

**Propuesta para el Establecimiento de un
Programa de Bachillerato en
Ciencia de Computación e Ingeniería de Software
en el Colegio de Ingeniería del
Recinto Universitario de Mayagüez de la
Universidad de Puerto Rico**

Revisado: 4/11/2002

**Colegio de Ingeniería
Universidad de Puerto Rico
Recinto Universitario de Mayagüez**

<http://ece.uprm.edu/~bvelez/projects/Computing>

Reconocimientos

Esta propuesta fue formulada por el Comité ICOM-Software del Departamento de Ingeniería Eléctrica y Computadoras del Recinto Universitario de Mayaguez.

Tabla de Contenido

1. INTRODUCCIÓN	1
1.1 TÍTULO DEL PROGRAMA Y GRADOS	1
1.2 DURACIÓN DEL PROGRAMA EN AÑOS PARA ESTUDIANTES A TIEMPO COMPLETO	1
1.3 BREVE EXPOSICIÓN DEL PROGRAMA	1
1.4 FECHA DE COMIENZO.....	2
2. JUSTIFICACIÓN	2
2.1 EL DESARROLLO DE LA INGENIERÍA DE SOFTWARE COMO UNA DISCIPLINA DE INGENIERÍA	4
2.2 RAZONES DE TIPO ACADÉMICO PARA EL ESTABLECIMIENTO DEL PROGRAMA	5
2.2 RAZONES DE TIPO ACADÉMICO PARA EL ESTABLECIMIENTO DEL PROGRAMA	6
2.3 NECESIDAD DEL NUEVO PROGRAMA	6
2.4 OPORTUNIDADES DE EMPLEO PARA LOS EGRESADOS DEL PROGRAMA	7
3. RELACIÓN DEL PROGRAMA CON EL PLAN ESTRATÉGICO SISTÉMICO DE LA UNIVERSIDAD DE PUERTO RICO	8
4. RELACIÓN DEL NUEVO PROGRAMA CON OTROS	9
4.1 EN EL RECINTO UNIVERSITARIO DE MAYAGÜEZ	9
4.2 EN OTRAS UNIDADES DEL SISTEMA UNIVERSITARIO DE LA U.P.R.....	9
4.3 EN OTRAS INSTITUCIONES DEL PAÍS	10
5. DESCRIPCIÓN DEL PROGRAMA	10
5.1 FILOSOFÍA, METAS Y OBJETIVOS ESPECÍFICOS DEL PROGRAMA.....	10
5.2 PERFIL DEL EGRESADO	11
5.3 COMPONENTES DEL PROGRAMA	13
5.3.1 <i>Distribución de los Cursos por Areas de Educación</i>	13
5.3.2 <i>Descripción de los Cursos</i>	15
5.3.2.1 Cursos Requeridos.....	15
5.3.2.2 Electivas Técnicas	17
5.3.3 <i>Modelo del Programa Propuesto</i>	21
5.3.4 <i>Metodología Educativa y Estrategias Instruccionales</i>	22
5.3.5 <i>Catálogo y Promoción</i>	22
6. ADMISIÓN Y MATRÍCULA	23
6.1 REQUISITOS DE ADMISIÓN.....	23
6.2 PROYECCIÓN DE LA MATRÍCULA.....	23
7. REQUISITOS ACADÉMICOS PARA OTORGAR EL GRADO	23
7.1 TOTAL DE HORAS-CRÉDITO QUE SE REQUIEREN	23
7.2 INDICES ACADÉMICOS MÍNIMOS	23
7.3 TOTAL DE CRÉDITOS A ACEPTARSE EN TRANSFERENCIA	23
7.4 REQUISITOS DE IDIOMA	24
7.5 TIEMPO LÍMITE PARA COMPLETAR EL GRADO.....	24

8. FACULTAD.....	24
8.1 FACULTAD NECESARIA PARA EL ESTABLECIMIENTO DEL PROGRAMA.....	24
8.1.1 <i>Facultad Necesaria para Ofrecer los Cursos de Ciencias de Computación e Ingeniería de Software del Programa</i>	24
8.1.2 <i>Facultad Necesaria para Ofrecer los Cursos de Otras Disciplinas</i>	25
8.2 PLAN DE RECLUTAMIENTO DE LA FACULTAD.....	25
8.2.1 <i>Reclutamiento de Facultad para Ofrecer los Cursos de Ciencias de Computación e Ingeniería de Software</i>	25
8.2.1.1 Reclutamiento de Facultad en el RUM.....	25
8.2.1.3 Reclutamiento de Facultad Externa al RUM.....	25
8.2.1.4 Reclutamiento de Ayudantes de Cátedra.....	26
8.2.2 <i>Facultad para Ofrecer los Cursos de Otras Disciplinas</i>	26
8.3 PLAN PARA EL ADIESTRAMIENTO DE LA FACULTAD.....	26
9. RECURSOS DEL APRENDIZAJE.....	26
9.1 INVENTARIO DE RECURSOS EXISTENTES.....	26
9.2 PLAN DE MEJORAMIENTO DE RECURSOS DISPONIBLES.....	27
9.2.1 <i>Obtención de Libros</i>	27
9.2.1 <i>Obtención de Materiales Audiovisuales</i>	27
9.2.3 <i>Obtención de Acceso Electrónico a Base de Datos</i>	27
9.2.4 <i>Uso de Otras Bibliotecas</i>	27
10. INSTALACIONES FÍSICAS Y EQUIPO.....	28
10.1 INVENTARIO DE FACILIDADES DISPONIBLES.....	28
10.2 IMPACTO DEL PROGRAMA SOBRE LAS INSTALACIONES FÍSICAS EXISTENTES.....	28
10.3 NECESIDAD Y DISPONIBILIDAD DE SERVICIOS DE CÓMPUTOS PARA EL NUEVO PROGRAMA.....	28
11. ACREDITACIÓN Y LICENCIA DEL PROGRAMA.....	30
11.1 ACREDITACIÓN PROFESIONAL.....	30
11.2 LICENCIAMIENTO POR EL CES.....	30
12. ADMINISTRACIÓN DEL NUEVO PROGRAMA.....	30
13. AYUDA ECONÓMICA PARA LOS ESTUDIANTES.....	30
14. PRESUPUESTO.....	31
15. INGRESOS.....	33
16. EVALUACION.....	34
APÉNDICE A: DEFINICIÓN DE CIENCIA DE COMPUTACIÓN SEGÚN CAC/ABET.....	35
APÉNDICE B: PRONTUARIOS DE LOS CURSOS NUEVOS.....	36

1. INTRODUCCIÓN

1.1 Título del Programa y Grados

El título del programa propuesto es

Programa de Bachillerato en Ciencia de Computación e Ingeniería de Software .

Los grados a otorgarse son:

- **Bachillerato en Ciencias en Ingeniería de Software (B.S. Software Engineering)**
- **Bachillerato en Ciencias en Ciencia de Computación. (B.S. Computer Science)**

1.2 Duración del Programa en Años para Estudiantes a Tiempo Completo

El programa propuesto tiene una duración normal de cinco años. El tiempo máximo para completar el grado es de diez años.

1.3 Breve Exposición del Programa

El programa que se propone en este documento está destinado a formar profesionales en ciencia de computación y en ingeniería de software. El programa está estructurado en torno a un tronco común de cursos de formación socio-humanística, fundamentos de ingeniería, ciencias y matemáticas y fundamentos de computación. Los grados profesionales de bachiller en ciencias en ciencia de computación y bachiller en ciencia en ingeniería de software resultan de un conjunto de 27 créditos académicos adicionales, especializados en áreas de ciencia de computación e ingeniería de software, respectivamente.

El programa propuesto adopta el modelo curricular del Colegio de Ingeniería del Recinto de Mayagüez de la Universidad de Puerto Rico, el cual ofrece una base de fundamentos de ingeniería común a todos los estudiantes del área junto a una amplia gama de posibilidades para la profundización y especialización profesional. Un claro reflejo de esta adopción es el hecho de que el programa propuesto ofrece dos títulos profesionales: el de bachillerato en ciencias en ciencia de computación y el de bachillerato en ciencias en ingeniería de software. El grado en ciencia de computación es adecuado para estudiantes interesados en una formación más amplia y flexible en computación, y especialización en áreas tales como: computación en redes, inteligencia artificial, sistemas computacionales, bases de datos, computación gráfica, diseño e implantación de lenguajes de programación, arquitectura de computadoras, entre otras. El grado en ingeniería de software es adecuado para estudiantes interesados en el desarrollo de software en todas sus fases: análisis de requisitos, especificación, diseño, implantación y pruebas. El ofrecimiento de un grado en esta área de especialidad responde a avances en el área de ingeniería de software que la definen como disciplina con características y mercado ocupacional propios. De acuerdo al "Institute of Electrical and Electronic Engineers" (IEEE), la Ingeniería de Software es la aplicación de una metodología disciplinada, sistemática y cuantificable en el desarrollo, operación y mantenimiento de programas de computación. El ingeniero de Software

tiene a su cargo tareas técnicas tales como el análisis, especificación, diseño, prueba, verificación y validación de programas de computadoras, al igual que tareas de gerencia tales como control de calidad, medición de procesos y administración de proyectos.

Ambos grados prepararan adecuadamente a los estudiantes interesados en proseguir estudios de maestría o doctorado en ciencia de computación o ingeniería de software.

En resumen, los egresados del programa que se propone estarán capacitados para diseñar, evaluar, mantener, renovar e implantar todo tipo de sistema de procesamiento informático, incluyendo sistemas de computación industriales, comerciales, científicos, tecnológicos y de informática médica. Estarán también preparados para continuar estudios superiores en computación o participar en proyectos de investigación y desarrollo en equipos multidisciplinarios. En particular, los egresados de ingeniería de software estarán especializados en el desarrollo de proyectos de software de gran envergadura, que por su complejidad y uso potencial, envuelvan un alto riesgo y responsabilidad social.

1.4 Fecha de Comienzo

Una vez aprobado por las instancias universitarias y administrativas pertinentes, el programa comenzará a ofrecerse tan pronto estén disponibles los recursos necesarios (humanos, físicos, y presupuestarios que se solicitan en esta propuesta. Se estima que estos recursos estarán disponibles para el primer semestre del año académico 2002-2003.

2. JUSTIFICACIÓN

2.1 Razones de tipo académico para el establecimiento de programa

2.1.1 El carácter actual de la ciencia de computación

La agencia acreditadora de programas académicos en ciencia de computación, “Computer Science Accreditation Board” ha sido recientemente transformada en una comisión: “Computing Accreditation Comisión” (CAC) del organismo acreditador de programas en ingeniería y tecnología, “Accreditation Board for Engineering and Technology” (ABET). Esta comisión está formada por las organizaciones de profesionales en ciencia de computación de mayor influencia y prestigio en Estados Unidos y el mundo, incluyendo la “Association for Computing Machinery” y la “IEEE Computer Society”. El siguiente es un fragmento de la definición de ciencia de computación propuesta por esta agencia (ver apéndice A para la definición completa):

Computer science is a discipline that involves the understanding and design of computers and computational processes ... The discipline ranges from theoretical studies of algorithms to practical problems of implementation in terms of computational hardware and software... Thus, the discipline spans both advancing the fundamental understanding of algorithms and information processes in general as well as the practical design of efficient reliable software and hardware to meet given specifications. (CSAB/ABET Definition of Computer Science as a profession)

Esta definición reafirma el carácter dual de ciencia de computación, en el cual la teoría y los métodos de la ingeniería mantienen una estrecha e indivisible relación. El aspecto teórico busca repuestas a preguntas relacionadas a la naturaleza de la computación, preguntas tales como: ¿Qué es computación? ¿Qué problemas son computables? ¿Qué problemas admiten una solución computacional eficiente?, ¿Cuáles son las métricas adecuadas para determinar la eficiencia de un proceso computacional?. Cada una de estas preguntas encuentra contrapartes en la experimentación y el uso práctico, en el diseño, construcción y validación de sistemas y procesos computacionales tanto en hardware como en software. Recíprocamente, la experimentación tecnológica pone nuevos desafíos teóricos a la disciplina. En el siguiente párrafo del documento de CSAB, la constante interacción entre teoría, experimentación e ingeniería es señalada como una característica distintiva de la ciencia de computación.

Computer science is a young discipline that is evolving rapidly from its beginnings in the 1940's. As such it includes theoretical studies, experimental methods, and engineering design all in one discipline. This differs radically from most physical sciences that separate the understanding and advancement of the science from the applications of the science in fields of engineering design and implementation. In computer science there is an inherent intermingling of the theoretical concepts of computability and algorithmic efficiency with the modern practical advancements in electronics that continue to stimulate advances in the discipline.

W. A. Wulf, un miembro distinguido del panel de la ACM para el estudio de la posible otorgación de una licencia profesional de ingeniería de software, ha expresado:

“...my discipline {computer science} spans a multidimensional spectrum from deep and elegant mathematics to crafty programming, from abstraction to solder joints, from deep truth to elusive human factors, from scholars motivated purely by the desire for knowledge to practitioners making my everyday life better. It embraces the ethos of the scholar as well as that of the professional”

El programa que se propone es este documento está destinado a formar profesionales de la computación dotados principalmente de:

1. Una visión global de la disciplina de la computación.
2. Una clara apreciación de la relación entre teoría y práctica.
3. Una amplia capacidad para evolucionar profesionalmente conforme a las cambiantes exigencias de la disciplina.

Los proponentes analizaron diversos modelos para la implantación de un programa académico capaz de materializar estos objetivos. Como resultado de ese análisis, se alcanzaron los siguientes puntos de consenso:

1. La tecnología de la información tiene un carácter universal, en el sentido de que sirve tanto para propósitos científicos, humanísticos, comerciales y tecnológicos. Por lo tanto, el experto en computación debe tener el beneficio de una formación lo más amplia y abarcadora posible, incluyendo ciencias, ingeniería, economía y humanidades.
2. La generación, validación y control de calidad del software desempeña un rol especial dentro de la tecnología informática, dada la complejidades propias de este tipo de empresa y la extrema dependencia de nuestra sociedad en sistemas basados en extensos y complejos programas de computación (ver informe PITAC). Por esta razón es necesario formar profesionales de la computación especialistas en software. Dichos profesionales tienen un mercado ocupacional definido como ingenieros de software, e incluso, potencialmente, una licencia profesional independiente. Estas consideraciones ameritan el que el programa otorgue un grado específico en ingeniería de software.
3. El currículo de ciencia de computación y el de ingeniería de software comparten un número significativo de cursos. El ingeniero de software necesita conocimientos de ciencia de computación lo suficientemente amplios como para aplicar los mismos crítica y creativamente en su ejercicio profesional. Estos conocimientos son la base del entendimiento, renovación y adecuada aplicación de técnicas y métodos de análisis, para la especificación, diseño, prueba, verificación y validación de programas de computación. Los aspectos éticos, sociales, humanísticos y las destrezas de comunicación y trabajo en equipo son también comunes a ambas profesiones. Por sobre esta base común, el ingeniero de software necesita conocimientos específicos de técnicas, metodologías de gerencia industrial y alguna especialización consistente con su misión profesional. Los proponentes estiman que plantear programas separados de ciencia de computación e ingeniería de software conduciría inevitablemente a duplicación y desperdicio de recursos, e iría en detrimento de la visión global de la disciplina de la computación que se pretende lograr para ambos profesionales.

Los proponentes concuerdan en que el modelo adoptado por la mayoría de las universidades líderes en computación en los Estados Unidos, quienes ubican sus programas de ciencia de computación en la facultad de ingeniería, es el más adecuado para alcanzar los objetivos antes descritos. Es importante observar que de las 18 universidades que, según el último informe anual de “US News & World Report”, cuentan con los programas en ciencia de computación más relevantes de los Estados Unidos, el 72% (13) ofrece dichos programas dentro de la entidad académica equivalente al Colegio de Ingeniería del Recinto Universitario de Mayaguez. En seis (6) de estas universidades, la administración de los programas de ciencia de computación es compartida con la de los programas de Ingeniería Eléctrica y/o de Ingeniería de Computadoras. Ejemplos de estas son MIT, UC Berkeley y Stanford.

La reciente consolidación de las agencias de acreditación académica para ciencia de computación e ingeniería antes mencionada, refuerza las bondades de este modelo.

2.1.2 El desarrollo de programas de bachillerato en Ingeniería de Software

El desarrollo de programas de bachillerato en ingeniería de software es un fenómeno relativamente reciente. No obstante, ya existen en la actualidad trece programas en Inglaterra, tres programas en Australia, y dos programas en Canadá. En Estados Unidos hay tres programas: el de Rochester Institute of Technology, el de Fort Monmouth University y el de Southern Politechnical State University. Existen además cerca de cuarenta programas de Maestría en Ingeniería de Software en Estados Unidos.

En noviembre de 1999, el Software Engineering Institute (SEI) , una entidad de la Universidad de Carnegie Mellon, publicó un documento pionero titulado "Guidelines for Software Engineering Education", el cual establece guías y criterios para el desarrollo de programas de bachillerato en ingeniería de software. Se espera que este documento estimule el desarrollo de nuevos programas a nivel nacional y mundial.

2.1.3 El establecimiento de ingeniería de software como una profesión independiente.

Entidades tales como el "IEEE Software Engineering Standards Committee", el Departamento de Defensa de los Estados Unidos. y el "Software Engineering Institute" han promovido exitosamente el establecimiento de estándares para la especificación de requerimientos y diseño, prácticas administrativas, pruebas y validación de programas computacionales. También, el "Texas Board of Professional Engineering" ha comenzado a otorgar licencias profesionales a ingenieros de software que cumplan con criterios de acreditación. La "IEEE Computer Society" y la "Association for Computer Machinery" han formado el grupo de trabajo: "Joint Steering Committee for the Establishment of Software Engineering as a Profession". El propósito de este comité de alto nivel es precisamente definir estándares de ética, gama de conocimientos requeridos, prácticas recomendadas y currículos necesarios para establecer la Ingeniería de Software como una profesión. Este comité ha hecho publico un documento de referencia titulado "Guide to the Software Engineering Body of Knowledge". Este comité estudia también los requisitos para el examen "Principles of Practice Examination" (PE) , los cuales serán puestos ante la consideración del "National Council of Examiners for Engineers and Surveyors" (NCEES). Cabe destacar que esta última organización ya ha incluido un cinco porciento de contenido de Ingeniería de Software en su examen de Ingeniería Eléctrica.

Uno de los objetivos del programa propuesto es preparar formalmente a profesionales para una eventual certificación como ingenieros profesionales de software. Para lograr este objetivo se espera colaborar con sectores de la industria, el comercio, el gobierno, y el Colegio de Ingenieros de Puerto Rico en la formulación de acciones tendientes al reconocimiento de la Ingeniería de Software como una profesión regulada. Los últimos avances de la profesión como disciplina de ingeniería hacen suponer que no es materia de si esta certificación se dé o no, sino más bien de cuándo comenzará el proceso de licenciamiento. El plan de desarrollo económico de Puerto Rico. contempla el desarrollo de la industria del software, lo cual impone presión adicional para la certificación de ingenieros de software. Los graduados del programa propuesto estarán en posición de obtener dicha certificación

2.1.4 Computación como actividad académica mundial

La computación es la disciplina de mayor actividad académica en el mundo. En su fase teórica, la computación plantea una amplia gama de problemas relacionados a los fundamentos de la disciplina, su filosofía, sus métodos y principios básicos. Todo esta nueva organización de conocimientos se desarrolla y valida experimentalmente, tanto en la academia como en la industria especializada. En su fase tecnológica, el producto central de la ciencia de la computación: la tecnología de la información, ha impactado la comunicación social, el comercio, el modo que se hace investigación científica y tecnológica, el modo de producción industrial y la cultura entera. Por esta razón, la carencia de programas académicos en ciencia de computación conlleva la potencial marginación cultural y económica de una sociedad.

2.1.5 Ingeniería de Software como actividad académica

Dentro de este contexto, la ingeniería de software se destaca como una naciente especialización de gran importancia. No hay duda de que uno de los problemas centrales en la cultura informática es el de la eficiente producción de software de alta calidad. De dicha calidad dependen la economía, el comercio mundial y en muchos casos, la vida de seres humanos. En el informe del “President’s Information Technology Advisory Committee” (PITAC-report) del año 1999, se señala:

Software - The demand for software has grown far faster than our ability to produce it. Furthermore, the Nation needs software that is far more usable, reliable, and powerful than what is being produced today. We have become dangerously dependent on large software systems whose behavior is not well understood and which often fail in unpredictable ways. Therefore, increases in research on software should be given a high priority. Special emphasis should be placed on developing software for managing large amounts of information, for making computers easier to use, for making software easier to create and maintain, and for improving the ways humans interact with computers.

2.2 Necesidad del Nuevo Programa

Además de lo expuesto en los puntos 2.1 y 2.2, está la demanda estudiantil. Esta puede ser estimada por el alto número de solicitudes de admisión presentadas al programa en Ingeniería de Computadoras y por una encuesta realizada por el Dr. Manuel A. Pérez, ex catedrático auxiliar del Departamento de Ingeniería Eléctrica y Computadoras. El número de solicitudes de admisión al programa de bachillerato en ingeniería de computadoras ha ido en aumento (665 en 1996-97, 637 en 1997-98, 808 en 1998-99). Dichas solicitudes corresponden tradicionalmente a los estudiantes de mayor índice académico del sistema de la UPR. Debido a limitaciones en recursos, el departamento de Ingeniería Eléctrica y Computadoras ha mantenido el número de admisiones constante en 75 estudiantes por año. De acuerdo a la encuesta hecha por el Dr. Pérez 32 de un total de 40 estudiantes (80%) de ingeniería en computadoras se manifestó interesado en estudiar ciencia de computación o ingeniería de software, si tales programas hubiesen existido en el departamento. Una simple proyección de esta encuesta nos permite conjeturar que aproximadamente el 80% de los estudiantes que solicitan admisión al programa de Ingeniería de

Computadoras- esto es, alrededor de 600- solicitarían admisión al programa de Ciencia de Computación o al de Ingeniería de Software.

El programa propuesto responderá también a la alta demanda de la industria por expertos en computación e ingenieros de software. Dicha demanda es un hecho reconocido en Estados Unidos. En Puerto Rico, diferentes comités asesores y compañías de consultores contratados por el gobierno han coincidido en que la tecnología de la información, la computación y la comunicación son los ejes fundamentales de una política de desarrollo de ciencia y tecnología para el país. Entre estos informes se destaca el estudio sobre Tecnologías de Comunicaciones e Información de la empresa consultora Arthur D. Little. Este plantea la creación de un *Centro para el Desarrollo de Software*, destinado a convertirse en el primer centro de América Latina como un hito fundamental en dicha política. Este centro tendría la misión de conducir proyectos de desarrollo de alta tecnología en colaboración con la industria. El programa que propone en este documento está directamente relacionado con objetivos de este tipo. De hecho, el estudio antes mencionado reconoce el rol protagónico que le cabe al Recinto Universitario de Mayagüez en el desarrollo de recursos humanos capaces de llevar a cabo este tipo de empresa.

2.3 Oportunidades de Empleo para los Egresados del Programa

La demanda de profesionales de la computación en Puerto Rico, Estados Unidos y el mundo es un fenómeno ampliamente reconocido. En la medida que el rol de la tecnología de la información como eje central de la comunicación social, la enseñanza, la investigación científica y tecnológica, el desarrollo de productos, y el comercio continúe en expansión, dicha demanda aumentará. Una evidencia clara de esta demanda es el alto número de reclutadores de compañías que visitan el Recinto en busca de candidatos a posiciones profesionales. Este hecho es particularmente notable en el Departamento de Ingeniería Eléctrica y Computadoras, de donde frecuentemente se reclutan estudiantes egresados de Ingeniería de Computadoras para posiciones relacionadas con Ciencia de Computación o Ingeniería de Software. Entre dichas compañías se destacan IBM Corp., Motorola, Oracle, Lucent Technologies, AT&T, Xerox Corp. y Raytheon. Cabe destacar que el programa de Ingeniería de Computadoras adolece de una preparación integral en ciencia de computación y no cuenta con especialización en el proceso de desarrollo de software, requisitos regularmente en demanda por los reclutadores. El programa propuesto al satisfacer dichos requerimientos, hará que los egresados sean más atractivos y competitivos en el mercado de trabajo.

3. RELACIÓN DEL PROGRAMA CON EL PLAN ESTRATÉGICO SISTÉMICO DE LA UNIVERSIDAD DE PUERTO RICO

Los objetivos del programa propuesto concuerdan con los del Plan Estratégico Sistémico de la Universidad de Puerto Rico. En particular, el programa guarda una estrecha relación con las direcciones estratégicas de los Asuntos Críticos 6 y 7 del Plan.

El Asunto Crítico 6 del Plan es PROCURAR LA ACTUALIZACIÓN CONTINUA DE LA OFERTA ACADÉMICA. Entre las Direcciones Estratégicas de este Asunto Crítico, las siguientes guardan una estrecha relación con el Programa propuesto:

- ?? Establecer un sistema continuo de investigación institucional y de revisión de los programas académicos que incorpore en su análisis los cambios en el conocimiento, en el entorno social y en el perfil estudiantil
- ?? Promover la revisión continua del contenido de los cursos para incorporar los resultados de la actividad investigativa y los nuevos desarrollos en las disciplinas

Por mucho tiempo el sistema de la universidad de Puerto Rico ha prescindido de un programa en ciencia de computación a tono con los cambios experimentados por la disciplina. La ausencia de un programa de computación acreditado por la agencia acreditadora de programas en computación ABET es un reflejo de esto. El programa en ciencia de computación que se propone tiene todos los elementos necesarios para llegar a ser un programa acreditable a tono con los cambios en la disciplina.

Por otro lado el desarrollo de la ingeniería de software como una disciplina de ingeniería es evidencia de un cambio en el conocimiento, el entorno social y el perfil estudiantil que debe motivar la actualización de la oferta académica para incorporar dichos cambios.

El Asunto Crítico 7 del Plan es MEJORAR CONTINUAMENTE LOS PROCESOS DE APRENDIZAJE - ENSEÑANZA. Entre las Direcciones Estratégicas de este Asunto Crítico, la siguiente guarda estrecha relación con el programa propuesto:

- ?? Incorporar experiencias de investigación y creación a los currículos graduados y subgraduados

La ciencia de la computación y la ingeniería de software proveen una amplia gama de oportunidades para la investigación subgraduada. Productos del impacto de Netscape han sido resultado directo de investigación subgraduada. Consciente de esta característica de la disciplina, el programa incentivará la creatividad a través del estilo de enseñanza, los laboratorios y la participación de sus estudiantes en foros de investigación subgraduada.

4. RELACIÓN DEL NUEVO PROGRAMA CON OTROS PROGRAMAS

En términos generales, la educación universitaria en computación está dividida entre programas dedicados a la formación de expertos en computación y programas dedicados a la capacitación en el uso de la computación en otras áreas profesionales. El siguiente análisis compara el programa propuesto con programas dedicados a la formación de expertos en computación.

4.1 En el Recinto Universitario de Mayagüez

El programa que se propone está íntimamente relacionado con el programa de bachillerato en Ingeniería de Computadoras (ICOM) del Departamento de Ingeniería Eléctrica y Computadoras. De hecho, el programa que se propone en este documento es la evolución del componente de Ciencia de Computación e Ingeniería de Software de ICOM hacia un programa independiente. Ambos programas compartirán un número significativo de cursos tanto en ciencia de computación como en ingeniería de software. La diferencia entre ingeniería de computadoras y ciencia de computación e ingeniería de software está marcada por los campos de especialización de cada disciplina y un énfasis ya sea en el aspecto software o hardware en el currículo. Tradicionalmente corresponden a ingeniería de computadoras áreas de especialización tales como percepción remota, procesamiento digital de señales, diseño de hardware, VLSI, y en general, el diseño de equipo computadorizado y firmware, es decir, software de carácter relativamente permanente.

La Opción de Ciencias de Computación del programa bachillerato en Matemáticas del Departamento de Matemáticas del Recinto Universitario de Mayagüez ofrece algunos cursos en Ciencia de Computación similares a algunos de los contemplados en esta propuesta. Este programa no cubre suficientes materias como para ser catalogado de programa en ciencia de computación.

El programa que se propone guarda una leve relación con el programa en Sistemas de Información que ofrece el Colegio de Administración de Empresas del Recinto Universitario de Mayagüez. Los programas en sistemas de información no son considerados programas en ciencia de computación ni en ingeniería de software y de hecho, tienen un conjunto de criterios de acreditación independiente

4.2 En Otras Unidades del Sistema Universitario de la U.P.R.

El programa propuesto está relacionado con los programas de bachillerato de Ciencias de Computadoras del Recinto de Río Piedras, Ciencias de Computadoras del Colegio Universitario Tecnológico de Bayamón, Ciencias de Computadoras del Colegio Universitario Tecnológico de Arecibo, y Matemáticas Computacionales del Colegio Universitario de Humacao. Todos estos son programas de bachillerato en Ciencia de Computación o contienen secuencias estructuradas de cursos en Ciencia de Computación a nivel de bachillerato. La diferencia principal entre estos programas y el que se propone en este documento está en la capacidad de este último para integrar aspectos científicos y de ingeniería efectivamente tanto dentro del currículo como en la experiencia de aprendizaje del estudiante y una mayor oferta de posibilidades de especialización profesional.

La Concentración en Sistemas Computadorizados de Información del programa de Bachillerato en Administración de Empresas del Recinto de Río Piedras contiene temas en Ciencia de Computación, dentro de un currículo orientado a la administración. Nuevamente, este tipo de programas no es normalmente clasificado ni acreditado con los mismos parámetros de un programa en ciencia de computación ni de ingeniería de software.

4.3 En Otras Instituciones del País

El programa propuesto está relacionado con el programa de bachillerato en Ciencias de Cómputos de la Universidad Metropolitana. Este es también un programa de bachillerato en ciencia de computación. El programa propuesto es más amplio en su cobertura de temas tanto del área de computación como en formación general y de ingeniería. El programa propuesto ofrece además una gama mayor de posibilidades de especialización.

5. DESCRIPCIÓN DEL PROGRAMA

5.1 Filosofía, Metas y Objetivos Específicos del Programa

La filosofía fundamental del programa propuesto en este documento es la de una visión integradora de la disciplina de la computación. Tal visión es fundamental en una disciplina emergente, plena en posibilidades de expansión y con una influencia decisiva en el desarrollo social y económico de la nación.

El programa propuesto tiene como meta principal la formación de expertos en ciencia de computación e ingeniería de software capacitados profesional, cultural y éticamente para actuar como agentes de desarrollo e investigación en computación.

Los objetivos específicos del programa son:

?? Liderar en la educación en ciencia de computación e ingeniería de software en Puerto Rico. Este liderazgo se reflejará en:

- ?? el desarrollo y mejoramiento continuo del currículo de acuerdo a nuevas tendencias en ciencia de computación e ingeniería de software,
- ?? el mantenimiento de equipos y recursos de enseñanza actualizados,
- ?? el mantenimiento y expansión de una facultad de excelencia,
- ?? la capacidad para atraer un número significativo de estudiantes sobresalientes al programa.

? ? Formar profesionales en ciencia de computación e ingeniería de software dotados de:

- ?? una visión global de la disciplina de la computación y un área de especialización claramente definida,
- ?? aprecio tanto por los asuntos teóricos de la disciplina, como los prácticos y la interrelación entre ellos,
- ?? capacidad para evolucionar de acuerdo a los cambios en la disciplina,
- ?? capacidad de comunicación técnica y profesional en español e inglés,
- ?? conciencia de los asuntos éticos y sociales relacionados a la práctica profesional e interacción social,
- ?? conocimientos en ciencia y economía.

? ? Servir de nexo entre la universidad, la industria y el gobierno. Esta capacidad deberá reflejarse en:

- ?? la existencia de un número significativo de proyectos de investigación y desarrollo con organismos del gobierno o la industria que involucren estudiantes y facultad del programa,
- ?? el establecimiento de relaciones formales con estas entidades y otros organismos profesionales como el Colegio de Ingenieros y Agrimensores, con el objetivo de colaborar en el logro de metas profesionales comunes.

? ? El programa debe ser acreditado por ABET

5.2 Perfil del Egresado

5.2.1. Rasgos comunes a ambos grados

- ?? Conocimiento amplio de ciencias de la computación. Familiaridad con el lenguaje y los temas y problemas básicos de la disciplina. Dominio de temas tales como algoritmos, programación, estructuras de datos, lenguajes de programación, sistemas de computación, bases de datos y arquitecturas de computadoras.
- ?? Conocimiento de los fundamentos de ingeniería, que lo capacitan para participar activamente en desarrollos de aplicaciones tales como sistemas de control, programas de análisis y programas de diseño, entre otras.
- ?? Conocimiento básico de los fundamentos de electrónica necesarios para el entendimiento de los aparatos físicos controlados por la programación y de los mecanismos internos de un sistema de computación.
- ?? Capacidad para aplicar creativa e integradamente estos conocimientos en la solución de problemas y en el diseño, evaluación y mantenimiento de sistemas informáticos.
- ?? Capacidad de analizar y determinar las especificaciones apropiadas a un problema específico y proponer estrategias de solución. Capacidad para determinar en que medida

un sistema de computación satisface los criterios definidos para su uso actual y desarrollo futuro.

- ?? Habilidad de comunicación oral y escrita en los idiomas inglés y español.
- ?? Capacidad de formular claramente sus objetivos a corto, mediano y largo plazo y de hacer sus ideas y resultados comprensibles para sus compañeros de trabajo. Capacidad para comunicar efectivamente los aspectos esenciales de un determinado problema y su solución, a una audiencia general.
- ?? Capacidad de aprender de sus experiencias y realizar trabajo en equipo. Capacidad para evolucionar de acuerdo a la naturaleza cambiante de la disciplina.
- ?? Conciencia del impacto de su trabajo en la calidad de vida de la sociedad, incluyendo un claro entendimiento y respeto por los asuntos legales, éticos, sociales y culturales pertinentes a la práctica de su profesión. Capacidad para identificar los riesgos potenciales asociados al uso u operación de sistemas de computación.
- ?? Visión global de los sistemas de computación. El egresado del programa debe entender la estructura de los sistemas de computación y los procesos involucrados en su construcción y análisis, más allá de los detalles asociados a una implantación en particular.
- ?? Apreciación de la relación entre teoría y práctica. El egresado debe apreciar tanto el valor de un buen diseño de ingeniería como el del marco teórico que lo sustenta. Esto es, debe entender el valor de la interrelación entre teoría, experimento y resultado y ser capaz usarla efectivamente en su practica profesional.

5.2.1. Rasgos propios del egresado del bachillerato en Ingeniería de Software

- ?? Conocimientos vastos en técnicas y metodologías para el desarrollo de software a gran escala, tales como interacción humano-computadora, especificación, diseño, codificación y prueba, y aquellos relacionados a la gerencia, tales como administración de proyectos y control de calidad.
- ?? Capacidad para integrar y aplicar sus conocimientos en el desarrollo de proyectos novedosos de software.

5.2.2. Rasgos propios del egresado del bachillerato en Ciencia de Computación

- ?? Conocimientos vastos de aspectos teóricos y principios fundamentales en ciencia de computación, tales como teoría de autómatas, modelos formales de computación, complejidad computacional, entre otros.
- ?? Conocimientos especializados en una o más áreas de ciencia de computación. Ejemplos de estas áreas son: computación en redes, inteligencia artificial, sistemas computacionales, bases de datos, computación gráfica, diseño e implantación de lenguajes de programación, arquitectura de computadoras, entre otras.
- ?? Capacidad para aplicar los principios fundamentales de la disciplina en la solución de una amplia gama de problemas computacionales e informáticos.

5.3 Componentes del Programa

5.3.1 Distribución de los Cursos por Área

Los estudiantes deberán aprobar un mínimo de 158 créditos distribuidos de la siguiente manera:

Formación General	33
Fundamentos de Ingeniería	21
Ciencias y Matemáticas	32
Fundamentos de Computación	33
Especialidad (Computación/Software)	27
Electivas libres	12

A continuación se presenta el desglose de los cursos requeridos por área:

Formación general: 33 créditos.	
Español	6
Inglés	12
Professional Conversation	1
Electivo en Educación Física	2
Electivas Sociohumanísticas	12

Ciencias y Matemáticas: 32 créditos.	
Cálculo I, II y III	11
Ecuaciones Diferenciales Ordinarias	3
Química I, II	8
Física I, II con Laboratorios	10

Fundamentos de ingeniería: 21 créditos.	
Gráficas de Ingeniería	2
Mecánica Aplicada – Estática y Dinámica	3
Termodinámica	3
Probabilidad y Estadística de Ingeniería	3
Materiales de Ingeniería	3
Economía de Ingeniería	3
Intro a Circuitos Eléctricos	3
Lab. Intro a Circuitos Eléctricos	1

Electivas libres: 12 créditos.

Fundamentos de Computación: 33 créditos.	
Introducción a la Programación	3
Programación Avanzada	3
Estructuras de Datos	3
Estructuras Discretas para Computación	3
Análisis y Diseño de Algoritmos	3
Lenguajes de Programación	3
Redes de Computadoras	3
Sistemas Operativos	3
Sistemas de Bases de Datos	3
Arquitectura de Computadoras I y II	6

Grado en Ciencia Computación: 27 créditos.	
Teoría de Computación	3
Introducción a Ingeniería de Software	3
Computación Científica	3
Inteligencia Artificial	
Compiladores	
Introducción Interacción entre Humano y Computadora	
Organización de Sistemas de Computación	
Admin de Proyectos de Programación	
Desarrollo de Sistemas Distribuidos	
Temas especiales en Computación	
Investigación subgraduada en Computación	

Grado en Ingeniería de Software: 27 créditos.	
Introducción a Ingeniería de Software	3
Introducción a Interacción Humano-Computadora	3
Especificación de Programas	3
Diseño de Programas	3
Prueba y Confiabilidad de Programas	3
Proyecto Ingeniería de Software I y II	3
Administración de Proyectos de Software	6
Inteligencia Artificial	
Desarrollo de Compiladores	
Desarrollo Sistemas Distribuidos	
Investigación subgraduada en Ingeniería de Software	

5.3.2 Descripción de los Cursos

5.3.2.1 Cursos requeridos y electivas técnicas para ambos grados

A continuación se describen los cursos de ciencia de la computación requeridos para ambos grados ofrecidos por el programa. Se propone una codificación común CCOM (ciencia de la computación)

Cursos Nuevos

CCOM3edc. Estructuras Discretas para Computación. 3 créditos. Prerreq: ninguno. Introducción a estructuras matemáticas discretas y su uso en Ciencia de la Computación. Incluye tópicos tales como la relación entre datos y conjuntos, técnicas de demostración, operadores y funciones, lógica y circuitos elementales, grafos y organización de procesos computacionales, elementos de probabilidad discreta y eventos aleatorios en Computación. Una (1) hora de laboratorio y tres horas de conferencia semanal.

CCOM3iap. Introducción a la Programación. (INGE 3016) 3 créditos. Prerreq: ninguno. Análisis de problemas y programación en un lenguaje de alto nivel. Sistemas numéricos, representaciones internas, constantes, variables y tipos de datos. Estructuras de control de decisión e iterativas. Funciones y mecanismos básicos de entrada y salida de datos. Recurrencia. Problemas de programación, con aumento progresivo en dificultad. Una (1) hora de laboratorio y tres horas de conferencia semanal.

CCOM4pa. Programación Avanzada. (ICOM 4015) 3 créditos. Prerrequisitos: CCOM3iap. Análisis de problemas y programación en un lenguaje de alto nivel. Tipos de dato abstractos. Conceptos avanzados de programación incluyendo encapsulación, modularidad, polimorfismo paramétrico y por subtipos, algoritmos recursivos, manejo de memoria dinámica. Énfasis en el control de la complejidad de sistemas complejos de software. Una (1) hora de laboratorio y tres horas de conferencia semanal.

CCOM3aci. Arquitectura de Computadoras I. 3 creds. Prereq: None. Conceptos fundamentales de análisis y diseño de circuitos lógicos; álgebra booleana y compuertas lógicas, circuitos combinatorios y secuenciales, unidades aritméticas y lógicas (ALU), memoria y dispositivos lógicos programables, transferencias entre registros y vías de datos, diseño de unidades de control. Tres (3) horas de conferencia semanal.

CCOM4acii. Arquitectura de Computadoras II. 3 creds. Prereq: CCOMcai. Fundamentos de conjuntos de instrucciones: operaciones, registros, formatos de instrucciones, tipos de datos, modos de direccionamiento a operandos, brincos, subrutinas, interrupciones, entrada/salida, acceso a memoria; programación en lenguaje de máquina; programación en lenguaje de ensamblaje; jerarquía de memoria: caches, memoria virtual, unidades de almacenamiento secundario; pipelining; sistema de entrada/salida; aritmética de computadoras; apoyo arquitectural a sistemas operativos y lenguajes de programación. Tres (3) horas de conferencia semanal.

CCOM4ada. Análisis y Diseño de Algoritmos. 3 Creds. Prereq: CCOMedc, CCOMedd. Formal algorithm analysis. Fundamental algorithm design techniques: greedy algorithms, divide-and-conquer, dynamic programming, backtracking. Fundamental graph algorithms. Searching and sorting. Una (1) horas de laboratorio y tres horas de conferencia semanal.

CCOM4rdc. Redes de Computadoras. 3 creds. Prereq: ICOM 5007. Modelo estratificado de redes de computadoras. Arquitecturas y topologías de redes de computadoras. Protocolos para los estratos de enlace, redes, transporte y aplicaciones. Énfasis en los protocolos utilizados por Internet: IP, TCP, HTTP, entre otros. Seguridad en redes de computadoras. Aplicaciones para el World Wide Web. Una (1) horas de laboratorio y tres horas de conferencia semanales.

CCOM 5ddsd. Desarrollo de Sistemas Distribuidos. 3 creds. Prereqs: CCOMso. Arquitectura de sistemas operativos distribuidos. Implantación eficiente de servicios de red. RPC. Métodos para comunicación, coordinación y sincronización eficiente entre procesos distribuidos. Agendas y transacciones distribuidas. Programación de seguridad de sistemas distribuidos. Introducción a algoritmos distribuidos. Tres horas de conferencia semanales.

Cursos Existentes

ICOM 4035. Data Structures. 3 creds. Prereq: ICOM 4015. Basic concepts of data. Representation of information as data inside and outside the computer. Lists in linear, orthogonal, strings and array forms. Tree structures. Techniques for storage allocation, distribution, collection, and sorting of data. Data structures in programming languages. Two hours of lecture and three hours of laboratory per week.

ICOM 4036. Lenguajes de Programación. (ICOM 4036) 3 creds. Prereq: CCOMedd. Estudio comparativo de lenguajes de programación, funcional, imperativo, orientado-objetos, y lógico. Conceptos de gramáticas de lenguajes, representación y tipos de data, procedimientos y funciones, procesamiento de símbolos, principios de programación concurrente y programación lógica. Una (1) horas de laboratorio y tres horas de conferencia semanales.

ICOM 5007. Operating Systems Programming. Four credit hours. Three hours of lecture and one-three hour laboratory per week. Prerequisites: ICOM 4035 and CCOMcaii. Concepts of operating systems, multiprogramming, multiprocessing, batch, partitioned, and real time. Organizational and processing of file systems. Study of queueing theory and information flow control.

ICOM 4017. Computer-based Information Systems. Three credit hours. Three hours of lecture per week. Prerequisite: ICOM 4035. Analysis and design of computer-based management information systems; communication theory and the flow of information within organizations; methods and procedures of gathering, disseminating and controlling information; integrated Electronic Data Processing versus batchcontrolled system; the development and installation of information processing systems.

ICOM5015. Inteligencia Artificial. 3 creds. Prereqs: CCOMedd. Introducción al campo de la inteligencia artificial. El lenguaje LISP y sus derivados. Técnicas de búsqueda, juegos, visión, representación del conocimiento, inferencia. Pruebas de teoremas. Entendimiento de lenguaje natural. Tres horas de conferencia semanales.

ICOM 4029. Desarrollo de Compiladores. (ICOM4029) 3 creds. Prereqs: CCOMldp. Técnicas de análisis de código fuente. Análisis léxico y gramático. Generación de código objeto eficiente. Componentes compiladores e interpretadores. Tres horas de conferencia semanales.

5.3.2.2 Cursos requeridos y electivas técnicas para ingeniería de software

Cursos Nuevos

INSO4sss. Introducción a la Ingeniería de Software. 3 creds. Prereqs: CCOM4ldp. Introducción al proceso de ingeniería de software. Modelos de procesos de desarrollo y medidas relacionadas. Modelo lineal, modelos evolutivos, en espiral y de dominios de aplicación. Introducción a las actividades de desarrollo, incluyendo planificación, análisis, especificación, diseño, codificación, prueba e instalación. Control de versiones. Administración de configuración de programación. Herramientas CASE. Administración de proyectos. Ética en la ingeniería de software. Una (1) hora bisemanal de laboratorio.

INSO4aaa. Administración de Proyectos de Programación. 3 creds. Prereqs: INSO4sss. Estimación, planificación, monitoreo, documentación, evaluación, refinamiento y control de calidad. Administración de personal. Proceso Personal de Software (PSP). Proceso de Software en Equipo (TSP). Modelo de Madurez de Capacidad (CMM). Análisis de Riesgos. Planificación de hitos. Método de senda crítica (CPM) y diagramas PERT. Herramientas de control de proyectos. Una (1) hora semanal de laboratorio.

INSO4sss. Introducción a la Ingeniería de Software. 3 creds. Prereqs: CCOM4ldp. Introducción al proceso de ingeniería de software. Modelos de procesos de desarrollo y medidas relacionadas. Modelo lineal, modelos evolutivos, en espiral y de dominios de aplicación. Introducción a las actividades de desarrollo, incluyendo planificación, análisis, especificación, diseño, codificación, prueba e instalación.

Control de versiones. Administración de configuración de programación. Herramientas CASE. Administración de proyectos. Ética en la ingeniería de software. Una (1) hora bisemanal de laboratorio.

INSO4ttt. Introducción a Interacción Humano-Computadora. 3 creds. Prereqs: INSO4sss. Introducción a los principios de interacción humano-computadoras; psicología cognoscitiva, factores humanos, estilos de interacción. Diseño e implementación de interfaces gráficas para usuarios; análisis de tareas, herramientas de desarrollo para interfaces gráficas, métodos de evaluación de usabilidad. Tres (3) horas de conferencia semanal.

INSO4uuu. Especificación de Programas. 3 creds. Prereqs: INSO4sss. Técnicas para determinar, identificar y representar necesidades y requisitos. Métodos formales de especificación. Análisis orientado a objetos. Estándares de especificación de requerimientos. El lenguaje UML. Verificación de requerimientos. Herramientas de especificación. Administración de requerimientos. Arquitecturas de Programación. Una (1) hora semanal de laboratorio.

INSO4vvv. Diseño de Programas. 3 creds. Prereqs: INSO4uuu. Elementos fundamentales y técnicas de diseño de software. Métodos formales de diseño. Diseño funcional y orientado a objetos. Estándares de especificación de diseño. El lenguaje UML. Validación y verificación de diseño. Métricas de diseño. Patrones de diseño. Diseño por contrato y el lenguaje Eiffel. Herramientas de diseño. Una (1) hora semanal de laboratorio.

INSO4www. Prueba y Confiab Programas. 3 creds. Prereqs: INSO4vvv. Pruebas y validación en el ciclo de vida de un sistema de software. Técnicas, herramientas y modelos formales de pruebas. Validación y verificación de especificaciones y diseño mediante pruebas en programas. Pruebas de unidad, integración, rendimiento, de esfuerzo y de tolerancia a fallas. Herramientas de prueba y depuración. Una (1) hora semanal de laboratorio.

INSO4yyy. Proyecto Ingeniería Software I. 3 creds. Prereqs: INSO4www, INSO4xxx. Primera parte de un curso de proyecto de ingeniería de software. Los estudiantes trabajarán en equipo para resolver problemas comerciales o industriales. Cada equipo será asignado un cliente interno o externo. El trabajo conlleva todo los aspectos del proceso de ingeniería de software al igual que presentaciones escritas y orales. La primera parte del curso termina con una presentación sobre el trabajo completado de las primeras etapas del proyecto.

INSO4zzz. Proyecto Ingeniería Software II. 3 creds. Prereqs: INSO4yyy. Segunda parte del curso de proyecto de ingeniería de software. Los estudiantes continúan trabajando sobre lo presentado en el primer curso. El trabajo termina con una presentación sobre el trabajo realizado y una evaluación del proyecto.

INSO 4bbb. Investigación Subgraduada en Ingeniería de Software. 3-6 Creds. Prereq: Permiso del Instructor. Desarrollo de un trabajo de investigación relacionado con ingeniería de software bajo la supervisión de un(a) miembro(a) de la facultad.

5.3.2.3 Cursos requeridos y electivos para ciencia de la computación

Cursos Nuevos

CCOM 5cso. Organización de Sistemas de Computación 3 Creds. Prereq: CCOM 4acii. Conceptos avanzados de diseño de procesadores y computadoras. Énfasis en métodos cuantitativos para la evaluación de "performance". Análisis y diseño de sistemas de computación. Dependencia organizacional en las computaciones a ser realizadas. Velocidad y costo de máquinas y sus componentes. Diseño de conjuntos de instrucciones. Arquitecturas "pipeline" y vectoriales. Diseño de jerarquías de memoria. Tres horas de conferencia semanales.

ICOM 4tdc. Teoría de Computación. 3 Creds. Prereq: CCOMedc. Discusión de varios aspectos de la teoría de la computación incluyendo: lógica, análisis combinatorio, análisis de algoritmos. Modelos de

computación. Tesis de Church. Halting problem. Clases de complejidad computacional en espacio y tiempo. NP-Completeness. Una (1) hora de laboratorio y tres horas de conferencia semanales.

CCOM 4isc. Computación Científica. 3 Creds. Prereq: CCOMedd. Conceptos fundamentales de computación científica. Sistemas numéricos de punto flotante. Análisis de error, estabilidad y convergencia. Métodos numéricos y simbólicos en ciencias e ingeniería. Tres horas de conferencia semanales.

CCOM 4tec. Temas Especiales en Computación. 3 Creds. Prereq: Permiso del Instructor. Estudio de temas especiales en computación.

CCOM 4isc. Investigación Subgraduada en Computación. 3 Creds. Prereq: Permiso del Instructor. Desarrollo de un trabajo de investigación relacionado con Ciencia de la Computación bajo la supervisión de un miembro de la facultad.

5.3.3.1 Modelo del Programa Propuesto: Tronco común a ambos grados (1^{ro} y 2^{do} años)

A continuación se presenta un modelo de los primeros dos años del programa propuesto. Estos dos años son comunes a ambos grados ofrecidos por el programa. El modelo está de acuerdo con la distribución de cursos mencionados en la sección 5.3.1.

Primer Año			
Primer Semestre			
Curso	Descripción	Créds	Prerrequisitos/Corequisitos
MATE 3031	Cálculo I	4	
QUIM 3001	Química General I	4	
INGL 3101	Inglés Básico I	3	
ESPA 3101	Español Básico I	3	
CCOM3iap	Introducción a la Programación	3	
EDFI ----	Electivo en Educación Física	1	
	Total	18	
Segundo Semestre			
MATE 3032	Calculo II	4	MATE 3031
QUIM 3002	Química General II	4	QUIM 3001
INGL 3102	Inglés Básico II	3	INGL 3101
ESPA 3102	Español BásicoII	3	ESPA 3101
CCOM 4pa	Programación Avanzada	3	CCOM 3iap
EDFI ----	Electivo en Educación Física	1	
	Total	18	

Segundo Año Primer Semestre			
Curso	Descripción	Créds	Prerrequisitos/Corequisitos
MATE 3063	Calculo III	3	MATE 3032
FISI 3171	Física I	4	
FISI 3173	Laboratorio de Física I	1	
INGE 3011	Gráficos de Ingeniería	2	
CCOM 3edc	Estructuras Discretas para Computación	3	
INGL 3201	Gramática, Composición y Lectura I	3	INGL3102
	Total	16	
Segundo Semestre			
MATE 4009	Ecuaciones Diferenciales Ordinarias	3	MATE3063
FISI 3172	Física II	4	FISI3171
FISI 3174	Laboratorio de Física II	1	FISI3173
ICOM 4035	Estructuras de Datos	3	CCOM 4pa, CCOM3edc
INGL 3235	Technical Report Writing	3	INGL3102
INEL 4icee	Intro Circuitos Eléctricos y Electrónicos	3	
INEL 4iceeL	Intro Circuitos Eléctricos y Electrónicos, Laboratorio	1	
	Total	18	

5.3.3.2 Modelo del Programa Propuesto: B.S. Ciencia de la Computación (3^{ro} a 5^{to} año)

A continuación se presenta un modelo de los últimos tres años para el grado en Ciencia de la Computación. Este modelo está de acuerdo con la distribución de cursos mencionados en la sección 5.3.1.

Tercer Año Primer Semestre			
Curso	Descripción	Créds	Prerrequisitos/Corequisitos
CCOM 4ada	Análisis y Diseño de Algoritmos	3	ICOM 4035
INSO 4sss	Intro Ingeniería de Software	3	ICOM 4036
CCOM aci	Arquitectura Computadoras I	3	
ININ 4010	Probabilidad y Estadística para Ing.	3	CCOM3iap
ICOM 4036	Lenguajes de Programación	3	ICOM 4035
INGL 3195	Profesional Conversation	1	INGL 3102
	Total	16	
Segundo Semestre			
CCOM 4tdc	Teoría de la Computación	3	CCOM 4ada
CCOM 4eee	Arquitectura Computadoras II	3	CCOM 3aci
ININ 4015	Economía de Ingeniería	3	ININ 4010
INME 4045	Termodinámica	3	QUIM3002, FISI3172
CCOM ----	Electivo Técnico**	3	
	Total	15	

Cuarto Año			
Primer Semestre			
Curso	Descripción	Crédts	Prerrequisitos/Corequisitos
ICOM 5007	Sistemas Operativos	3	CCOM4 edd, CCOM 4caii
CCOM 4cs	Computación Científica	3	CCOM edd
CCOM ----	Electivo Técnico**	3	
---- ----	Electivo libre	3	
---- ----	Electivo Sociohumanístico	3	
	Total	15	
Segundo Semestre			
ICOM 4017	Sistemas de Bases de Datos	3	ICOM 5007
CCOM 4rdc	Redes de Computadoras	3	ICOM 5007
CCOM ----	Electivo Técnico**	3	
---- ----	Electivo Sociohumanístico*	3	
---- ----	Electivo Libre	3	
	Total	15	

Quinto Año			
Primer Semestre			
Curso	Descripción	Crédts	Prerrequisitos/Corequisitos
CCOM ----	Electivo Técnico**	3	
CCOM ----	Electivo Técnico**	3	
---- ----	Electivo Sociohumanístico*	3	
---- ----	Electivo Libre	3	
	Total	12	
Segundo Semestre			
CCOM ----	Electivo Técnico**	3	
INGE 4001	Materiales de Ingeniería	3	
---- ----	Electivo Sociohumanístico*	3	
---- ----	Electivo Libre	3	
INGE 3035	Mecánica para Ingeniería	3	
	Total	15	

Total de créditos	
158	
* Electivo sociohumanístico seleccionado de la lista de cursos recomendados.	
** Electivo técnico seleccionado de la lista de cursos recomendados (sec 5.3.2.2)	

5.3.3.3 Modelo del Programa Propuesto: B.S. Ingeniería de Software

A continuación se presenta un modelo del programa. Este modelo está de acuerdo con la distribución de cursos mencionados en la sección 5.3.1.

Tercer Año			
Primer Semestre			
Curso	Descripción	Créds	Prerrequisitos/Corequisitos
CCOM 4ada	Análisis y Diseño de Algoritmos	3	ICOM 4035
INSO 4sss	Intro Ingeniería de Software	3	ICOM 4036
CCOM aci	Arquitectura Computadoras I	3	
ININ 4010	Probabilidad y Estadística para Ing.	3	CCOM3iap
ICOM 4036	Lenguajes de Programación	3	ICOM 4035
INGL 3195	Profesional Conversation	1	INGL 3102
	Total	16	
Segundo Semestre			
INSO 4ttt	Interacción Humano Computadora	3	INSO 4sss
INSO 4uuu	Especificación de Programas	3	INSO 4sss
CCOM 4eee	Arquitectura Computadoras II	3	CCOM 3aci
ININ 4015	Economía de Ingeniería	3	ININ 4010
INME 4045	Termodinámica	3	QUIM3002, FISI3172
	Total	15	

Cuarto Año			
Primer Semestre			
Curso	Descripción	Créds	Prerrequisitos/Corequisitos
CCOM 5007	Sistemas Operativos	3	CCOM4 edd, CCOM 4caii
INGE 3035	Mecánica para Ingeniería	3	
INSO 4vvv	Diseño de Programas	3	INSO4uuu
---- ----	Electivo libre	3	
---- ----	Electivo Sociohumanístico*	3	
	Total	15	
Segundo Semestre			
ICOM 4017	Sistemas de Bases de Datos	3	ICOM 5007
CCOM 4rdc	Redes de Computadoras	3	ICOM 5007
INSO 4www	Prueba y Confiabilidad Programas	3	INSO 4vvv
---- ----	Electivo Sociohumanístico*	3	
---- ----	Electivo Libre	3	
	Total	15	

Quinto Año			
Primer Semestre			
Curso	Descripción	Créds	Prerrequisitos/Corequisitos
INSO 4yyy	Proyecto Ingeniería Software I	3	INSO 4www***, INSO 4ttt
CCOM ----	Electivo Técnico**	3	
---- ----	Electivo Sociohumanístico*	3	
---- ----	Electivo Libre	3	
	Total	12	
Segundo Semestre			
INSO 4zzz	Proyecto Ingeniería Software II	3	INSO 4yyy
INGE 4001	Materiales de Ingeniería	3	
---- ----	Electivo Sociohumanístico*	3	
---- ----	Electivo Libre	3	
CCOM ----	Electivo Técnico**	3	
	Total	15	

Total de créditos	
158	
* Electivo sociohumanístico seleccionado de la lista de cursos recomendados.	
** Electivo técnico seleccionado de la lista de cursos recomendados (sec 5.3.2.2)	
*** Mínimo 4to año y haber completado 30 créditos en CCOM	

5.3.4 Metodología Educativa y Estrategias de Instrucción

La metodología educativa estará dirigida a cumplir con los objetivos del programa propuesto. Se utilizarán conferencias regulares, trabajo independiente, aprendizaje en colaboración y trabajo en equipo. La capacidad de trabajo individual será estimulada mediante asignaciones y preparación de informes, en algunos cursos. La capacidad de trabajo en equipo y de aprendizaje en colaboración, será incentivada y evaluada a través de proyectos que, por su complejidad, demanden la efectiva colaboración entre varios estudiantes. En particular, se proveerán actividades de aprendizaje que simulen efectivamente procesos de producción industrial y amplíen la visión global del estudiante sobre el proceso de desarrollo de un producto. Se entrenará a los estudiantes en el uso de herramientas aplicadas al desarrollo de programación mediante sesiones de laboratorio conducidas por el profesor o un ayudante de cátedra cualificado.

En términos generales, el estilo de enseñanza enfatizará el aprendizaje individual y activo por sobre la enseñanza pasiva. Los estudiantes serán continuamente incitados a pensar individualmente y criticar sus propias soluciones. Se promoverá el trabajo en equipo y el aprendizaje en grupo, incluyendo el uso efectivo de tecnologías de comunicación para la interacción entre los estudiantes y sus pares.

5.3.5 Catálogo y Promoción

La descripción del programa según se indica en la sección 5.3.1 y la descripción de los cursos que se indican en la sección 5.3.2 se incluirán en el catálogo de ofrecimientos académicos del Recinto Universitario de Mayagüez. Una vez aprobado el programa se prepararán un boletín informativo y afiches de promoción.

6. ADMISIÓN Y MATRÍCULA

6.1 Requisitos de Admisión

Los requisitos de admisión son haber aprobado la escuela secundaria y tener el índice de ingreso mínimo requerido. El índice de ingreso es función del promedio general de escuela superior y la puntuación obtenida en el "College Board". El índice de ingreso mínimo se computará basado en una proyección de oferta/demanda dado por el número de solicitudes y la proyección de matrícula.

6.2 Proyección de la Matrícula

Los recursos solicitados en esta propuesta están calculados para atender una matrícula de cincuenta (50) estudiantes de nuevo ingreso por año para cada uno de los dos grados de bachillerato que se proponen.

7. REQUISITOS ACADÉMICOS PARA OTORGAR EL GRADO

Los requisitos académicos para otorgar el grado de bachillerato en ingeniería de software y el grado de bachillerato de ciencia de computación se indican en las siguientes secciones.

7.1 Total de Horas-Crédito que se Requieren

Se requiere la aprobación de un total de ciento sesenta (158) créditos para otorgar el grado según se indica en la sección 5.3.1.

7.2 Índices Académicos Mínimos

Se requiere un índice académico no menor de 2.0 para graduarse del programa.

7.3 Total de Créditos a Aceptarse en Transferencia

Los cursos a aceptarse en transferencia son determinados por los directores de los departamentos a los que corresponden los cursos a ser acreditados. No obstante el estudiante debe tomar los últimos 28 créditos en el Recinto Universitario de Mayagüez. Este requisito puede ser obviado en casos excepcionales con la autorización del Decano de Ingeniería y el Decano de Asuntos Académicos.

7.4 Requisitos de Idioma

No hay.

7.5 Tiempo Límite para Completar el Grado

El tiempo límite para completar el grado es de diez (10) años como estudiante a tiempo completo.

8. FACULTAD

8.1 Facultad Necesaria Para el Establecimiento del Programa

8.1.1 Facultad Necesaria para Ofrecer los Cursos de Ciencias de Computación

Para ofrecer los cursos requeridos y los cursos electivos en ciencia de computación se debe reclutar facultad con especialidad en diferentes áreas. El número de profesores a reclutar por año, durante los primeros cinco años aparece en la tabla 8.1. Estos cálculos asumen una matrícula constante (para cada bachillerato) de 50 estudiantes de nuevo ingreso por año y una carga docente promedio de tres cursos por año. De este modo, la facultad dedicará un 50% de su tiempo a la enseñanza (tres cursos de bachillerato y uno graduado anualmente) y otro 50% a la investigación académica.

Tabla 8.1 Facultad Necesaria por Área de Especialidad en Ciencia de Computación e Ingeniería en Software Durante los Primeros Cinco Años

Área	Año				
	1	2	3	4	5
Sistemas operativos			1	1	
Bases de datos			1	1	
Lenguajes	2	3	1		
Interacción humano-computadora		1	1		
Ingeniería en Software			3	2	1
Procesamiento paralelo			1	1	
Computación Científica			1		
Redes de Computación			1	1	
Arquitectura de Computadoras			2		1

8.1.2 Facultad Necesaria para Ofrecer los Cursos de Otras Disciplinas

Para poder ofrecer el programa se necesitará la contratación de facultad en disciplinas como español, inglés, humanidades, ciencias sociales, matemáticas, física, química, educación física, ingeniería eléctrica, ingeniería general, ingeniería industrial y economía. En la figura 8.1 se presenta la proyección de la facultad necesaria para ofrecer los cursos de otras disciplinas durante los primeros 5 años. Se asume una carga académica anual de 8 cursos por profesor(a).

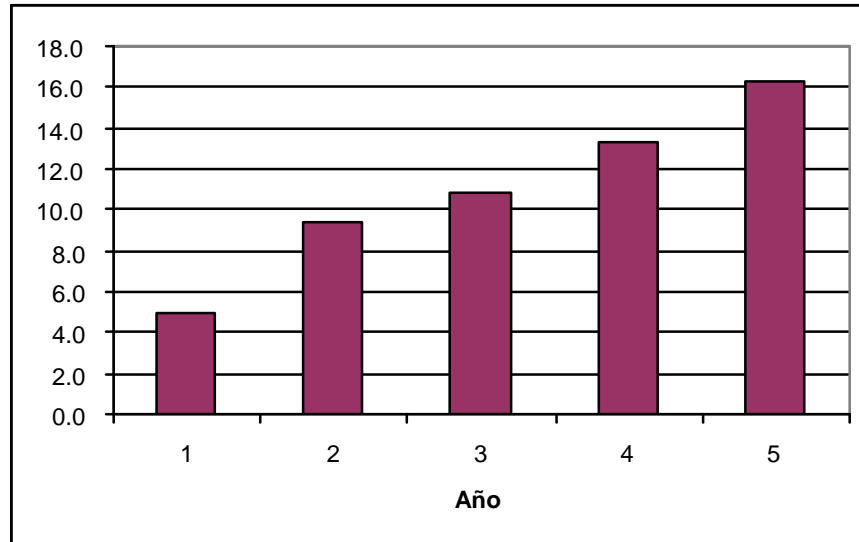


Figura 8.1 Facultad Necesaria para Ofrecer los Cursos de Otras Disciplinas Durante los Primeros Cinco (5) Años

8.2 Plan de Reclutamiento de la Facultad

8.2.1 Reclutamiento de Facultad para Ofrecer los Cursos de Ciencias de Computación

8.2.1.1 Reclutamiento de Estudiantes Talentosos

Tradicionalmente en la facultad de ingeniería el método de reclutamiento más efectivo ha sido el de enviar estudiantes talentosos a obtener doctorados en Estados Unidos. Por esto se propone el desarrollo de una campaña agresiva para el reclutamiento de estos estudiantes. Mediante esta campaña se seleccionará un número no inferior a diez estudiantes. Dichos estudiantes serán subvencionados para hacer estudios doctorales en Ciencias de Computación o Ingeniería en Computación. Los estudiantes se especializarán en una de las áreas indicadas en la tabla 8.1.

Para que el plan de reclutamiento de estudiantes pueda tener éxito es necesario ofrecer estipendios que cubran adecuadamente el costo de realizar estudios de doctorado en Estados Unidos. Los estipendios que se proveen actualmente son inadecuados

8.2.1.2 Reclutamiento de Nueva Facultad

Para completar la facultad se reclutarán profesores con especialidades en las áreas que se indican en la tabla 8.1. A modo de incentivo, cada profesor reclutado recibirá dinero semilla para investigación durante sus primeros dos años. Este dinero semilla proveerá fondos para la adquisición de equipo, viajes de investigación, una ayudantía de investigación y compensación salarial para el período de verano. Además se le proveerá una descarga docente de seis créditos con el objeto de facilitar su trabajo de investigación.

8.2.1.4 Reclutamiento de Ayudantes de Cátedra

Además de reclutar facultad con maestría y doctorados es necesario reclutar ayudantes de cátedra para asistir al profesorado en la enseñanza de los cursos, particularmente en las sesiones de laboratorio. Se estima que será necesario reclutar un estudiante de cátedra por cada cuatro secciones. Tomando como base una matrícula de 100 estudiantes de nuevo ingreso por año se requerirá un total de 10 ayudantes de cátedra al cabo de cinco años. Estos ayudantes serán reclutados de los programas graduados de ciencias e ingeniería de computadoras.

8.2.2 Facultad para Ofrecer los Cursos de Otras Disciplinas

El reclutamiento de esta facultad se llevará a cabo por los respectivos departamentos. Para este propósito se incluye una partida presupuestaria como parte del presupuesto recurrente del programa.

8.3 Plan para el Adiestramiento de la Facultad

La facultad del programa se reclutará debidamente adiestrada. Sin embargo, se espera que el profesorado continúe manteniéndose al día en su campo asistiendo a conferencias y participando en los diferentes talleres de capacitación en métodos y estrategias de enseñanza efectiva que se ofrecen en el RUM.

9. RECURSOS DEL APRENDIZAJE

9.1 Inventario de Recursos Existentes

La Facultad de Ingeniería tiene a su disposición la Biblioteca General del Recinto Universitario de Mayagüez. Esta biblioteca dispone de una colección de libros que puede servir para comenzar el programa propuesto. Hemos determinado, sin embargo que es necesario presentar un plan de mejoramiento bibliotecario con el propósito de asegurar que la biblioteca general provea la ayuda necesaria al estudiantado de este programa durante los próximos cinco años.

Especial atención se le dará a todo el sistema informático con que la biblioteca general dispone para la comunicación dentro del Recinto Universitario de Mayagüez como fuera de este de tal manera que este sistema informático provea al usuario(a) métodos eficientes para el acceso a los repositorios electrónicos con que cuenta dicha biblioteca. Pensamos que la biblioteca general del

Recinto Universitario de Mayagüez debe convertirse en el repositorio y centro de comunicación de información electrónica.

9.2 Plan de Mejoramiento de Recursos Disponibles

El siguiente plan de cinco años está destinado a mejorar el estado actual de la Biblioteca General del RUM, con el objetivo de atender las necesidades inmediatas del programa propuesto. El plan consiste en cuatro partes principales: obtención de libros, obtención de materiales audiovisuales, obtención de acceso electrónico a documentación y base de datos.

9.2.1 Obtención de Libros

Al comienzo de cada semestre se solicitará a la Biblioteca General información sobre la disponibilidad de fondos para la adquisición de libros. Se enviará a la Biblioteca General la lista de libros que desea adquirir para el próximo semestre. La preparación de cada lista será coordinada con el representante de la biblioteca ante la facultad de ingeniería.

9.2.2 Obtención de Materiales Audiovisuales

Al principio de cada semestre se enviará a la biblioteca general una lista de recursos audiovisuales a adquirir para el próximo semestre. La preparación de cada lista será coordinada con el representante de la biblioteca ante la facultad.

9.2.3 Obtención de Acceso Electrónico a Base de Datos

La Biblioteca General del RUM está desarrollando infraestructura destinada a facilitar el acceso electrónico a documentación y bases de datos a través de la red electrónica de comunicación de nuestro recinto. Se evaluará periódicamente dicha infraestructura con el objeto de asegurar que posea los requisitos mínimos para sostener un ambiente activo de acceso a la biblioteca.

9.2.4 Uso de Otras Bibliotecas

Actualmente la Biblioteca General cuenta con un programa de préstamos interbibliotecarios. Además, mediante Internet y servicios como el "World Wide Web" se tiene acceso a un caudal de publicaciones e información científica.

10. INSTALACIONES FÍSICAS Y EQUIPO

10.1 Inventario de Facilidades Disponibles

Actualmente no existen facilidades físicas disponibles para ofrecer el programa que se propone. Esto hace indispensable la construcción de al menos 20,400 pies cuadrados, distribuidos de la siguiente manera:

Cantidad	Tipo de Espacio	Area
8	salones de clases	4,000
3	laboratorios	6,000
30	oficinas de profesores	3,000
10	oficinas de ayudantes de cátedra	1,000
	oficinas administrativas	2,000
	Salas de estudio y organizaciones estudiantiles	2,000
4	baños	600
3	salas de reuniones	1,800
	Total	20,400

Las facilidades propuestas deberán tener iluminación apropiada y sistemas de acondicionamiento de línea y UPS para todos los tomacorrientes eléctricos destinados a sistemas de comunicación de redes y sistemas de computación. Además, se instalarán cables y equipos de comunicación de redes digitales en todas las facilidades. En los laboratorios se instalarán cerraduras electrónicas. Las facilidades se habilitarán apropiadamente con mobiliario de oficina o equipos de computación según sea el caso. Los salones de clase contarán con un equipo de proyección de computadoras.

Además de las facilidades físicas antes descritas se propone la construcción de 25 espacios de estacionamiento.

10.2 Impacto del Programa Sobre las Instalaciones Físicas Existentes

Puesto que se deberán construir nuevas facilidades físicas, el programa propuesto no afecta las instalaciones existentes en el RUM.

10.3 Necesidad y Disponibilidad de Servicios de Cómputos para el Nuevo Programa

10.3.1 Laboratorios

Para ofrecer el programa se propone el establecimiento de tres laboratorios. Estos laboratorios se describen a continuación.

Laboratorio de Computación - Este laboratorio servirá para apoyar los cursos de fundamentos de computación. Este laboratorio contará con 60 estaciones de trabajo tipo PC, dos servidores PC, una impresora láser de alta velocidad y un sistema de proyección de computadoras del alta iluminación. Las estaciones de trabajo al igual que los servidores PC operarán con el sistema operativo Linux. El software necesario consistirá de compiladores de lenguajes como C, C++, Java, etc.

Laboratorio de Software - Este laboratorio servirá para apoyar los cursos de que requieren desarrollo de software. El laboratorio contará con 100 estaciones de trabajo tipo PC, dos servidores tipo PC, un servidor de alto rendimiento tipo Unix, una impresora láser de alta velocidad y un sistema de proyección de computadoras del alta iluminación. Las estaciones de trabajo al igual que los servidores PC operarán con el sistema Windows 2010. El servidor de alto rendimiento operará con el sistema operativo Unix. El software necesario consistirá de programas para desarrollar bases de datos como Oracle y Access, programas para desarrollar interfaz de usuario(a) como Delphi, JBuilder y Visual Basic, y programas para desarrollar aplicaciones de Web como Front Page y Page Mill.

Laboratorio de Ingeniería de Software - Este laboratorio servirá para apoyar los cursos de ingeniería de software. El laboratorio contará con 40 estaciones de trabajo tipo PC, un servidor PC, un servidor de alto rendimiento tipo Unix, una impresora láser de alta velocidad y un sistema de proyección de computadoras del alta iluminación. Las estaciones de trabajo al igual que el servidor PC operarán con el sistema Windows Millenium. El servidor de alto rendimiento operará con el sistema operativo Unix. El software necesario consistirá de programas de administración de proyectos y diseño de programas.

10.3.2 Personal Docente y Administrativo

Además de los sistemas de computación antes mencionados se requieren 5 computadoras tipo PC para el personal administrativo, 26 estaciones de trabajo (PC o Macintosh) para la facultad y 10 estaciones de trabajo tipo PC para ayudantes de cátedra. Todas estas estaciones de trabajo deberán tener un micrófono y una mini cámara para facilitar videoconferencia y un sistema de interacción verbal. El software mínimo para estas estaciones consistirá de un paquete de productividad de oficina como Office 2010, un “browser” como Netscape, un programa de correo electrónico, un convertidor de formatos PDF (Acrobat) y un sistema de videoconferencia.

11. ACREDITACIÓN Y LICENCIA DEL PROGRAMA

11.1 Acreditación Profesional

Como todos los programas de ingeniería del RUM es deseable que los dos nuevos bachilleratos sean eventualmente acreditado por la “Accreditation Board of Engineering and Technology” (ABET). Una vez graduada la primera clase del programa se solicitará una visita de acreditación a esta agencia.

11.2 Licenciamiento por el CES

Una vez aprobado por los cuerpos correspondientes de la Universidad de Puerto Rico el programa que se propone, requiere la aprobación del Consejo de Educación superior.

12. ADMINISTRACIÓN DEL NUEVO PROGRAMA

El programa será inicialmente administrado por el Departamento de Ingeniería Eléctrica y Computadoras del Recinto Universitario de Mayagüez. Para estos efectos se propone la creación del cargo académico administrativo de Director Asociado para el Área de Computación. Dicho director asociado tendrá bajo su responsabilidad:

- ?? currículum,
- ?? desarrollo de infraestructura,
- ?? planificación,
- ?? matrícula de estudiantes y reclutamiento de personal

Este director asociado debe ser nombrado antes del comienzo oficial del programa.

El programa requerirá además de la contratación del siguiente personal de apoyo:

- un(a) secretario(a)
- un(a) oficial administrativo
- un(a) consejero(a) académico

13. AYUDA ECONÓMICA PARA LOS ESTUDIANTES

Las becas y préstamos usuales otorgados por agencias del gobierno de Puerto Rico y del gobierno federal, están a disposición de estudiantes cualificados. Aquellos estudiantes de rendimiento académico sobresaliente tendrán matrícula de honor y con ello, la eximición del pago de la misma.

14. PRESUPUESTO

En la tabla 14.1 se presenta el presupuesto recurrente estimado para los primeros 5 años del programa. El presupuesto anual a partir del sexto año será similar al del quinto año con los debidos ajustes salariales. En la tabla 14.2 se presenta el presupuesto no recurrente del programa para los primeros cinco años. En la tabla 14.3 se indican los criterios utilizados para calcular los diferentes renglones del presupuesto.

Tabla 14.1 Presupuesto Recurrente del Programa Propuesto Durante los Primeros Cinco Años.

Renglón	Año				
	1	2	3	4	5
Facultad de la disciplina	161,903	442,254	1,187,750	1,582,226	1,706,476
Facultad de otras disciplinas	224,596	419,247	486,626	598,924	733,682
Ayudantes de cátedra	7,200	18,000	46,800	64,800	72,000
Personal administrativo	78,174	82,083	86,187	90,496	95,021
Administrador(a) de sistemas	55,170	57,929	121,650	191,599	201,178
Estaciones de trabajo (workstation)	171,000	162,000	301,000	298,000	301,000
Servidores	4,000	85,000	8,000	58,000	85,000
Periferales	20,000	20,000	20,000	20,000	20,000
Software	77,000	99,000	137,000	161,000	162,000
Recursos bibliotecarios	5,000	5,000	5,000	5,000	5,000
Materiales didácticos	20,000	20,000	20,000	20,000	20,000
Materiales de oficina	10,000	15,000	20,000	25,000	25,000
Total	834,044	1,425,512	2,440,012	3,115,045	3,426,357

Tabla 14.2 Presupuesto No Recurrente del Programa

Renglón	Año				
	1	2	3	4	5
Construcción de Facilidades	2,264,400				
Estacionamientos	200,000				
Equipo de oficina	35,000				
Equipo de laboratorio	51,000	51,000	61,000	10,000	
Dinero semilla	50,000	150,000	250,000	350,000	250,000
Total	2,600,400	201,000	311,000	360,000	250,000

Tabla 14.3 Criterios para determinar el presupuesto

Renglón	Criterio
Facultad de la disciplina	Salario y beneficios marginales según escala de ingeniería. La facultad se reclutará según se indica en la tabla 8.1.
Facultad de otras disciplinas	Salario y beneficios marginales según escala básica. (Se asume mitad Instructores con MS y mitad Catedráticos Auxiliares con Ph.D.). La facultad se reclutará según la proyección que se indica en la figura 8.1.
Ayudantes de cátedra	Un ayudante por cada cuatro secciones por semestre @ \$7200 por ayudante
Personal administrativo	Salario y beneficios marginales según escalas salariales para personal no docente y compensaciones para directores(as) asociados para el siguiente personal: una(o) secretaria(o) un(a) consejero(a) académico un oficial administrativo un(a) director(a) asociado Se asume un incremento salarial de un 5% anual.
Administrador(a) de sistemas	Salario y beneficios marginales para dos administradores más beneficios marginales. Se asume un incremento salarial de un 5% anual.
Estaciones de trabajo (workstations)	\$2,500 por computadora de laboratorio, \$3,000 por cada computadora para personal administrativo y facultad. Además de proveer computadoras para el personal docente y administrativo se establecerán tres laboratorios de computación con un total conjunto de 200 computadoras. Se asume que cada estación de trabajo se reemplaza al cabo de dos años.
Servidores	\$10,000 por server PC y \$75,000 por server Unix. Se adquirirá y mantendrá cuatro servidores PC y dos servidores Unix. Se asume que cada servidor se reemplaza al cabo de tres años.
Equipos periféricos	\$20,000 anuales
Software	\$1,000 por computadora, \$10,000 por servidor PC y \$25,000 por servidor Unix. Este costo se contabiliza solamente en el año en que se compra cada computadora.
Recursos bibliotecarios	\$5000 por año
Materiales didácticos	\$20,000 por año
Materiales de oficina	Incremental de \$10,000 a \$25,000 por año
Construcción de facilidades	20,400 pies cuadrados @ \$111 el pie cuadrado
Estacionamientos	25 @ \$8000 por espacio
Equipo de oficina (muebles, anaqueles, fotocopiadoras, etc.)	\$35,000
Equipos de laboratorios	\$500 por mueble de computadora, \$5000 por equipo de proyección para laboratorios y salones de clases (distribuido en cuatro años)
Dinero semilla	\$25,000 por facultad por los primeros dos años de contratación

15. INGRESOS

Además de ingresos por concepto de matrícula, el Colegio de Ingeniería frecuentemente recibe donaciones de compañías privadas para desarrollar laboratorios académicos, didácticos y de investigación. Se espera que el programa que se propone atraiga un número significativo de fondos por este concepto lo cual reduciría las la cantidad de fondos institucionales necesarios para ofrecer el programa.

Recientemente la agencia gubernamental PRIDCO aprobó una propuesta sometida por el RUM titulada: *El Recinto Universitario de Mayagüez: Eje Académico del Corredor Tecnológico del Oeste*. Mediante esta dádiva PRIDCO contribuirá al programa de Bachillerato en Ingeniería de Software \$3,011,868 (a modo de pareo) en un periodo de cinco años. En adición, la dádiva de PRIDCO incluye una partida de \$3,000,0000 (a modo de pareo) para la construcción de facilidades físicas para ofrecer los programas que fueron incluidos en la propuesta original. Al presente no se ha determinado que porción de esos fondos corresponderá al programa de ingeniería de software pero se espera que sea alrededor de una cuarta parte ya que la propuesta a PRIDCO incluía cuatro programas académicos.

La distribución por año de los fondos de esta dádiva para el programa de ingeniería de software se indica en la tabla 15.1. La dádiva de PRIDCO tendrá el efecto de reducir la contribución requerida por la UPR al presupuesto del programa indicado en la sección 14.

Tabla 15.1 Distribución de Fondos de la Dádiva de PRIDCO para el Programa de Bachillerato de Ingeniería de Software

Año				
1	2	3	4	5
359,862	512,875	532,869	764,539	841,723

16. EVALUACION

Para determinar el grado de éxito del programa se realizarán una evaluación del mismo al cabo de 6 años. En esta evaluación se utilizaran los siguientes indicadores:

Indicador	Expectativas
Admisiones	100 estudiantes por año con índices de ingreso superior al 95% de los/las estudiantes admitidos/as al RUM
Retención	90% de retención
Graduandos	90% de los/las estudiantes originalmente admitidos/as graduándose en cinco años
Empleos	85% de los/las egresados/as con ofertas de trabajo en campos afines con la ingeniería de software
Estudios Graduados	15% de los/las egresados/as admitidos/as a universidades reconocidas para proseguir estudios graduados
Acreditación	El programa deberá estar acreditado al cabo del sexto año.
Donativos	\$500,000 en el periodo de 6 años

Apéndice A. Definición de Ciencia de Computación según CAC/ABET

Computer science is a discipline that involves the understanding and design of computers and computational processes. In its most general form it is concerned with the understanding of information transfer and transformation. Particular interest is placed on making processes efficient and endowing them with some form of intelligence. The discipline ranges from theoretical studies of algorithms to practical problems of implementation in terms of computational hardware and software. A central focus is on processes for handling and manipulating information. Thus, the discipline spans both advancing the fundamental understanding of algorithms and information processes in general as well as the practical design of efficient reliable software and hardware to meet given specifications. Computer science is a young discipline that is evolving rapidly from its beginnings in the 1940's. As such it includes theoretical studies, experimental methods, and engineering design all in one discipline. This differs radically from most physical sciences that separate the understanding and advancement of the science from the applications of the science in fields of engineering design and implementation. In computer science there is an inherent intermingling of the theoretical concepts of computability and algorithmic efficiency with the modern practical advancements in electronics that continue to stimulate advances in the discipline. It is this close interaction of the theoretical and design aspects of the field that binds them together into a single discipline.

Because of the rapid evolution it is difficult to provide a complete list of computer science areas. Yet it is clear that some of the crucial areas are theory, algorithms and data structures, programming methodology and languages, and computer elements and architecture. Other areas include software engineering, artificial intelligence, computer networking and communication, database systems, parallel computation, distributed computation, computer-human interaction, computer graphics, operating systems, and numerical and symbolic computation.

A professional computer scientist must have a firm foundation in the crucial areas of the field and will most likely have an in-depth knowledge in one or more of the other areas of the discipline, depending upon the person's particular area of practice. Thus, a well educated computer scientist should be able to apply the fundamental concepts and techniques of computation, algorithms, and computer design to a specific design problem. The work includes detailing of specifications, analysis of the problem, and provides a design that functions as desired, has satisfactory performance, is reliable and maintainable, and meets desired cost criteria. Clearly, the computer scientist must not only have sufficient training in the computer science areas to be able to accomplish such tasks, but must also have a firm understanding in areas of mathematics and science, as well as a broad education in liberal studies to provide a basis for understanding the societal implications of the work being performed.

Apéndice B. PRONTUARIOS DE LOS CURSOS NUEVOS

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4isc
Course Title: Scientific Computing
Credit-Hours: 3

2. Course Description:

Fundamental concepts of scientific computing including floating-point numerical systems, error analysis, stability and convergence and numerical and symbolic methods in science and engineering.

3. Pre-requisites:

CCOM4ada Analysis and design of Algorithms.

4. Textbook, Supplies and Other Resources:

C. Van Loan, Introduction to Scientific Computing, Second Edition, Prentice Hall, 2000

G. Farin, Curves and Surfaces for Computer Aided Geometry Design, Academic Press, 1992.

5. Purpose:

Undergraduate computer science core course. The purpose is to provide students with underlying concepts of scientific computing.

6. Course Goals:

At the end of this course, students will have the following skills:

1. Knowledge of mathematical and software principles of scientific computing.
2. Ability to critique and judge different methods applied in science and engineering.
3. Ability to apply knowledge of error analysis and convergence in the solution of problems.
4. Ability to identify, formulate and solve scientific computing problems.
5. Understanding of the effect of limited precision in computing and nature of the different sources of error.
6. Knowledge of numerical solution for linear and nonlinear problems
7. Understanding of dynamic problems and adaptive methods
8. Understanding of computational geometry concepts in computing and their application in science and engineering.
9. Knowledge of visualization techniques

7. Requirements:

Students are expected to have a working knowledge in a high level language, data structures and analysis of algorithms.

8. Laboratory/Field Work (If applicable):

Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog for further information.

10. General Topics

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
Sources of error Round-off error and computer arithmetic, ill-conditioned problems and unstable algorithms, truncation error, extrapolation and convergence analysis, differentiation, interpolation, integration.	1-5	Lectures Discussion Homework Collaborative learning
Algebraic equations Solving linear systems of equations: condition number, special problems (symmetric, positive definite, least squares, sparse), matrix factorizations, eigenvalues, software packages. Solving nonlinear equations: Newton's method, robust optimization, line search, descent directions, iterative methods.	1-4,6	Lectures Discussion Homework Collaborative learning
Dynamics Time stepping methods, Adams and Runge Kutta methods, error analysis and modified equations. conservations principles, adaptive methods, convergence studies, application to computational fluid dynamics.	1-4, 7	Lectures Discussion Homework Collaborative learning
Computational Geometry	1-4, 8	Lectures

<p>Basic theory of curves and surfaces, interpolation and approximation techniques, theory of splines, B-splines, NURBS, Computer Aided geometric Design (CAGD), application to science and engineering problems (Grid generation, visualization and solution of nonlinear PDEs).</p>		<p>Discussion Homework Collaborative learning</p>
<p>Graphics and Scientific visualization</p>	<p>1-4, 9</p>	<p>Lectures Discussion Homework Collaborative learning</p>

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4tdc
Course Title: Theory of Computation
Credit-Hours: 3

2. Course Description:

Discussion of various aspects in theory of computation including basic mathematical models of computation, finite automata and regular languages, context-free languages, Turing machines, Church's thesis, reducibility and completeness, time complexity and NP-completeness, space complexity, and probabilistic computation.

3. Pre-requisites:

ICOM 4035 Data Structures

4. Textbook, Supplies and Other Resources:

M. Sipser, Introduction to the Theory of Computation, PWS Publishing, 1997

D. Kozen: Automata and Computability. Springer-Verlag, New York. 1997.

J. E. Hopcroft, and J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.

M. A. Harrison, Introduction to Formal Language Theory, Addison-Wesley, 1978.

H. R. Lewis, and C. H. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1998.

5. Purpose:

Undergraduate computer science core course. The purpose is to provide students with underlying concepts of the theory of computation.

6. Course Goals:

At the end of this course, students will have the following skills:

1. Ability to understand mathematical definitions and proofs at a high level.
2. Knowledge of modular arithmetic, probability, and asymptotic notation and terminology.
3. Ability to apply knowledge of theory of computing to categorize problems in science and engineering.
4. Knowledge and practical understanding of time and space complexity.
5. Ability to critique and judge different methods from the computability point of view.
6. oral and written communication skills.

7. Requirements:

Students are expected to have a working knowledge in a high level language and data structures.

8. Laboratory/Field Work (If applicable):

Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog for further information.

10. General Topics

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
Mathematical background Sets, sequences and tuples, functions and relations, graphs, strings and languages, boolean logic. Proof by construction, proof by contradiction, proof by induction	1-6	Lectures Discussion Homework Collaborative learning

<p>Regular Languages Formal definition of a finite automaton, examples, formal definition of computation, designing finite automata, regular operations, nondeterminism, closure under the regular operations, formal definition of a regular expression, equivalence with finite automata, the pumping lemma for regular languages.</p>	1-6	Lectures Discussion Homework Collaborative learning
<p>Context-Free Languages Formal definition of a context-free grammar, examples, designing context-free grammars, ambiguity, Chomsky normal form, Formal definition of a pushdown automaton, examples, equivalence with context-free grammars, the pumping lemma for context-free languages</p>	1-6	Lectures Discussion Homework Collaborative learning
<p>The Church-Turing Thesis Formal definition and examples of Turing machines, multitape Turing machines, nondeterministic Turing machines, enumerators, equivalence with other models, Hilbert's problems.</p>	1-6	Lectures Discussion Homework Collaborative learning
<p>Decidability Decidable languages, The Halting problem, undecidable problems from language theory, mapping reducibility.</p>	1-6	Lectures Discussion Homework Collaborative learning
<p>Time Complexity Big-O and small-o notation, analyzing algorithms, complexity relationships among models, The class P, polynomial time, the class NP, NP Completeness, polynomial time reducibility, the Cook-Levin Theorem, the vertex cover problem, the Hamiltonian path problem, the subset sum problem</p>	1-6	Lectures Discussion Homework Collaborative learning
<p>Space Complexity Savitch's theorem, Pspace Completeness, the TQBF problem - Winning strategies for games, generalized geography,</p>	1-6	Lectures Discussion Homework Collaborative learning

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM3aci
Course Title: Computer Architecture I
Credit-Hours: 3

2. Course Description:

Fundamental concepts of logic circuits analysis and design; boolean algebra and logic gates, combinational and sequential circuits, arithmetic logic units (ALU), memory and programmable logic devices, register transfers and data paths, control units design.

3. Pre/Co-requisites:

INEL4075 – Introduction to Electric Circuits

4. Textbook, Supplies and Other Resources:

M. Morris Mano & Charles R. Kime, “Logic and Computer Design Fundamentals” 2nd. Edition, Prentice Hall, 2000.

5. Purpose:

This is a core course for majors in computing science and software engineering. It introduces fundamental concepts of logic circuits analysis and design.

6. Course Goals:

After completing the course, the student should be able to:

- a- have a knowledge of Boolean algebra
- b- analyze and design combinational circuits
- c- analyze and design sequential circuits
- d- design logic circuits using programmable logic devices
- e- design datapaths and control units

A. COURSE SYLLABUS

7. Requirements:

All students are expected to :

Do all assigned readings and related homework.

Come to class regularly and on time.

Take all the exams and complete all the projects to receive credit for the course.

8. Laboratory/Field Work (If applicable):

In order to do some of the homework and projects the students will use logic design tools.

These tools will be available at the Computing Laboratory.

9. Department/Campus Policies:

9a. Class attendance: Class attendance is compulsory. The University of Puerto Rico, Mayagüez Campus, reserves the right to deal at any time with individual cases of non-attendance. Professors are expected to record the absences of their students. Frequent absences affect the final grade, and may even result in total loss of credits. Arranging to make up work missed because of legitimate class absence is the responsibility of the student. (Bulletin of Information Undergraduate Studies, pp 39 1995-96)

9b. Absence from examinations: Students are required to attend all examinations. If a student is absent from an examination for a justifiable reason acceptable to the professor, he or she will be given a special examination. Otherwise, he or she will receive a grade of zero of "F" in the examination missed. (Bulletin of Information Undergraduate Studies, pp 39, 1995-96)

9c. Final examinations: Final written examinations must be given in all courses unless, in the judgment of the Dean, the nature of the subject makes it impracticable. Final examinations scheduled by arrangements must be given during the examination period prescribed in the Academic Calendar, including Saturdays. (see Bulletin of Information Undergraduate Studies, pp 39, 1995-96).

9d. Partial withdrawals: A student may withdraw from individual courses at any time during the term, but before the deadline established in the University Academic Calendar. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9e. Complete withdrawals: A student may completely withdraw from the University of Puerto Rico, Mayagüez Campus, at any time up to the last day of classes. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9f. Disabilities: All the reasonable accommodations according to the Americans with Disability Act (ADA) Law will be coordinated with the Dean of Students and in accordance with the particular needs of the student.

9g. Ethics: Any academic fraud is subject to the disciplinary sanctions described in article 14 and 16 of the revised General Student Bylaws of the University of Puerto Rico contained in Certification 018-1997-98 of the Board of Trustees. The professor will follow the norms established in articles 1-5 of the Bylaws.

10. General Topics:

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
1. Combinational logic circuits analysis <i>boolean algebra, logic gates</i>	Learn to analyze combinational logic circuits with boolean algebra and logic gates.	Lectures, homework
2. Combinational logic design <i>truth tables, Karnaugh Maps, sum of products, products of sums, tri-state buffers, code converters, decoders, encoders, multiplexers, binary adders and subtractors, binary multipliers, time delays</i>	Learn to design combinational logic circuits using truth tables and Karnaugh Maps	Lectures, discussion, homework
3. Sequential circuits analysis <i>latches, level trigger flip-flops, edge trigger flip-flops, delay times</i>	Learn to analyze sequential circuits based on latches and flip-flops. Understand the effect of physical aspects such as delay times.	Lectures, discussion, homework
4. Sequential circuits design <i>state diagrams, excitation tables, registers, shift register, counters</i>	Learn to develop state diagrams, excitation tables. Learn the methodology to design sequential circuits	Lectures, discussion, homework
5. Memory and programmable logic devices <i>Static and dynamic random access memories (RAM), read-only memory (ROM), programmable logic arrays (PLA), programmable array logic (PAL), VLSI programmable logic devices</i>	Understand the operation of memory devices and programmable logic devices. Learn to use commercial programmable logics devices in the design of logic circuits	Lectures, discussion, homework, laboratory simulations
6. Register transfer and datapaths <i>microoperations, multiplexer-based transfer, bus-based transfer, datapaths, arithmetic logic units (ALU), shifters, control words</i>	Understand concepts of register transfer. Learn to design datapaths using registers, multiplexers and tri-state buffers	Lectures, discussion, homework, laboratory simulations
7. Control unit design <i>Sequencing, algorithmic state machine charts (ASM), hardwire control, microprogram control, pipeline control</i>	Learn to design control units for datapaths	Lectures, discussion, design project, teamwork, laboratory simulations

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM3acii
Course Title: Computer Architecture II
Credit-Hours: 3

2. Course Description:

Fundamentals of instruction sets: operations, registers, instruction formats, data types, addressing modes, branches, subroutines, interrupts, input/output, memory access; machine language programming; assembly language programming; memory hierarchy: caches, virtual memory, secondary storage; pipelining; input/output system; computer arithmetic; architectural support for operating systems.

3. Pre/Co-requisites:

CCOM3aci - Computer Architecture I

4. Textbook, Supplies and Other Resources:

William Stallings, Computer Organization and Architecture, (fifth edition), Prentice Hall, 2000.

5. Purpose:

This is a core course for majors in computing science and software engineering. It introduces fundamental concepts of computer architecture.

A. COURSE SYLLABUS

6. Course Goals:

After completing the course, the student should be able to:

- a- understand the fundamentals of instruction sets
- b- code programs at the machine language level
- c- write programs using assembly language
- d- understand the operation of cache memories, virtual memory, pipelining, and input/output system
- e- learn computer arithmetic algorithms and integer and floating point numbers representation
- f- understand how operating systems are supported by the architecture

7. Requirements:

All students are expected to :

Do all assigned readings and related homework.

Come to class regularly and on time.

Take all the exams and complete all the projects to receive credit for the course.

8. Laboratory/Field Work (If applicable):

In order to do some of the homework and projects the students will need to write programs in machine language and assembly language. The software needed to generate these programs will be available at the Computing Laboratory.

A. COURSE SYLLABUS

9. Department/Campus Policies:

9a. Class attendance: Class attendance is compulsory. The University of Puerto Rico, Mayagüez Campus, reserves the right to deal at any time with individual cases of non-attendance. Professors are expected to record the absences of their students. Frequent absences affect the final grade, and may even result in total loss of credits. Arranging to make up work missed because of legitimate class absence is the responsibility of the student. (Bulletin of Information Undergraduate Studies, pp 39 1995-96)

9b. Absence from examinations: Students are required to attend all examinations. If a student is absent from an examination for a justifiable reason acceptable to the professor, he or she will be given a special examination. Otherwise, he or she will receive a grade of zero of "F" in the examination missed. (Bulletin of Information Undergraduate Studies, pp 39, 1995-96)

9c. Final examinations: Final written examinations must be given in all courses unless, in the judgment of the Dean, the nature of the subject makes it impracticable. Final examinations scheduled by arrangements must be given during the examination period prescribed in the Academic Calendar, including Saturdays. (see Bulletin of Information Undergraduate Studies, pp 39, 1995-96).

9d. Partial withdrawals: A student may withdraw from individual courses at any time during the term, but before the deadline established in the University Academic Calendar. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9e. Complete withdrawals: A student may completely withdraw from the University of Puerto Rico, Mayagüez Campus, at any time up to the last day of classes. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9f. Disabilities: All the reasonable accommodations according to the Americans with Disability Act (ADA) Law will be coordinated with the Dean of Students and in accordance with the particular needs of the student.

9g. Ethics: Any academic fraud is subject to the disciplinary sanctions described in article 14 and 16 of the revised General Student Bylaws of the University of Puerto Rico contained in Certification 018-1997-98 of the Board of Trustees. The professor will follow the norms established in articles 1-5 of the Bylaws.

10. General Topics:

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
1. Instruction sets <i>operations, registers, instruction formats, data types, addressing modes, branches, subroutines, interrupts, input/output, memory access</i>	Understand the fundamentals of instruction sets.	Lectures, discussion
2. Machine code and	Learn to code programs in machine	Lectures, discussion,

assembly language programming	language Learn to write programs using an assembler	programming homework
3. Memory hierarchy <i>caches, virtual memory, secondary storage units</i>	Understand the organization of the memory hierarchy. Understand the operation of caches and the virtual memory system	Lectures, discussion, simulation homework
4. Pipelining <i>stages, interlocks</i>	Understand the concept of pipelined instruction execution	Lectures, discussion
5. Input/Output system <i>I/O modules, programmed I/O, interrupt-driven I/O, direct memory access (DMA), I/O channels</i>	Understand the operation of input/output system	Lectures, discussion
6. Computer arithmetic <i>integer addition and subtraction, multiplication and division, floating point number representation and operations</i>	Learn algorithms for implementing integer arithmetic Learn the IEEE 754 floating point number representation and operations	Lectures, discussion, homework
7. Architectural support for operating systems <i>scheduling, memory management, interrupts and exception handling</i>	Understand how the architecture provides support for the implementation of operating systems	Lectures, discussion

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4ttt

Course Title: Introduction to Human-Computer Interaction

Credit-Hours: 3

2. Course Description:

Introduction to the principles of human-computer interaction; cognitive psychology, human factors, interaction styles. Design and implementation of graphical user interfaces (GUI); task analysis, GUI development tools, usability evaluation methods.

3. Pre/Co-requisites:

ICOM 4035 – Data Structures

4. Textbook, Supplies and Other Resources:

Dix, A., Finlay, J., Abowd, G., and Russell, B., Human-Computer Interaction, 2nd edition, Prentice Hall, 1998.

5. Purpose:

This is a core course for majors in software engineering. It introduces fundamental concepts of human-computer interaction.

6. Course Goals:

After completing the course, the student should be able to:

- a- understand the effect of cognitive psychology and human factors in human-computer interaction
- b- understand the advantages and disadvantages of different interaction styles
- c- conduct a task analysis
- d- design user-centered GUIs
- e- learn to use user interface development tools effectively
- f- learn and apply usability engineering evaluation methods

A. COURSE SYLLABUS

7. Requirements:

All students are expected to :

Do all assigned readings and related homework.

Come to class regularly and on time.

Take all the exams and complete all the projects to receive credit for the course.

8. Laboratory/Field Work (If applicable):

In order to do some of the homework and projects the students will use user interface design tools. These tools will be available at the Software Laboratory.

9. Department/Campus Policies:

9a. Class attendance: Class attendance is compulsory. The University of Puerto Rico, Mayagüez Campus, reserves the right to deal at any time with individual cases of non-attendance. Professors are expected to record the absences of their students. Frequent absences affect the final grade, and may even result in total loss of credits. Arranging to make up work missed because of legitimate class absence is the responsibility of the student. (Bulletin of Information Undergraduate Studies, pp 39 1995-96)

9b. Absence from examinations: Students are required to attend all examinations. If a student is absent from an examination for a justifiable reason acceptable to the professor, he or she will be given a special examination. Otherwise, he or she will receive a grade of zero of "F" in the examination missed. (Bulletin of Information Undergraduate Studies, pp 39, 1995-96)

9c. Final examinations: Final written examinations must be given in all courses unless, in the judgment of the Dean, the nature of the subject makes it impracticable. Final examinations scheduled by arrangements must be given during the examination period prescribed in the Academic Calendar, including Saturdays. (see Bulletin of Information Undergraduate Studies, pp 39, 1995-96).

9d. Partial withdrawals: A student may withdraw from individual courses at any time during the term, but before the deadline established in the University Academic Calendar. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9e. Complete withdrawals: A student may completely withdraw from the University of Puerto Rico, Mayagüez Campus, at any time up to the last day of classes. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9f. Disabilities: All the reasonable accommodations according to the Americans with Disability Act (ADA) Law will be coordinated with the Dean of Students and in accordance with the particular needs of the student.

9g. Ethics: Any academic fraud is subject to the disciplinary sanctions described in article 14 and 16 of the revised General Student Bylaws of the University of Puerto Rico contained in Certification 018-1997-98 of the Board of Trustees. The professor will follow the norms established in articles 1-5 of the Bylaws.

10. General Topics:

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
1. Cognitive psychology <i>input-output channels, memory, reasoning and problem solving</i>	Understand the effect of cognitive psychology in human-computer interaction	Lectures, discussion, homework
2. Human factors <i>visual perception, hearing, touch, movement</i>	Understand the effect of cognitive human factors in human-computer interaction	Lectures, discussion, homework
3. Interaction styles <i>Command-based, menu-driven, WIMP, Web-based (hypertext-based)</i>	Understand the advantages and disadvantages of different interaction styles	Lectures, discussion, homework
4. Task Analysis <i>task decomposition, data collection, entity-relation-based techniques, knowledge-based analysis</i>	Learn how to conduct a task analysis.	Lectures, discussion, homework
5. GUI development tools <i>visual programming, widget classes, application development</i>	Learn to develop applications using GUI development tools	Lectures, discussion, programming homework, project
6. Usability evaluation methods <i>usability heuristics, user tests, think aloud</i>	Learn to apply usability engineering techniques in the evaluation of GUIs.	Lectures, discussion, project

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4tec

Course Title: Special Topics in Computing

Credit-Hours: 3

2. Course Description:

Studies in special topics in computing.

3. Pre/Co-requisites:

To be specified by the instructor.

4. Textbook, Supplies and Other Resources:

To be specified by the instructor.

5. Purpose:

This is an elective course for majors in computing and software engineering.

6. Course Goals:

To be specified by the instructor.

7. Requirements:

To be specified by the instructor.

8. Laboratory/Field Work (If applicable):

To be specified by the instructor.

A. COURSE SYLLABUS

9. Department/Campus Policies:

9a. Class attendance: Class attendance is compulsory. The University of Puerto Rico, Mayagüez Campus, reserves the right to deal at any time with individual cases of non-attendance. Professors are expected to record the absences of their students. Frequent absences affect the final grade, and may even result in total loss of credits. Arranging to make up work missed because of legitimate class absence is the responsibility of the student. (Bulletin of Information Undergraduate Studies, pp 39 1995-96)

9b. Absence from examinations: Students are required to attend all examinations. If a student is absent from an examination for a justifiable reason acceptable to the professor, he or she will be given a special examination. Otherwise, he or she will receive a grade of zero of "F" in the examination missed. (Bulletin of Information Undergraduate Studies, pp 39, 1995-96)

9c. Final examinations: Final written examinations must be given in all courses unless, in the judgment of the Dean, the nature of the subject makes it impracticable. Final examinations scheduled by arrangements must be given during the examination period prescribed in the Academic Calendar, including Saturdays. (see Bulletin of Information Undergraduate Studies, pp 39, 1995-96).

9d. Partial withdrawals: A student may withdraw from individual courses at any time during the term, but before the deadline established in the University Academic Calendar. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9e. Complete withdrawals: A student may completely withdraw from the University of Puerto Rico, Mayagüez Campus, at any time up to the last day of classes. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9f. Disabilities: All the reasonable accommodations according to the Americans with Disability Act (ADA) Law will be coordinated with the Dean of Students and in accordance with the particular needs of the student.

9g. Ethics: Any academic fraud is subject to the disciplinary sanctions described in article 14 and 16 of the revised General Student Bylaws of the University of Puerto Rico contained in Certification 018-1997-98 of the Board of Trustees. The professor will follow the norms established in articles 1-5 of the Bylaws.

10. General Topics:

To be specified by the instructor.

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4isc
Course Title: Undergraduate Research in Computing
Credit-Hours: 1 - 6

2. Course Description:

Research work in computing under the supervision of a professor.

3. Pre/Co-requisites:

To be specified by the professor supervising the student.

4. Textbook, Supplies and Other Resources:

N/A

5. Purpose:

This is an elective course for majors in computing and software engineering.

6. Course Goals:

Provide an experience in research.

7. Requirements:

To be specified by the professor supervising the student.

8. Laboratory/Field Work (If applicable):

To be specified by the professor supervising the student.

A. COURSE SYLLABUS

9. Department/Campus Policies:

9a. Class attendance: Class attendance is compulsory. The University of Puerto Rico, Mayagüez Campus, reserves the right to deal at any time with individual cases of non-attendance. Professors are expected to record the absences of their students. Frequent absences affect the final grade, and may even result in total loss of credits. Arranging to make up work missed because of legitimate class absence is the responsibility of the student. (Bulletin of Information Undergraduate Studies, pp 39 1995-96)

9b. Absence from examinations: Students are required to attend all examinations. If a student is absent from an examination for a justifiable reason acceptable to the professor, he or she will be given a special examination. Otherwise, he or she will receive a grade of zero of "F" in the examination missed. (Bulletin of Information Undergraduate Studies, pp 39, 1995-96)

9c. Final examinations: Final written examinations must be given in all courses unless, in the judgment of the Dean, the nature of the subject makes it impracticable. Final examinations scheduled by arrangements must be given during the examination period prescribed in the Academic Calendar, including Saturdays. (see Bulletin of Information Undergraduate Studies, pp 39, 1995-96).

9d. Partial withdrawals: A student may withdraw from individual courses at any time during the term, but before the deadline established in the University Academic Calendar. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9e. Complete withdrawals: A student may completely withdraw from the University of Puerto Rico, Mayagüez Campus, at any time up to the last day of classes. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9f. Disabilities: All the reasonable accommodations according to the Americans with Disability Act (ADA) Law will be coordinated with the Dean of Students and in accordance with the particular needs of the student.

9g. Ethics: Any academic fraud is subject to the disciplinary sanctions described in article 14 and 16 of the revised General Student Bylaws of the University of Puerto Rico contained in Certification 018-1997-98 of the Board of Trustees. The professor will follow the norms established in articles 1-5 of the Bylaws.

10. General Topics:

N/A

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4edc

Course Title: Discrete Structures for Computing

Credit-Hours: 3

2. Course Description:

Discussion of discrete mathematical structures frequently encountered in computer science and engineering. Topics include relationship between data and sets, proof techniques, operators and functions, basic logic and circuits, graph theory and organization of computational processes, elements of discrete probability and random events in computing.

3. Pre-requisites:

MATE 3005 Pre-calculus

4. Textbook, Supplies and Other Resources:

K. H. Rosen, "Discrete Mathematics and Its Applications," McGraw-Hill Higher Education, Fourth Edition, 1998.

R. Johnsonbaugh, Discrete Mathematics, Prentice Hall, 1997.

D. J. Velleman, How to Prove It, Cambridge University Press, 1995.

W. K. Grassmann and J-P Tremblay, Logic and Discrete Mathematics - a Computer Science Perspective, Prentice Hall, 1996.

5. Purpose:

Undergraduate computer science and software engineering core course. The purpose is to provide students with fundamentals in discrete mathematics.

6. Course Goals:

At the end of this course, students will have the following skills:

10. Understanding of fundamental concepts of discrete structures used in computer science and software engineering.
11. Good position to understand properly advance concepts in computer architecture and networking as well as computing modeling in which discrete mathematics is relevant.
12. Ability to critique and judge different methods applied in science and engineering.
13. Ability to apply knowledge of discrete mathematics in the solution of problems.
14. Ability to identify, formulate and solve discrete mathematics-related problems.
15. oral and written communication skills.

7. Requirements:

None

8. Laboratory/Field Work (If applicable):

Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog for further information.

10. General Topics

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
<i>Introduction</i>		Motivation and overview Discussion

<p>Logic and Set Theory</p> <p>Logic</p> <p>Propositional Equivalences</p> <p>Predicates and Quantifiers</p> <p>Boolean Algebra</p> <p>Sets and operations</p>	1-6	<p>Lectures</p> <p>Discussion</p> <p>Teamwork in class</p> <p>Theoretical Homework</p> <p>Computer Homework</p>
<p>Algorithms and Mathematical Reasoning</p> <p>Algorithms</p> <p>Complexity of Algorithms</p> <p>Applications of Number Theory</p> <p>Methods of Proof</p> <p>Recursive Definitions</p> <p>Recursive Algorithms</p> <p>Program Correctness</p>	1-6	<p>Lectures</p> <p>Discussion</p> <p>Teamwork in class</p> <p>Theoretical Homework</p> <p>Computer Homework</p>
<p>Counting</p> <p>Permutations and Combinations</p> <p>Discrete Probability</p> <p>Probability Theory</p> <p>Relations and Properties</p> <p>Recurrence Relations</p> <p>Inclusion-Exclusion</p>	1-6	<p>Lectures</p> <p>Discussion</p> <p>Teamwork in class</p> <p>Theoretical Homework</p> <p>Computer Homework</p>
<p>Graph Theory</p> <p>Introduction and definitions</p> <p>Graph Isomorphism</p> <p>Connectivity</p> <p>Trees cycles and paths</p> <p>Euler and Hamilton Paths</p> <p>Shortest Path Problems</p> <p>Critical path Analysis</p> <p>Graph Coloring</p>	1-6	<p>Lectures</p> <p>Discussion</p> <p>Teamwork in class</p> <p>Theoretical Homework</p> <p>Computer Homework</p>
<p>Modeling Computation</p> <p>Languages and Grammars</p> <p>Finite-State Machines</p> <p>Language Recognition</p> <p>Turing Machines</p>	1-6	<p>Lectures</p> <p>Discussion</p> <p>Teamwork in class</p> <p>Theoretical Homework</p> <p>Computer Homework</p>

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4ada
Course Title: Analysis and Design of Algorithms
Credit-Hours: 3

2. Course Description:

Formal algorithm analysis. Fundamental algorithm design techniques: greedy algorithms, divide-and-conquer, dynamic programming. Fundamental graph algorithms, Searching and sorting.

3. Pre-requisites:

ICOM 4035 Data Structures

4. Textbook, Supplies and Other Resources:

R. Sedgewick, P. Flajolet, "An Introduction to the Analysis of Algorithms", Addison-Wesley, 1996.

D. Kozen. The Design and Analysis of Algorithms. Springer, 1992.

T. Cormen, C. Leiserson, R. Rivest. Introduction to Algorithms. McGraw-Hill, 1990.

A. Aho, J. Hopcroft, J. Ullman. The Design and Analysis of Computer Algorithms. Addison-Wesley, 1974.

5. Purpose:

Undergraduate computer science and software engineering core course. The purpose is to provide students with fundamentals in analysis and design of algorithms.

6. Course Goals:

At the end of this course, students will have the following skills:

1. Understanding of basic analysis techniques: Approximating functions asymptotically, bounding sums, and solving recurrences.
2. Knowledge of basic design techniques: Divide-and-conquer, dynamic programming, and greed.
3. Knowledge of advanced data structures and basic graph algorithms: Amortized analysis, Fibonacci heaps, and disjoint sets, strongly-connected components, minimum spanning trees, shortest paths, and maximum flow.
4. Ability to deal with intractability: Reductions, approximation algorithms, and branch-and-bound.
5. Ability to synthesize and design new algorithms for science and engineering problems.
6. Ability to critique and judge different algorithms applied in science and engineering.
7. Ability to apply knowledge of analysis and design of algorithms in the implementation of problem solutions.
8. oral and written communication skills.

7. Requirements:

Students are expected to have a working knowledge in a high level language and data structures.

8. Laboratory/Field Work (If applicable):

Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog for further information.

10. General Topics:

TOPIC	SKILLS/GOALS	TEACHING/LEARNING STRATEGIES
Mathematical background Illustration of algorithm design and analysis. Analysis techniques: approximating functions asymptotically, bounding sums, solving recurrences.	1,5,7,8	Lectures Discussion Homework laboratory
Design techniques Divide and conquer Dynamic programming Greed Huffman coding.	2,5-8	Lectures Discussion Homework laboratory

<p>Search structures heaps, balanced trees, hash tables</p>	3, 5-8	Lectures Discussion Homework laboratory
<p>Amortized data structures Amortized analysis: aggregate, charging, and potential methods. Fibonacci heaps: maintaining a heap under merges and promotions of elements. Disjoint sets: maintaining a partition under unions.</p>	3, 5-8	Lectures Discussion Homework laboratory
<p>Graph algorithms Fundamental algorithms: depth- and breadth-first search, topological sorting, strongly-connected components. Minimum spanning trees: Prim's and Kruskal's algorithms. Single-source shortest paths: Dijkstra's and Bellman-Ford's algorithms. All-pairs shortest paths: Johnson's algorithm. Maximum flow: Goldberg and Tarjan's algorithm.</p>	3, 5-8	Lectures Discussion Homework laboratory
<p>Dealing with intractability Reductions: NP- completeness and polynomial-time reducibility. Approximation algorithms: vertex cover, traveling salesman, set cover, and subset sum. Branch and bound : maximum cut.</p>	4-8	Lectures Discussion Homework laboratory

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4aaa

Course Title: Software Engineering Project Management

Credit-Hours: 3

2. Course Description:

Estimation, planning, monitoring, documentation, evaluation, refinement and quality control in software engineering projects. Personnel management. Personal Software Process (PSP). Team Software Process (TSP). Capability Maturity Model (CMM). Risk analysis. Critical Path Method (CPM) and PERT diagrams. Project management tools.

3. Pre-requisites: INSO4sss.

4. Textbook, Supplies and Other Resources:

Bob Hughes and Mike Cotterrel, Software Project Management, International Thompson Press, 1995

Watts S. Humphrey, Introduction to the Personal Software Process, Addison-Wesley, 1996

Watts S. Humphrey, Introduction to the Team Software Process, Addison-Wesley, 1999

Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 5th Edition, 2001.

5. Purpose:

Technical elective course for software engineering and computer science students. The purpose is to prepare students with software engineering project management skills.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

1. Knowledge of project management principles and techniques
2. Evaluation of a software development project
3. Selection of appropriate development model
4. Estimation of effort in software development projects
5. Planning of activities in software development projects
6. Analyze and assess risks in software development projects
7. Manage resources in software development projects
8. Monitor and control software development projects
9. Contract management
10. Manage people in software projects
11. Awareness of software quality standards
12. Awareness of ethics in project management

7. Requirements:

All students are expected to bring a solid computer science background (programming, data structures and languages).

8. Laboratory/Field Work (If applicable):

Students will use the Applied Database and Software Engineering Laboratory. Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

Topic	No. of Classes	Goals (Sec. 6)	Teaching/Learning Skills
<i>Introduction</i>	1		Motivation and overview Lectures and discussion
<i>Overview of software project planning</i>	2	1	Lectures and discussion
<i>Software project evaluation</i> Assessment Cost analysis and forecasting Cash flow analysis and forecasting Risk analysis	6	2	Lectures Discussion Homework
<i>Development process models</i> Waterfall model RAD model V Model Spiral model Prototyping models Incremental models Tools	3	3	Lectures Discussion Homework
<i>Effort estimation</i> Estimation techniques Function-point techniques Object-code techniques Procedural techniques COCOMO	6	4	Lectures Discussion Homework Collaborative learning Laboratoy
<i>Exam # 1</i>	1		Written exam
<i>Project planning</i> Scheduling Gantt Charts Network planning CPM / PERT Project management tools Ethical issues	9	5,12	Lectures Discussion Homework Collaborative learning Laboratoy
<i>Risk management</i> Risk analysis Risk management plans Risk related to scheduling Z values	3	6	Lectures Discussion Homework
<i>Resource allocation</i>	3	7	Lectures Discussion

Resource scheduling Cost scheduling			Homework
<i>Exam # 2</i>	1		Written exam
<i>Monitoring and control</i> Process control framework Data collection Progress control PSP and TSP	5	8	Lectures Discussion Homework Laboratory
<i>Contract management</i>	2	9	Lectures and discussion
<i>Personnell management</i>	2	10,12	Lectures and discussion
<i>Software quality standards</i>	1	11	Lectures and discussion

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4sss
Course Title: Introduction to Software Engineering
Credit-Hours: 3

2. Course Description:

Introduction to the software engineering process. Software lifecycle models and related metrics. Linear models, evolutionary models, spiral and domain-specific models. Introduction to lifecycle activities, including planning, analysis, specification, design, testing and deployment. The UML language. Version control. Software Configuration Management. CASE tools. Software Engineering Ethics.

3. Pre-requisites: CCOM4lpd.

4. Textbook, Supplies and Other Resources:

Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 5th Edition, 2001.

Tom DeMarco, The Deadline: A Novel About Project Management, DorsetHouse Publishing, 1997

Rational Software, "UML Notation Guide", UML Specification v.1.3, Sec.3, Object Management Group (OMG), 1999

Pankaj Palote, An Integrated Approach to Software Engineering, 2nd Edition, Springer-Verlag, 1997

5. Purpose:

Required course for software engineering and computer science students. The purpose is to expose students to the complete software development process.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

13. Knowledge of the software lifecycle
14. Distinction among different lifecycle models and ability to decide for a particular one
15. Awareness of software development process metrics, especially size, effort, duration, cost and staffing
16. Knowledge of the UML language
17. Awareness on software development planning and project management
18. Awareness of different requirements analysis and specification techniques, including system analysis and use case analysis and specification
19. Awareness of different design techniques, especially object-oriented techniques
20. Awareness of version control and Software Configuration Management
21. Awareness of testing techniques, including unit, integrated, system and validation
22. Awareness Knowledge of the use of a CASE tool for requirements and design specification
23. Awareness of software engineering ethics
24. Apply acquired knowledge in the development of a project

7. Requirements:

All students are expected to bring a solid computer science background (programming, data structures and languages).

8. Laboratory/Field Work (If applicable):

Students will use the Applied Database and Software Engineering Laboratory. Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

Topic	No. of Classes	Goals (Sec. 6)	Teaching/Learning Skills
<i>Introduction</i>	1		Motivation and overview Lectures and discussion
<i>Software Lifecycle</i> Software lifecycle models and standards Linear lifecycle models Evolutionary lifecycle models Spiral lifecycle models Domain-specific lifecycle models	4	1,2	Lectures Discussion Homework
<i>Project description</i>	1	14	Discussion
<i>Development Process Metrics</i> Size estimation: LOC and FP Effort and duration Cost estimation, COCOMO	5	3,12	Lectures Discussion Laboratory Homework / Project
<i>Software Planning</i> Process and group organization Resource allocation, Staffing curve Scheduling and milestones Gantt Charts Risk analysis Ethical aspects	7	5,13	Lectures Laboratory Collaborative Learning Homework / Project
<i>“The Deadline” Novel Exam</i>	1	5	Written exam
<i>The UML Language</i> Use-case diagrams Class diagrams Sequence diagrams	4	4,12	Lectures Laboratory Homework / Project
<i>Exam # 1</i>	1		
<i>Requirements Analysis and</i>	8	6,12	Lectures Laboratory

Specification Analysis activities Requirements elicitation techniques System analysis Requirements specification standards Use case analysis Use case specification			Homework / Project Collaborative learning
Project discussion	1	14	Discussion
Design Design activities Design goals System design Software architectures Ethical aspects	5	7,13	Lectures Homework/Project
Exam # 2	1		Written exam
Testing Testing activities Unit and integrated testing	2	10	Lectures
Software Configuration Management Version control Software Configuration Management	4	11	Lectures Homework/Project

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4uuu
Course Title: Software Requirements Engineering
Credit-Hours: 3

2. Course Description:

Techniques to elicit, identify, determine and specify software requirements. Formal specification methods. Object-oriented analysis. Requirements specification standards. The UML language. Requirements verification and validation. Specification tools. Requirements management. Quality metrics. Requirements configuration management. Ethics in requirements elicitation.

3. Pre-requisites: INSO4sss

4. Textbook, Supplies and Other Resources:

Dean Leffingwell and Don Widrig, Managing Software Requirements: A Unified Approach, Addison-Wesley, 2000.

Bernd Bruegge and Allen H. Dutoit, Object-Oriented Software Engineering: Conquering Complex and Changing Systems, Prentice Hall, 1999

Rational Software, "UML Notation Guide", UML Specification v.1.3, Sec.3, Object Management Group (OMG), 1999

Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 5th Edition, 2001.

5. Purpose:

Required course for software engineering students; open to computer science students. The purpose is to teach students the software requirements engineering process, principles and techniques.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

1. Knowledge of the requirements engineering process
2. Knowledge of requirements analysis principles and techniques
3. Knowledge of requirements elicitation principles and techniques
4. Knowledge of requirements specification principles and techniques
5. Management of scope and change in the requirements engineering process
6. Generate and use of quality metrics
7. Knowledge of validation and verification principles and techniques
8. Use of CASE tools for requirements specification and management
9. Application of requirements management in a development project
10. Awareness of ethics in software requirements engineering

7. Requirements:

All students are expected to bring: (i) a solid computer science background (programming, data structures and languages); (ii) knowledge of the complete software development process.

8. Laboratory/Field Work (If applicable):

Students will use the Applied Database and Software Engineering Laboratory. Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

Topic	No. of Classes	Goals (Sec. 6)	Teaching/Learning Skills
<i>Introduction</i>	1		Motivation and overview Lectures and discussion
<i>The Requirements Engineering Process</i> Overview of the RE process Team management	3	1	Lectures Discussion Homework
<i>Requirements analysis</i> Requirements analysis activities Business modeling Systems engineering	6	2	Lectures Discussion Homework/Project
<i>Project discussion</i>	1	9	Discussion
Requirements elicitation Challenges System feature extraction Interview techniques Workshop techniques Brainstorming techniques Use case specification Storyboarding Role playing / CRC Prototyping techniques Ethical aspects	12	3,8,10	Lectures Discussion Collaborative learning Laboratory Homework/Project

Exam # 1	1		Written exam
Requirements specification Organizing and documenting requirements Use case specification Finite state machines Decision tables and trees Activity diagrams and workflows Object modeling Dataflow models Maintenance specifications	10	4	Lectures Discussion Collaborative learning Laboratory Homework/Project
Exam # 2	1		Written exam
Project discussion	1	9	Discussion
Scope Change management Managing scope Requirements configuration management	4	5	Lectures Discussion Laboratory
Quality metrics SRS Metrics Use Case Metrics	3	6	Lectures Laboratory
Validation and Verification Validation Verification	2	7	Lectures

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4vvv
Course Title: Software Design
Credit-Hours: 3

2. Course Description:

Fundamental principles and techniques of object-oriented software design. Formal design methods. Design specification standards. The UML language. Design validation and verification. Design metrics. Software architectures. Design patterns. Design by contract. The Eiffel language. Design tools.

3. Pre-requisites: *INSO4uuu*

4. Textbook, Supplies and Other Resources:

Meilir Page-Jones , Fundamentals of Object-Oriented Design in UML , Addison-Wesley, 2000

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides and Grady Booch, Design Patterns, Addison-Wesley, 1995

Bernd Bruegge and Allen H. Dutoit, Object-Oriented Software Engineering: Conquering Complex and Changing Systems, Prentice Hall, 1999

Rational Software, "UML Notation Guide", UML Specification v.1.3, Sec.3, Object Management Group (OMG), 1999

Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 5th Edition, 2001.

5. Purpose:

Required course for software engineering students; open to computer science students. The purpose is to teach students the principles and techniques of software design.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

11. Knowledge of the software design process
12. Knowledge of object-oriented principles
13. Knowledge of standard design notations
14. Knowledge of design principles
15. Knowledge of design patterns
16. Knowledge of system design principles and techniques
17. Knowledge of detailed design principles and techniques
18. Use of a CASE tool for software design
19. Application of software design in a development project
20. Awareness of ethics in software design

7. Requirements:

All students are expected to bring: (i) a solid computer science background (programming, data structures and languages); (ii) knowledge of the complete software development process.

8. Laboratory/Field Work (If applicable):

Students will use the Applied Database and Software Engineering Laboratory. Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

Topic	No. of Classes	Goals (Sec. 6)	Teaching/Learning Skills
<i>Introduction</i>	1		Motivation and overview Lectures and discussion
<i>The software design process</i>	1	1	Lectures
Review of object-oriented principles	1	2	Lectures, Discussion Homework
Standard design notation Class diagrams State and activity diagrams Sequence diagrams Collaboration diagrams Package diagrams Deployment diagrams	8	3,8	Lectures Homework Laboratory
Project Description	1	9	Discussion
Design principles Encapsulation and connascence Domains, encumbrance and cohesion State space and behavior Robust class hierarchies Inheritance and polymorphism Quality metrics	6	4	Lectures Discusssion Homework
Exam # 1	1		Written exam
Design patterns and software architectures	4	5	Lectures Homework
System design Design goals and tradeoffs Concurrency design Boundary conditions Hardware/software mappings Persistence design Access control policies Control flow tradeoffs Real-time constraints Ethical issues	12	6,8,9,10	Lectures Discussion Laboratory Homework / Project
Exam # 2	1		Written exam

Project discussion	1	9	Discussion
Detailed design Detailed design concepts Design by contract / Eiffel Detailed object design Component design	8	7	Lectures Homework / Project Laboratory

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4www
Course Title: Software Testing and Reliability
Credit-Hours: 3

2. Course Description:

Testing and validation in the software lifecycle. Techniques, tools and formal models of testing. Validation and verification of design through testing. Unit, integrated testing. Performance and stress testing. Usability testing. Fault tolerance. Test and debugging tools.

3. Pre-requisites: *INSO4vvv*

4. Textbook, Supplies and Other Resources:

Edward Kit, Software Testing in the Real World: Improving the Process, Addison-Wesley, 1995

Roger S. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 5th Edition, 2001.

5. Purpose:

Required course for software engineering students; open to computer science students. The purpose is to teach students the principles and techniques of software testing and reliability.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

21. Knowledge of software testing philosophy
22. Knowledge of the software testing process.
23. Knowledge of testing methods and techniques.
24. Knowledge of test management
25. Use of a CASE tool for software testing
26. Application of knowledge in a development project
27. Awareness of ethics in software testing

7. Requirements:

All students are expected to bring: (i) a solid computer science background (programming, data structures and languages); (ii) knowledge of the complete software development process; (iii) knowledge of software requirements specification and software design.

8. Laboratory/Field Work (If applicable):

Students will use the Applied Database and Software Engineering Laboratory. Laboratory sessions will be given at class time.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

Topic	No. of Classes	Goals (Sec. 6)	Teaching/Learning Skills
Introduction	1		Motivation and overview Lectures and discussion
Software testing philosophy and ethics	3	1,7	Lectures and discussion
The software testing process Overview of the process Risk management Validation and verification Cost/effectiveness tradeoffs Planning Configuration management Testing standards Test documents Test and reliability metrics	12	2,5	Lectures Discussion Homework Laboratory
Exam # 1	1		Written exam
Project description	1	6	Discussion
Testing methods	20	3	Lectures

Unit testing and test-case generation Integrated testing Verification Validation Usability testing Stress and performance testing Cost containment Testing tools / automated testing Testing metrics and release criteria			Discussion Homework / Project Collaborative learning Laboratory
Exam # 2	1		Written exam
Project discussion	1	6	Discussion
Test management Test documentation Test configuration management Planning and organization Current practices	4	5	Lectures Discussion
Project discussion	1	6	Discussion

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4yyy
Course Title: Software Engineering Project I
Credit-Hours: 3

2. Course Description:

First part in a two-course project sequence. Students will work in teams to solve academic, government, commercial or industrial software engineering problems. Each team will be assigned a customer. Work in the project includes the activities of the development process, up to design. Oral and written presentations are required. At the end of the course, students will make an oral presentation of the work completed, up to design. All the required documentation up to the end of the design phase are to be submitted for evaluation.

3. Pre-requisites: *INSO4www*

4. Textbook, Supplies and Other Resources:

No textbook.

5. Purpose:

Required course for software engineering students; open to computer science students. This is a two-semester capstone course to expose students to a major design and implementation experience, as required by ABET.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

- 28. Application of knowledge in a development project
- 29. Use of a variety of CASE tools for development

7. Requirements:

All students are expected to have completed the sequence of courses of the Software Engineering major.

8. Laboratory/Field Work (If applicable):

Student will make regular visits to the customer's site. The Applied Database and Software Engineering Laboratory will be used for development; implementation in the customer's site may also be required.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

This is a practice course, guided by the instructor. There are no lectures. At least three hours of discussion per week with the instructor are required.

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: INSO4zzz

Course Title: Software Engineering Project II

Credit-Hours: 3

2. Course Description:

Second part in a two-course project sequence. Students will work in teams to solve academic, government, commercial or industrial software engineering problems. Each team will be continue working with the customer assigned in the first part, or might be assigned another project already designed. Work in the project includes the activities of the development process, from implementation to deployment. Oral and written presentations are required. At the end of the course, students will make an oral presentation of the work completed, up to deployment. All the required documentation up to the end of the project are to be submitted for evaluation.

3. Pre-requisites: INSO4yyy.

4. Textbook, Supplies and Other Resources:

No textbook.

5. Purpose:

Required course for software engineering students; open to computer science students. This is a two-semester capstone course to expose students to a major design and implementation experience, as required by ABET.

6. Course Goals:

At the end of this course, students are expected to have the following knowledge and skills:

30. Application of knowledge in a development project

31. Use of a variety of CASE tools for development

7. Requirements:

All students are expected to have completed the sequence of courses of the Software Engineering major.

8. Laboratory/Field Work (If applicable):

Student will make regular visits to the customer's site. The Applied Database and Software Engineering Laboratory will be used for development; implementation in the customer's site may also be required.

9. Department/Campus Policies:

Please see the Campus Undergraduate Catalog.

10. Campus Resources

Some reading materials will be placed in the reserve section at the campus library.

11. General Topics

This is a practice course, guided by the instructor. There are no lectures. At least three hours of discussion per week with the instructor are required.

**University of Puerto Rico
Mayagüez Campus
College of Engineering**

Syllabus & Instructor Information Sheet Form

A. COURSE SYLLABUS

1. General Information:

Course Number: CCOM 4isc

Course Title: Undergraduate Research in Software Engineering

Credit-Hours: 1 - 6

2. Course Description:

Research work in Software Engineering under the supervision of a professor.

3. Pre/Co-requisites:

To be specified by the professor supervising the student.

4. Textbook, Supplies and Other Resources:

N/A

5. Purpose:

This is an elective course for majors in computing and software engineering.

6. Course Goals:

Provide an experience in research.

7. Requirements:

To be specified by the professor supervising the student.

8. Laboratory/Field Work (If applicable):

To be specified by the professor supervising the student.

A. COURSE SYLLABUS

9. Department/Campus Policies:

9a. Class attendance: Class attendance is compulsory. The University of Puerto Rico, Mayagüez Campus, reserves the right to deal at any time with individual cases of non-attendance. Professors are expected to record the absences of their students. Frequent absences affect the final grade, and may even result in total loss of credits. Arranging to make up work missed because of legitimate class absence is the responsibility of the student. (Bulletin of Information Undergraduate Studies, pp 39 1995-96)

9b. Absence from examinations: Students are required to attend all examinations. If a student is absent from an examination for a justifiable reason acceptable to the professor, he or she will be given a special examination. Otherwise, he or she will receive a grade of zero of "F" in the examination missed. (Bulletin of Information Undergraduate Studies, pp 39, 1995-96)

9c. Final examinations: Final written examinations must be given in all courses unless, in the judgment of the Dean, the nature of the subject makes it impracticable. Final examinations scheduled by arrangements must be given during the examination period prescribed in the Academic Calendar, including Saturdays. (see Bulletin of Information Undergraduate Studies, pp 39, 1995-96).

9d. Partial withdrawals: A student may withdraw from individual courses at any time during the term, but before the deadline established in the University Academic Calendar. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9e. Complete withdrawals: A student may completely withdraw from the University of Puerto Rico, Mayagüez Campus, at any time up to the last day of classes. (see Bulletin of Information Undergraduate Studies, pp 37, 1995-96).

9f. Disabilities: All the reasonable accommodations according to the Americans with Disability Act (ADA) Law will be coordinated with the Dean of Students and in accordance with the particular needs of the student.

9g. Ethics: Any academic fraud is subject to the disciplinary sanctions described in article 14 and 16 of the revised General Student Bylaws of the University of Puerto Rico contained in Certification 018-1997-98 of the Board of Trustees. The professor will follow the norms established in articles 1-5 of the Bylaws.

10. General Topics:

N/A