

# Enseñando Ingeniería de Software en un Programa de Ciencias de la Computación

**Marcelo Jenkins**

Escuela de Ciencias de la Computación e Informática

Universidad de Costa Rica

San Pedro, Costa Rica 2060

mjenkins@cariari.ucr.ac.c.r

## *Resumen*

*Este artículo describe la estrategia que ha seguido la Escuela de Ciencias de la Computación e Informática de la Universidad de Costa Rica para enseñar ingeniería de software en un programa de bachillerato en ciencias de la computación e informática. Se explica en detalle el diseño de los cursos del área de ingeniería de software y cómo encajan éstos dentro del currículum global de la carrera.*

*Los aspectos descritos en este artículo pueden interesar profesores e instructores que desean formar ingenieros de software a nivel de pregrado.*

**Palabras claves:** enseñanza de ingeniería de software.

## 1. Introducción

Enseñar ingeniería de software en un programa de ciencias de la computación e informática no es fácil debido al poco número de créditos que tradicionalmente se dedican a esta área dentro del pensum total de un programa de ciencias de la computación.

Este artículo describe la estrategia que ha seguido la Escuela de Ciencias de la Computación e Informática de la Universidad de Costa Rica para enseñar ingeniería de software en un programa de bachillerato en ciencias de la computación e informática. Se explica en detalle el diseño de los cursos del área de ingeniería de software y cómo encajan éstos dentro del currículum global de la carrera.

Los aspectos descritos en este artículo pueden interesar profesores e instructores que desean formar ingenieros de software a nivel de pregrado.

## 2. Antecedentes

La Escuela de Ciencias de la Computación e Informática (ECCI) de la Universidad de Costa Rica (UCR) llevó a cabo un proceso de revisión y actualización completo del pensum del Programa de Bachillerato en Computación e Informática. Este proceso culminó en 1999 con el programa actual que se implementó a partir del 2000.

El último estudio de carácter integral dirigido a modificar el plan de estudios de la carrera de Bachillerato en Computación e Informática se realizó en 1987, culminando con el plan de estudios que estuvo vigente hasta finales de 1999. Sin embargo, la computación e informática es una disciplina que está constantemente sujeta a vertiginosas modificaciones, debido a cambios constantes en las tecnologías que la sustentan. Adicionalmente, el mercado laboral costarricense para profesionales en esta disciplina ha evolucionado grandemente en los últimos años. La profesión se ha expandido y a la vez se ha diversificado, lo que ha provocado que la práctica de la profesión haya crecido en

complejidad y diversidad. Desde 1988, se habían hecho algunas modificaciones menores a este plan de estudios, pero no es sino hasta 1999 que se propone una reforma integral.

Aspectos tales como la evolución de la disciplina, la creciente demanda de profesionales calificados, y la especialización de la disciplina, nos obligaron a plantear esta propuesta de reforma el plan de estudios del programa de bachillerato con el objetivo de responder eficazmente a esta nueva realidad.

La ECCI consiste actualmente de 25 profesores de tiempo completo y otros 20 de tiempo parcial. Cuenta con 500 estudiantes de pregrado y 100 de posgrado. El programa de Bachillerato es de 4 años de estudio y el de Licenciatura es 1 año adicional sobre el Bachillerato. La UCR trabaja en años académicos de 2 semestres de 16 semanas lectivas cada uno, con cursos de 4 horas lectivas semanales (4 créditos) y 2 horas lectivas semanales (2 créditos).

En 1999, el área de ingeniería de software del programa de bachillerato de la ECCI adolecía de varios problemas:

1. Los cursos estaban desactualizados tecnológicamente y metodológicamente.
2. No contábamos con las herramientas de software adecuadas para enseñar las últimas metodologías.
3. Existía una diferencia entre lo enseñado en los cursos y las prácticas de ingeniería de software utilizadas la industria que contrataba a nuestros graduados.
4. Los cursos de ingeniería de software requerían un mayor componente práctico en el uso de metodologías y herramientas modernas disponibles en el mercado.

### 3. Rediseño curricular

Para resolver los problemas descritos arriba, se diseñó el nuevo pensum que se describe en la Tabla 2. Los cursos de ingeniería de software se muestran en color gris.

La Tabla 1 contiene una comparación cuantitativa de las diferentes áreas curriculares entre al plan vigente hasta 1999 y la nueva propuesta que entró a regir en 2000.

Área curricular	Plan vigente en 1999		Plan 2000	
	# cursos	# créditos	# cursos	# créditos
Cursos de humanidades	7	21	7	21
Cursos de inglés	0	0	1	4
Cursos de matemática (sigla MA- )	5	20	3	12
Cursos de matemática discreta e investigación de operaciones (sigla CI- )	5	20	4	16
Cursos de computación e informática (excluyendo laboratorios y cursos electivos)	16	63	15	60
Cursos de laboratorio	0	0	6	6
Cursos electivos (de computación e informática)	4	16	5	20
<b>Total</b>	<b>37</b>	<b>140</b>	<b>41</b>	<b>139</b>

**Tabla 1.** Resumen cuantitativo de las modificaciones a las áreas curriculares del plan de estudios.

Las principales modificaciones que contiene el nuevo plan con respecto al plan de estudios anterior son:

1. Se incorporó un curso de inglés básico (LM-1030 Inglés Intensivo I), reconociendo así la necesidad que tienen los estudiantes de computación e informática de tener un dominio básico de este idioma.
2. Se crearon seis nuevos cursos de laboratorio (de un crédito cada uno) asociados como co-requisito de varios cursos de computación e informática, incorporando así un componente práctico adicional a estos cursos. Esto representa una innovación pedagógica en nuestra carrera.
3. Se incluyó un curso electivo adicional, lo que aumenta la flexibilidad curricular.

- Se redujo el número de cursos del área de matemática (sigla MA- ) de cinco a cuatro, eliminando a la vez algunas duplicidades de contenidos entre éstos y los cursos de matemáticas discretas (sigla CI- ).
- Se redujo el número de cursos del área de matemática discreta e investigación de operaciones de cinco a cuatro cursos, eliminando algunos contenidos y comprimiendo otros en diferentes cursos.
- Aunque se sustituyeron algunos cursos por otros, el número de cursos y créditos de computación e informática permanecen sin mayor variación.

### 3.1 Descripción Detallada del Nuevo Plan de Estudios

La Tabla 2 resume el nuevo plan de estudios de bachillerato. Cada caja representa un curso con su sigla, nombre, y número de créditos en la parte superior derecha. Las siglas inmediatamente arriba de cada caja representan los cursos requisitos de ese curso. Las cajas divididas en dos partes representan los cursos teóricos que tienen asociado un curso de laboratorio como co-requisito.

I A	CI-1104 <span style="border: 1px solid black; padding: 2px;">4</span> Estructuras Discretas Aplicadas I	CI-1010 <span style="border: 1px solid black; padding: 2px;">4</span> Introducción a Computación e Informática	LM-1030 <span style="border: 1px solid black; padding: 2px;">4</span> Inglés Intensivo I	EG <span style="border: 1px solid black; padding: 2px;">6</span> Humanidades I	EF- <span style="border: 1px solid black; padding: 2px;">0</span> Actividad Deportiva	1er Sem 18	
	CI-1104 MA-0229 <span style="border: 1px solid black; padding: 2px;">4</span> Matemática para Computación II	CI-1010 CI-1101 <span style="border: 1px solid black; padding: 2px;">4</span> Programación I	CI-1104 o MA-0150 CI-1204 <span style="border: 1px solid black; padding: 2px;">4</span> Estructuras Discretas Aplicadas II	EG- <span style="border: 1px solid black; padding: 2px;">6</span> Humanidades II			2do Sem 18
II A	MA-0329 <span style="border: 1px solid black; padding: 2px;">4</span> Matemática para Computación III	CI-1101 CI-1201 <span style="border: 1px solid black; padding: 2px;">4</span> Programación II	CI-1210 <span style="border: 1px solid black; padding: 2px;">4</span> Diseño de Circuitos Digitales	CI-1211 <span style="border: 1px solid black; padding: 2px;">1</span> Laboratorio Diseño de Circ. Dig.	SR - <span style="border: 1px solid black; padding: 2px;">2</span> Seminario Realidad Nacional I	EG- <span style="border: 1px solid black; padding: 2px;">2</span> Actividad Artística	
	MA-0329 MA-0429 <span style="border: 1px solid black; padding: 2px;">4</span> Matemática para Computación IV	CI-1101 CI-1201	CI-1101/CI-1204 CI-1201/CI-1204	MA-0329/CI-1204	S.R. N. I	4to Sem 18	
III A	MA-0429/ CI-1221/ CI-1352 CI-1453 <span style="border: 1px solid black; padding: 2px;">4</span> Investigación de Operaciones III	CI-1220 <span style="border: 1px solid black; padding: 2px;">4</span> Ensambladores y Microprocesadores	CI-1221 <span style="border: 1px solid black; padding: 2px;">4</span> Estructuras de Datos y Análisis de Algoritmos	CI-1352 <span style="border: 1px solid black; padding: 2px;">4</span> Investigación de Operaciones I	SR - <span style="border: 1px solid black; padding: 2px;">2</span> Seminario Realidad Nacional II		5to Sem 17
	CI-1221/CI-1220 CI-1221/CI-1453 CI-1310 <span style="border: 1px solid black; padding: 2px;">4</span> Sistemas Operativos I	CI-1311 <span style="border: 1px solid black; padding: 2px;">1</span> Laboratorio Sistemas Operativos I	CI-1221 CI-1312 <span style="border: 1px solid black; padding: 2px;">4</span> Bases de Datos I	CI-1313 <span style="border: 1px solid black; padding: 2px;">1</span> Laboratorio Bases de Datos I	EG <span style="border: 1px solid black; padding: 2px;">3</span> Repertorio	6to Sem 18	
IV	CI-1322 <span style="border: 1px solid black; padding: 2px;">4</span> Autómatas y compiladores	CI-1320 <span style="border: 1px solid black; padding: 2px;">4</span> Redes de Computadoras I	CI-1321 <span style="border: 1px solid black; padding: 2px;">1</span> Lab. Redes de Computadoras	CI-1323 <span style="border: 1px solid black; padding: 2px;">4</span> Arquitectura de Computadoras	CI-1330 <span style="border: 1px solid black; padding: 2px;">4</span> Ingeniería de Software I		CI-1331 <span style="border: 1px solid black; padding: 2px;">1</span> Laboratorio Ingeniería de Software I
	CI-1322 CI-1441 <span style="border: 1px solid black; padding: 2px;">4</span> Paradigmas	R.E. CI- <span style="border: 1px solid black; padding: 2px;">4</span> Electiva	R.E. CI- <span style="border: 1px solid black; padding: 2px;">4</span> Electiva	CI-1330/CI-1331 CI-1430 <span style="border: 1px solid black; padding: 2px;">4</span> Ingeniería	CI-1431 <span style="border: 1px solid black; padding: 2px;">1</span> Laboratorio		

A	Computacionales			de Software II	Ingeniería de Software II	17
Ñ	S.R..N. II/CI-1330	R.E.	R.E.	R.E.		
O	CI-1440 Informática y Sociedad	CI- Electiva	CI- Electiva	CI- Electiva		8vo Sem 16
						<b>TOTAL 139</b>

**Tabla 2.** Nuevo Plan de Bachillerato en Computación e Informática..

En sombreado y negrita se destacan los cursos del área de ingeniería de software incluidos en el pensum. Los cursos *CI-1330 Ingeniería de Software I* y *CI-1430 Ingeniería de Software II* (junto con sus respectivos laboratorios) constituyen los únicos dos cursos obligatorios de esta área en todo el pensum de bachillerato.

Los cursos electivos deben ser escogidos por el estudiante de la lista de cursos del Programa de Licenciatura en Computación e Informática. La Tabla 3 contiene la lista de cursos electivos relacionados con ingeniería de software (todos ellos pertenecen al Plan de Licenciatura en Computación e Informática).

<b>ÁREA DE SISTEMAS DE INFORMACIÓN</b>	
CI-2303	ADMINISTRACIÓN Y FORMUL. DE PROY.
CI-2306	AUDITORÍA DE SISTEMAS COMPUTAC.
CI-2307	ECONOMÍA DE LA COMPUTACIÓN
CI-2309	GERENCIA COMPUTACIONAL
CI-2310	ESTRATEGIA Y TECNOL.DE LA INFORM.
CI-2311	SIST.DE INFORMAC.Y TOMA DE DECISIONES
CI-2312	TÉCNICAS DE APLICAC.ADM.Y COMERCIAL
<b>ÁREA DE INGENIERÍA DE SISTEMAS</b>	
CI-2400	TEMAS ESPECIALES DE ING.DE SISTEMAS
CI-2411	INTRODUCCIÓN A LA MULTIMEDIA
CI-2412	SISTEMAS DE INFORM.GEOGRÁFICOS
CI-2413	DESARROLLO DE APLICAC.PARA INTERNET
CI-2414	RECUPERACIÓN DE INFORMACIÓN

**Tabla 3.** Lista de cursos electivos.

## 4. Área de Ingeniería de Software

El área de ingeniería de software es una de las más importantes dentro de nuestro programa de Bachillerato pues debido a que estimamos que más del 75% de nuestros graduados se emplean en la industria del desarrollo de sistemas.

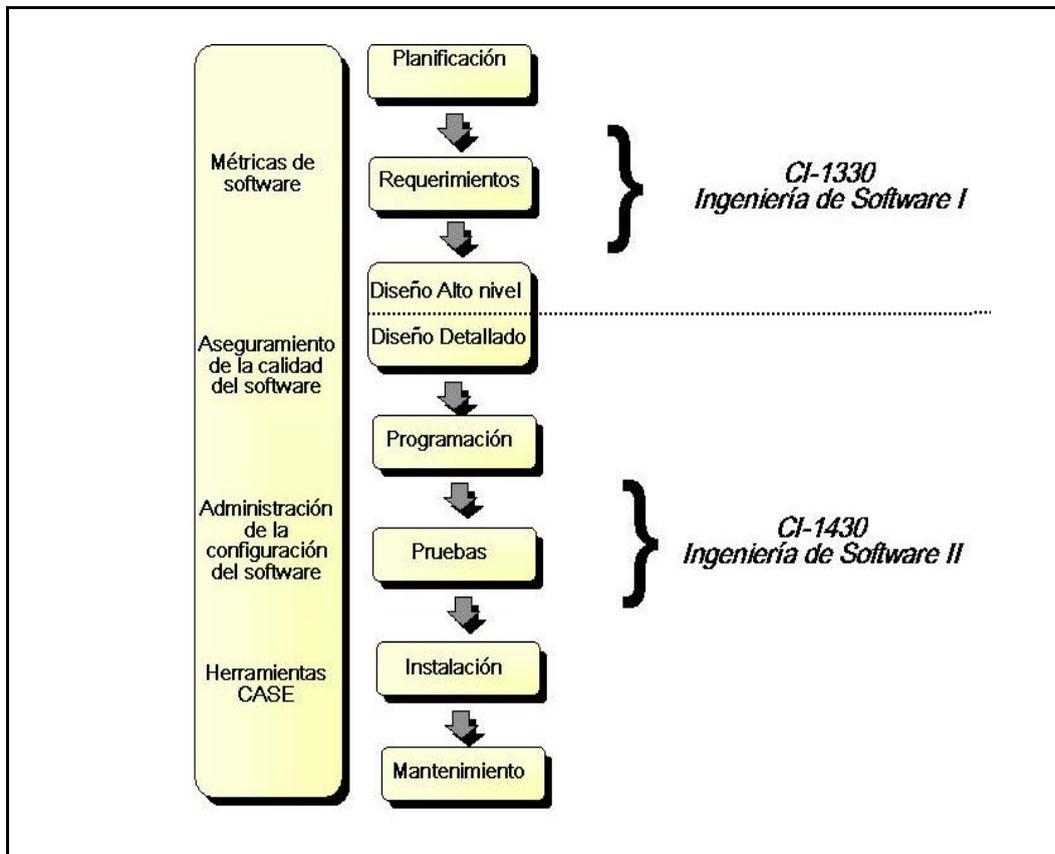
En el diseño de estos dos cursos, planteamos los siguientes objetivos:

1. Cubrir metodologías y herramientas modernas de ingeniería de software disponibles en el mercado laboral.
2. Crear cursos muy prácticos donde el estudiante no sólo aprenda las metodologías sino también el uso eficaz de diversas herramientas CASE.
3. Cubrir todo lo básico de la disciplina de ingeniería de software en únicamente 2 cursos (y sus laboratorios).
4. Complementar cualquier falta de contenidos con los cursos electivos (por escoger por cada estudiante).

La Figura 1 muestra cómo se cubren las diferentes fases del ciclo tradicional de desarrollo de sistemas en cada uno de los dos cursos de ingeniería de software. El curso *CI-1330 Ingeniería de Software I* cubre los temas de planificación y estimación de proyectos, análisis de requerimientos y diseño de alto nivel. El curso *CI-1430 Ingeniería de Software II* abarca los temas de diseño detallado, programación, pruebas, instalación y mantenimiento de sistemas.

Los procesos de soporte tales como aseguramiento de la calidad del software (SQA), métricas de software, administración de la configuración del software, y utilización de herramientas CASE se cubren en ambos cursos.

Es importante destacar que los estudiantes realizan un sólo proyecto práctico de mediano tamaño a través de ambos cursos, de tal manera que planifican el proyecto, especifican los requerimientos, y elaboran el diseño de alto nivel en el primer curso. En el semestre siguiente, en el segundo curso retoman el mismo proyecto y a partir del diseño conceptual derivan el diseño detallado, generan los programas, realizan las pruebas e instalan el sistema final. De esta manera, se cubre el ciclo de vida del desarrollo de sistemas en dos semestres utilizando un mismo proyecto.



**Figura 1.** Como los temas de ingeniería de software son cubiertos en ambos cursos.

Este proyecto lo llevan a cabo en grupos de 3 o 4 estudiantes. Uno de los elementos que pretendemos enseñar en estos cursos es el trabajo en equipo. Para esto, cada equipo de 3 o 4 estudiantes debe definir un Líder del Proyecto, quien hace de administrador del proyecto. Los demás miembros del equipo, deben tener roles definidos (e.g., analista, programador, documentador, etc.). Los estudiantes aprenden técnicas para coordinar un equipo técnico de 3 o 4 personas que está trabajando concurrentemente en la solución de un problema durante un año lectivo completo. Por ejemplo, aprenden cómo distribuir tareas y asignar responsables en el cronograma del proyecto, y cómo coordinar la actualización y consistencia de los productos técnicos de software que se producen a través del proceso de desarrollo (e.g., el diseño del sistema).

Los cursos de laboratorio, que son co-requisito, se desarrollan en un laboratorio de cómputo en el que todos los estudiantes tengan acceso a software necesario de acuerdo al tema del laboratorio. El profesor entrega al inicio de cada sesión una descripción del trabajo a realizar y da seguimiento y guía continua a los estudiantes en el desarrollo de las tareas asignadas para ese día. Cada uno de estas asignaciones tiene como objetivo específico la comprensión de las herramientas utilizadas. La asistencia a los laboratorios es obligatoria y se lleva a cabo un laboratorio de 2 horas cada semana. Al estudiante se le evalúa por la realización de los trabajos prácticos y por la asistencia a las sesiones de laboratorio.

Adicionalmente, los estudiantes pueden tomar el curso *CI-2300 Proyecto* como curso electivo de 4to año para complementar sus conocimientos en el área de ingeniería de software. Este es un curso práctico en donde se desarrolla en proyecto de software real en grupos de 3 o 4 estudiantes, es decir, a cada grupo se le asigna un proyecto con un usuario final (cliente) cuyo producto final deberá ser puesto en producción en las oficinas del cliente.

En nuestra experiencia, preferimos que este tipo de curso-proyecto sea tomado por los estudiantes una vez que hayan aprobado el curso *CI-1430 Ingeniería de Software II* pues de esta manera ya cuentan con las herramientas mínimas para enfrentarse a un proyecto real.

A continuación se describen en detalle los programas de ambos cursos y sus respectivos laboratorios.

## 4.1 Ingeniería de software I

<b>Sigla:</b> CI-1330								
<b>Nombre:</b> Ingeniería de Software I								
<b>Créditos:</b> 4								
<b>Horas:</b> 4 (4 teoría)								
<b>Nivel:</b> 6to ciclo								
<b>Periodo:</b> Semestral								
<b>Requisitos:</b> CI-1312 Bases de Datos I, CI-1313 Laboratorio Bases de Datos I								
<b>Co-requisitos:</b> CI-1331 Laboratorio Ingeniería de Software I								
<b>Objetivo General:</b> Introducir métodos, técnicas, herramientas y estándares modernos de ingeniería de software para resolver problemas de análisis de requerimientos y diseño de sistemas de información.								
<b>Objetivos específicos:</b> Al finalizar el curso el estudiante será capaz de: <ol style="list-style-type: none"><li>4 Utilizar los métodos, técnicas y herramientas más apropiadas, de acuerdo al dominio del problema y al contexto de su solución, para el análisis de requerimientos y diseño de sistemas de información.</li><li>5 Elaborar especificaciones de requerimientos de sistemas de información de alta calidad, siguiendo métodos y técnicas avanzadas, y aplicando algún estándar apropiado y válido internacionalmente, siguiendo los métodos, técnicas y estándares apropiados.</li><li>6 Aplicar las técnicas más avanzadas de verificación de especificaciones de requerimientos y diseño de sistemas de información.</li><li>7 Utilizar eficazmente alguna herramienta CASE moderna para el análisis y diseño de sistemas de información.</li></ol>								
<b>Contenidos:</b> <ol style="list-style-type: none"><li>1. Introducción a la ingeniería de software: definiciones, modelos del ciclo de vida de sistemas.</li><li>2. Planificación de proyectos de desarrollo de software.</li><li>3. Estimación de proyectos mediante puntos de función.</li><li>4. Aseguramiento de la calidad del software.</li><li>5. Métricas de software.</li><li>6. Análisis de requerimientos orientado a objetos utilizando UML..</li><li>7. Estándares y procedimientos para la elaboración de especificaciones de requerimientos del software.</li><li>8. Técnicas de verificación de especificaciones de requerimientos de software.</li><li>9. Diseño de software orientado a objetos utilizando UML.</li><li>10. Técnicas de verificación de especificaciones de diseño de software.</li><li>11. Herramientas CASE y Uso de Rational Rose 2000.</li></ol>								
<b>Metodología:</b> La mayor parte del curso se llevará a cabo mediante el sistema de lecciones magistrales. Como material de lectura se utilizará un compendio de capítulos de diferentes libros y algunos artículos de revistas. El profesor complementará el material del compendio con ejemplos de otras fuentes. En el curso co-requisito CI-1331 Lab. Ing. Soft. I se desarrollarán prácticas cortas en el laboratorio para poner en práctica los conceptos vistos en clase.								
<b>Evaluación:</b> <table><tr><td>I Examen Parcial</td><td>25%</td></tr><tr><td>II Examen Parcial</td><td>25%</td></tr><tr><td>Exposición</td><td>20%</td></tr><tr><td>Proyecto práctico</td><td>30%</td></tr></table>	I Examen Parcial	25%	II Examen Parcial	25%	Exposición	20%	Proyecto práctico	30%
I Examen Parcial	25%							
II Examen Parcial	25%							
Exposición	20%							
Proyecto práctico	30%							



## 4.2 Laboratorio Ingeniería de software I

<b>Sigla:</b> CI-1331
<b>Nombre:</b> Laboratorio Ingeniería de Software I
<b>Créditos:</b> 1
<b>Horas:</b> 2 (2 Laboratorio)
<b>Nivel:</b> 6to ciclo
<b>Periodo:</b> Semestral
<b>Requisitos:</b> CI-1312 Bases de Datos I, CI-1313 Laboratorio Bases de Datos I
<b>Co-requisitos:</b> CI-1330 Ingeniería de Software I
<b>Objetivo General:</b> Utilizar y evaluar herramientas automáticas herramientas CASE y lenguajes de cuarta generación para implementar los conceptos cubiertos en el curso CI-1330 Ingeniería de software I.
<b>Contenidos:</b> <ol style="list-style-type: none"><li>1. Realización de prácticas dirigidas en el uso y evaluación de una herramientas CASE orientada a objetos para tareas de análisis de requerimientos y diseño de sistemas (Rational ROSE 2000).</li><li>2. Demostración de herramientas alternativas para el desarrollo de software.</li></ol>
<b>Metodología:</b> <p>El curso se desarrollará en un laboratorio de cómputo en el que todos los estudiantes tengan acceso a software necesario de acuerdo al tema del laboratorio.</p> <p>El profesor entregará al inicio de cada sesión una descripción del trabajo a realizar y dará seguimiento y guía continua a los estudiantes en el desarrollo de sus proyectos. Cada uno de estos proyectos tiene un objetivo específico en la comprensión de las herramientas utilizadas.</p> <p>La asistencia a los laboratorios es obligatoria. Las fechas específicas de cada sesión serán avisadas con anticipación en clase.</p> <p>Al estudiante se le evaluará por la realización de los trabajos prácticos y por la asistencia a las sesiones de laboratorio</p>
<b>Evaluación:</b> N Prácticas de laboratorio 50% Asistencia 50%

### 4.3 Ingeniería de software II

<b>Sigla:</b> CI-1430								
<b>Nombre:</b> Ingeniería de Software II								
<b>Créditos:</b> 4								
<b>Horas:</b> 4 (4 teoría)								
<b>Nivel:</b> 7to ciclo								
<b>Periodo:</b> Semestral								
<b>Requisitos:</b> CI-1330 Ingeniería de Software I, CI-1331 Laboratorio Ingeniería de Software I								
<b>Co-requisitos:</b> CI-1431 Laboratorio Ingeniería de Software I								
<b>Objetivo General:</b> Introducir métodos, técnicas, herramientas y estándares modernos de ingeniería de software para resolver problemas de diseño detallado, programación, y prueba de sistemas de información.								
<b>Objetivos Específicos</b> Al finalizar el curso el estudiante será capaz de: <ol style="list-style-type: none"><li>1. Comprender especificaciones de diseño de software elaboradas con los métodos, técnicas y herramientas más avanzadas disponibles.</li><li>2. Comprender, seleccionar y combinar los métodos, técnicas y herramientas de diseño de software más apropiadas, de acuerdo al dominio del problema y al contexto de su solución.</li><li>3. Generar varias opciones de diseño ante un problema específico, valorarlas y escoger razonadamente una.</li><li>4. Aplicar las técnicas más avanzadas de verificación de diseños de software.</li><li>5. Producir diseños de software de alta calidad, aplicando un estándar actualizado, apropiado y válido internacionalmente.</li><li>6. Producir diseños de interfaz humano-sistema de alta calidad, que sean apropiados a las características específicas del software y a las restricciones definidas por el usuario.</li><li>7. Sistematizar su experiencia como diseñador de software mediante la aplicación de estándares de documentación.</li><li>8. Aplicar las técnicas más avanzadas de prueba de software, así como algún estándar actualizado y válido internacionalmente, en pruebas modulares, de integración de sistemas y de aceptación de software.</li></ol>								
<b>Contenidos:</b> <ol style="list-style-type: none"><li>1. Marco de referencia del diseño orientado a objetos: abstracción; modularización (subsistemas, frameworks, clases), ocultamiento de información; cohesión y acoplamiento, algún estándar de especificación de diseños de software, actualizado y válido internacionalmente.</li><li>2. Técnicas de diseño detallado orientado a objetos.</li><li>3. Técnicas de medición de la cohesión y del acoplamiento: métricas tradicionales de la complejidad del producto, orientadas a módulos procedimentales, métricas orientadas a objetos de la complejidad del producto.</li><li>4. Heurísticas para un buen diseño.</li><li>5. Métricas técnicas de software.</li><li>6. Programación de sistemas y lenguajes de cuarta generación.</li><li>7. Técnicas de prueba de software: algún estándar de prueba actualizado y válido internacionalmente, pruebas de modulares, pruebas de integración, pruebas de aceptación, diseño de la interfaz humano-sistema.</li><li>8. Administración de la configuración del software.</li><li>9. Mantenimiento de software.</li></ol>								
<b>Evaluación:</b> <table><tr><td>I Examen Parcial</td><td>25%</td></tr><tr><td>II Examen Parcial</td><td>25%</td></tr><tr><td>Exposición</td><td>20%</td></tr><tr><td>Proyecto práctico</td><td>30%</td></tr></table>	I Examen Parcial	25%	II Examen Parcial	25%	Exposición	20%	Proyecto práctico	30%
I Examen Parcial	25%							
II Examen Parcial	25%							
Exposición	20%							
Proyecto práctico	30%							



#### 4.4 Laboratorio Ingeniería de software II

<b>Sigla:</b> CI-1431				
<b>Nombre:</b> Laboratorio Ingeniería de Software II				
<b>Créditos:</b> 1				
<b>Horas:</b> 2 (2 Laboratorio)				
<b>Nivel:</b> 7to ciclo				
<b>Periodo:</b> Semestral				
<b>Requisitos:</b> CI-1330 Ingeniería de Software I, CI-1331 Laboratorio Ingeniería de Software I				
<b>Co-requisitos:</b> CI-1430 Ingeniería de Software II				
<b>Objetivo General:</b> Utilizar y evaluar herramientas automáticas (e.g., herramientas CASE, lenguajes de cuarta generación) para implementar los conceptos cubiertos en el curso CI-1430 Ingeniería de software II.				
<b>Contenidos:</b> <ol style="list-style-type: none"><li>1. Realización de prácticas dirigidas en el uso y evaluación de una o más herramientas CASE para tareas de diseño detallado, programación, prueba y administración de la configuración (e.g., Rational ROSE, System Architect).</li><li>2. Demostración de herramientas alternativas para el desarrollo de software.</li></ol>				
<b>Metodología:</b> <p>El curso debe desarrollarse en un laboratorio en el que todos los estudiantes tengan acceso a equipo necesario de acuerdo al tema del laboratorio.</p> <p>El profesor dará seguimiento y guía continua a los estudiantes en el desarrollo de sus proyectos. Cada uno de estos proyectos tiene un objetivo específico en la comprensión de las herramientas utilizadas.</p> <p>La asistencia es obligatoria.</p> <p>Al estudiante se le evaluará por la ejecución de los laboratorios prácticos, y por la presentación de reportes de laboratorio.</p>				
<b>Evaluación:</b> <table><tr><td>N Prácticas de laboratorio</td><td>50%</td></tr><tr><td>Asistencia</td><td>50%</td></tr></table>	N Prácticas de laboratorio	50%	Asistencia	50%
N Prácticas de laboratorio	50%			
Asistencia	50%			

## Conclusiones

El cambio curricular descrito en este artículo se dio como respuesta a las exigencias de la industria. Hemos aprendido que es necesario enseñar lo que la industria necesita. Esto implica tener profesores capacitados y laboratorios actualizados con las herramientas adecuadas.

Los cursos *CI-1330 Ingeniería de Software I* y *CI-1430 Ingeniería de Software II* se han impartido todos los semestres desde principios de 2000 (6 semestres en total) con un promedio de 30 estudiantes por semestre. Es decir, aproximadamente unos 180 estudiantes han tomado ambos cursos a la fecha.

Hemos notado una mejoría cualitativa en la calidad académica de nuestros graduados de bachillerato en el área de ingeniería de software. Debido a los cambios implementados en 2000, creemos que actualmente están mejor preparados para afrontar las exigencias del mercado laboral en el área de ingeniería de software.

La experiencia nos ha demostrado que cursos de laboratorio prácticos son necesarios para lograr una mejor asimilación de los conceptos cubiertos en los cursos teóricos. Creemos que los cursos de laboratorio complementan muy bien el curso teórico con prácticas dirigidas orientadas hacia la enseñanza del buen uso de las herramientas CASE que se utilizan para hacer software. En nuestro caso, utilizamos herramientas tales como Microsoft Project 2000 para la planificación y control del proyecto, Rational Rose 2000 para la especificación de los requerimientos y el diseño del sistema y generación de código, Microsoft Source Safe para administración de la configuración, y Jbuilder para la programación. En el futuro cercano queremos incorporar herramientas adicionales tales como Rational Requisite-Pro para administración de requerimientos y Rational Team Test para pruebas de sistemas.

## Referencias

- [1] Catapult Inc. *Microsoft Project 98*. McGraw-Hill, 1998.
- [2] Garmus D., Herron D. *Measuring the Software Process*. Yourdon Press, 1996.
- [3] A. Abran, J.W. Moore (edts.). *Guide to the Software Engineering Body of Knowledge SEBOK*. IEEE Inc., 2001.
- [4] Pressman, R. *Ingeniería de Software: Un enfoque práctico*. 5ta edición, McGraw-Hill, 2000.
- [5] Quatrani, T. *Visual Modeling with Rational Rose 2000 and UML*. Addison-Wesley, 2000.
- [6] *The International Journal of Education and Computing*. Elsevier Publishing, Vol. 7, No. 1 & 2, 1991.
- [7] The Samuel Neaman Institute. *Engineering Education 2001*. Technical Report, 1987.
- [8] Turner, J. Computing Curricula 1991: A summary of the ACM/IEEE Joint Curriculum Task Force Report. *Communications of the ACM*, 34(6), June 1991, pags. 69-84.
- [9] Vicerrectoría de Docencia UCR. *Políticas y Normas para la Actualización de Planes de Estudio*. 1995.
- [10] Vicerrectoría de Docencia UCR. *Guía para la Elaboración y Presentación de Modificaciones a Planes de Estudio*. 1991.