



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

INGENIERIA DE SOFTWARE 3

Carrera/Plan: *Licenciatura en Sistemas*
Plan 2003-07/Plan 2012

Año 2014

Año: 4º

Régimen de Cursada: *Semestral (indicar 1º o 2º)*

Carácter: *Obligatoria*

Correlativas: *Matemática III - Sistemas y Organizaciones*

Profesores: *Gregorio Perichinsky*

Hs. Semanales: *6 hs.*

FUNDAMENTACIÓN

En forma breve explicar la importancia de la asignatura para la formación del futuro profesional y el tipo de aporte específicos que realizará la misma.

OBJETIVOS GENERALES

Completar el ciclo de temas básicos de la Ingeniería de Software dados en los dos cursos anteriores, con el estudio de aspectos de evaluación y seguimiento de proyectos, control de calidad de sistemas de software y auditoría de los mismos.

Desarrollar el estudio de casos concretos.

CONTENIDOS MINIMOS

- Calidad de software.
- Auditoría y peritaje de sistemas.
- Costeo.
- Seguimiento y evaluación de proyectos.

PROGRAMA ANALÍTICO

Sistema de información (SI) es un conjunto de elementos orientados al tratamiento y administración de datos e información.

Objetivos de la Organización y la Tecnología Informática. Actividades Técnicas de Trabajo. Información. Recursos. Personas. Tipificación de los Sistemas de Información. Estructura. Comportamiento. Tipo Diana Scott. Dominios. Atributo-Valor. Conocimiento. Matemática. Sociología. Ciclos de Vida de los SI. Modelo Pirámide y Triangular. Sistema de procesamiento (TPS). Sistemas de información gerencial (MIS). Sistema de soporte de decisiones (DSS). Sistemas de información ejecutiva (EIS). Sistemas de automatización (OAS). Sistemas de Planificación de Recursos (ERP). Sistema experto (SE). Sistemas Estratégicos. Era Post Industrial. Paquetes de Conexión Física y Lógica. Redes. Evolución ARPANET-Internet. WWW y Protocolos y Servicios. Congestión de Red. Bucle Abierto y Cerrado. Variables de Diseño. Conmutación de Paquetes y Circuitos. Puertas. Rutas. Colas. Cabecera y Cola de paquetes y segmentos. Función transaccional. Productos. Gestión de un SI. Sistemas de Soporte de Decisiones. Modelado. Simulación. Sistemas de



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Decisión Ejecutiva. Tipología de los SI a partir del Paradigma Orientado a Objetos. Características de O-O: Identidad, Abstracción, Clasificación, Encapsulamiento, Herencia, Polimorfismo y Persistencia. Tipología de SI por Clases de Sub-Características. Clases formales de Beckner. Estado Operacional. Sistemas-S. Sistemas-P. Sistemas-E. Mundo Real.

Perfil de los de Sistemas y su relación con el desarrollo.

Sistemas de Ciencias de la Computación – Informática - Computacional, Sistemas en Teoría de Sistema, Sistemas en Seguridad Computacional, Sistemas de Información Geográfica, Sistemas de Representación del Conocimiento, Sistemas en Teoría de los Dominios, Sistemas de Atributo-Valor, Sistemas en Ciencias Sociales, Sistemas Competitivos, Sistemas Cooperativos, Sistemas que modifican el estilo de operación del negocio, Sistemas empresariales (modelo de pirámide o jerarquía orgánico funcional), [Sistema de procesamiento de transacciones](#) (TPS), [Sistemas de información gerencial](#) (MIS), [Sistemas de soporte de las decisiones](#) (DSS), [Sistemas de información ejecutiva](#) (EIS). Interfaces, Paquetes, Internet y congestión. Ontología Informática.

Modelos y etapas del ciclo de vida del Software

Modelo de Cascada, Modelos Evolutivos, Modelo Iterativo Incremental, Modelo en Espiral, Modelo en Espiral WINWIN y no usual. Modelo de prototipo, Paradigmas, Modelos con Técnicas de Cuarta Generación, Modelo de Especificación Operacional, Modelos con Métodos Formales, Modelo Transformativo, Combinación de Modelos, Criterios para la elección de un Modelo.

Ingeniería de Requerimientos

Conceptos generales, Proceso de la Ingeniería de Requerimientos, Ingeniería de Requerimientos Orientada a Aspectos, Modelos de Ciclo de vida en Cascada y en Espiral aplicadas a la Ingeniería de Requerimientos, Actividades: Extracción, Análisis, Especificación, Validación, Entrevistas y Cuestionarios, Herramientas: Brainstorming, Arqueología de Documentos, Aprendiz, Talleres Basados en Casos de Uso, Prototipos, Análisis de FODA, Cadena de Valor, Modelo de Clase Conceptual, conceptos y clases, Diagrama de pescado de Ishikawa o Fishbone, Diagrama de Operaciones y de Actividades, Documento de especificación de Requerimientos y Casos de Uso (ESRE), Especificación de Casos de Uso, Lista de Requerimientos y de Control, Casas de Calidad, Quality Function Deployment (QFD) y Casos de Uso, Modelo Conceptual, Procesos Iterativos. Elicitación de requerimientos y jerarquías de metas.

Administración y Gestión de Proyectos

Marco Teórico, Planeación, Organización, Dirección y Control de los recursos, Factores críticos, Alcance, Programa y Costo de un proyecto, Satisfacción del cliente, Señales de un proyecto que producen fallas, Análisis pre-implantación, implantación y pos-implantación y de pos-mortem, Tipos de factores críticos organizativo-estratégico, organizativo-táctico, tecnológico-estratégico y tecnológico-táctico, Planificación de proyectos, Pasos: Descomposición del producto y el proceso, Estimación de tiempos, Definición de resultados con identificación de hitos, Análisis de interdependencias, Planificación temporal (Diagramas de Gantt, Redes PERT y CPM) y Re planificación, Herramientas CASE, Formación, criterios y Perfiles, Equipos, Integrantes, Compromisos, Roles, Funciones, Líder, Analista de sistemas, Ingeniero de proyecto, Administrador de diseño, Ingeniero de diseño, Equipo de programadores, Equipo de mantenimiento, Auditores, Toma de decisiones y árbol de niveles.

Metodologías de Desarrollo

Evolución de la Solución de Problemas de POLYA al Ciclo de Deming-Shewhart, Metodologías Ágiles, El Manifiesto Ágil, Proceso de Desarrollo Unificado (RUP), Principios de desarrollo, Ciclo de Vida, Contexto de



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Aplicación, Programación Extrema (XP), Valores y Principios de XP, Ciclo de vida, SCRUM, Roles en SCRUM, Reuniones en Scrum, Sprint, BackLog, Documentación, Dynamic Systems Development Methods (DSDM), Principios del DSDM, Lean Software Development (LSD), Lean de TOYOTA, Desarrollo Rápido de Aplicaciones (DRA), Modelado de Gestión, Modelado de Datos, Modelado de Procesos, Generación de Aplicaciones, Adaptación del SCRUM al DRA. Cada Sprint es un DRA con Producto Probado. Ciclo de Ciclo de Deming-Shewhart de Requerimientos, Análisis y Diseño y de Implementación con IT.

Métricas, Calidad y Riesgo.

Métricas orientadas al tamaño, Métricas orientadas a la función, Métricas de calidad, Métricas orientadas a la complejidad lógica, Métricas orientadas a las estructuras de datos, Métricas orientada objetos, Métricas de proyectos web, Índice de madurez del software, Fiabilidad, Ubicación en el ciclo de vida, Proceso de recopilación de métricas, Calidad de las métricas, Confiabilidad, Validez, Utilidad de las métricas post-mortem, Problemas sociales para implantar las mediciones, Estimación y Modelo Constructivo de Costo (COCOMO), Recursos humanos, del hardware, de software y Reusabilidad, Estimación del proyecto de software, Técnicas de descomposición y Estimación de Líneas de Código y de Puntos de Función, Modelos Univariable y multivariable estático y dinámico, Análisis de Riesgo, Estrategias de Riesgo, Identificación del Riesgo, Evaluación y Proyección del Riesgo, Calidad, Calidad Del Software, Factores de Calidad, Corrección, Robustez, Extensibilidad, Reutilización, Compatibilidad, Eficiencia, Portabilidad, Funcionalidad, Oportunidad, Documentación, Control de Calidad del Software, Aseguramiento de la calidad (SQA), Enfoque Socio-Técnico, Cultura Organizacional, Política de Calidad, Estructura Organizacional, Modelos de Aseguramiento de la Calidad, Calidad Total, Factores esenciales para introducir el Control Total de Calidad, Verificación y validación del software (V&V). Certificación de Calidad y normas ISO.

Métricas con Modelos de Construcción

Modelo estático conceptual, máximo nivel de abstracción, clasificación, cantidad total de atributos, agregación, complejidad del sistema, Longitud número de líneas de código, Funcionalidad, Complejidad, reusabilidad, Ciencia de Halstead, número de operadores, número de operandos monádicos, ocurrencias totales de operadores, ocurrencias totales de operandos, longitud, vocabulario, volumen, dificultad, esfuerzo, β del tiempo de Halstead, Planificación, Provisión de personal, Eficiencia, factor de complejidad o grado ciclomático de Mc Cabe, caminos tipo: Principal (Requerido), Secuencia, Repetitivo y Anidado, Número Ciclomático, Garantía de calidad, Modelo de Mc Call, Factores técnicos, Características de operación, Capacidad para soportar cambios (ser modificado), Adaptabilidad a nuevos entornos, Coeficientes de Regresión, Factor de Calidad de Mc Call, Métodos de análisis de Puntos de Función (Albrecht) PF, Léxico Extendido del Lenguaje, Escenarios, Elicitación de Requerimientos, Episodios, Componentes de Transacciones (DET), Índices de Puntos de Función (FPI), Pesos Promedios Industriales, Constante Industrial, Influencia Total de Factores técnicos de Mc Call (TDI), Ajuste de Complejidad Técnica (TCA), Índice de Puntos de Función Ajustado (AFPI), Alternativas de Cálculo de Puntos de Función con Escenarios y Episodios y el Modelo Empírico de Albrecht clásico y las transacciones, Resultados Variacionales, Restricciones y Reglas Transaccionales, Modelo de Costos Constructivos (COCOMO), Líneas de Código, Influencia del Lenguaje de Programación, Estimar Tiempo Hombre, Estimar Tiempo Total, Estimar Tiempo de Entrega (Scheduler), Estimar Cantidad de Profesionales, Estimar Costo Total, Modelo Orgánico, Semi-Libre y Libre, Relación con Puntos de Función, Puntos de Función Ajustados, Razón con las Líneas de Código, Líneas de Código, COCOMO II, Multiplicadores Pre y Pos Arquitectura, Acoplamiento, Cohesión, Modularidad, Mantenimiento, Sensibilidad, Estimación del número esperado de defectos, Inclusión y Remoción de Defectos, COCOMO Quality Models (COQUALMO).

Modelo de capacidad de madurez del software

Características de una organización inmadura y de una organización madura, Madurez del Proceso de Software, Los Cinco Niveles de Madurez del Proceso de Software, Caracterización del comportamiento de los



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

niveles de la madurez CMM: Nivel 1 – Inicial, Nivel 2 – Repetible, Nivel 3 – Definido, Nivel 4 – Gestionado, Nivel 5 – Optimización, Comprensión de los Niveles de Madurez, Comprender el nivel Inicial, Comprender los niveles Repetible y Definido, Comprender los niveles Gestionado y Optimizado, Omisión de los Niveles de Maduración, Áreas Claves, Prácticas Claves, Integración de modelo de capacidad de madurez CMMI: Nivel 0 – No realizado, Nivel 1 – Realizado informalmente, Nivel 2 – Planificado y seguido, Nivel 3 – Bien definido, Nivel 4 – Cuantitativamente controlado, Nivel 5 – Mejoramiento continuo, Evolución y Ventajas del CMMI, Un Modelo en vez de múltiples Modelos para la administración de procesos, Evaluación de los procesos, Construcción, selección y uso de instrumentos y herramientas de evaluación, Realizar mejoras en el diseño, Determinar la capacidad del proceso de suministros, Modelo SPICE (Software Process Improvement and Capability Determination), Elementos Principales de SPICE, Ventajas y Desventajas. SCAMPI: (Standard CMMI Appraisal Method for Process Improvement) Método de Evaluación Estándar del CMMI para la Mejora del Proceso

METODOLOGÍA DE ENSEÑANZA

Las clases teóricas se desarrollan a través de la explicación de los contenidos mínimos. En ambas instancias, tanto teórica como práctica, los alumnos cumplen con las etapas epistemológicas del conocimiento científico, primero con la detección y planteo de problemas a partir de las hipótesis desarrolladas en la teoría, análisis documental y bibliográfico para determinar el marco teórico, procedimientos deductivos a partir del análisis documental, consecuencias contrastables en la “práctica”, modelo experimental y experiencia con validación y verificación del qué y el cómo para determinar errores y falsedades y realizar un back tracking. De ser necesario plantear una nueva hipótesis en un modelo iterativo.

Esto se materializa a través de una monografía realizada por los alumnos en donde esté el problema, el marco teórico y las conclusiones. Puede ser individual o grupal y las correcciones son sucesivas e interactivas.

Experimentación Práctica:

Sobre el desarrollo de parte de algunos sistemas de características distintas, se intenta aplicar el desarrollo de: técnicas de ingeniería de requerimientos, elegir el o los ciclos de vida adecuados, recorridos e inspecciones, herramientas de análisis y diseño que conozcan, modelado utilizando por ejemplo UML.

Las características de los sistemas son:

- 1) Sistema administrativo, al mismo tiempo se explican distintos circuitos administrativos de: facturación, compras, contabilidad, etc. Casos de empresas pequeñas, medianas y grandes.
- 2) Sistema de cálculo matemático: aproximación de integrales, raíces, etc. Como obtener soluciones cuando por ejemplo se necesitan precisiones mas allá de la que da un lenguaje de programación, etc.
- 3) Sistema de control automático de procesos industriales. Introducción al estudio de micro controladores, distintas gamas de PIC y PLC.

Particularmente se desarrolla un software que captura desde un puerto. El producto se asemeja a uno real que mide la rugosidad de rutas calculando entre otras cosas el índice de rugosidad internacional (IRI), ciertas estadísticas, etc.

Planteo de casos similares y se propone encontrar soluciones para cada uno. Estos casos pueden ser: casas inteligentes, sistema de garaje (facturación, control de barrera, seguridad, categorías de estacionamiento, etc.)

La idea en general de toda la práctica es que el alumno propone una solución en función de las situaciones planteadas.

En todos los casos los requerimientos son extraídos de casos reales.

Exponer en público las propuestas y debatir las distintas alternativas. Todos aprenden de todos.

Al tratarse de una materia de los últimos años creemos importante poner al alumno en contacto con la realidad laboral. Por ello alentamos la exposición por parte de empresas o empresarios de sus experiencias y vivencias laborales.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Perspectiva- Visión, considerada en principio más ajustada a la realidad, que viene favorecida por la observación ya distante, espacial o temporalmente de cualquier hecho o fenómeno.

Visión- Punto de vista particular sobre un tema, un asunto

Táctica- Método o sistema para ejecutar o conseguir algo

Estrategia-En un proceso regulable, conjunto de las reglas que aseguran una decisión óptima en cada momento.

Recursos y equipamientos: PCs y Red homogénea, proyectores varios de diapositivas (cañón), pizarrones varios, un mínimo de multimedia.

EVALUACIÓN

La evaluación de la cursada se realiza a través del desarrollo de proyectos individuales y/o grupales (máximo tres (3) alumnos, con revisiones en la práctica, y se realizan dos parciales de desarrollo conceptual.

Para la aprobación final de la materia deben entregar y defender el trabajo final que comenzaron a desarrollar durante la cursada, en donde toman tópicos de la teoría para armar el marco teórico con el que van a resolver el problema planteado.

Hay temas que obligan a tener un cuidadoso desarrollo del overview y el review en la Ingeniería del software, por ser muy abarcativo y muy frecuentemente desarrollado por los alumnos, en cuyo caso se debe ser más o menos novedoso en el tratamiento.

Se definen en cada caso puntos de control por etapas.

Las limitaciones se imponen por las etapas de la investigación científicas, que si bien hay muchas, para diversos autores, se comentan sucintamente algunas:

1- Que sorprendió, para resolver o explicar, en un suceso o varios fenómenos, etc.: Sorpresa, Inquietud, Alerta, etc. Produjo interés, deseo de estudiarlo para verlo mejor o de otra manera.

2- Problema (planteo).

3- Conjetura, hipótesis Heurística, Hipótesis de trabajo: se trata de resolver artísticamente un problema, que es lo mejor que le queda a la informática.

4- Marco teórico que en el plan de aprobación de la materia, que les adjunto los denomino Análisis Documental, Bibliografía.

5- Procedimientos de deductivos, que es la monografía de lo que estudiaron en el análisis documental.

5- Consecuencias contrastables, por la práctica, experimentación y hasta experiencia y habilidades: validación y/o verificación.

6- Procedimientos de Contrastación o validación y/o verificación, el que y el cómo.

7- Conclusiones, una final o varias, para cada parte diferenciable, comenzando por una comprobación de los resultados, pues si dan (a) refutación, falsedad o no contrastados hay que hacer un back tracking y plantear hasta una nueva hipótesis de trabajo. Según hasta donde retrocedieron iterar a partir de ese punto dado; (b) Si no, está terminado:

If <falso> then (a) else (b).

De la organización: Título, Resumen del Trabajo, Índice, Análisis de la Documentación, Monografía a partir del Análisis, Conclusiones (pueden ser parciales y total final o solo la final) y Referencias bibliográficas.

BIBLIOGRAFÍA OBLIGATORIA

- Boehm, Barry W., Software Engineering Economics, Prentice Hall, 1981.
- Brooks, Frederick P., The Mythical Man-Month, Addison Wesley, 1995.
- Date, C. J., Introducción a los Sistemas de Bases de Datos 1986.
- De Marco, Tom, Controlling Software Projects: Management, Measurement, Prentice Hall, 1982.
- Flavin, Matt, Fundamental Concepts of Information Modelling, 1981.
- Humphrey, Watts S., Managing the Software Process. Addison-Wesley Massachusetts, 1989.
- Ivar Jacobson Object-Oriented Software Engineering. Addison-Wesley. 1994.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

- Mac Menamin & Palmer, Essential System Analysis, 1984.
- Meyer, Bertrand, Object-oriented software construction, Prentice Hall, 1988.
- Myers Glenford J., The Art of Software Testing, Wiley, 1979.
- Page-Jones, Meilir, The Practical Guide To Structured Desing, 1988.
- Managing code inspection information, Jack Barnard, ART Price AT&T Bell Laboratories, 1992
- Software Inspection. An industry Best Practice, Wheeler, DA, Brykezyski, B, Meeson, RN, IEEE Computer Society Press, 1996
- Jones, T.C., Programming Productivity, McGraw-Hill, 1986
- "Implementating Software Inspections", Notas del curso, IBM Systems Sciences Institute, IBM Corporation, 1981
- Handbook of walkthroughs, Inspections and Technical Reviews, 3° edición, Freeman, D.P., Weimberg, G.M., Dorset House, 1990
- Software Requirements, Alan M. Davis, Prentice Hall, 1993
- El Lenguaje Unificado de Modelado, G. Booch, J.Rumbaugh, I.Jacobson – Addison Wesley, 1999
- Ingeniería de Software, un enfoque práctico 7° edición – Roger S.. Pressman – Mc Graw ,Hill, 2006.
- Software Testing Technics, Van Nostrand Reinholds, 1990.
- Fundamental of software Enginnering, Ghezzi et al., Prentice Hall, 1992.
- Gonzalo Cuevas Agustín: Una Guía del CMM. Para Comprender el Modelo de Madurez de Capacidad del Software. Traducción del Inglés "A Guide to the CMM" de Kenneth M. Dymond. 1998.
- Constantine, L. and Lockwood, L. "Software for use". A Practical Guide to the Models and Methods of Usage - Centred Design. Addison - Wesley. 1999.
- Ghezzi, C., Jazayeri, M., Mandrioli, D. "Fundamentals of Software Engineering". Prentice Hall. 1991.
- Grimaldi, Ralph P. "Matemáticas discreta y combinatoria. Introducción y aplicaciones". Addison-Wesley Iberoamericana, 1989.
- IEEE STD 1074-1991 Customer Request IEEE Standard fo developing Software Life Cycles Process, Identify Ideas or Needs, Preliminary Statement of Need, Feasibility Studies, Statement of need.
- Jacobson, Ivar; Booch, Grady; Rumbaugh, James. "The Unified Software Development Process". Addison-Wesley ISE. 1999.
- Pfleeger, S. "Software Engineering: Theory and Practice". 7° edición - Prentice Hall. 2002.
- Sommerville, I. "Software Engineering". 7° edición - Addison - Wesley. 2005.
- Thomas, I. & Nejmeh, B. A., 1992, Definitions of Tool Integration for Environments, IEEE Software, 9, 2 (Mar), 29-35.
- Fenton, N.E., Pfleeger, Sh.L. "Software Metrics". PWS Publishing Company. Last Edition.
- Kitchenham, B., Pickard, L., Pfleeger, S.L. "Case studies for method and tool evaluation". IEEE Software, 12(4) pp 52-62. 1995.
- Probasco, L., "Dear Dr. Use Case: What About Function Points and Use Cases?", *RationalEdge*, August 2002.
- B. Boehm, C. Abts, A. Brown, S. Chulani, B. Clark, E. Horowitz, R. Madachy, D. Reifer, B. Steece: *Software Cost Estimation with COCOMO II*. Prentice-Hall, 2000.
- Principios de Administración. 1ª Edición. Terry & Franklin. CECSA, México. 2001.
- Project Management: A Systems Approach to Planning, Scheduling, and Controlling. 8va Edición. Harold, K. Ohio, EUA. 2003.
- Administración de proyectos exitosos. Robert Graham - Randall England. Prentice-Hall . 1999.
- Kent Beck. "Extreme Programming Explained": Embrace Change. Addison-Wesley, Pearson Education, 2000. ISBN 201-61641-6.
- Schwaber K., Beedle M., Martin R.C. "Agile Software Development with SCRUM". Prentice Hall. 2001.
- Cockbun, A. "Agile Software Development". Addison-Wesley. 2001.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

- Stapleton J. "Dsdm Dynamic Systems Development Method: The Method in Practice". Addison-Wesley. 1997.
- Highsmith J., Orr K. "Adaptive Software Development: A Collaborative Approach to Managing Complex Systems". Dorset House. 2000.
- Stylianou, A y Kumar, R An Integrative Framework for IS Quality Management, Communications of the ACM 43, 9 Septiembre 2000
- Desarrollo, notas y apuntes de la Teoría de la cátedra.

BIBLIOGRAFÍA COMPLEMENTARIA

- Guidelines for Undergraduate Degree Programs in Software Engineering, IEEE Computer Society Press and ACM Press, August 2004. <http://www.computer.org/curriculum/> / <http://www.acm.org/education/curricula.html>
- **[IT2006]** The ACM SIGITE Task Force on IT Curriculum. Information Technology 2006, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology. As this is written, the IT curriculum report is in development. We expect the review and development process to be completed in 2006. At the moment, the current draft is called *IT2005* and is available for public review and comment. The most current link to the IT volume can be found at: <http://www.acm.org/education/curricula.html>.
- SEI - Software Engineering Institute
- ESI - European Software Institute
- System Security Engineering CMM
- Finkelstein's Capability Immaturity Model paper (PDF file)
- Tom Schorsch's Capability Immaturity Model study
- Weske Mathias, "Business Process Management: Concepts, Languages, Architectures". Springer, Pag 3-67. ISBN 978-3-540-73521-2. 2008
- "Extensión UML para el MODELADO de Mapas Navegacionales de Aplicaciones WEB Basado en MDA". Humberto Javier Cortés Benavides. Máster en Investigación en Informática, Facultad de Informática, Universidad Complutense de Madrid. Trabajo Final de Máster en Sistemas Inteligentes. 8 de septiembre de 2011.
- Arquitectura Cliente/Servidor. SCS - Sistemas Cliente/Servidor. 4to Informática-<http://ccia.ei.uvigo.es/docencia/SCS-27> de septiembre de 2009
- Ingeniería de Requisitos en Aplicaciones para la Web –Un estudio comparativo, María José Escalona, Universidad de Sevilla, Lenguajes y Sistemas Informáticos, España, Nora Koch, Universidad de Munich y F.A.S.T. GmbH, Munich, Alemania.



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

CRONOGRAMA DE CLASES Y EVALUACIONES

Clases	Contenidos/Actividades	Evaluaciones previstas
Perfil de los de Sistemas y su relación con el desarrollo. Modelos y etapas del ciclo de vida del Software		Evolución Práctica casos reales
	Caracterización de Sistemas Práctica Tema 1	Elección del Proyecto
	Transacciones	Avances del Proyecto y Tópicos Evolución Práctica
Ingeniería de Requerimientos. Metodología de desarrollo. Administración y Gestión de Proyectos.	Identificación de los ciclos de vida Práctica Tema 2	Caracterización del Proyecto Evolución Práctica
		Introducción Evolución Práctica Estado Pre teórico
Métricas, Calidad y Riesgo. Modelo de capacidad de madurez del software	Métricas con Modelos de Construcción Práctica Tema 3	Primer desarrollo teórico y conclusión. Evolución Práctica, parcial, entrega de trabajo y un coloquio.
	Control de Calidad y Madurez Práctica Etapas de actividades de ingeniería de requerimientos.	Segundo desarrollo teórico y conclusión.
		Final del desarrollo teórico y conclusión.

Fechas de evaluación 24-07-2014 y 31-07-2014.

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

nestor@piovoso.com.ar Lic. Néstor Piovoso Jefe de Trabajos Prácticos.

gregorio@perichinsky.com.ar Dr. Gregorio Perichinsky Profesor Titular.

Firmas del/los profesores responsables:



**UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA**

Prof. Dr. Gregorio Perichinsky