

UNIVERSIDAD NACIONAL DEL SUR BAHIA BLANCA		1 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
DISEÑO Y DESARROLLO DE SOFTWARE	CÓDIGO: 5587	
	ÁREA N°: III	

CARRERAS

Licenciatura en Ciencias de la Computación

PROFESOR RESPONSABLE

Dra. Marcela Capobianco – Profesor Adjunto con Dedicación Exclusiva

Dr. Gerardo I. Simari – Profesor Adjunto con Dedicación Semi-Exclusiva

CARGA HORARIA	Teoría 64	Práctica 64	Laboratorio	CANTIDAD DE SEMANAS	16
---------------	--------------	----------------	-------------	---------------------	----

CORRELATIVAS

PARA CURSAR LA MATERIA		PARA APROBAR LA MATERIA	
APROBADAS	CURSADAS	APROBADAS	CURSADAS
Análisis y Diseño de Sistemas	Bases de Datos	Bases de Datos	

DESCRIPCIÓN

Diseño y Desarrollo de Software es precedida en el plan de estudios por las asignaturas "*Análisis y Diseño de Sistemas*" y "*Bases de Datos*" por lo cual los alumnos que cursan esta materia ya poseen habilidades para realizar la extracción de requerimientos y especificación de requerimientos del software, proponer modelos de datos adecuados y realizar diseños de alto nivel. El objetivo principal de este curso es el estudio de los métodos, técnicas y herramientas para diseño, desarrollo y mantenimiento del software, de forma tal de obtener un producto confiable, seguro y con un bajo costo de mantenimiento.

Además de los conceptos especificados en el programa, se espera que los alumnos adquieran y/o refuercen aptitudes para:

- Demostrar conocimiento y habilidades en el área de ingeniería de software.
- Aplicar los modelos y técnicas aprendidas a un proyecto de software asignado, usando el criterio personal para decidir qué herramientas resultan más apropiadas dependiendo de una situación concreta.
- Trabajar como parte de un equipo para producir un producto de software de mediana complejidad que posea una calidad apropiada.
- Encontrar soluciones que presenten un buen compromiso entre calidad y las restricciones existentes de tiempo y recursos.
- Documentar en forma adecuada las distintas etapas de construcción del software.
- No depender de una tecnología en particular y ser capaces de aprender por cuenta propia nuevos modelos, técnicas y tecnologías, apreciando la necesidad de continuar su formación profesional de manera independiente.

65
CPB

UNIVERSIDAD NACIONAL DEL SUR BAHIA BLANCA		2 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
DISEÑO Y DESARROLLO DE SOFTWARE	CÓDIGO: 5587	
	ÁREA N°: III	

METODOLOGÍA DE ENSEÑANZA

Siguiendo el esquema del resto de las materias del plan de estudios de la carrera, la modalidad de enseñanza incluye clases teóricas y prácticas. Se dictarán 8 horas semanales (4 de cada modalidad); además, la cátedra estará disponible en un horario fuera de clase para contestar consultas. En las clases teóricas, el profesor brindará un primer acercamiento al material bibliográfico relacionado con cada uno de los tópicos de la materia, ofreciendo a su vez ejemplos ilustrativos tanto pedagógicos como también basados en casos reales. Por otra parte, las clases prácticas incluirán el desarrollo de trabajos prácticos (al menos uno por unidad, ver Programa Analítico) con el objetivo de afianzar los conceptos estudiados en las clases teóricas.

Como complemento a las clases teóricas y prácticas tradicionales, cada cuatrimestre se invita a distintas empresas de SW de la localidad para que ofrezcan charlas informativas sobre las prácticas que aplican en sus empresas. La intervención de profesionales de la industria genera un espacio de discusión en el que los alumnos pueden comprobar la importancia de la aplicación de los conceptos vistos durante el cuatrimestre en el ámbito profesional e indagar sobre las dificultades que enfrentan las distintas organizaciones al momento de su implementación, confirmando beneficios que surgen de la aplicación de estos conceptos. Por último, el contacto con profesionales ya establecidos le permitirá al alumno despejar dudas acerca de la transición próxima entre la universidad y el ámbito laboral.

MECANISMO DE EVALUACIÓN

Dado el carácter de la materia, una actividad de máxima importancia es la realización de un proyecto de cursado durante el período de dictado de la materia. El conjunto de alumnos se dividirá en comisiones de igual tamaño, cuyo número total dependerá de la cantidad de alumnos en el curso, y cada comisión recibirá una consigna basada en un caso real que requiera la producción de un sistema de software. La consigna será simplificada adecuadamente para asegurar que se puede llevar a cabo durante el período de dictado de la materia, no siendo en todos los casos necesaria la implementación completa del sistema. Con el objetivo de desarrollar y afianzar las habilidades para trabajar en un entorno real de producción de software, en cada etapa del ciclo de vida del proyecto se evaluará la realización de una "rotación" de la asignación de consignas a comisiones; es decir, luego de finalizar cada etapa, cada comisión tomará el trabajo realizado por otra. Al finalizar el proyecto, cada comisión realizará una defensa del mismo, que consistirá en una breve presentación de la experiencia y una discusión con los miembros de la cátedra, quienes en este momento tendrán la oportunidad de evaluar la contribución individual de cada alumno.

Por último, para complementar la evaluación de los alumnos con respecto a su desempeño en el proyecto, se tomarán exámenes parciales durante el cuatrimestre. Para cursar la materia será

65

[Handwritten signature]

UNIVERSIDAD NACIONAL DEL SUR BAHIA BLANCA		3 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
DISEÑO Y DESARROLLO DE SOFTWARE	CÓDIGO: 5587	
	ÁREA N°: III	

necesario aprobar tanto el proyecto como los parciales. La nota global de cada alumno será fijada luego de un examen final escrito; se ofrecerá un sistema de promoción para aquellos alumnos que se desempeñen en forma destacada tanto en el proyecto como en los parciales.

PROGRAMA SINTÉTICO

- 1) Calidad del Software: Del producto y del proceso.
- 2) Responsabilidad y ética profesional.
- 3) Conceptos y principios del diseño. Arquitectura y Diseño. Patrones. Lenguajes de diseño arquitectónico. *Frameworks*. Diseño para requisitos no funcionales. Privacidad, integridad y seguridad en sistemas de información.
- 4) Diseño centrado en el usuario.
- 5) Pruebas del software. Técnicas de prueba. Estrategias de prueba del software. Pruebas orientadas a objetos.
- 6) Software libre. Propiedad intelectual, licenciamiento de software y contratos informáticos. Aspectos legales.
- 7) Ingeniería del software basada en componentes.
- 8) Nociones de sistemas colaborativos. Sistemas de control de versiones.
- 9) Ingeniería de Software de Sistemas de Tiempo Real.
- 10) Reingeniería de software.

PROGRAMA ANALÍTICO

1. Calidad del Software: Del producto y del proceso.

El producto. Evolución del software. Características y aplicaciones del software. Crisis del software. Mitos del software. El proceso. Ingeniería de software: procesos, métodos y herramientas. Visión general. Proceso del software. Modelos del proceso. Modelo lineal-secuencial. Construcción de prototipos. Modelos RAD. Modelos evolutivos: incremental, espiral, desarrollo concurrente, desarrollo ágil. Desarrollo basado en componentes. Métodos formales. Técnicas de cuarta generación. Tecnologías de proceso.

2. Ética del ingeniero de software.

Responsabilidad profesional y ética del ingeniero de software. Confidencialidad. Competencia. Derechos de propiedad intelectual. Sociedades profesionales existentes. Objetivos de las mismas. Código de ética del ACM. Análisis de sus principios. Ejemplos de situaciones concretas de la vida profesional.

3. Conceptos y principios de diseño.

Diseño e ingeniería del software. Proceso de diseño: calidad y evolución del software. Principios

de diseño. Conceptos: abstracción, refinamiento, modularidad, arquitectura del software, jerarquía de control, división estructural, estructura de datos, procedimientos, ocultamiento de información. Diseño modular efectivo: independencia funcional, cohesión y acoplamiento. Heurísticas de diseño. Modelo del diseño. Documentación del diseño. Diseño up-front y ágil. Arquitectura y Diseño. Ejemplos concretos: arquitectura de la Web. Arquitectura en nuestro escritorio. Contexto dentro del proceso de desarrollo. Conceptos básicos. Arquitectura. Componentes. Conectores. Configuración. Estilos arquitectónicos. Patrones arquitectónicos. Modelos arquitectónicos. Diseñando arquitecturas. El proceso de diseño. Concepción de la arquitectura. Herramientas conceptuales fundacionales. Experiencia en acción: estilos y patrones. Arquitecturas específicas de un dominio (DSSA). Patrones. Estilos. Clasificación. Estilos simples: programa principal y subrutina, orientados a objetos. En capas. Máquinas virtuales. Cliente-Servidor. De flujos de datos. Cañerías y filtros. *Blackboards*. Intérpretes. Código móvil. Invocación implícita: *publish/subscribe*, basada en eventos, *peer-to-peer*. Estilos mixtos. Estilos complejos: C2, objetos distribuidos, REST. Recuperación de la arquitectura. Conectores. Tipos de conectores. Selección de conectores apropiados. Arquitecturas específicas de un dominio (DSSA). Concepto. Problemas similares con soluciones similares. Dominio, negocio y tecnología. Conocimiento del dominio. *Frameworks*. Frameworks de implementación de arquitecturas. Evaluando distintos *frameworks*. Diseño para requisitos no funcionales. Privacidad, integridad y seguridad en sistemas de información. Lenguajes de Diseño Arquitectónico. Técnicas de modelado. Lenguaje natural. Ventajas y desventajas. Modelado gráfico informal. Ventajas y desventajas. UML y derivados. Ventajas y desventajas. Lenguajes de descripción arquitectónicos (ADLs). Concepto de ADL. Diseñando para requisitos no funcionales. Confiabilidad, eficiencia, portabilidad, privacidad, integridad y seguridad.

4. Diseño centrado en el usuario.

Diseño de interfaz. Reglas: control al usuario, reducir carga de memoria, construcción de interfaces consistentes. Diseño de interfase de usuario: modelos de diseño y proceso del diseño. Análisis y modelado de tareas. Actividades: definición de objetos y acciones. Problemas. Herramientas de implementación. Evaluación del diseño. Diseño centrado en el usuario. Modelos de Proceso centrados en el usuario. Principios Fundamentales. Actividades. Ingeniería de la usabilidad (IU). Métodos de evaluación de IU.

65


5. Pruebas del software. Técnicas de prueba del software.

Fundamentos: Objetivos, principios y facilidad. Diseño de casos de prueba. Prueba de caja blanca. Prueba del camino básico: notación de grafo, complejidad ciclomática, obtención de casos, matrices de grafos. Prueba de la estructura de control: de condición, de flujo de datos, de bucles. Prueba de caja negra: métodos basados en grafos, partición equivalente, valores límites, comparación. Prueba de entornos especializados. Arquitectura y aplicaciones: interfaces gráficas de usuario, arquitecturas cliente/servidor, documentación y facilidades de ayuda, sistemas de tiempo real. Pruebas en el desarrollo ágil. Estrategias de prueba del software. Enfoque estratégico: verificación y validación, organización, estrategia, criterios de completamiento. Aspectos estratégicos. Prueba de unidad: consideraciones y procedimientos. Prueba de integración: descendente, ascendente, de regresión, de humo, comentarios sobre la prueba. Prueba de validación: criterios, revisión de la configuración, pruebas

UNIVERSIDAD NACIONAL DEL SUR BAHIA BLANCA		5 6
DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN		
DISEÑO Y DESARROLLO DE SOFTWARE	CÓDIGO: 5587	
	ÁREA N°: III	

alfa y beta. Prueba del sistema: de recuperación, de seguridad, de resistencia, de rendimiento. El arte de la depuración: proceso, consideraciones psicológicas, enfoques. Pruebas orientadas a objetos. Ampliando la visión de las pruebas. Prueba de los modelos de DOO: exactitud de los modelos, consistencia de los modelos. Estrategia de pruebas orientadas a objetos: pruebas de unidad, pruebas de integración y pruebas de validación. Diseño de casos de prueba para software OO: implementación de los conceptos de OO al diseño de casos de prueba, aplicabilidad de los métodos convencionales al diseño de casos de prueba, pruebas basadas en errores, impacto de la POO en las pruebas, casos de prueba y jerarquía de clases, diseño de pruebas basados en escenarios, estructuras de prueba superficiales y profundas. Métodos de prueba aplicables al nivel de clases: verificación al azar, prueba de partición. Diseño de casos de prueba inter-clases.

6. Software libre. Propiedad intelectual, licenciamiento de software y contratos informáticos. Aspectos legales. Licencias de software. Historia. Distintos tipos de licencias. Clasificación. Propiedad Intelectual. Contratos informáticos. Aspectos Legales. Software Libre. Software de código abierto. Software privativo. Software como servicio (SAAS). Ingeniería del software libre. Qué es el software libre. Motivaciones. Historia. Ingeniería del software libre. Diferencias con el acercamiento tradicional. La catedral y el bazar. Toma de decisiones en el bazar. Procesos en el software libre. Críticas a la catedral y el bazar. Estudios sobre el software libre.

7. Ingeniería de software basada en componentes. Ingeniería de software basada en componentes. Su actualidad y relevancia. Procesos de ISBC. Ingeniería del dominio. Desarrollo basado en componentes. Clasificación y recuperación de componentes. Economía.

8. Nociones de Sistemas Colaborativos. Sistemas de control de versiones. Motivaciones. Funciones básicas. Historia. Modelo optimista y pesimista de los sistemas de control de versiones. Caso de estudio: CVS. Limitaciones de CVS. Otros sistemas. Modelo centralizado. Caso de estudio: Subversion. Modelo distribuido. Caso de estudio: Darcs. Software libre y de código abierto. Caso de estudio: Git.

9. Ingeniería de Software de Sistemas de Tiempo Real. Introducción a los sistemas de tiempo real. Programación de los sistemas de tiempo real. Programación concurrente. Entorno de ejecución. Fiabilidad y tolerancia a fallos.

10. Reingeniería del software. Reingeniería de procesos de negocios. Reingeniería del software. Ingeniería inversa. Reestructuración. Ingeniería directa.

BIBLIOGRAFÍA

- R. Pressman: *Software Engineering: A Practitioner's Approach*, 8h Ed. McGraw-Hill, 2015.
- R. Taylor, N. Medvidovic, E. Dashofy: *Software Architecture: Foundations, Theory, and Practice*. Wiley, 2010.

UNIVERSIDAD NACIONAL DEL SUR
BAHIA BLANCA

6

6

DEPARTAMENTO DE CIENCIAS E INGENIERÍA DE LA COMPUTACIÓN

DISEÑO Y DESARROLLO DE SOFTWARE

CÓDIGO: 5587

ÁREA N°: III

- P. Jorgensen: *Software Testing: A Craftsman's Approach*, 4th Ed. Auerbach Publications, 2013.
- H. van Vliet: *Software Engineering: Principles and Practice*, 3rd ed. Wiley, 2008.
- I. Sommerville: *Software Engineering*, 10th ed. Addison-Wesley, 2015.
- Z. Qin, J. Xing, X. Zheng: *Software Architecture*. Springer, 2008.
- E. Raymond: *The Cathedral and the Bazaar*, O'Reilly and Associates, 1999.
- M. Bar, K. Fogel: *Open Source Development with CVS*, 3rd Ed. GNU Public License, 2015.
- B. Collins-Sussman, B. W. Fitzpatrick, C. M. Pilato: *Version Control with Subversion – For Subversion 1.7 (Compiled from r5658)*. Creative Commons License, 2011.
- H. Kniberg: *Scrum and XP from the Trenches*, 2nd Ed. Lulu.com, 2015.
- Association for Computing Machinery: *ACM Code of Ethics and Professional Conduct*. Disponible en: <https://ethics.acm.org/code-of-ethics/softwareengineering-code/>
- Consejo Profesional de Ciencias Informáticas de la Provincia de Buenos Aires: Código de Ética Profesional. Disponible en: http://www.cpciba.org.ar/archivos/documentos/codigo_etica_profesional.pdf

AÑO

2018

FIRMA PROFESOR RESPONSABLE



VISADO

COORDINADOR ÁREA

SECRETARIO ACADÉMICO

DIRECTOR
DEPARTAMENTO



Dr. DIEGO C. MARTINEZ
SECRETARIO ACADÉMICO
DPTO. DE CS. E ING. DE LA COMP.
UNIVERSIDAD NACIONAL DEL SUR

Dr. MARCELO A. FALAPPA
DIRECTOR DECANO
DPTO. DE CS. E ING. DE LA COMP.
UNIVERSIDAD NACIONAL DEL SUR