



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

SAN LUIS, 12 MAR 2010

VISTO:

El Expediente EXP-USL 0004956/2009 mediante el cual el Director de la Carrera de posgrado **Maestría en Ingeniería de Software** solicita la modificación del Plan de Estudio de la Carrera; y

CONSIDERANDO:

Que la carrera de posgrado **Maestría en Ingeniería de Software** fue creada por Ordenanza CS 35/97.

Que el Ministerio de Cultura y Educación de la Nación otorgó el reconocimiento oficial y consecuente validez nacional al Título de "Magíster en Ingeniería de Software" para la Carrera de referencia, mediante Resolución Ministerial N° 142/99.

Que la Carrera **Maestría en Ingeniería de Software** fue acreditada por CONEAU, según Resolución N° 435/01, y categorizada Cn.

Que la experiencia acumulada durante el desarrollo de la Maestría en Ingeniería de Software, indica que existe una demanda específica de formación de posgrado, acorde a la evolución del conocimiento en la región y en el país, que requiere una actualización de la Carrera.

Que existe disponibilidad de un estándar industrial internacionalmente aceptado respecto de los contenidos de carreras de este tipo, dado por el Software Engineering Institute – Carnegie Mellon University, que mantiene permanentemente en estudio y mejora continua su propio plan de maestría. Las modificaciones propuestas respetan dicho estándar y lo presentan adaptado al caso específico de nuestro entorno en un todo de acuerdo con las normativas vigentes de posgrado a nivel nacional, Resolución Ministerial N° 1168/97, y de la Universidad Nacional de San Luis, Ordenanza N° 23/09-CS.

Que existe un crecimiento de la demanda de profesionales formados específicamente para el desarrollo de software con criterios industriales.

Que existe un incremento de los requisitos de certificación de calidad para las empresas radicadas en nuestro país.

Que se observa un incremento en la tendencia a instalar "software factories" internacionales en nuestro país.

Que existe un incremento en la tendencia a exportar software desde nuestro país, que responda a estándares de calidad internacionalmente aceptados.

CORRESPONDE ORDENANZA N°

005 - 10

Dr. FELIX D. NIETO QUINTAS
DECANO
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

Que es necesario incrementar la formación de profesionales en el área de ingeniería de software permitiéndole la culminación de la maestría con el desarrollo de una tesis orientada a un proyecto o desarrollo profesional donde el alumno pueda describir, analizar y evaluar una actividad profesional diseñada y ejecutada permitiéndole resolver un problema real particular de la práctica de la ingeniería de software que represente un aporte al desarrollo de esta área.

Que existe en la Facultad de Ciencias Físico Matemática y Naturales como política institucional la articulación entre sus carreras de posgrado afines a Informática.

Que el Plan de Estudios de la carrera Maestría en Ingeniería de Software debe ser modificado para su adecuación a la Reglamentación General de Posgrado vigente de la Universidad Nacional de San Luis, Ordenanza N° 23/09 –CS.

Que el Consejo del Departamento de Informática, en su Sesión del día 19 de noviembre de 2009, decidió avalar la propuesta de modificación de Plan de Estudios de la Maestría en Ingeniería de Software.

Que la Comisión Asesora de Investigación de la Facultad de Ciencias Físico-Matemáticas y Naturales, actuando como Comisión Asesora de Posgrado, en su Reunión del día 18 de febrero de 2010 aconseja aprobar la modificación del **Plan de Estudios de la Maestría en Ingeniería de Software**.

Que es atribución del Consejo Directivo decidir sobre los Planes de Estudio conforme a lo establecido por el artículo 97°, inciso k, del Estatuto Universitario.

Que en su Sesión del día 11 de Marzo de 2010, el Consejo Directivo de la Facultad de Ciencias Físico Matemáticas y Naturales resuelve aprobar la modificación del Plan de Estudios de la Maestría en Ingeniería de Software, propuesta por el Director de la Carrera.

Que corresponde su protocolización.

Por ello, en virtud de lo acordado en su sesión del 11 de Marzo de 2010 y en uso de sus atribuciones

**EL CONSEJO DIRECTIVO DE LA FACULTAD DE CIENCIAS
FISICO, MATEMATICAS Y NATURALES
ORDENA:**

ARTÍCULO 1°: Solicitar al Consejo Superior la modificación de la Ordenanza CS 35/97 en lo referido al Plan de Estudios de la Carrera de Posgrado Maestría en Ingeniería de Software, para ser reemplazado por el texto que obra en el ANEXO único de la presente Ordenanza.

CORRESPONDE ORDENANZA N°

005-10

FELIX D. NESTO QUINTAS
DECANO
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.

DR. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

ARTÍCULO 2°: Aprobar la modificación del Plan de Estudios de la carrera de posgrado Maestría en Ingeniería de Software, de acuerdo al ANEXO de la presente norma.

ARTICULO 3°: Elevar al Consejo Superior de la Universidad Nacional de San Luis para su ratificación.

ARTICULO 4°.- Comuníquese, insértese en el Libro de Ordenanzas, publíquese en el Digesto Administrativo y archívese.-

ORDENANZA N°

005-10

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat
U.N.S.L

Dr. FÉLIX D. NIETO QUINTAS
DECANO
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

ANEXO

ARTÍCULO 1º: Los fundamentos de la Carrera de Posgrado Maestría en Ingeniería de Software son:

- Crecimiento de la demanda de profesionales formados específicamente para el desarrollo de software con criterios industriales.
- Incremento de los requisitos de certificación de calidad para las empresas radicadas en nuestro país.
- Incremento de la tendencia a instalar "software factories" internacionales en nuestro país.
- Incremento de la tendencia a exportar software desde nuestro país. Dicho software debe responder a estándares de calidad internacionalmente aceptados.
- Experiencia recogida durante el dictado por más de 10 años de esta Carrera, mostrando que existe una demanda específica de formación de posgrado.
- Actualización de los contenidos del plan de estudio respecto de los contenidos de carreras de este tipo (Software Engineering Institute – Carnegie Mellon University).
- Articulación con otras carreras de posgrado afines a Informática de la Facultad de Ciencias Físico, Matemáticas y Naturales.

ARTÍCULO 2º: Establecer para la carrera de posgrado Maestría en Ingeniería de Software, los siguientes objetivos:

- Introducir al alumno en la aplicación del método y conocimiento científicos en el diseño y construcción de Productos de Software destinados a optimizar la gestión integral de organizaciones de naturaleza diversa.
- Promover la producción de software a través de la aplicación de metodologías que permitan obtener un producto de software de alta calidad.
- Formar profesionales que tengan la capacidad para integrar y coordinar equipos dedicados al Planeamiento Estratégico de Sistemas de Información, a la Programación del Desarrollo de Sistemas, a la Definición de Especificaciones y Diseño de Sistemas de Información, a la Programación e Implantación de Productos de Software, al Mantenimiento, Re-ingeniería y Auditoría de Sistemas de Información.
- Profundizar en métodos que llevan a una alta productividad en la actividad de desarrollar software.
- Orientar la formación de grado del profesional hacia el desarrollo de Sistemas de Información para facilitar la integración de graduados en disciplinas diversas en equipos multidisciplinarios conformados para la producción de software.

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Capacitar al profesional para producir Software con enfoques metodológicos estrictos y con un adecuado soporte automatizado en las tareas de definición y análisis de requerimientos, desarrollo, programación, implantación y mantenimiento incluyendo competencia en la utilización de la Tecnología CASE.
- Promover la aplicación de herramientas de evaluación de proyectos a la producción de software.
- Identificar los criterios y principios de la Ingeniería de Software, asumiéndola como la disciplina dedicada a la producción sistemática de Productos de Software que serán desarrollados y modificados, necesariamente, en correspondencia con el cronograma y el presupuesto del proyecto.
- Utilizar conceptos de las Ciencias de la Computación, de la Administración y de la Economía combinados, en un marco metodológico adecuado, con la finalidad de obtener Productos de Software de Alta Calidad.

ARTÍCULO 3º: Establecer el perfil profesional del egresado:

La Maestría en Ingeniería de Software de la Universidad Nacional de San Luis aporta los conocimientos y habilidades necesarios para llevar a cabo esta construcción de software confiable destinado a mercados de alta exigencia y a organizaciones complejas. Las capacidades, en principio, puede agruparse en:

- Capacidades eminentemente tecnológicas que permitirán desarrollar sistemas complejos utilizando las técnicas o métodos de desarrollo de software más innovadoras.
- Capacidades de Gestión que permitirán identificar y seleccionar elegir las mejores alternativas tecnológicas en cuanto a costo, confiabilidad y adaptabilidad para desarrollar software en las mejores condiciones en mercado exigentes y complejo.
- Capacidades de comunicación que faciliten transmitir, convencer de la necesidad de desarrollar aplicaciones útiles en entornos complejos y para presentar a los proyectos de software como inversiones atrayentes.

Además,

El Magister en Ingeniería de Software es un profesional de post grado con sólidas bases metodológicas en el desarrollo de software a mediana y gran escala, en la tecnología de inteligencia de negocios y en las principales herramientas de programación, así como en la interacción con los líderes de distintos tipos de organizaciones.

- El Magister en Ingeniería de Software será capaz de desempeñarse en cualquiera de los roles involucrados en un proceso de desarrollo de software como líder de proyecto, arquitecto del software, analista, programador, ingeniero de requerimientos, ingeniero de pruebas, etc.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- El Magister en Ingeniería de Software podrá además definir alcances, costos, tiempos, recursos y factibilidad para un proyecto de software, así como proponer soluciones de software, globales o parciales, que permitan el control de los procesos, la mejora en el proceso de toma de decisiones o soluciones innovadoras para la industria y el entretenimiento
- Los rasgos distintivos del perfil del Magister en Ingeniería de Software pueden delinearse mediante la enumeración de sus capacidades:
 - Capacidad para la gestión integral de proyectos de software de alta complejidad
 - Capacidad para participar y coordinar equipos dedicados al Planeamiento Estratégico de Sistemas de Información, a la Programación del Desarrollo de Sistemas, a la Definición de Especificaciones y Diseño de Sistemas de Información, a la Programación e Implantación de Productos de Software, al Mantenimiento, Re-ingeniería y Auditoría de Sistemas de Información.
 - Capacidad de incidir significativamente en la Calidad y Productividad de la actividad producción de Software.
 - Posibilidad de orientar su formación de grado hacia el desarrollo de Sistemas de Información al facilitar la integración de graduados en disciplinas diversas en equipos multidisciplinarios conformados para la producción de Software.
 - Capacidad de abordar la producción de Software con enfoques metodológicos estrictos y un adecuado soporte automatizado en tareas de definición y análisis de requerimientos, desarrollo, programación, implantación y mantenimiento incluyendo competencia en la utilización de la Tecnología CASE.
 - Capacidad para aplicar a la producción de Software las herramientas habituales de gestión de proyectos.
 - Capacidad para integrar y coordinar equipos dedicados al desarrollo y producción de Software de alta confiabilidad.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.

ARTÍCULO 4º: Establecer los alcances formativos del Magister en Ingeniería de Software. Entenderá en el estudio, factibilidad, proyecto, planificación, dirección, realización, instalación, puesta en marcha, operación, ensayo, mantenimiento, reparación, modificación, transformación e inspección de:

CORRESPONDE ORDENANZA Nº 005 - 10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Arquitecturas de Software incluyendo a los sistemas embebidos (“embeded systems”) y su aplicación en gestión de la información y el conocimiento, la automatización de control y adquisición de datos, interfaces con otros sistemas de diversa naturaleza (mecánica, eléctrica, humana, etc.), medición y control del rendimiento de los sistemas de software de mediana y alta complejidad.
- Sistemas de Teleinformáticos en general, interfaces para establecer enlaces entre aplicaciones, software de Redes de Computadoras en sus diferentes formas y topologías.
- Interacción del software de aplicación con el Software de Base, en particular teniendo en cuenta la Arquitectura y Componentes de Sistemas Operativos de Computadoras.
- Seguridad informática en aplicaciones de mediana y alta complejidad.
- Administración, Planificación y Organización de Recursos Computacionales en general y en particular a lo que hace a los componentes de software de los sistemas de información.
- Análisis, diseño e implementación de Sistemas de Software para aplicaciones específicas por ejemplo las vinculadas al hardware de computadoras y dispositivos de comunicación, incluyendo a sistemas embebidos, sistemas de control automático y adquisición de datos.
- Capacitación en temas relacionados con los incisos anteriores.
- Pericias, arbitraje y tasaciones relacionadas con incisos anteriores.
- Intervenir en el diseño, implementación, operación y mantenimiento de:
 - Sistemas de Software de Aplicación, abarcando temas de Ingeniería en Software, Bases de Datos y Algoritmos en general y optimización en el uso de los recursos computacionales en general y de software en particular.
 - Sistemas de Procesamiento Digital de la información, incluyendo las interfaces correspondientes.
 - Productos de software clasificados como de Inteligencia Artificial, recursos para sistemas expertos, tratamiento y reconocimiento de imágenes y patrones.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Físico Matemáticas y Nat.
U.N.S.L.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Físico Matemáticas y Nat.
U.N.S.L.

ARTICULO 5º: Los requisitos de admisión son los que establecen las reglamentaciones vigentes de la Universidad Nacional de San Luis.

En particular, la maestría está orientada a:

- Profesionales en Informática (Licenciados en Ciencias de la Computación, en Sistemas, Ingenieros en Sistemas y de carreras afines con planes de estudio de cuatro o más años de duración).

CORRESPONDE ORDENANZA Nº

005-10.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Profesionales en Ciencias Económicas (Licenciados en Administración, Actuarios, Licenciados en Economía) que aspiren a obtener la aptitud de integrarse a equipos multidisciplinarios de desarrollo de sistemas de software.
- Profesionales en Ingeniería (en las diversas especialidades) que deseen poseer la capacidad de integrarse a equipos multidisciplinarios de desarrollo de software.
- Profesionales en Ciencias Exactas o Naturales que deseen desarrollar la capacidad de formar parte de equipos de desarrollo de software.
- Profesionales universitarios, graduados en universidades de gestión pública o privada de nuestro país o del exterior, en carreras de grado de cuatro o más años de duración, que manifiesten una muy firme decisión de integrarse a equipos multidisciplinarios de desarrollo de software.

Los profesionales o graduados con formación de grado no específica en Informática y que no demuestren los conocimientos base necesarios, deberán cursar seminarios/talleres como requisito para la confirmación de su aceptación como alumno de la carrera, acorde a lo que establezca el Comité Académico. Dicho Comité determinará los talleres que el alumno deberá aprobar con el fin de nivelar sus conocimientos para poder cursar la Maestría.

Los talleres se realizarán a los efectos de asegurar que el postulante tenga los conocimientos necesarios para garantizar el aprovechamiento máximo de la formación que brinda la carrera. La dedicación de cada alumno será determinada de acuerdo con la formación y capacitación con la que inicien la Maestría. La pauta rectora será lograr un nivel de conocimientos que le permita participar del desarrollo de los módulos de la Maestría. Los siguientes son un listado de talleres, los cuales se irán actualizando según los avances tecnológicos y necesidades de los alumnos:

- Taller 1: Tópicos Básicos de Arquitectura de Computadores
- Taller 2: Fundamentos de Algoritmos y Programación en Java
- Taller 3: Construcción de Modelos usando UP/UML
- Taller 4: Introducción a la Tecnología de Base de Datos
- Taller 5: Desarrollo de Aplicaciones Web.

ARTÍCULO 6°: Se establece un cupo de 50 alumnos, que podrá ser modificado acorde con las disponibilidades de recursos humanos y materiales del Departamento de Informática de la Facultad de Ciencias Físico, Matemáticas y Naturales.

ARTÍCULO 7°: Se establece los siguientes tipos de evaluaciones:

De los talleres y asignaturas:

CORRESPONDE ORDENANZA N° 005 - 10

Dr. FERNANDO BULNES QUINTERO
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- a) Desarrollo de casos prácticos correspondientes a cada tipo de metodología.
- b) Comprobación formal mediante test de evaluación.

De la Tesis de Maestría:

La Tesis de Maestría, según la reglamentación de posgrado vigente de la Universidad Nacional de San Luis, es una **Tesis de Desarrollo o de Proyecto Profesional**, la cual describe, analiza y evalúa una actividad profesional diseñada y ejecutada, por el alumno, para resolver un problema particular de la práctica especializada que constituyan un aporte al desarrollo de la profesión.

Conceptualmente el Informe de Tesis de Maestría deberá presentar con claridad:

- Descripción del Problema (de lo general a lo particular y destacando su relevancia)
- Descripción detallada de la solución que se propone respecto del problema presentado
- Justificación y sustento de la consistencia y robustez de la solución propuesta

Formalmente la Tesis estará organizada:

- I. Presentación
 - Portada
 - Prefacio
 - Reconocimiento
 - Abreviaturas
- II. Cuerpo
 - Introducción
 - Partes (optativas)
 - Capítulos
 - Sub capítulos
- III. Final
 - Apéndice
 - Notas
 - Referencias
 - Bibliografía
 - Figuras
 - Tablas
 - Glosario

[Handwritten signature]
Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Físico, Matem. y Nat.
U.N.S.L.

Las Tesis, en su organización, deberán regirse según los lineamientos del "Manual de Redacción de Escritos de Investigación", Eduardo Scarano - Editorial Macchi - 2004

CORRESPONDE ORDENANZA N° 005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

ARTÍCULO 8°: El alumno deberá cursar 36 créditos (Artículo 11° del Título Preliminar – Disposiciones Genereales, Ordenanza N° 23/09-CS) en asignaturas, de las cuales serán 20 créditos de asignaturas obligatorias, 8 créditos de asignaturas optativas y, podrá optar con 8 créditos por realizar una tutoría en investigación o bien 8 créditos por la aprobación de las asignaturas Metodología de la Investigación Científica I y Metodología de la Investigación Científica II.

Se otorgará el Título de "Magister en Ingeniería de Software", a aquellos alumnos que hayan obtenido los 36 créditos y hayan aprobado la Tesis de Desarrollo o de Proyecto Profesional de la Maestría.

ARTICULO 9°: Establecer el ordenamiento curricular, créditos, crédito horario total y crédito horario total presenciales y crédito horario de prácticas de la carrera.

Nro.Mat	C	Asignatura	Creds.	CHT	CHP
PRIMER AÑO					
1	1	Modelos de Sistemas de Software	4	80	30
2	1	Métodos de la Ingeniería de Software	4	80	30
3	1	Metodología de la Investigación Científica I*	4	80	25
4	2	Administración de Proyectos de Desarrollo de Software	4	80	30
5	2	Metodología de la Investigación Científica II*	4	80	25
SEGUNDO AÑO					
6	1	Análisis de Artefactos de Software	4	80	30
7	1	Arquitectura de Sistemas de Software	4	80	30
8	2	Optativas	8	160	60
9		Tutoría en Proyecto de Investigación acreditado*	8	160	
		Tesis			
Total			36	720	

Fernando M. Bulnes
Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TGA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N° 005 - 10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

* El Maestrando podrá optar por cursar Metodología de la Investigación Científica I y II o bien realizar la Tutoría.

NOMENCLATURA USADA

Nro. Mat: Número de Materia

C: Cuatrimestre

Creds.: Créditos

CHT: Crédito Horario Total Presencial

CHP: Crédito Horario de Formación Práctica que ya están incluidos en el Crédito Horario Total Presencial de la Asignatura

ARTICULO 10°: Establecer el régimen de correlatividades de la carrera.

Nro. Mat	C	Asignatura	P/Cursar		P/Rend.
			Reg.	Aprob	Aprob.
PRIMER AÑO					
1	1	Modelos de Sistemas de Software	-	-	-
2	1	Métodos de la Ingeniería de Software	-	-	-
3	1	Metodología de la Investigación Científica I*	-	-	-
4	2	Administración de Proyectos de Desarrollo de Software	2	-	2
5	2	Metodología de la Investigación Científica II*	3	-	3
SEGUNDO AÑO					
6	1	Análisis de Artefactos de Software	2, 4	1	1,2,4
7	1	Arquitectura de Sistemas de Software	-	1,2	1,2

[Handwritten signature]
Dr. Fernando M. Bulnes
Fac. de Cs. Físico, Mat. y Nat.
U.N.S.L.

[Handwritten signature]
DR. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Físico, Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N° 005 - 10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

OPTATIVAS				
	Sistemas de tiempo real	7	6	6,7
	Sistemas financieros	-	4	4
	Análisis comparativos de paradigmas de programación	-	6	6
	Tópicos avanzados de especificación y verificación formal de software	6	7	6,7
	Ingeniería Web	6	7	6,7
	Auditoría informática	-	4	4
	Temas avanzados de calidad del software	7	6	6,7

NOMENCLATURA USADA

Nro. Mat: Número de Materia
C: Cuatrimestre
P/Cursar Reg.: Materias correlativas que el alumno debe tener regular para poder cursar la materia
P/Cursar Aprob.: Materias correlativas que el alumno debe tener aprobadas para poder cursar la materia
P/Rend. Aprob.: Materias correlativas que el alumno debe tener aprobadas para poder rendir la materia

ARTICULO 11°: Establecer los siguientes objetivos, contenidos conceptuales mínimos, crédito horario total, crédito horario de formación práctica considerando que estas horas están dentro del crédito horario total, metodología de cursado, tipo de evaluación y bibliografía básica, complementaria y de nivelación.

CORRESPONDE ORDENANZA N°

005 - 10

Felipe D. Nieto Quintana
Dr. FELIPE D. NIETO QUINTANA
DECANO
Fac. de Cs. Físico, Mat. y Nat.
UNSL

Fernando M. Bulnes
Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Físico, Mat. y Nat.
UNSL



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

1. Modelos de sistemas de software

1.1. Objetivos

Transferir, a los alumnos de la Maestría en Ingeniería de Software, los conocimientos y habilidades instrumentales relacionados con el diseño, la implantación y el uso de modelos abstractos para la caracterización y estudio de propiedades de sistemas de software.

1.2. Contenidos conceptuales mínimos

Fundamentos científicos del uso de modelos abstractos para la caracterización y estudio de propiedades de sistemas de software. Distintos criterios para la clasificación de los enfoques y herramientas de modelado disponibles. Estudio comparativo de las fortalezas y debilidades de los distintos enfoques y herramientas asociadas. Correspondencia entre los enfoques y herramientas y los distintos espacios de problema. Máquinas de estado. Relaciones de abstracción, invariantes, no determinismo y definiciones inductivas y denotacionales Redes de Petri. Métodos Formales. Especificación y verificación formal utilizando el lenguaje Haskell. Estudio de la verificación de modelos y de la consistencia interna de artefactos de software. Mecanismos de composición / descomposición.

1.3. Crédito horario total: 80 horas

1.4. Crédito horario de formación práctica: 30 horas

1.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio análogos a los correspondientes a la Industria de Software.

1.6. Tipo de evaluación

- c) Desarrollo de casos prácticos correspondientes a cada tipo de metodología.
- d) Comprobación formal mediante test de evaluación.

1.7. Bibliografía

Básica:

Kühne, Thomas, "Models in Software Engineering", Springer, 2006

Magee, Jeff, Kramer, Jeff, "Concurrency: State Models & Java Programs", Wiley, Segunda Edición

- Kan, Stephen, "Metrics and Models in Software Quality Engineering", Addison-Wesley Professional, 1995

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
J.N.S.L.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
J.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- The RAISE Language Group. "The RAISE Specification Language". Prentice-Hall International, 1992.
- The RAISE Method Group. "The RAISE Development Method". Prentice-Hall International, 1995.
- International Standard ISO/IEC 15909, High-level Petri Nets - Concepts, Definitions and Graphical Notation, 2000
- Bjørner, Dines, Software Engineering 2: Specification of Systems and Languages, Springer, 2006.
- www.haskell.org
- Wordsworth, John, Software Development with Z: A Practical Approach to Formal Methods in Software Engineering, Addison-Wesley, 1992.

Complementaria:

- Beydeda, Sami, Model-Driven Software Development (Journals Papers), Gruhn, Volker (Eds.) 2005.
- Berard, B.; Bidoit, M.; Finkel, A.; Laroussinie, F.; Petit, A.; Petrucci, L.; Schnoebelen, P., Systems and Software Verification: Model-Checking Techniques and Tools, Springer, 2001
- Ghezzi, Mandrioli, Morascas and Pezze, A general way to put time in Petri nets. Proceeding 5th International Workshop on Software Specification and Design, Pittsburgh, May 89, pages 60-67, IEEE Computer Society Press, 1989.
- Zuberek, M., Timed Petri nets: Definitions, Properties, and applications. Microelectronics and Reliability, 31(4):627-644, 1991.
- Ghezzi, Mandrioli, Morascas and Pezze, A unified high-level Petri Net model for time critical system. IEEE Transactions on Software Ingeniering 17(2), 160-172. 1991.

Página Web del Instituto de Ingeniería de Software (CMU),
<http://www.mse.cd.cmu.edu>,

Bibliografía y herramientas RAISE descargables desde el sitio de la United Nations University - UNU/IIST: <http://www.iist.unu.edu/raise/>

- Dasso, A.; Funes, A., Formal Methods in Software Engineering, Encyclopedia of Information Science and Technology, Idea Group, 2005.
- Funes, A.; George, C., UML and the Unified Process, Idea Group, 2003.
- Luqi Goguen, J., Formal methods: Promises and problems, IEEE Software, 14(1), págs. 73-85., Ene-Feb 1997.

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Bowen, J.; Hinchey, M., Seven more myths of formal methods. IEEE Software, 12(3). Julio, 1995.
- Bowen, J.; Hinchey, M., Ten Commandments of formal methods. IEEE Computer. Abril, 1995

De nivelación:

- Peterson J., Petri Net Theory and the Modelling of Systems, Prentice Hall, Englewood Cliffs, New Jersey, 1981.
- Arthur, Lowell Jay, Rapid Evolutionary Development, Wiley, 1992
- Martin, James, Information Engineering, Prentice Hall, 1990
- Connell, J. and L. Shafer, Object-Oriented Rapid Prototyping. Yourdon Press / Prentice Hall, 1995.
- Ian Sommerville, Software Engineering (Fifth Edition), Addison-Wesley, 1996
- Pressman, Roger, The 6th edition of Software Engineering: A Practitioner's Approach I, McGraw-Hill, 6ta Edición, 2005
- Rational Software, SQA Suite.
<http://www.rational.com/products/sqa/prodinfo/index.jtmpl>
- Rational Software, ClearCase,
<http://www.rational.com/products/clearcase/prodinfo/index.jtmpl>
- Rational Software, Performance Studio,
<http://www.rational.com/products/pstudio/prodinfo/index.jtmpl>
- Fenton, N.E., Pfleeger, S.L., Software Metrics: a Rigorous and Practical Approach, 2nd Ed., PWS Publishing Company, 1997.

2. Métodos de la Ingeniería de Software

2.1. Objetivos

Posibilitar, a los alumnos de la Maestría en Ingeniería de Software, la comprensión de la esencia y el uso de los métodos que “salvan” la brecha existente entre el problema del mundo real a ser resuelto y el sistema de información que opera para solucionar dicho problema.

CORRESPONDE ORDENANZA N° . 005 - 10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

2.2. Contenidos conceptuales mínimos

Comprensión de la esencia y uso de los métodos que “salvan” la brecha existente entre el problema del mundo real a ser resuelto y el sistema de información que opera para solucionar dicho problema. Métodos formales vs. Métodos semi-formales. Métodos para desarrollo de sistemas de tiempo real. Metodologías Ágiles. Métodos y Modelado específicos para la Web. Singularidades para el caso de sistemas basados en el conocimiento, los agentes inteligentes y otros paradigmas para el diseño de sistemas. Fortalezas y debilidades de cada uno de ellos. Fases desde el requerimiento hasta la obtención del código con el nivel de calidad requerido. Habilidades requeridas por los distintos enfoques metodológicos. Desarrollo de las habilidades necesarias distinguir las verdaderas diferencias entre las distintas propuestas metodológicas. Desarrollo de las habilidades para seleccionar el método más adecuado ante un problema específico a ser resuelto y un entorno particular. Definición de Procesos de Software.

2.3. **Crédito horario total:** 80 horas

2.4. **Crédito horario de formación práctica:** 30 horas

2.5. **Metodología de cursado:** presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

2.6. **Tipo de evaluación**

- e) Desarrollo de casos prácticos correspondientes a cada tipo de metodología.
- f) Comprobación formal mediante test de evaluación.

2.7. **Bibliografía**

Básica:

- Gustafson, David, Outline of Theory and Problems of Software Engineering, NetLibrary Inc, 2002
- Hasselbring, W., Giesecke, S., Research Methods in Software Engineering, Gito, Berlin, Germany, 2005
- Jackson, Michael, Software Requirement & Specifications, Addison Wesley, 1995.
- Henderson-Seller, B., Edwards, J., (dos tomos) Object-Oriented Knowledge: The Working Object : Object-Oriented Software Engineering : Methods and Management, Prentice Hall Object-Oriented Series, 1994

CORRESPONDE ORDENANZA N° 005 - 10


DR. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Gonzalez, Avelino, Dankel Douglas, "Engineering of Knowledge-Based Systems", ISBN:9780132769402, Publisher:Prentice Hall, 1993
- Petrie, Charles, Agent-Based Engineering, the Web, and Intelligence, Stanford Center for Design Research, <http://www-cdr.stanford.edu/NextLink/Expert.html>
- Moore, James, The Road Map to Software Engineering: A Standards-Based Guide Wiley-IEEE Computer Society Press, 2006.

Complementaria:

- Mendes, Emilia, Web Engineering, Springer, 2005
- Liu, Jane, Real-Time Systems, Prentice Hall, 2005
- Monin, Jean-Francois Understanding Formal Methods, Springer, 2003
- B. Thuraisingham, Web Data Management and Electronic Commerce, CRC Press, 2000
- Alistair, Desarrollo de Software Ágil, Addison-Wesley Professional; 1st edition, 2001
- Dujmovic, J.J., A Method for Evaluation and Selection of Complex Hardware and Software Systems, 22nd International Conference for the Resource Management and Performance Evaluation of Enterprise Computing Systems. CMG 96 Proc., V1, 1996.
- Schwabe, D.; Rossi, G.; Emerald, L.; Lyardet, F., Engineering Web applications for reuse, IEEE Multimedia, Springer 2001.
- Springer. International Conference on Web Engineering (ICWE), 2003, 2004, 2005. <http://www.springer.com>.
- Rosenfeld, L., Morville, P., Information Architecture for the World Wide Web, O'Reilly, 1998.
- Zuse, H., A Framework of Software Measurement, Walter de Gruyter, Berlín-NY, 1998.
- Dujmovic, J.J., CSc841 Reader: Computer Performance Evaluation. SFSU, 1996.
- Lazowska, E.D.; et al, Quantitative System Performance. Prentice-Hall, 1984.
- OMG, UML Semantics, <http://www.omg.org>
- OMG, Object Constraint Language Specification, <http://www.omg.org>

CORRESPONDE ORDENANZA N°

005-10

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Scharl, A., Evolutionary Web Development, Springer, 2000
- Furht, B., Handbook of Internet Computing, CRC, 2000
- Cockburn, Alistair, Characterizing People as Non-Linear, First-Order Components in Software Development,
<http://alistair.cockburn.us/crystal/articles/cpanfocisd/characterizingpeopleasnonlinear.html>
- Alianza Ágil, <http://www.agilealliance.org>
- Manifiesto para el Desarrollo de Software Ágil, <http://www.agilemanifesto.org>
- Beck, Kent, Extreme Programming Explained, Addison-Wesley, 1999.
- Paulk, Mark, XP desde la Perspectiva del CMM,
<http://www.sei.cmu.edu/cmm/papers/xp-cmm-paper.pdf>
- Crystal, <http://members.aol.com/acockburn/>
- Highsmith, Jim, Adaptive Software Development, Dorset House Publishing Company, Incorporated, 1999
- Software and Systems Modeling, <http://www.springerlink.com/>.
- Palmer, Stephen ; Felsing, John, A Practical Guide to Feature-Driven Development (The Coad Series), Prentice Hall PTR, 2002
- Coad, Peter "Java Modeling In Color With UML", Innovations in Systems and Software Engineering, <http://www.springerlink.com/>.
- Schwaber, Ken; Beedle, Mike, Agile Software Development with Scrum, Prentice Hall; 1st edition, 2001
- TogetherSoft, <http://www.togethersoft.com>
- Stapleton, Jennifer; Constable, Peter, DSDM, Addison-Wesley Professional; 1st edition, 1997
- Ambler, Scott, Agile Modeling: Effective practices for Extreme Programming and the Unified Process, John Wiley & Sons, 2002.

De nivelación:

- Pressman, Roger, The 6th edition of Software Engineering: A Practitioner's Approach I, McGraw-Hill, 6ta Edición, 2005

CORRESPONDE ORDENANZA N° 005-10

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TGA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Sommerville, Ian, Software Engineering, Pearson, 7ma edición
- Booch, Rumbaugh, Jacobson, El Proceso de Desarrollo de Software Unificado, Addison-Wesley, última edición disponible.
- Booch, Rumbaugh, Jacobson, El Lenguaje de Modelado Unificado, Addison-Wesley, última edición disponible.
- Booch, Grady, Object-Oriented analysis and design with applications, The Benjamin/Cummings Publishing Company Inc. 1994.
- Fenton, N.E., Pfleeger, S.L., 1997, Software Metrics: a Rigorous and Practical Approach, 2nd Ed., PWS Publishing Company
- Conallen J, Building Web Applications with UML, Addison-Wesley Pub Co, 1999.

3. Metodología de la Investigación Científica I

3.1. Objetivos

Los objetivos son lograr que los alumnos:

- Obtengan conocimiento de la Investigación Científica, con especial aplicación a las estrategias metodológicas.
- Aprendan a organizar la actividad de investigación.
- Adopten posturas críticas desde el punto de vista metodológico de las investigaciones publicadas.
- Asocien los conocimientos de Metodología con los otros adquiridos en las distintas asignaturas de la Maestría, para realizar un trabajo de investigación creativo y riguroso.

3.2. Contenidos conceptuales mínimos

Enfoque e instrumentos.

El conocimiento y todo lo que le atañe: la definición de conocimiento, su adquisición y transmisión.

Qué es conocimiento. Diferentes tipos: científico, artístico, común, etc. ¿Existe un conocimiento particular a una ciencia o área? ¿Hay un conocimiento universal, verdadero para todas las personas, o uno para cada uno?

CORRESPONDE ORDENANZA N°

005-10

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

El enfoque científico: Conocimiento: Ordinario y científico. El Método Científico. La táctica científica. Las ramas de la ciencia. Objetivo y Alcance de la ciencia. Pseudociencia.

Concepto: Lenguajes científicos. Sistemática de conceptos.

Dilucidación: Vaguedad y casos limítrofes. Precisión. Definición. Interpretación.

Las ideas científicas.

Problema: La fuente de la ciencia. Problemas científicos. Heurística. Problemas científicos.

Hipótesis: Significaciones de "hipótesis". Formulación. Clases: forma y contenido. Clases: punto de vista gnoseológico. Fundamento. Contrastabilidad. Requisitos. Funciones. Hipótesis filosóficas de la ciencia.

Ley: variables e invariantes. Clases. Fórmulas y pautas. Requisitos.

Teoría: Unidad conceptual. Deducibilidad. Construcción de teorías. Matematización. Referencia y evidencia.

3.3. Crédito horario total: 80 horas

3.4. Crédito horario de formación práctica: 25 horas

3.5. Metodología de cursado: presencial. Dictado con exposición conceptual.

3.6. Método de Evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Evaluación formal mediante test de comprobación. Presentación y aprobación de un proyecto.

3.7. Bibliografía

M. Bunge, La Investigación Científica, Ed. Ariel, Barcelona, 1989.

M. Bunge, Epistemología, Ciencia de la Ciencia. Ed. Ariel, Barcelona, 1981.

K. Popper, La Lógica de la investigación científica, Ed. Tecnos, Madrid, 1962.

- Stephen Thornton, "Karl Popper", The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.), URL = <<http://plato.stanford.edu/archives/win2002/entries/popper/>>
- Thomas Kuhn, The Structure of Scientific Revolutions, University of Chicago Press, 1962.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Mosterín, Jesús and Roberto Torretti (2002). Diccionario de Lógica y Filosofía de la Ciencia. Madrid: Alianza Editorial, 2002. 670 pp. ISBN: 84-206-3000-4.
- John Preston, "Paul Feyerabend", The Stanford Encyclopedia of Philosophy (Summer 2002 Edition), Edward N. Zalta (ed.),
- URL = <http://plato.stanford.edu/archives/sum2002/entries/feyerabend/>
- B. Russell, "The Scientific Outlook", Allen and Unwin, London, 1954.
- Hendricks, Vincent F. Mainstream and Formal Epistemology, New York: Cambridge University Press. 2006.
- Contemporary Debates in Epistemology, Peter Sosa and Matthias Steup (eds.). Blackwell Publishing. 2005.

4. Administración de proyectos de desarrollo de software

4.1. Objetivos

Transferir, a los alumnos de la Maestría en Ingeniería de Software, los conocimientos y habilidades instrumentales relacionados con los métodos y herramientas de mayor aceptación para la gestión de proyectos en general y la gestión de proyectos de software en particular. Efectuar un estudio comparativo de las fortalezas y debilidades de diversas propuestas respecto del proceso del software y de cada uno de los modelos de ciclo de vida. Facilitar la gestión del alcance, la programación de las actividades y el presupuesto de proyectos de software, la gestión de los recursos humanos del proyecto, la administración de las comunicaciones internas y externa del proyecto, la administración de riesgos del proyecto, la gestión de las configuraciones, el aseguramiento de la calidad en proyectos de software y, fundamentalmente, una efectiva aplicación de un enfoque integrado en la gestión de proyectos.

4.2. Contenidos conceptuales mínimos

Recursos requeridos en un proyecto de software: Su administración específica a lo largo del ciclo de vida del proyecto. Conformación y liderazgo de equipos de desarrollo de software. El concepto de Proceso de Software: Su aplicación. Modelos de ciclo de vida de productos de software. Estudio comparativo de las fortalezas y debilidades de cada modelo de ciclo de vida. Programación y presupuesto de proyectos de software. Administración de riesgos. Gestión de las configuraciones. Aseguramiento de la calidad en proyectos de software.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

Definición de las áreas clave de proceso asociadas al Capability Maturity Model Integration y sus correspondientes prácticas y componentes tecnológicas. Relación del producto de software con el producto integrado con especial énfasis en el software embebido. Aspectos legales del desarrollo de software: copyright, patentes, etc. La documentación de un proyecto de software: Estándares vigentes. La ingeniería de requerimientos a lo largo de un proyecto.

4.3. Crédito horario total: 80 horas

4.4. Crédito horario de formación práctica: 30 horas

4.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

4.6. Tipo de evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Caso práctico integrador
- c) Test de evaluación formal

4.7. Bibliografía

- David I. Cleland, Roland Gareis (2006). Global project management handbook. McGraw-Hill Professional, 2006. ISBN 0071460454.
- Nokes, Sebastian. The Definitive Guide to Project Management. 2nd Ed.n. London (Financial Times / Prentice Hall): 2007. ISBN 978 0 273 71097 4.
- Lewis R. Ireland (2006) Project Management. McGraw-Hill Professional, 2006. ISBN 007147160X.
- Paul C. Dinsmore et al (2005). The right projects done right!, John Wiley and Sons, 2005. ISBN 0787971138.
- Lewis R. Ireland (2006). Project Management. McGraw-Hill Professional, 2006. ISBN 007147160X.
- Dennis Lock (2007). Project management (9ed.) Gower Publishing, Ltd., 2007. ISBN 0566087723 .

CORRESPONDE ORDENANZA N° 005-10


DR. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Young-Hoon Kwak (2005). "A brief history of Project Management". In: The story of managing projects. Elias G. Carayannis et al. (eds), Greenwood Publishing Group, 2005. ISBN 1567205062
- David I. Cleland, Roland Gareis (2006). Global project management handbook. McGraw-Hill Professional, 2006. ISBN 0071460454.
- Martin Stevens (2002). Project Management Pathways. Association for Project Management. APM Publishing Limited, 2002. ISBN 190349401X
- Morgen Witzel (2003). Fifty key figures in management. Routledge, 2003. ISBN 0415369770.
- F. L. Harrison, Dennis Lock (2004). Advanced project management: a structured approach. Gower Publishing, Ltd., 2004. ISBN 0566078228.
- Bjarne Kousholt (2007). Project Management – Theory and practice. Nyt Teknisk Forlag. ISBN 8757126038.
- Berczuk, Appleton; (2003). Software Configuration Management Patterns: Effective Team Work, Practical Integration (1st ed.). Addison-Wesley. ISBN 0-201-74117-2.
- Bersoff, E.H. (1997). Elements of Software Configuration Management. IEEE Computer Society Press, Los Alamitos, CA, 1-32
- Futrell, R.T. et al. (2002). Quality Software Project Management. 1st ed. Prentice-Hall.
- International Organization for Standardization (2003). ISO 10007: Quality management systems – Guidelines for configuration management (preview). Retrieved Oct, 2009, from http://www.iso.org/iso/catalogue_detail.htm?csnumber=36644
- Alain Abran, James W. Moore (2004). Software configuration management. In: Guide to Software Engineering Body of Knowledge. IEEE Computer Society.
- Marakas, George, "Decision Support Systems", Prentice Hall, 1999
- Rivas, Alonso, "Auditoría Informática", Díaz de Santos, 1997
- Sage, Andrew, "Systems Management for Information Technology and Software Engineering (Wiley Series in Systems Engineering)", John Wiley, 1995
- Reifer, Donald, "Software Management", IEEE, 1997

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Jenner, Michael, "Software Quality Management", Wiley, 1995
- Ginac, Frank, "Customer Oriented Software Quality Assurance", Prentice Hall, 1998
- Martin, James, "Information Engineering", Prentice Hall, 1990
- Dujmovic J., Informes de Auditoría del Programa 41 del banco Mundial, 1999
- Debnath, N., Uzal, R. et al "Software Quality Assurance in a Reengineering Project Based on Rapid Evolutionary Prototyping Methodology", 2000
- Arthur, Lowell Jay, "Rapid Evolutionary Development", Wiley, 1992
- Dujmovic, J., Uzal, R. et al "Software Quality Assurance issues in a project based on Rapid Evolutionary Prototyping Methodology, SFSU web page, 1999
- Uzal, R. et al "Rapid Evolutionary Prototyping of Data Base Applications", Software Engineering IASTED Software Engineering Conference, Las Vegas, 1998.
- Connell, J. and L. Shafer, "Object-Oriented Rapid Prototyping". Yourdon Press / Prentice Hall, 1995.
- Página Web del Instituto de Ingeniería de Software (CMU) <http://www.cmu.edu>
- James Womack, Daniel Jones y Daniel Roos, "The machine that changed the world: The story of Lean Production", HarperBusiness, 1991.
- <http://www.biblioteca.secyt.gov.ar>. Journal "Advances in Engineering Software"
- <http://www.biblioteca.secyt.gov.ar>. Book. Editorial Elseiver. Data Communications, the Internet, and Electronic Commerce
- <http://www.springerlink.com/>. Annals of Software Engineering
- <http://www.springerlink.com/>. Book. Encyclopedia of Information Systems
- <http://www.ieee.org/>. Journals and Conferences in IEEE Xplore.

5. Metodología de la Investigación Científica II

5.1. Objetivos

Profundizar los conceptos específicos y desarrollar habilidades necesarias para elaborar la Tesis correspondiente a la Maestría en Ingeniería de Software. Estudiar y adquirir la capacidad de aplicar los aspectos centrales de un proceso de investigación y conocer los mecanismos para su transformación en una Tesis de Maestría en Ingeniería de Software.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

Posibilitar una definición consistente del ámbito de la temática o espacio de problema a ser investigado y seleccionar las estrategias teórico-metodológicas adecuadas. Adquirir los conocimientos básicos para la presentación de la Tesis de la Maestría en Ingeniería de Software y para encarar su defensa

5.2. Contenidos conceptuales mínimos

El Método Científico y su aplicación a diferentes ciencias. Adquisición del Conocimiento Científico. Validación (Corroboración) en el Método Científico. Teorías, Teoremas, Hipótesis, etc. en la Metodología de la Ciencia. Técnica y Ciencia. El Lenguaje Científico. Ciencia y Lenguajes Formales. El Método Científico en Computación. La Educación y el Método Científico.

La aplicación de las ideas científicas: de la explicación a la acción.

Explicación: Contestación a los porqués. Explicación no científica. Subsunción científica. Explicación interpretativa. Explicación interpretativa y reducción a leyes. Poder explicativo. Funciones y alcances.

La contrastación de las ideas científicas: de la observación a la inferencia.

Observación: Hecho. Observabilidad. Objetivación. Datos y evidencia. Funciones.

Medición: Cuantificación numérica. Valor medido. Cómputo. Escala y unidad. Técnicas. Resultado.

Experimento: Cambio planificado. Control. Proyecto. Significación. Prueba de la contrastación. Funciones.

La inferencia científica: Inferencia. Contrastación de proposiciones observacionales.

Contrastación de hipótesis. Confirmación y refutación. Una historia ejemplar: Torricelli.

Contrastación de teorías. Examen de teorías.

5.3. Crédito horario total: 80 horas

5.4. Crédito horario de formación práctica: 25 horas

5.5. Metodología de cursado: presencial. Dictado con exposición conceptual.

5.6. Método de Evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Evaluación formal mediante test de comprobación. Presentación y aprobación de un proyecto.

CORRESPONDE ORDENANZA N°

005-10

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

5.7. Bibliografía

- B. Russell, Significado y verdad, Ed. Ariel, Barcelona, 1990.
- M. Bunge, La Investigación Científica, Ed. Ariel, Barcelona, 1989.
- K. Popper, La Lógica de la investigación científica, Ed. Tecnos, Madrid, 1962.
- Stephen Thornton, "Karl Popper", The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.), URL = <<http://plato.stanford.edu/archives/win2002/entries/popper/>>
- Thomas Kuhn, The Structure of Scientific Revolutions, Univ. of Chicago Press, 1962
- Feyerabend, Paul. "Diálogo sobre el método", en: Estructura y desarrollo de la Ciencia, de Feyerabend, Radnitzky, Stegmüller, y otros. Alianza, Madrid, 1984.
- Lakatos, Imre. "La metodología de los Programas de investigación científica". Alianza. Madrid. 1993.
- John Preston, "Paul Feyerabend", The Stanford Encyclopedia of Philosophy (Summer 2002 Edition), Edward N. Zalta (ed),
URL = <<http://plato.stanford.edu/archives/sum2002/entries/feyerabend/>>.
- B. Russell, "The Scientific Outlook", Allen and Unwin, London, 1954.
- Morton, Adam. A Guide Through the Theory of Knowledge (Third Edition) Oxford: Blackwell Publishing. 2002. ISBN 1405100125
- Contemporary Debates in Epistemology, Peter Sosa and Matthias Steup (eds.). Blackwell Publishing. 2005.

6. Análisis de artefactos de software

6.1. Objetivos

Posibilitar, a los alumnos de la Maestría en Ingeniería de Software, análisis de fortalezas y debilidades de productos del desarrollo de software y utilizados en dicho desarrollo. Realizar un estudio de fortalezas y debilidades de las herramientas para la definición de especificaciones, herramientas para el diseño de productos de software, herramientas de generación de código, lenguajes y ambientes de programación correspondientes a los diversos paradigmas (lógico, algorítmico, orientado a objetos y funcional). Efectuar un estudio comparativo de herramientas de testing, de ingeniería reversa y de mantenimiento de productos de software.

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

6.2. Contenidos conceptuales mínimos

Análisis de fortalezas y debilidades de productos del desarrollo de software y utilizados en dicho desarrollo. Estudio de fortalezas y debilidades de las herramientas para la definición de especificaciones, herramientas para el diseño de productos de software, herramientas de generación de código, lenguajes y ambientes de programación correspondientes a los diversos paradigmas (lógico, algorítmico, orientado a objetos y funcional). Estudio comparativo de herramientas de testing, de ingeniería reversa y de mantenimiento de productos de software. Uso intenso del concepto de "slicing" de programas en el aseguramiento de calidad, en el testing y en el mantenimiento de productos de software. Estudio detallado del lenguaje funcional Haskell y de sus posibilidades en el ámbito de la especificación, implantación y verificación.

6.3. Crédito horario total: 80 horas

6.4. Crédito horario de formación práctica: 30 horas

6.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

6.6. Tipo de evaluación

- Desarrollo de casos prácticos correspondientes a cada tópico
- Evaluación formal mediante test de comprobación.

6.7. Bibliografía

Básica:

Neilsen, F., Neilsen, H. y Hankin, C., "Principles of Sprogram Análisis", Springer-Verlag New York, Inc, 1999

N.D. Singpurwalla and S.P. Wilson Statistical Methods in Software Engineering (Reliability and Risk), Springer-Verlag, New York, Inc., 1999

- ISO/IEC 9126-1991 International Standard, Information technology – Software product evaluation – Quality characteristics and guidelines for their use
- ISO/IEC 14598-5:1998 International Standard, "Information technology -- Software product evaluation" -- Part 5: Process for evaluators, R. Jain, The Art of computer systems Performance Analysis. J. Wiley, 1991

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

Complementaria:

- Paul Jorgensen, "Software Testing- A Craftsman's approach", CRC Press, 1995
- B. Marick, "The Craft of Software Testing - Object based and Object oriented Testing", Prentice Hall, 1995.
- J.J. Dujmovic, CSc841 Reader: Computer Performance Evaluation. SFSU, 1996.A.O. Allen, Probability, Statistics, and Queueing Theory with Computer Science Applications. Second Edition, Academic Press, 1990.
- A.O. Allen, Introduction to Computer Performance Analysis with Mathematica. Academic Press, 1994.
- T. Browning, Capacity Planning for Computer Systems. Academic Press, 1995.
- R. Grace, The Benchmark Book. Prentice Hall, 1996. M.K. Molloy, Fundamentals of Performance Modeling. Macmillan, 1989.
- E.D. Lazowska et al., Quantitative System Performance. Prentice-Hall, 1984.
- L. Lipsky, Queuing Theory. Maxwell Macmillan, 1992.
- K.S. Trivedi, Probability and Statistics with Reliability, Queuing, and Computer Science Applications. Prentice-Hall, 1982.
- D. Ferrari et al., Measurement and Tuning of Computer Systems. Prentice-Hall, 1983.
- Cem Kaner, James Bach, Bret Pettichord, "Lessons Learned in Software Testing", Wiley; 1st edition, 2001
- Brian Marick, Testing Foundations, <http://www.testing.com>
- Brian Hambling, Managing Software Quality, Mc Graw-Hill, 1996.
- Bret Pettichord, <http://pettichord.com>
- <http://www.biblioteca.secyt.gov.ar>. Journal "Environmental Modelling and Software"
- <http://www.biblioteca.secyt.gov.ar>. Book. Editorial Elseiver. Book Systems Analysis and Design
- Annals of Software Engineering, <http://www.springerlink.com>
- Requirements Engineering, <http://www.springerlink.com>



Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Software and Systems Modeling, <http://www.springerlink.com>
- International Journal on Software Tools for Technology Transfer (STTT), <http://www.springerlink.com>
- Journals and Conferences in IEEE Xplore, <http://www.ieee.org>

7. Arquitectura de sistemas de software

7.1. Objetivos

Transferir, a los alumnos de la Maestría en Ingeniería de Software, los conocimientos y habilidades instrumentales necesarios para encarar proyectos de software de alta complejidad, para el manejo de los niveles de abstracción de la arquitectura de dichos sistemas, para diseñar la arquitectura de productos de software de alta complejidad. Modelos y para utilizar eficazmente métodos formales orientados a la arquitectura. Posibilitar la utilización de herramientas de generación de instancias específicas de la arquitectura las destinadas a la evaluación de la arquitectura de sistemas complejos existentes

7.2. Contenidos conceptuales mínimos

Los proyectos de software de alta complejidad: El nivel de abstracción de la arquitectura. El diseño de la arquitectura de productos de software de alta complejidad. Modelos y métodos formales orientados a la arquitectura. Herramientas de generación de instancias específicas de la arquitectura. Evaluación de la arquitectura de sistemas complejos existentes. Estilos de arquitectura de sistemas de software de mayor difusión. El rol del dominio del espacio de problema en la especialización de una arquitectura general a una familia de aplicaciones en particular. Diseño de la arquitectura de sistemas basados en el conocimiento. Consideración de agentes inteligentes en el diseño de la arquitectura. Otros paradigmas para el diseño de la arquitectura de sistemas de información: Web Services y Service-Oriented Architectures.

7.3. **Crédito horario total: 80 horas**

7.4. **Crédito horario de formación práctica: 30 horas**

7.5. **Metodología de cursado:** presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

CORRESPONDE ORDENANZA N°

005-101


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
J.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

7.6. Tipo de evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Test de verificación formal de conocimientos

7.7. Bibliografía

Básica:

- Bass, Len; Clements, Paul; Kazman, Rick; Software Architecture in Practice, 2nd Edition, Addison-Wesley Professional; 2nd edition, 2003
- Clements, Paul; Bachmann, Felix; Bass, Len; Garlan, David; Ivers, James; Little, Reed; Nord, Robert; Stafford, Judith, Documenting Software Architectures: Views and Beyond, Addison-Wesley Professional; 1st edition, 2002
- Shaw, Mary; Garlan, David; Software Architecture: Perspectives on an Emerging Discipline, Prentice Hall; 1 edition, 1996
- Gonzalez, Avelino, Dankel Douglas, "Engineering of Knowledge-Based Systems", ISBN:9780132769402, Publisher:Prentice Hall, 1993
- Petrie, Charles, Agent-Based Engineering, the Web, and Intelligence, Stanford Center for Design Research, <http://www-cdr.stanford.edu/NextLink/Expert.html>
- Leary, Timothy, Intelligence Agents, ISBN 0-452-27025-1); Ronin Publishing, 1996
- Barry, Douglas, Web Services and Service-Oriented Architectures, Editorial : Morgan Kaufmann, 2003, ISBN: 978-1-55860-906-8

Complementaria:

- Erl, Thomas; Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall PTR, 2004
- Hohmann, Luke; Beyond Software Architecture: Creating and Sustaining Winning Solutions, Addison-Wesley Professional; 1st edition, 2003
- Buschmann, Frank; Meunier, Regine; Rohnert, Hans; Sommerlad, Peter; Stal, Michael; Sommerlad, Peter; Stal, Michael, Pattern-Oriented Software Architecture, Volume 1: A System of Patterns, John Wiley & Sons; 1 edition, 1996.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Austin, Robert D., *Measuring and Managing Performance in Organizations*, Dorset House Publishing Company, Incorporated, 1996
- Fowler, Martin, *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional; 1st edition, 2002
- Kerievsky, Joshua, *Refactoring to Patterns (Addison-Wesley Signature Series)*, Addison-Wesley Professional, 2004
- Kruchten, Philippe, *The Rational Unified Process*, Addison-Wesley Professional; 2nd edition, 2000
- Leary, Timothy, *Intelligence Agents*, ISBN 0-452-27025-1); Ronin Publishing, 1996
- *Journal Environmental Modelling and Software*, <http://www.biblioteca.secyt.gov.ar>
- *Theories, Methodologies, and Foundations*, Elsevier, <http://www.biblioteca.secyt.gov.ar>
- *Annals of Software Engineering*, <http://www.springerlink.com>
- *Programming and Computer Software*, <http://www.springerlink.com>
- *Software and Systems Modeling*, <http://www.springerlink.com>
- *Lecture Notes in Computer Science*, <http://www.springerlink.com>
- *International Journal on Software Tools for Technology Transfer (STTT)*, <http://www.springerlink.com>
- *Encyclopedia of Information Systems*, <http://www.springerlink.com>
- *Journals and Conferences in IEEE Xplore*, <http://www.ieee.org>
- *International Journal of Knowledge-Based and Intelligent Engineering Systems* ISSN: 1327-2314 <http://www.iospress.nl/loadtop/load.php?isbn=13272314>
- *International Journal of Knowledge-Based and Intelligent Engineering Systems* ISSN: 1327-2314 <http://www.iospress.nl/loadtop/load.php?isbr=13272314>

8. Ciclo de cursos optativos

Los siguientes cursos propuestos se dictarán como optativos, los cuales dos de ellos deberán ser seleccionados por los alumnos para cubrir la cantidad total de cursos necesarios según la estructura general de la carrera.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
J.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

Es dable destacar que el siguiente listado no es limitativo a que puedan agregarse en el futuro nuevos cursos optativos conforme al avance de la tecnología y la ciencia.

8.1. Sistemas de tiempo real

8.1.1. Objetivos

Posibilitar, a los alumnos de la Maestría en Ingeniería de Software la real comprensión losl conceptos de Tiempo Real y de Sistemas de Tiempo Real. Analizar en profundidad las características de los Sistemas de Tiempo Real. Encarar la especificación de Requerimientos, Análisis, Diseño e Implementación de Sistemas de Tiempo Real. Efectuar la Especificación y Verificación Formal de Sistemas de Tiempo Real. Enacarar, efectivamente, el Modelado y Simulación de Sistemas de Tiempo Real.

8.1.2. Contenidos conceptuales mínimos

Concepto de Tiempo Real y de Sistema de Tiempo Real. Las características de los Sistemas de Tiempo Real. Requerimientos, Análisis, Diseño e Implementación de Sistemas de Tiempo Real. Especificación y Verificación Formal de Sistemas de Tiempo Real. Modelado y Simulación de Sistemas de Tiempo Real.

8.1.3. Crédito horario total: 80 horas

8.1.4. Crédito horario de formación práctica: 30 horas

8.1.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.1.6. Tipo de evaluación

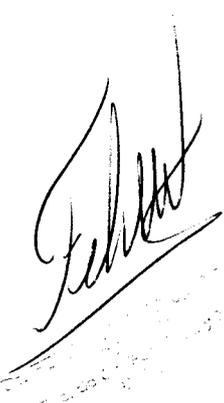
- Desarrollo de casos prácticos correspondientes a cada tópico
- Evaluación formal mediante test de comprobación.

8.1.7. Bibliografía

Básica:

- Real-Time Systems Design and Analysis. Philip A. Laplante. Wiley-IEEE Press; 3 edition (April 20, 2004).
- Qing Li, Yao, Caroline "Real-Time Concepts for Embedded Systems", CMPBooks, 2003.

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
J.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Sifakis, J., Yovine, S., "Compositional Specification of Timed Systems", In 13th Annual Symposium on Theoretical Aspects of Computer Science, pages 347-359, Grenoble, France, LNCS 1046, Springer-Verlag, February 1996
- E. Clarke, O. Grumberg and D. Long, "Model Checking", Springer-Verlag Nato ASI Series F, Volume 152, 1996.
- Bouyssounouse, Bruno, Embedded Systems Design: The ARTIST Roadmap for Research and Development (Lecture Notes in Computer Science), Springer, 2005)
- M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis y S. Yovine, KRONOS: A model-checking tool for real-time systems, Computer Aided Verification (CAV 98), Lectures Notes in Computer Science 1427, Springer Verlag. 1998.
- 25 Years of Model Checking: History, Achievements, Perspectives (Lecture Notes in Computer Science) Publisher: Springer Publication: 2008.

Complementaria:

- R. Alur, C. Courcoubetis, N. Halbwachs, T. Henzinger, P. Ho, X. Nicollin, A. Olivero, J. Sifakis y S. Yovine. The Algorithmic Analysis of Hybrid Systems. Theoretical Computer Science B vol. 137. IEEE Enero 1995.
- C. Daws, A. Olivero, S. Tripakis, and S. Yovine, The tool KRONOS. In Hybrid Systems III, Verification and Control, pages 208-219. Lecture Notes in Computer Science 1066, Springer-Verlag, 1996.
- T.A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. Information and Computation, 111(2):193--244, 1994.
- J. Sifakis and S. Yovine, Compositional specification of timed systems. In 13th Annual Symposium on Theoretical Aspects of Computer Science, STACS'96, pages 347--359, Grenoble, France, Fe'vrier 1996. Lecture Notes in Computer Science 1046, Springer-Verlag.
- Peterson J., Petri Net Theory and the Modelling of Systems. Prentice Hall, Englewood Cliffs, New Jersey, 1981.
- Ghezzi, Mandrioli, Morascas and Pezze, A general way to put time in Petri nets. Proceeding 5th International Workshop on Software Specification and Design, Pittsburgh, May 89, pages 60-67, IEEE Computer Society Press, 1989.

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

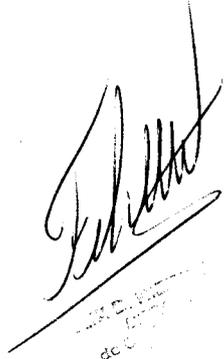
- M. Zuberek., Timed Petri nets: Definitions, Properties, and applications. Microelectronics and Reliability, 31(4):627-644, 1991.
- Ghezzi, Mandrioli, Morascas and Pezze. A unified high-level Petri Net model for time critical system. IEEE Transactions on Software Ingeniering 17(2), 160-172. 1991.
- Buy and Sloan. Analysis of Real-Time Programs with Simple Time Petri Nets. ACM 1994.
- High-level Petri Nets – Concepts, Definitions and Graphical Notation. ISO/IEC 15909, 1997, Version 3.4.
- Eric Y. T. Juan, Jeffrey J. P. Tsai, Tadao Murata, Yi Zhou, Reduction Methods for Real Time Systems Using Delay Time Petri Nets. IEEE Transaction on Software Engineering. Vol 27, No. 5. May 2001.
- Henzinger, Nicollin, Sifakis and Yovine. Symbolic model checking for real-time systems. Information and Computation, 111(2):193-244, 1994.
- Ian Sommerville, Ingeniería de Software (6ª Edición), Pearson Educación, 2002
- Lecture Notes in Computer Science, <http://www.springerlink.com>
- Journals and Conferences in IEEE Xplore, <http://www.ieee.org>

8.2. Sistemas financieros

8.2.1. Objetivos

Transferir, a los alumnos de la Maestría en Ingeniería de Software, los conocimientos y habilidades instrumentales relacionadas con la Evaluación Financiera de Proyectos de Inversión. Posibilitar el estudio del proyecto de inversión como flujo de contado (“cash flow”). Aplicar adecuadamente los criterios de evaluación de un proyecto de inversión tales como el período de recupero, el Valor Actual Neto (VAN), utilizar el VAN para evaluar proyectos alternativos. Calcular y usar adecuadamente la Tasa Interna de Retorno (TIR) de un proyecto y realizar un estudio comparativo de herramientas computacionales orientadas a la evaluación de proyectos de inversión. Introducir al alumnos en los elementos básicos de administración financiera tales como riesgo y rendimiento, cartera de inversiones, rendimiento esperado de una cartera, entre otros.

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

8.2.2. Contenidos conceptuales mínimos

Evaluación Financiera de Proyectos de Inversión. El proyecto de inversión como flujo de contado ("cash flow"). Criterios de evaluación de un proyecto de inversión. El período de recupero. Características. El Valor Actual Neto (VAN) de un Proyecto de Inversión. Cálculo del VAN de distintos proyectos. Utilización del VAN para evaluar proyectos alternativos. La Tasa Interna de Retorno (TIR) de un proyecto. Estudio comparativo de herramientas computacionales orientadas a la evaluación de proyectos de inversión. Riesgo y rendimiento. El rendimiento esperado de una cartera. Modelo de fijación de precios de los activos financieros. Línea del Mercado de Capitales. El intercambio compensatorio riesgo – rendimiento. Cálculo de Beta. Beta de una cartera. La diversificación. Herramientas computacionales necesarias para la gestión del riesgo. Concepto de Cuadro de Mando Integral (BSC). Definición de Indicadores en el Cuadro de Mando Integral (BSC). Indicadores Financieros y no Financieros en el BSC. Ejemplos de BSC en distintos tipos de organizaciones. Selección de herramientas computacionales para implantar el concepto de BSC. Metodología de implantación del concepto de BSC. Ejemplos de BSC en distintos tipos de organizaciones. Selección de herramientas computacionales para implantar el concepto de BSC. Metodología de implantación del concepto de BSC.

8.2.3. Crédito horario total: 80 horas

8.2.4. Crédito horario de formación práctica: 30 horas

8.2.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.2.6. Tipo de evaluación

- Desarrollo de casos prácticos correspondientes a cada tópico
- Evaluación formal mediante test de comprobación.

8.2.7. Bibliografía

Básica:

- Brealey, Richard, Myers, Stewart "Dirección Financiera", Mc Graw Hill, 1996
- Kaplan R., Norton D. "Cuadro de Mando Integral" ("Balanced Scorecard"), Gestión 2000, 1997.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

Complementaria:

- Olve, Nils-Göran, et al "Implantando y Gestionando el Cuadro de Mando Integral", Gestión 2000, HBS Press, mayo del 2000
- Marakas, George, "Decision Support Systems", Prentice Hall, 1999
- Emery, Douglas, Finnerty, "Administración Financiera Corporativa", Prentice Hall, 2000
- Cissell, Robert; Cissel Helen, Flaspohler, David "Matemáticas Financieras", CECSA, 1998
- Emery, Douglas, Finnerty, "Administración Financiera Corporativa", Prentice Hall, 2000
- Mondino, Diana; Pendás, Eugenio "Finanzas para Empresas Competitivas" Granica, 1994
- Welsch, Glenn A. et al. Presupuestos. Planificación y control de utilidades. 2ª edición en español. México. Prentice-Hall Hispanoamericana. 1990.
- Van Horne, James C. Administración Financiera. 2ª edición en español. México. Prentice-Hall Hispanoamericana. 1993.
- Weston, J. F., Copeland, T. E. Finanzas en administración. (2 volúmenes). 3ª edición en español. México. McGraw-Hill. 1988.

8.3. Análisis comparativos de paradigmas de programación

8.3.1. Objetivos

Posibilitar, a los alumnos de la Maestría en Ingeniería de Software la real comprensión las características fundamentales de los paradigmas de programación algorítmico, funcional, lógico y orientado a objetos. Realizar un análisis comparativo de las fortalezas y debilidades de cada uno de los paradigmas. Desarrollar de aplicaciones con un enfoque multiparadigma y lograr avances significativos en programación multiparadigma.

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///..

8.3.2. Contenidos conceptuales mínimos

Paradigmas que implican una solución procedural u operacional, paradigmas que implican solución demostrativa y paradigmas que incluyen tipos de soluciones declarativas. Paradigmas con “efecto de lado” (variables estrechamente relacionadas con direcciones de la memoria del computador): Paradigma imperativo y paradigma orientado a objetos. Paradigmas “sin efecto de lado” (excluyendo los que tradicionalmente son denominados paradigmas funcionales). Soluciones funcionales procedurales y soluciones funcionales declarativas. Paradigmas declarativos: Hechos, reglas, restricciones, ecuaciones, transformaciones y otras propiedades derivadas del conjunto de valores que configuran la solución. Aspectos innovadores y relevantes de la programación funcional y de la programación lógica. Programación multiparadigma. Desarrollo de casos comparando fortalezas y debilidades de los diversos paradigmas en espacios de problema diversos. Procesamiento paralelo asíncrono (cooperación de procesos paralelos) y procesamiento paralelo síncrono (procesos simples aplicados simultáneamente a múltiples objetos). Desarrollo de casos de estudio de la sintaxis de los lenguajes de programación: criterios generales, elementos sintácticos, modelos de traducción. Casos de estudio del concepto y de la necesidad de las descripciones semánticas, de la equivalencia de programas, de corrección y completitud. Casos de estudio de semánticas denotacional, axiomática y operacional. Para cada lenguaje o tipo de lenguaje se estudiará conceptualmente la interrelación entre “diseño e implantación”. Cada lenguaje abordado será estudiado individualmente en término de conceptos para luego encarar un análisis comparativo consistente.

8.3.3. Crédito horario total: 80 horas

8.3.4. Crédito horario de formación práctica: 30 horas

8.3.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.3.6. Tipo de evaluación

- Desarrollo de casos prácticos correspondientes a cada tópico
- Evaluación formal mediante test de comprobación.

8.3.7. Bibliografía

Básica


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N°

005 - 10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Van Roy, Peter, Haridi, Seif, Concepts, Techniques, and Models of Computer Programming, MIT Press, 2004
- Norvig, Peter, Paradigms of Artificial Intelligence Programming, Morgan Kaufmann, 1992
- Thompson, Simon, Haskell: The Craft of Functional Programming, Addison Wesley; 2nd edition, 1999
- Maluszynski, Jan, Logic, Programming and Prolog, John Wiley & Sons Ltd, 2da edición (electrónica) en <http://www.ida.liu.se/~ulfni/lpp/>
- Piattini, Mario, Garzás, Javier, Object-oriented Design Knowledge: principles, heuristics, and best practices, Idea Group, 2007
- Van Roy, Peter, Multiparadigm Programming in Mozart/Oz: Second International Conference, MOZ 2004, Charleroi, Belgium, October 7-8, 2004, Revised Selected Papers (Lecture Notes in Computer Science), Springer; 1st edition, 2005.

Complementaria:

- Harrison, Neil; Foote, Brian; Rohnert, Hans, Pattern Languages of Program Design 4, Software Patterns Series, Addison Wesley Publishing Company; 1st edition, 1999
- Daume III, Hal, Yet Another Haskell Tutorial, 2002-2004, <http://www.isi.edu/~hdaume/htut/>
- Apt, K.R. From Logic Programming to Prolog, Prentice Hall, 1996.
- Bird, R. Introducción a la Programación Funcional, Prentice Hall, 2000.
- Pratt, T.W.; Zelkowitz, M.V. Lenguajes de programación: diseño e implementación, Prentice-Hall, 1998
- Scott, M.L. Programming Language Pragmatics, Morgan Kaufmann Publishers, 2000
- Horowitz, E. Fundamentals of Programming Languages, Computer Sc. Press, 1984.
- Loudon, K.C. Programming Languages: Principles and Practice, PWS Publishing, 1993.
- Sethi, R. Lenguajes de programación: conceptos y constructores, Addison Wesley, 1992.
- Programming and Computer Software, <http://www.springerlink.com>


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Lecture Notes in Computer Science, <http://www.springerlink.com>
- Journals and Conferences in IEEE Xplore, <http://www.ieee.org>

8.4. Tópicos avanzados de especificación y verificación formal de software

8.4.1. Objetivos

Realizar un estudio en profundidad de los pro y contra de los Métodos Formales de los estilos de especificaciones, de los estilos de desarrollo y de los grados de rigor. Incrementar las habilidades de los alumnos en especificaciones formales, especificaciones orientadas a la propiedad, especificaciones orientadas al modelo. Estudiar los estilos aplicativo, imperativo y concurrente. Estudiar las funciones totales y parciales y sus respectivas posibilidades. Considerar definiciones explícitas e implícitas. Estudiar operadores, módulos, esquemas y objetos. Lograr una combinación útil y armoniosa de de Técnicas Formales y semi-formales.

8.4.2. Contenidos conceptuales mínimos

Métodos Formales. Pro y contra de los Métodos Formales. Estilos de especificaciones Estilos de desarrollo. Grados de rigor. Especificaciones formales. Especificaciones orientadas a la propiedad. Especificaciones orientadas al modelo. Estilos: aplicativo, imperativo, concurrente. Funciones totales y parciales. Definiciones explícitas e implícitas. Operadores. Módulos. Esquemas. Objetos. Lógica de Orden Superior.. Combinación de Técnicas Formales y semi-formales. Captura de requerimientos. Semántica formal. Lógica Temporal .Chequeo de Modelos.

8.4.3. Crédito horario total: 80 horas

8.4.4. Crédito horario de formación práctica: 30 horas

8.4.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.4.6. Tipo de evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Evaluación formal mediante test de comprobación.

CORRESPONDE ORDENANZA N° , 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac de Cs. Fco. Mat. y Nat
U.N.S.L



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

8.4.7. Bibliografía

- Richard Sharpe, "Formal Methods Start to Add up Again", *Computing*, 8 January 2004.
- Axel van Lamsweerde, "Formal specification: a roadmap", Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, Pages: 147 – 159, 2000. ACM New York, NY, USA.
- Ann Sobel and Michael R. Clarkson, "Formal Methods Application: An Empirical Tale of Software Development", *IEEE Transactions on Software Engineering*, Vol. 28, N° 3, Marzo 2002.
- J.P. Bowen and M.G. Hinchey, "Ten Commandments of Formal Methods ... Ten Years Later". *IEEE Computer*, 39(1):40-48, January 2006.
- Bowen, J., & Hinchey, M., "Ten commandments of formal methods". *IEEE Computer*. Abril, 1995.
- Philip E. Ross, The Exterminators, *IEEE Spectrum*, pages 36–41, September 2005.
- Dasso, A., Funes, A., "Formal Methods in Software Engineering" en Encyclopedia of Information Science and Technology, Idea Group, Enero de 2005.
- Sten Agerholm and Peter G. Larsen, "A Lightweight Approach to Formal Methods", In *Proceedings of the International Workshop on Current Trends in Applied Formal Methods*, Boppard, Germany, Springer-Verlag, October 1998.
- Daniel Jackson and Jeannette Wing, "Lightweight Formal Methods", *IEEE Computer*, April 1996.
- Daniel Jackson, "Alloy: A Lightweight Object Modelling Notation", *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Volume 11, Issue 2 (April 2002), pp. 256-290.
- John Derrick and Eerke Boiten. Refinement in Z and Object-Z, Foundations and Advanced Applications. Springer-Verlag, 2001.
- Graeme Smith. The Object-Z Specification Language. Kluwer Academic Publishers, 2000.
- Roger Duke and Gordon Rose. Formal Object-Oriented Specification Using Object-Z. MacMillan, 2000.



Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- The Maude System. SRI International. University of Illinois. <http://maude.cs.uiuc.edu/>
- Dang Van, H.; George, C.; Janowski, T.; Moore, R. (Eds.). *Specification Case Studies in RAISE*. Springer. 2002.
- The RAISE Language Group. "The RAISE Specification Language". Prentice-Hall International, 1992.
- The RAISE Method Group. "The RAISE Development Method". Prentice-Hall International, 1995.
- Steve Schneider, *The B-Method: An Introduction*, Palgrave, Cornerstones of Computing series, October 2001. ISBN 0-333-79284-X.
- Kevin Lano, *The B Language and Method: A Guide to Practical Formal Development*, Springer-Verlag, FACIT series, 1996. ISBN 3-540-76033-4.
- A. T. Nakagawa, T. Tamai, A.T. Nakagawa, "CAFE: An Industrial-Strength Algebraic Formal Method". Kokichi Futatsugi (Editor), Elsevier Science, 2000.
- Deri Sheppard, "An Introduction to Formal Specification with Z and VDM" (The McGraw-Hill International Series in Software Engineering), McGraw Hill, 1994.
- Klaus Schneider. *Verification of Reactive Systems: Formal Methods and Algorithms*. Texts in Theoretical Computer Science. 2003.
- E. Clarke, Bernd-Holger Schlingloff. "Model Checking", *Handbook of Automated Reasoning*, pp. 1635-1790, 2001.
- E. Clarke, O. Grumberg, D. Peled. "Model Checking". MIT Press, 1999.
- B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen. *Systems and Software Verification: Model-Checking Techniques and Tools*. Springer. 1999.
- Michael Huth and Mark Ryan. "Logic in Computer Science: Modelling and Reasoning About Systems", Cambridge University Press, 2004.
- Venema, Yde. "Temporal Logic", in Goble, Lou, ed., *The Blackwell Guide to Philosophical Logic*. Blackwell. 2001.
- Z. Manna and Amir Pnueli, *The Temporal Logic of Reactive and Concurrent Systems: Specification*, Springer-Verlag, New York, 1991.



Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fís. Mat. y Nat.
(UNSL)

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- John Fitzgerald, Peter G. Larsen. Validated Designs for Object-oriented Systems. Springer. 2005.
- Funes, A., George, C., "Formalizing UML class diagrams", "capítulo 8 de UML and the Unified Process, Idea Group, 2003.
- Richard Denney, *Succeeding with Use Cases: Working Smart to Deliver Quality*, Addison-Wesley Professional Publishing, 2005, ISBN 0-321-31643-6.
- Bowen, J., & Hinchey, M., "Seven more myths of formal methods". IEEE Software, 12(3). Julio, 1995.
- J.A. Hall, Seven Myths of Formal Methods. *IEEE Software*, 7(5):11-19, September 1990.

8.5. Ingeniería Web

8.5.1. Objetivos

Analizar la evolución del diseño en entorno web. Lograr la aplicación efectiva de los principios y criterios de la Ingeniería de Software en entorno web. Analizar la naturaleza específica de las aplicaciones web. Estudiar y aplicar las características específicas del ciclo de vida del software en entorno web. Encarar la Especificación de Requerimientos, Análisis y Diseño en entorno web. Concretar la implementación de aplicaciones web. Realizar el testing de aplicaciones web. Estudiar y aplicar las singularidades del mantenimiento de aplicaciones web.

8.5.2. Contenidos conceptuales mínimos

Ingeniería Web entendida como un proceso para crear, implantar y mantener aplicaciones y sistemas Web de alta calidad. Lenguajes Web. Plataformas de desarrollo Web. Caso práctico: Análisis comparativo Plataforma Java vs. Microsoft .NET. Estándares en la Web. Arquitecturas de aplicaciones en la Web. Cliente/Servidor "n capas". Sistemas Distribuidos. Testing de aplicaciones Web. Web sintáctica y Web semántica. Web Semántica: Representación del Conocimiento. Ontologías.

8.5.3. Crédito horario total: 80 horas

8.5.4. Crédito horario de formación práctica: 30 horas

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
UNSL



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

8.5.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.5.6. Tipo de evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Evaluación formal mediante test de comprobación.

8.5.7. Bibliografía

Básica

- Mendes, Emilia, Xavier Mendes, María Emilia, Mosley, "Web Engineering" Springer, 2006
- Suh, Woojong, Web Engineering: Principles and Techniques, Idea Group Inc., 2005 ISBN 1-59140-432-0
- Kappel, Geti, Proll, Birgit, Reich, Siegfried, Retschitzegger, Werner, "Web Engineering: The Discipline of Systematic Development of Web Applications", John Wiley & Sons, 1st Edition, June 2006 ISBN 0-470-01554-3
- Andersson, Eve, Greenspun, Philip, Grumet, Andrew, "Software Engineering for Internet Applications, MIT Press 2006; ISBN 0262511916

Complementaria

- Xueyan Tang , Jianliang Xu, Samuel T. Chanson, "Web Content Delivery (Web Information Systems Engineering and Internet Technologies Book Series), Springer, 2005, ISBN-13: 978-0387243566
- Fraternali, Piero, Koch, Nora, Wirsing, Martin "Web Engineering: managing diversity and complexity of web application development : [hot topics]" Spinger, 2005
- Díaz, Ma Paloma, Montero, Susana, Aedo, Ignacio,"Ingeniería Web y Patrones de Diseño", Pearson Educación, 2004, ISBN: 8420546097. ISBN-13: 9788420546094
- Powell, Thomas, "Web Site Engineering", Prentice Hall, 1999
- Nielsen, J., , Designing Web Usability: The Practice of Simplicity; ISBN: 156205810X, New Riders Publishing ,2000

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat
J.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Powell, T.; Jones, D; Cutts, D., "Web Site Engineering: Beyond Web Page Design", Prentice Hall PTR., 1998
- Rosenfeld, L., Morville, P., Information Architecture for the World Wide Web, O'Reilly, 1998

8.6. Auditoría informática

8.6.1. Objetivos



Lograr que los alumnos de la Maestría en Ingeniería de Software distingan realmente entre Control y Auditoría y establezcan las responsabilidades en cada caso. Estudiar y encarar la Auditoría como verificación de la eficacia de los mecanismos de Control Interno. Distinguir entre Auditoría Interna y Externa. Analizar las características de la Auditoría de Estados Contables. Analizar y aplicar la Auditoría Informática. Estudiar el Ámbito de Incumbencia de la Auditoría Informática. Analizar los capítulos de la Auditoría Informática: Evaluación de los Controles y del Riesgo; Ajuste a los Procedimientos Establecidos, Auditoría de la Seguridad y de la Confidencialidad. Considerar el punto de vista "Auditoría Informática como soporte a la Auditoría Contable - Financiera". Estudiar y aplicar las herramientas de la Auditoría de Fraudes (forense). Estudiar y aplicar el estándar de la "Information System Audit and Control Association" (ISACA): COBIT.

8.6.2. Contenidos conceptuales mínimos



Control y Auditoría: Responsabilidades. La Auditoría como verificación de la eficacia de los mecanismos de Control Interno. Auditoría Interna y Externa. La Auditoría de Estados Contables. La Auditoría Informática: Ámbito de Incumbencia. Capítulos de la Auditoría Informática: Evaluación de los Controles y del Riesgo; Ajuste a los Procedimientos Establecidos, Auditoría de la Seguridad y de la Confidencialidad, Auditoría Informática como soporte a la Auditoría Contable - Financiera; Auditoría de Fraudes (forense). Estándar de la "Information System Audit and Control Association" (ISACA): COBIT. El "Institute of Internal Auditors". El estándar de la "Association of Certified Fraud Examiners" (ACFE). El "American Institute of Certified Public Accountants (AICPA): La certificación CIPT. Normas generales de Auditoría y Marco Legal vigentes en nuestro país. Normas específicas del BCRA.

8.6.3. Crédito horario total: 80 horas

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

8.6.4. Crédito horario de formación práctica: 30 horas

8.6.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.6.6. Tipo de evaluación

- a) Desarrollo de casos prácticos correspondientes a cada tópico
- b) Evaluación formal mediante test de comprobación.

8.6.7. Bibliografía

Básica:

- Hunton, James "et al", "Core Concepts of Information Tecnology Auditing", Wiley, 2004
- Piattini, Mario, "Auditoría Informática: Un enfoque práctico", Díaz de Santos, 2001
- Estándar COBIT de Control y de Auditoría <http://www.isaca.org/cobit>
- El estándar ISO 900x.3 <http://www.quality.org/ISO9000>

Complementaria:

- Estándares de Calidad específicos (casos de estudio) <http://www.hipaa.org>; <http://www.hl7.org>
- Capability Maturity Model Integration <http://www.sei.cmu.edu/cmml/>
- Dujmovic J., Informes de Auditoría del Programa 41 del banco Mundial, 1999
- Editorial Elseiver. Book Theories, Methodologies, and Foundations, <http://www.biblioteca.secyt.gov.ar>.
- Annals of Software Engineering, <http://www.springerlink.com>
- Lecture Notes in Computer Science, <http://www.springerlink.com>
- Journals and Conferences in IEEE Xplore, <http://www.ieee.org>

CORRESPONDE ORDENANZA N°

005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

8.7. Temas avanzados de calidad del software

8.7.1. Objetivos

Analizar: "Control de Calidad vs. Aseguramiento de Calidad en un Proyecto de Software". Estudiar y aplicar a la Gestión de la Calidad en las diversas fases del Ciclo de Vida de un Proyecto. Utilizar herramientas automatizadas de Aseguramiento de la Calidad durante un Proyecto de Software. Estudiar y aplicar los aspectos conceptuales e instrumentales del estándar ISO 900.x.3. Estudiar y analizar la aplicación de la estructura, ámbito de aplicación, fortalezas y debilidades del CMMI. Estudiar el real ambito de aplicabilidad del estándar HL7. Su evolución. Estudiar del estándar HIPAA. Estudiar y analizar la plicación del "Gerenciamiento de la Calidad" visto como un proceso de mejora continua: Aplicar este concepto en el ámbito de la Ingeniería de Software y al de la Tecnología Informática. Adquirir los conceptos y habilidades para el gerenciamiento del proceso de certificación correspondiente a los estándares de Calidad más difundidos.

8.7.2. Contenidos conceptuales mínimos

Control de Calidad vs. Garantía de Calidad vs Aseguramiento de Calidad en un Proyecto de Software. Gestión de la Calidad en las diversas fases del Ciclo de Vida de un Proyecto. Herramientas automatizadas de Aseguramiento de la Calidad durante un Proyecto de Software. Aspectos conceptuales e instrumentales del estándar ISO 900.x.3. Estructura, ámbito de aplicación, fortalezas y debilidades del CMMI. Ámbito de aplicabilidad del estándar HL7. Su evolución. Estudio del estándar HIPAA. Gerenciamiento de la Calidad: Calidad como un proceso de mejora continua: Aplicación de este concepto en el ámbito de la Ingeniería de Software y al de la Tecnología Informática. Gerenciamiento del proceso de certificación correspondiente a los estándares de Calidad más difundidos.

8.7.3. Crédito horario total: 80 horas

8.7.4. Crédito horario de formación práctica: 30 horas

8.7.5. Metodología de cursado: presencial. Dictado con exposición conceptual y realización de trabajos en el laboratorio.

8.7.6. Tipo de evaluación

- Desarrollo de casos prácticos correspondientes a cada tópico
- Evaluación formal mediante test de comprobación.

CORRESPONDE ORDENANZA N° 005-10


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
J.N.S.L.



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

8.7.7. Bibliografía

Básica:

- Evans, Isabel, "Achieving Software Quality through Teamwork", Artech House, 2004
- Godbole, Nina "Software Quality Assurance: Principles And Practice, Alpha Science , 2004
- Lewis, Robert, "Independent Verification and Validation: A life cycle Engineering Process for Quality Software", Wiley, 1992
- Capability Maturity Model Integration <http://www.sei.cmu.edu/cmmi/>
- El estándar ISO 900x.3 <http://www.quality.org/ISO9000>

Complementaria:

- Tian, Jeff, "Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement", Wiley-IEEE Computer Society Press , 2005
- Estándares de Calidad específicos (casos de estudio) <http://www.hipaa.org;>
<http://www.hl7.org>
- Gerard O'Regan, A Practical Approach to Software Quality, Springer; 2002
- Gerald M. Weinberg, Quality Software Management: Anticipating Change, Quality Software Management, Dorset House Publishing Company, 1997
- Frank P. Ginac, Customer Oriented Software Quality Assurance Prentice Hall, 1997
- Osten Oskarsson, Robert L. Glass, An Iso 9000 Approach to Building Quality Software, Prentice Hall, 1995
- Michael G. Jenner, Software Quality Management and ISO 9001: How to Make Them Work for You, Wiley, 1995
- Ronald Kirk Kandt , Software Engineering Quality Practices, Applied Software Engineering, Auerbach, 2005
- Gerard O'Regan, Mathematical Approaches to Software Quality, Springer; 2006


Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat.
U.N.S.L.

CORRESPONDE ORDENANZA N°

005-10



Universidad Nacional de San Luis
Facultad de Ciencias Físico
Matemáticas y Naturales

///...

- Rick Lutowski, Software Requirements: Encapsulation, Quality, and Reuse, Auerbach, 2005
- Neufelder, Ensuring Software Reliability, Quality and Reliability, CRC, 1992
- Michael Haug, Eric W. Olsen, Luisa Consolini, Software Quality Approaches: Testing, Verification and Validation, Springer; 2001
- Richard E. Nance, James D. Arthur, Managing Software Quality, Springer; 2002
- Panagiotis Sfetsos, Agile Software Develop. Quality Assurance, IGI Global, 2007
- Joc Sanders, Eugene Curran, Software Quality: A Framework for Success in Software Development and Support, ACM Press, Addison Wesley Longman; 1994
- Martyn A. Ould, Managing Software Quality and Business Risk, Wiley, 1999
- Isabel Evans, Achieving Software Quality through Teamwork, Artech House Publishers, 2004
- Dirk Meyerhoff, Begona Laibarra, Rob van der Pouw Kraan, Alan Wallet, Software Quality and Software Testing in Internet Times, High-tech Software Quality Management, Springer; 2002
- <http://www.biblioteca.secyt.gov.ar>. Book. Editorial Elseiver. Book Theories, Methodologies, and Foundations
- <http://www.springerlink.com/>. Annals of Software Engineering
- <http://www.springerlink.com/>. Lecture Notes in Computer Science
- <http://www.springerlink.com/>. Innovations in Systems and Software Engineering
- <http://www.ieee.org/>. Journals and Conferences in IEEE Xplore.

9. Tutoría en un Proyecto de Investigación acreditado

Los alumnos podrán optar por realizar una tutoría en un proyecto de investigación acreditado, tal como establece el artículo 26 inciso b de la ordenanza 23/09-CS, o aprobar las asignaturas Metodología de la Investigación Científica I y Metodología de la Investigación Científica II.

ORDENANZA N°

005-10

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat
U.N.S.L

Dr. FERNANDO M. BULNES
SECRETARIO DE CIENCIA Y TCA
Fac. de Cs. Fco. Mat. y Nat
U.N.S.L