entre las asignaturas Análisis y Diseño de Sistemas, Base de Datos y Diseño de Algoritmos pertenecientes al último año de la carrera de Analista en Computación. La propuesta consiste en el diseño y ejecución de de un proyecto-taller integrador en donde cada asignatura pueda llevar a la práctica y en forma conjunta y coordinada, los conceptos más relevantes de su currícula.

1. Introducción

Los procesos de enseñanza-aprendizaje que tienen lugar en las Ciencias de la Computación no escapan a los problemas recurrentes al resto de las ciencias. La problemática de la integración y articulación de contenidos se ha planteado como un elemento principal al momento de formar a un profesional informático. La evolución continua de las tecnologías informáticas, directamente vinculadas a las ciencias de la computación establece un desafío constante. Los alumnos, futuros profesionales, deberán adquirir la capacidad de aprender, integrar y articular nuevos conocimientos para hacer frente a los desafíos que el mercado les imponga.

Según [1] “La integración curricular hace referencia a la forma en que se organizan los contenidos temáticos del currículo en actividades que favorecen la globalización de los saberes. Se pretende superar la separación por asignaturas de las áreas del conocimiento, la fragmentación de los aprendizajes, de manera que el aprendizaje sea funcional. Es decir, que el alumno lo vea funcionando en una situación o problema real y construya las estrategias que le permitan establecer nuevas relaciones significativas entre contenidos diversos siendo capaz de realizar aprendizajes

significativos por sí mismo, en una amplia gama de situaciones y circunstancias.”

Desde el año 2006 hasta la fecha, el cuerpo docente autor de este trabajo y encargado del dictado de asignaturas del último año de la carrera de Analista en Computación pertenecientes al Departamento de Computación de la Universidad Nacional de Río Cuarto (UNRC) ha propuesto una serie de iniciativas con el objetivo de mejorar los procesos de enseñanza-aprendizaje. Este trabajo ha sido enmarcado institucionalmente dentro de los Proyectos de Innovación e Investigación para la Mejora de la Enseñanza de Grado (PIIMEG) de la UNRC. Los trabajos realizados hasta el momento han sido orientados a las siguientes temáticas:

- Investigación y utilización de herramientas para la automatización de la gestión de proyectos de software [2]. Se logró fomentar el uso de metodologías de desarrollo de software en proyectos reales, promoviendo la división de roles operativos y gerenciales entre los integrantes de un grupo de desarrollo de software. Los grupos fueron capacitados para la evaluación y selección de diferentes herramientas aplicables a las actividades de gestión de software.
- Diseño de un framework genérico [3]. En este trabajo se presentó una solución genérica a los problemas de inserción, eliminación, modificación y búsqueda de elementos, incluyendo tecnologías distribuidas, a través del uso de servicios web. Con este trabajo se fomentó la utilización de patrones de diseño de una manera sistemática y disciplinada.
- Investigación y desarrollo de herramientas destinadas a la automatización de las actividades de un proceso de producción de software para funcionalidades

específicas [4]. Este trabajo fue realizado por un grupo de tesis bajo la supervisión de los autores de este trabajo. Los alumnos estudiaron a fondo la metodología a automatizar para posibilitar la generación de todos los artefactos correspondientes a cada etapa.

- Aplicación de técnicas de ingeniería inversa [5]. Con este trabajo se logró capacitar a los alumnos para el análisis y evaluación técnicas de ingeniería inversa aplicables a proyectos de software, promoviendo e incentivando la reutilización de componentes de software.

Continuando con esta misma línea de trabajo, y en concordancia con los objetivos planteados inicialmente, presentamos una experiencia orientada a articular e integrar los contenidos curriculares entre las asignaturas Análisis y Diseño de Sistemas de Software, Base de Datos y Diseño de Algoritmos, del último año de la carrera de Analista en Computación de la UNRC. La propuesta radica principalmente en el diseño y ejecución de un proyecto-taller integrador en donde cada asignatura pueda llevar a la práctica y de forma conjunta y coordinada los conceptos más relevantes de su currícula. El resto del trabajo está organizado de la siguiente manera: en la sección 2 se muestran los contenidos básicos dictados en cada una de las asignaturas Análisis y Diseño de Sistemas de Software, Base de Datos y Diseño de Algoritmos, junto con un breve resumen de los objetivos. En la sección 3, se presenta la nuestra experiencia. La sección 4, muestra una evaluación del proyecto-taller integrador realizada por docentes y alumnos. Finalmente en la última sección, las conclusiones y trabajos futuros.

2. Asignaturas de 3er. año de la carrera de Analista en Computación

La carrera de Analista en Computación de la UNRC posee tres asignaturas en el primer cuatrimestre del 3er. año: Análisis y Diseño de Sistemas, Base de Datos y Diseño de Algoritmos. Dichas materias conciben al desarrollo de software como un proceso ingenieril [8,9] para lo cual se estudian metodologías de desarrollo, se aplican conocimientos de bases de datos y se utilizan técnicas de diseño de algoritmos a fin de incrementar el valor agregado del producto final.

Las tres asignaturas poseen trabajos finales en donde el alumno pueden llevar a una práctica concreta lo estudiado durante el cursado.

Probablemente, la naturaleza de los problemas abordados en cada asignatura haya contribuido a la falta de integración en los contenidos. Por ejemplo, si en la asignatura de Análisis y Diseño de Sistemas de Software se plantea el desarrollo de un software de gestión aplicando una metodología determinada, los alumnos rápidamente podrán integrar los conceptos de la asignatura Base de Datos al momento de requerir la persistencia de la información, pero, difícilmente lo podrán hacer con los contenidos de Diseño de Algoritmos. Esto se debe a que para este problema en particular, quizás, no se requiera la utilización algoritmos de mayor complejidad. Del mismo modo ocurriría si en Diseño de Algoritmos se plantea el desarrollo de un sistema que requiera la utilización de algoritmos más complejos y no requiera persistencia de los datos.

La modalidad de dictado en las tres asignaturas es similar, cuenta con clases teóricas, prácticas y proyectos.

Nuestra propuesta plantea un proyecto de desarrollo de software en donde sea posible involucrar los conocimientos prácticos provenientes de las tres asignaturas. A continuación se mencionan los contenidos y objetivos de cada asignatura.

2.1 Análisis y Diseño de Sistemas

La asignatura tiene como finalidad introducir al alumno en los conceptos básicos de Ingeniería de Software, métodos de desarrollo y su evolución. Para ello se estudian cada una de las etapas del ciclo de vida, se utilizan técnicas de especificación de problemas, a través de un lenguaje gráfico de modelado estándar como por ejemplo UML [6]. Se busca que el alumno adquiera fuertemente conceptos de diseño y construcción de modelos genéricos dando solución a problemas con similares característica utilizando patrones de diseño. También se pretende que el alumno adquiera conocimientos de re-ingeniería del software: ingeniería inversa e ingeniería directa.

2.2 Base de Datos

Los contenidos de Base de Datos están orientados a que el alumno adquiera experiencia en la modelización y diseño de bases de datos, especialmente haciendo uso del modelo relacional. Se estudian contenidos vinculados con el diseño de base de datos relacionales basado en la teoría de dependencia funcionales, esto otorga una fuerte base teórica al alumno a la hora de poder diseñar o analizar la calidad de un diseño ya existente. Es fundamental que el alumno de tercer año adquiera una experiencia en el manejo de bases de datos relacionales, por esto es que se incluye en los contenidos de la materia el Structured Query Language (SQL). Los prácticos se realizan sobre dos

motores de bases de datos libres: PostgreSQL [10] y MySQL[11]. El objetivo de utilizar dos motores diferentes es otorgar experiencia en más de un sistema de base de datos relacional, evaluando ventajas y desventajas de cada uno de ellos. También se incluyen contenidos básicos para el manejo de transacciones y de bases de datos avanzadas.

2.3 Diseño de Algoritmos

La asignatura Diseño de Algoritmos tiene por objetivo principal la familiarización del alumno con técnicas de diseño de algoritmos, aprovechando los mecanismos disponibles para su implementación en lenguajes de programación particulares. Se espera sin embargo que los alumnos puedan independizarse del lenguaje de programación. Por otro lado, se pretende que los alumnos puedan aplicar estas técnicas en problemas de distinta complejidad. Las temáticas abordadas en la asignatura están fuertemente vinculadas al uso adecuado de tipos abstractos de datos, su definición, y de anotaciones de programas tales como invariantes y contratos en términos de pre- y post-condición. Se utilizan también las estructuras de datos más relevantes, vistas en las asignaturas anteriores del área de algoritmos.

3 – Proyecto- taller

Para la definición de las temáticas a trabajar dentro del proyecto-taller se realizaron una serie de reuniones en donde participaron los docentes responsables y auxiliares de las tres asignaturas involucradas. Desde la asignatura Análisis y Diseño de Sistemas se planteó la idea de trabajar con un software ya existente. El objetivo fue poner al alumno en una situación muy común en la actualidad, que es incorporarse a un proyecto que ya viene

realizando un equipo de desarrollo y donde se desean incorporar nuevas funcionalidades. Para ello el alumno, futuro profesional, debe contar con conocimientos en ingeniería inversa y re-ingeniería. La ingeniería inversa [12] es el proceso de analizar un sistema para construir un modelo con mayor nivel de abstracción. La ingeniería inversa es concebida como un proceso que recorre en sentido contrario las etapas del ciclo de vida de un software, partiendo del código fuente. Una vez que se logra definir un modelo a partir del código, es cuando se aplican las técnicas de re-ingeniería para realizar actividades de corrección de fallas, mejoramiento y evolución del software.

Luego de analizar varias aplicaciones disponibles en repositorios de acceso público se decidió trabajar con la aplicación denominada “Plarpebu”. La aplicación “Plarpebu”<http://sourceforge.net/projects/plarpebu/> es una aplicación “*Open Source*” (código abierto), desarrollada en Java, carece de documentación y permite a un usuario reproducir archivos de audio almacenados en un soporte persistente. Por otro lado, la aplicación implementa un karaoke. Plarpebu es gratuita y multiplataforma. Funciona para Linux, Mac Os X, Windows y, en general, cualquier sistema que soporte Java.

3.1 Metodología de trabajo

Para la ejecución del proyecto-taller se organizó a los alumnos en grupos de tres o cuatro. El trabajo en grupo es el escenario más habitual que deberán afrontar los alumnos al momento de incorporarse al mercado. Estos grupos fueron mantenidos durante el cursado de las tres asignaturas. En los casos en que un alumno no haya cursado alguna de las tres materias, se buscó una alternativa de trabajo en cada asignatura.

Los grupos tuvieron dos horas semanales adicionales al cursado destinadas al proyecto-taller. En estos encuentros se presentó la aplicación sobre la cual se iba a trabajar, las metas del trabajo, se clarificaron dudas guiando a los alumnos en cada una de las etapas del proyecto. El trabajo fue realizado de manera incremental a lo largo de todo el cuatrimestre. La metodología de desarrollo establecida por el cuerpo docente fue la del Proceso Unificado [7].

A continuación se detalla la lista de actividades solicitada a cada uno de los grupos.

Actividad Nro.1: Captura de Requerimientos

Objetivos

- Análisis y estudio de la aplicación Plarpebu.
- Determinación de las funcionalidades de la aplicación.
- Obtención de un documento con la especificación de los requerimientos.

Tareas

1. Descargar e instalar la aplicación Plarpebu. <http://sourceforge.net/projects/plarpebu>
2. Utilizar la aplicación, observar su funcionamiento, analizarla en detalle.
3. Realizar ingeniería inversa y obtener un documento que refleje la Captura de Requerimientos, incluyendo los siguientes items:
 - a) Introducción teórica a la etapa de Captura de Requerimientos.
 - b) Diagrama de Casos de Uso del Sistema con las funcionalidades existentes en el Plarpebu.
4. Confeccionar una descripción breve de cada caso de uso detectado.

5. Confeccionar la descripción detallada de los casos de uso: Add File, Add Dir, Load List y Save List.

Actividad Nro. 2: Nuevas funcionalidades

Objetivos

- Extensión de las funcionalidades detectadas en la aplicación Plarpebu.
- Incorporación de una nueva funcionalidad propuesta por el cuerpo docente orientada a la aplicación de uno de los algoritmos estudiados en la asignatura Diseño de Algoritmos.
- Extensión del documento de la especificación de requerimientos generado en la actividad anterior.

Tareas

1. Proponer una lista de funcionalidades que se podrían incorporar a la aplicación.
2. Como se ha explicado anteriormente, la aplicación Plarpebu permite crear y manipular listas de temas musicales. El cuerpo docente desea que el sistema permita realizar lo siguiente: dado un tiempo t y un conjunto de temas musicales donde cada uno posee una duración y valuación, el sistema deberá producir una nueva lista en donde la suma de las duraciones de cada tema no supere el tiempo t maximizando las valuaciones de los temas.
3. Extender el Diagrama de Casos de uso de sistema Plarpebu, agregando las nuevas funcionalidades junto con una descripción breve de cada una de ellas.

Actividad Nro. 3: Diseño

Objetivos

- Utilización de técnicas de Ingeniería Inversa.

- Elaboración del diseño persistente de datos.
- Elaboración e implementación de una base de datos relacional.

Tareas

- 1 Aplicar técnicas de Ingeniería Inversa y obtener el diseño actual de la aplicación. Producir un diagrama de clases de diseño.
- 2 Generar el Diagrama de Clases Persistentes, considerando el diseño actual de la aplicación Plarpebu, más las estructuras necesarias para soportar las nuevas funcionalidades planteadas.
- 3 Diseñar una base de datos relacional que permita almacenar la información persistente modelada en el diagrama de clases del ítem anterior.

Actividad Nro. 4: Implementación

Objetivos

- Implementación e integración de las nuevas funcionalidades de la aplicación.
- Ejecución de la etapa de Prueba o Testing de las nuevas funcionalidades.

Tareas

- 1 Junto con el cuerpo docente, se seleccionará uno de los casos de uso propuestos por el Grupo.
- 2 Implementar la base de datos relacional diseñada en la actividad nro.3 utilizando los Sistemas Gestores de Base de Datos (SGBD) MySQL o PostgreSQL.
- 3 Extender y/o modificar el diseño actual de la aplicación Plarpebu considerando las nuevas funcionalidades a implementar.
- 4 Implementar el caso de uso seleccionado en el primer ítem junto con el solicitado

por el cuerpo docente expresamente en la actividad nro.2.

- 5 Realizar la prueba de las nuevas funcionales implementadas.

Las actividades planteadas dentro de este proyecto-taller fueron planificadas y coordinadas temporalmente de acuerdo a los contenidos trabajados en las clases teóricas y prácticas de las tres asignaturas.

La asignatura Análisis y Diseño de Sistemas estableció la estructura general del desarrollo del proyecto-taller a través de la metodología de desarrollo del Proceso Unificado[7]. Además, fue necesaria la aplicación de técnicas de ingeniería inversa y re-ingeniería de acuerdo a la naturaleza del trabajo. Por otro lado, Plarpebu no poseía base de datos. La única información persistente que administraba (mediante archivos de texto) eran listas de temas musicales. Es por ello que se estableció la generación de un diagrama de clases persistentes y su posterior mapeo a un SGBD. Por su parte y para integrar los contenidos de Diseño de Algoritmos el cuerpo docente incorporó una funcionalidad a la lista de aquellas identificadas por cada grupo. Dicha funcionalidad fue dirigida para que los alumnos pudieran abstraerse del problema y utilizaran el algoritmo KP (*Knapsack problem*) [13] o comúnmente conocido como el problema de la mochila.

4 – Resultados

La evaluación de los resultados fue realizada por el grupo de docentes encargado del dictado de las asignaturas junto con aportes de alumnos, docentes de otras asignaturas, asesores pedagógicos y directivos.

Los instrumentos de recolección de información utilizados fueron los siguientes:

- Reuniones de discusión entre los docentes de la asignatura.
- Evaluación de los informes de los talleres presentados por los alumnos.
- Encuesta y entrevistas individuales a alumnos, auxiliares y docentes participantes.

Los alumnos participaron en forma activa y muy entusiasta, lo que permitió obtener buenos resultados. En los informes parciales presentados por los alumnos (al finalizar cada etapa), se destacó el empeño y dedicación.

Al finalizar el proyecto-taller los alumnos completaron una encuesta, que estaba orientada a examinar cuestiones vinculadas a la integración de los temas de las tres materias involucradas, junto con la coordinación de las tres asignaturas, tiempo dedicado a cada una de ellas, la organización interna del grupo, etc. De las encuestas se ha podido obtener la siguiente información.

Respecto de la integración de los contenidos: los alumnos percibieron una fuerte integración entre Análisis y Diseño de Sistemas y Base de Datos, no así con Diseño de Algoritmos. Según nuestro análisis, esto pudo deberse a que la naturaleza del proyecto planteado no se adecuó totalmente a la aplicación de diversas técnicas y algoritmos estudiados en la asignatura Diseño de Algoritmos. Si bien todos los grupos implementaron la funcionalidad que les exigía la aplicación del algoritmo KP (*Knapsack problem*), éstos no pudieron establecer una relación directa con la asignatura Diseño de Algoritmos, tal vez por ser este uno de los tantos algoritmos que se estudian en esta asignatura. Lo ideal hubiese sido diseñar un proyecto de mayor complejidad, en donde fuese necesaria la aplicación de un mayor conjunto de técnicas y algoritmos estudiados en la asignatura Diseño de Algoritmos. El problema aquí es que resulta complicado el desarrollo de un proyecto-taller de mayor dificultad, principalmente por el

tiempo que se dispone. Probablemente, elevar el grado de complejidad del trabajo sería contraproducente para los alumnos ya que se sobrecargarían de actividades, lo que en un primer momento era una de las cuestiones que se quería evitar.

Con respecto a Base de Datos y Análisis y Diseño de Sistemas la relación fue muy buena debido a que la aplicación Plarpebu no poseía base de datos, por lo que cada grupo necesariamente debió diseñar e implementar un soporte de datos persistente para almacenar la información deseada.

De las reuniones de los docentes de las tres asignaturas, es posible destacar que se debieron aunar esfuerzos ajustando las planificaciones de cada materia a la del proyecto-taller. Esto fue beneficioso también ya que permitió una mayor coordinación y cooperación entre las asignaturas a la hora de establecer horarios de exámenes parciales, consultas y clases prácticas.

5- Conclusiones y trabajos futuros

Una vez finalizada la ejecución del proyecto-taller integrador y luego de analizar los resultados de las encuestas y las prácticas docentes, es posible mencionar algunas conclusiones.

El proyecto fue planteado con el objetivo de definir una actividad práctica organizando el temario de las asignaturas de tercer año de la carrera de Analista en Computación de la UNRC para facilitar la integración de los contenidos. Se buscó superar la separación por asignaturas con el fin de generar un aprendizaje funcional. Es decir, que el alumno lo vea funcionando en una situación o problema real y construya las estrategias que le permitan establecer nuevas relaciones significativas entre contenidos.

La estructura del taller fue definida en su mayoría por la asignatura Análisis y Diseño de

Sistemas mediante la aplicación de una metodología de desarrollo de software concreta. A partir de allí se integraron los temas de Base de Datos y en menor medida los de Diseño de Algoritmos. De acuerdo a los resultados obtenidos de las encuestas, sería interesante en años siguientes, el planteo de un proyecto de otra naturaleza. Dicho proyecto debería promover en los alumnos la utilización de metodologías de desarrollo, conocimientos de base de datos y, fuertemente, la aplicación de técnicas y algoritmos más complejos. Si bien este fue el objetivo inicial del proyecto-taller, la carga horaria requerida era muy grande, por lo que el proyecto-taller debió ajustarse a una carga horaria factible de afrontar por los alumnos. Se deberá trabajar fuertemente en diseñar un proyecto-taller integrador que pueda equilibrar las demandas de las tres asignaturas sin sobrecargar en demasía a los alumnos.

Los alumnos trabajaron con un sistema ya implementado. La aplicación Plarpebu no contaba con documentación alguna (salvo escasos comentarios en el código). Durante el cursado de la asignatura de Análisis y Diseño de Sistemas de Sistemas se hace mucho hincapié en la importancia de la utilización de metodologías de desarrollo de software. Esto permite, además de concebir a la producción de software como un proceso ingenieril, establecer una forma ordenada y disciplinada para fabricar un sistema informático. Los alumnos pudieron vivenciar los problemas ocasionados por la falta de documentación del sistema Plarpebu. Para poder obtener el diseño de la aplicación se utilizaron técnicas de ingeniería inversa, posteriormente, los alumnos pudieron realizar el análisis de la arquitectura de la aplicación. Algunos grupos lograron aplicar actividades de refactorización de código, por ejemplo detectando código duplicado.

La incorporación de nuevas funciones motivó a los alumnos a realizar tareas de

investigación, proponiendo funciones muy interesantes. Por ejemplo, un grupo incorporó la posibilidad de interactuar con una red social, de forma tal que cuando un tema musical se reproducía dentro del Plarpebu, el propio sistema enviaba información a un servidor remoto registrando una serie de estadísticas del usuario. Otro grupo implementó la funcionalidad de buscar en la web la letra del tema que se estaba reproduciendo. Para ello se utilizó una API de Yahoo. Otro grupo agregó la reproducción de videos.

Como trabajos futuros se tiene planificado repetir la experiencia durante el año 2012. También se plantea la realización de un seguimiento a los grupos que hayan realizado el taller al momento de dar comienzo a su trabajo final de carrera. Nuestra hipótesis es que la realización de un taller integrador favorece en gran medida el desarrollo del trabajo final, debido a que el alumno ya tiene incorporada la experiencia del trabajo dentro de un sistema real.

Bibliografía

- [1] Águila, Venían. Apunte conceptos básicos sobre integración curricular, (2002). <http://www.worldbank.org/wbi/ictforeducation/efa/docs/nicaragua>.
- [2] Daniele M., Uva M. Martelloto P., Picco G. "Aplicación de herramientas CASE a la enseñanza de Ingeniería de Software: Gestión de la Configuración de Software y Testing Funcional" . TE&ET 2010. El Calafate, 6-7 de mayo de 2010.
- [3] Daniele M., Romero D. "Construcción de un framework para la enseñanza de patrones de diseño". World Congress on Computer Science, Engineering and Technology Education (WCCSETE2006). Itanhaém / Santos, marzo de 2006.

- [4] Daniele M.,Uva M., Martelloto P. “Hacia una automatización de los procesos de desarrollo de software” Workshop de Investigadores en Ciencias de la Computación (WICC) 2010. El Calafate, 6 y 7 de mayo de 2010.
- [5] Daniele M., Martellotto P.,Uva M.,Montovani P. “Ingeniería Inversa en la enseñanza deAnálisis y Diseño de Sistemas”. CIESC 2009. XVII Congreso Iberoamericano de. Educación Superior en Computación. Pelotas, Brasil, 22-23 de Septiembre de 2009.
- [6] Booch G., Rumbaugh J., Jacobson I., The Unied Modeling Language. 2.0. Addison Wesley, Second Edition, 2005.
- [7] Jacobson I., Booch G., Rumbaugh J.. El Proceso Unificado de Desarrollo de Software.
- [8] Roger Pressman-Software Engineering. A Practitioner’s Approach. Mc-GrawHill-5ta E 2001.
- [9] Ian Sommerville - Software Engineering. Addison-Wesley, 5ta Edición. 1996.
- [10] Postgresql <http://www.postgresql.org/>
- [11] MySQL <http://www.mysql.com/>
- [12] Pressman, Roger S. (2002). “Ingeniería del Software”. (Quinta Edición) Madrid: McGraw Hill.
- [13] Knapsack Problems H. Kellerer, U. Pferschy,D.Pisinger. Knapsack Problems, Springer Verlag ISBN 3-540-40286-1.