

REGLAMENTO INTERNO PROGRAMA DOCTORADO EN INGENIERÍA INFORMÁTICA

Aprobado por CCDIP de fecha noviembre 28 de 2013.

"Dada la naturaleza del trabajo académico y en pos de un mejoramiento continuo, el presente reglamento será revisado y sancionado por el CCDIP anualmente. Si se registraren cambios esenciales, éstos aplicarán solamente a nuevas cohortes de estudiantes."

INTRODUCCIÓN

- Art. 1** El programa de **Doctorado en Ingeniería Informática** fue creado con fecha 11 de julio del año 2002, por acuerdo del Consejo Superior en su sesión n° 127, como consta en la resolución Acuerdo N° 442.
- Art. 2** El programa de Doctorado en Ingeniería Informática, en adelante DII, se regirá por estas normas y por el Reglamento General N° 47 que regula los programas doctorales en la Universidad Técnica Federico Santa María.
- Art. 3** Estas normas complementan el Reglamento General N° 47, en todas aquellas materias no contempladas en él, o que se han establecido allí, expresamente, como materias a ser reguladas por normas específicas de cada programa doctoral.

TÍTULO I: DISPOSICIONES GENERALES

- Art. 4 Objetivos del Programa:** El objetivo del DII es formar especialistas que deseen adquirir conocimientos avanzados en el área de la Ingeniería Informática, a través de un programa sistemático de cursos y seminarios, y la realización de una investigación teórica o aplicada conducente a la tesis doctoral, de modo que, a través de dicha investigación original e independiente, contribuya al desarrollo científico de la Ingeniería Informática.
- Art. 5 Áreas de especialización del Programa:** El DII ofrece especialización en:
- a) Computación Científica.
 - b) Inteligencia Computacional.
 - c) Tecnologías de Software.
 - d) Sistemas de Computación y Base de Datos.



Art. 6 Perfil del graduado: El graduado del programa DII debe ser capaz de:

- a) Desarrollar una investigación pura o aplicada, en el área de la ingeniería informática.
- b) Analizar un problema de investigación específico en alguna área de la ingeniería informática, estudiar el estado del arte asociado a este problema, y proponer una solución original e independiente.
- c) Evaluar la calidad o el desempeño de la solución propuesta al problema de investigación.
- d) Resumir y difundir los resultados obtenidos durante la investigación y divulgarlos en revistas y conferencias internacionales de su especialidad.

Art. 7 El estudiante debe tener una permanencia activa mínima en el DII equivalente a 150 SCT en la Institución (2,5 años) en régimen de jornada completa.

Art. 8 La permanencia máxima en el DII, para un alumno regular, es de 6 años.

TITULO II: ADMINISTRACIÓN DEL PROGRAMA

Art. 9 La tuición académica del programa DII dependerá, de manera exclusiva, del Departamento de Informática, o DI, de la Universidad Técnica Federico Santa María.

Art. 10 El DII será dirigido por el Comité Académico del DII, en adelante CADII, integrado por al menos 4 miembros del Cuerpo de Directores de Tesis del Programa, en adelante CDTP, definido en el Título III, pertenecientes a las dos jerarquías académicas superiores, y será presidido por el Director del Programa. Todos ellos serán propuestos por el Director del DI, al Consejo de Departamento de Informática para su designación como integrantes del CADII y Director del Programa (Anexo N° 1).



Art. 11 Le corresponde al CADII, además de las funciones establecidas en el Art. 16 del Reglamento General N° 47 de los Estudios de Postgrado:

- a) Programar cursos y profesores encargados
- b) Aprobar, en primera instancia, programas de nuevas asignaturas o cualquier otra modificación del plan de estudios.
- c) Actualizar, periódicamente, el cuerpo de profesores y el CDTP, de acuerdo a los criterios establecidos en los artículos 15 al 18.
- d) Sesionar al menos 3 veces al año, quedando sus acuerdos consignados en actas de acuerdo al reglamento interno de sala del DI, que regula el funcionamiento de las sesiones de los comités académicos de programas de postgrado, sin perjuicio de las atribuciones del Consejo del Departamento de Informática.
- e) Aplicar los mecanismos establecidos de evaluación del DII.
- f) Participar en las actualizaciones de los planes de desarrollo del Departamento de Informática.
- g) Exponer ante el cuerpo académico del DII las situaciones de conflicto académico o disciplinario que se presentaren, para una adecuada resolución.
- h) Otras competencias o actos, de índole académico, necesarios para la buena marcha del programa DII, corresponden al Director del Programa.

Art. 12 El Director del Programa será un profesor del CADII, propuesto por el Director del DI al Consejo del DI para su designación.

Art. 13 Las sesiones del CADII serán convocadas por el Director del Programa. Este Comité deberá sesionar con la concurrencia de más de la mitad de sus miembros. En caso de que alguna materia deba decidirse por votación, se aplicará el criterio de mayoría absoluta. De producirse un empate en la votación, le corresponderá al Director del Programa decidir.

Art. 14 El Director del Programa designará entre los miembros del CADII quien habrá de subrogarlo en su ausencia.



TITULO III: DE LOS PROFESORES DEL PROGRAMA

- Art.15** Podrán pertenecer al Cuerpo de Profesores del programa DII (Colaboradores) sólo académicos de jornada completa del DI, que tengan el grado de Doctor, y que pertenezcan, preferentemente, a las jerarquías de profesor adjunto o titular. (Anexo N° 2)
- Art. 16** Excepcionalmente, y por acuerdo del CADII, podrán ser parte del Cuerpo de Profesores del DII investigadores que no pertenezcan al DI, pero que tengan reconocida trayectoria en una línea de investigación relacionada con un tema específico de interés para el Programa. Dichos profesores deberán cumplir con las mismas exigencias que los miembros del CDTP.
- Art. 17** Podrán pertenecer al CDTP (Claustro) sólo académicos de jornada completa del DI que tengan el grado de Doctor, que pertenezcan, preferentemente a las jerarquías de profesor adjunto o titular, y que cumplan con la siguiente exigencia:
- Índice mínimo de productividad científica anual equivalente a 1,0 en los últimos 5 años. Considerando las publicaciones en Journal ISI (ponderación 1,0) y en Proceedings indexados por ISI y/o Scopus (ponderación 0,5), con al menos 4 publicaciones en Journals ISI en el período de evaluación.
- Art. 18** Excepcionalmente, y por acuerdo del CADII, podrá ser co-director de tesis doctoral un académico que no pertenezca al CDTP, pero que tenga reconocida trayectoria en una línea de investigación relacionada con el tema de tesis específico. En tal caso, el CADII designará un Director de tesis de entre los miembros del CDTP.
- Art. 19** Quienes dejen de cumplir las condiciones de productividad establecidas para pertenecer al CDTP, no podrán guiar nuevas tesis, pero continuarán dirigiendo aquellas tesis patrocinadas que ya hayan sido aprobadas a través de los correspondientes Exámenes de Calificación.

TITULO IV: DE LA ADMISIÓN

- Art. 20** Las postulaciones se canalizarán a través de la Dirección de Postgrado (en adelante DP), y deberán seguir los procedimientos y formalidades establecidos en el Reglamento General N° 47. Será requisito esencial para postular al programa DII el estar en posesión del grado de licenciado en ciencias de la ingeniería o en ciencias. El CADII podrá autorizar excepcionalmente, mediante resolución fundada, el ingreso interino de personas que se encuentren en trámites finales de graduación o



situaciones académicas que lo ameriten, estando su aceptación definitiva supeditada a la obtención del grado correspondiente, de conformidad al procedimiento que al efecto apruebe el CADII.

- Art. 21** Una solicitud de admisión será analizada por el CADII sólo cuando el postulante haya hecho llegar todos los antecedentes requeridos, establecidos en el procedimiento de admisión de la DP.
- Art. 22** Una vez enviados los antecedentes completos del postulante por la DP, el Director del Programa los pondrá a disposición del CADII, el que resolverá por mayoría absoluta sobre la aceptación o rechazo de la postulación. Para tomar la decisión, el CADII considerará las calificaciones de pregrado, carta de interés del postulante, cartas de recomendación, publicaciones en revistas y conferencias, participación en proyectos científico-tecnológicos, y otras consideraciones académicas.
- Art. 23** Un postulante será admitido sólo si el CDTP incluye un especialista en la línea de investigación de interés del postulante, cuidando que exista un adecuado equilibrio entre el número de estudiantes aceptados y el total de recursos disponibles.
- Art. 24** El CADII, o el Director del Programa, podrá exigir que el postulante entregue antecedentes adicionales y/o que participe en una entrevista, presencial o remota, para evaluar aspectos relevantes que permitan decidir en mejor forma sobre la solicitud de admisión.
- Art. 25** El CADII podrá exigir que el postulante apruebe una etapa de nivelación, antes de ser aceptado definitivamente en el DII.
- Art. 26** Una vez que el CADII resuelva sobre la aceptación o rechazo del postulante en el DII, el Director del DII informará dicha decisión a la DP. Si el postulante es admitido en el DII, el Director del Programa designará un Tutor, el que también será informado a la DP.
- Art. 27** El Tutor propondrá un plan de estudios específico al CADII, para su aprobación; el que incluirá, cuando corresponda, la convalidación de hasta 30 SCT en asignaturas del Programa de Estudios. Ambas decisiones deberán ser informadas a la DP.



TITULO V: SOBRE EL DESARROLLO DEL PROGRAMA

- Art. 28** El estudiante de doctorado deberá aprobar al menos 60 créditos SCT del DII en asignaturas de nivel postgrado (Programa de Estudios), y una Tesis (Actividad de Graduación) equivalente a 180 créditos SCT correspondientes a los Seminarios de Tesis. (Anexo N° 3).
- Art. 29** Las asignaturas del DII serán evaluadas con nota de 0 a 100, siendo 70 la nota mínima de aprobación. (Anexo N° 4)
- Art. 30** Una vez que el estudiante haya aprobado los 60 créditos SCT correspondientes a asignaturas de nivel postgrado, deberá informar al Director del Programa DII el nombre del académico miembro del CDTP que será su Director de Tesis. Esta información deberá ser confirmada por escrito por el Director de Tesis, quien a partir de ese momento será también el Tutor del estudiante dentro del DII. El Director del Programa comunicará esta decisión a la DP y al anterior Tutor, si correspondiere.
- Art. 31** Durante la realización de la tesis, el alumno de doctorado deberá inscribir las asignaturas Seminario de Tesis I, de 60 créditos SCT, y Seminario de Tesis II, de 120 créditos SCT. El profesor de estas asignaturas será el Director de Tesis del alumno.

TITULO VI: EL EXAMEN DE CALIFICACIÓN

- Art. 32** El Examen de Calificación estará basado en la defensa de su proyecto de tesis doctoral y en evaluación de conocimientos relevantes en los temas fundamentales de su especialidad, atingentes al proyecto de tesis doctoral, adquiridos por el estudiante durante el desarrollo del Programa de Estudios.
- Art. 33** El tema de tesis deberá ser propuesto por el estudiante y su Director de Tesis en el formato establecido por el CADII, el que estará disponible en la página web del Programa. Dicha propuesta deberá llevar la firma del estudiante y del Director de Tesis, y será entregada al Director del Programa, con antelación de al menos 15 días hábiles respecto del Examen de Calificación.
- Art. 34** Este examen será rendido ante la Comisión de Examen de Calificación, nominada por el CADII, la que estará integrada por:
- a) El Director de Tesis
 - b) Un Profesor/Investigador nominado por el CADII, que deberá pertenecer al CDTP.



- c) Un Profesor/Investigador externo a la UTFSM, experto en el área, nominado por el CADII.
- d) El Director del Programa, o quién designe de entre los miembros del CADII. Éste preside la Comisión de Examen, no pudiendo recaer esta responsabilidad en el Director de Tesis.

Art. 35 El Director del Programa deberá hacer llegar el proyecto de tesis doctoral a los miembros de la Comisión de Examen de Calificación, con una antelación de, al menos, 10 días hábiles.

Art. 36 En caso de reprobación, este examen podrá repetirse sólo una vez, dentro de un plazo de 6 meses, previo acuerdo de la Comisión. En caso de reprobación por segunda vez, el estudiante deberá abandonar el DII.

Art. 37 El examen de Calificación deberá ser aprobado por el estudiante durante sus dos primeros años de permanencia en el DII.

TITULO VII: DE LA TESIS DE DOCTORADO Y EXAMEN DE GRADO

Art. 38 El Seminario de Tesis I se calificará en escala de 0-100, una vez que el estudiante ha aprobado su Examen de Calificación.

Art. 39 El Seminario de Tesis II será calificado en escala de 0-100, una vez que el estudiante haya culminado su trabajo de tesis doctoral y se encuentre en condiciones de rendir su examen de grado.

Art. 40 El Comité de Tesis, encargado de evaluar el Trabajo de Tesis y el Examen de Grado, estará integrado por:

- a) El Director de Tesis
- b) Un Profesor/Investigador nominado por el CADII, que deberá pertenecer al CDTP.
- c) Al menos dos Profesores/Investigadores externos a la Universidad, expertos en el área, nominados por el Comité de Coordinación y Desarrollo de Investigación y Postgrado de la UTFSM, a proposición del CADII.
- d) El Director del Programa, o quién designe de entre los miembros del CADII. Éste preside el Comité de Tesis, no pudiendo recaer esta responsabilidad en el Director de Tesis.

Art. 41 El Examen de Grado es público y consiste en una presentación y defensa oral de la Tesis por parte del estudiante.

Art. 42 El Trabajo de Tesis de Doctorado consistirá en un trabajo personal de investigación en la línea de especialidad del estudiante; el que deberá conformar un cuerpo escrito novedoso y significativo de conocimientos y



generar, al menos, una publicación en alguna revista indexada de corriente principal, la que deberá estar aceptada antes de la entrega del trabajo de Tesis.

Además, el candidato deberá adjuntar una declaración simple de originalidad, independencia, exclusividad y reproducibilidad de los antecedentes contenidos en su Trabajo de Tesis (Formato Declaración Simple en Anexo N°6).

El escrito correspondiente al trabajo de Tesis deberá ser redactado en español o inglés.

Art. 43 El Examen de Grado se dará por aprobado con calificación mayor o igual a 85, en escala de 0 a 100.

Art. 44 Si la calificación del Examen de Grado fuese menor que 85, el Comité de Tesis, dentro de los 5 días hábiles siguientes a la realización del Examen, determinará conceder o no una última oportunidad para que el candidato al Grado rinda el Examen nuevamente.

TITULO VIII: DEL GRADO ACADÉMICO

Art. 45 Una vez cumplidas por parte del estudiante todas las exigencias de Graduación, la Universidad otorgará el grado académico de “**Doctor en Ingeniería Informática**”.

TITULO IX: DE LA RESPONSABILIDAD DEL PRESENTE REGLAMENTO

Art. 46 La responsabilidad de la aplicación de las disposiciones contenidas en el presente reglamento al interior del Programa, será del Director del Programa.

ARTÍCULOS TRANSITORIOS

Art. T1 Los académicos que, a la fecha de aprobación de estas normas por parte del CCDIP, se encuentren guiando alguna tesis, podrán continuar haciéndolo con todas las atribuciones y obligaciones asociadas a esta tarea, aún si no cumplieren los requisitos establecidos en estas normas para conformar el CDTP.



Art. T2 Aquellos estudiantes de doctorado, aún no graduados, que hubiesen ingresado al DII en una modalidad distinta a la establecida en este reglamento, se acogerán a lo dispuesto en el Reglamento General N° 18, aprobado por el Consejo Académico el 17 de junio de 2008 y por el Reglamento No.2 Departamental de Doctorado en Ingeniería Informática, aprobado por el Consejo de departamento el 6 de noviembre de 2008 y vigente al 28 de Diciembre de 2010.



ANEXO Nº 1: ADMINISTRACIÓN DEL PROGRAMA

DIRECCIÓN DEL PROGRAMA

- ALLENDE OLIVARES, HÉCTOR (Director)
- ASTUDILLO ROJAS, HERNÁN (Director Alterno)

COMITÉ DE PROGRAMA (CP) o COMITÉ ACADÉMICO (CADII)

- ALLENDE OLIVARES, HÉCTOR Miembro Titular (Director del Programa DII)
- ASTUDILLO ROJAS, HERNÁN Miembro Titular (Director Alterno DII)
- MOREIRA WENZEL, ANDRÉS Miembro Titular
- SALINAS CARRASCO, LUIS Miembro Titular
- CASTRO VALDEBENITO, CARLOS Miembro Suplente

CUERPO DE DIRECTORES DE TESIS DEL PROGRAMA (CDTP)

- ALLENDE OLIVARES, HÉCTOR Máquinas de Aprendizaje e Inteligencia Computacional
- ARROYUELO, DIEGO Análisis y Diseño de Algoritmos
- ASTUDILLO, HERNÁN Arquitectura de Software
- BONNAIRE, XAVIER Sistemas de Computación Distribuida
- CASTRO, CARLOS Optimización Combinatorial y Metaheurísticas
- DOMBROVSKAIA, LIOUBOV Interfaces Hombre-Máquina
- MENDOZA, MARCELO Recuperación de Información
- MOREIRA, ANDRÉS Informática Teórica, Computación Científica
- RIFF, MARÍA CRISTINA Computación Evolutiva y Metaheurísticas,
- SALINAS, LUIS Computación Científica
- SOLAR, MAURICIO Computación Paralela y Distribuida
- ZAMYATNIN, ALEXANDER A. Bioinformática



ANEXO Nº 2

NÓMINA DE PROFESORES DEL PROGRAMA DII

Nº	NOMBRE	GRADO ACADÉMICO	ÁREA ESPECIALIZACION
01	ALLENDE OLIVARES, HÉCTOR	Doctor. rer. nat. Statistik, Technische Universität Dortmund, Dortmund, Alemania. (1988)	Máquinas de Aprendizaje e Inteligencia Computacional
02	ARROYUELO BILLIARDI, DIEGO	Doctor en Ciencias, mención Computación, Universidad de Chile, Chile. (2009)	Análisis y Diseño de Algoritmos
03	ASTUDILLO ROJAS, HERNÁN	Ph.D. Information and Computer Science, Georgia Institute of Technology, Atlanta, GA, EEUU. (1996)	Arquitectura de Software
04	BONNAIRE, XAVIER	Doctor en Informática, Université Pierre et Marie Curie, Francia (1998)	Sistemas de Computación Distribuida
05	CASTRO VALDEBENITO, CARLOS	Doctor en Informática, INRIA, Université Henri Poincaré, Nancy, Francia. (1998)	Optimización Combinatorial y Metaheurísticas
06	DOMBROVSKAIA, LIOUBOV	Doctor en Ciencias de la Ingeniería, Universidad Católica de Chile, Santiago, Chile. (1998)	Interfaces Hombre-Máquina
07	HOFFMANN NAGEL, HUBERT	Dr. Ing. Informatik, TU Berlín, Alemania. (1987)	Computación Gráfica
08	LOBOS YÁÑEZ, CLAUDIO	Dr. In Models, Methods and Algorithms in Biology, health and environment. Université Joseph Fourier- Grenoble I, Francia. (2009)	Computación Gráfica



09	MENDOZA ROCHA, MARCELO	Doctor en Ciencias, mención Computación, Universidad de Chile, Chile. (2007)	Recuperación de Información
10	MONGE ANWANDTER, RAÚL	Dr. Ing.Informatik (Doctor Ingeniero), Universität Erlangen- Nürnberg, Alemania (1992)	Sistemas de Computación Distribuida
11	MOREIRA WENZEL, ANDRÉS	Doctor en Ciencias, mención Informática y Matemáticas, Universidad de Chile, Chile. (2003)	Informática Teórica, Computación Científica
12	RIFF ROJAS, MARÍA CRISTINA	Dr. in Mathematics and Informatics. École Nationale de Ponts et Chaussées, París, Francia. (1997)	Computación Evolutiva y Metaheurísticas, Inteligencia Computacional
13	SALINAS CARRASCO, LUIS	Dr. rer. nat. Mathematik, Universität des Saarlandes, Saarbrücken, Alemania. (1976)	Computación Científica
14	SIMMONDS WAGEMANN, JOCELYN	Ph.D. Computer Science, University of Toronto, Canadá. (2011)	Ingeniería de Software
15	SOLAR FUENTES, MAURICIO	Doctor en Ciencias en Ing. de Sistemas y Computación. COPPE, Universidade Federal de Rio de Janeiro, Brasil. (1992)	Computación Paralela y Distribuida
16	VISCONTI ZAMORA, MARCELLO	Ph.D. in Computer Science, Oregon State University, EEUU. (1993)	Ingeniería de Software
17	VON BRAND SKOPNIK, HORST	Ph.D. in Computer Science, Louisiana State University, Baton Rouge, LA, EEUU. (1987)	Sistemas de Computación



18	ZAMYATNIN, ALEXANDER A.	Ph.D. in Physic- Mathematic Sciences: Biophysics, Institute of Biological Physics, Academy of Sciences of the URSS (1971)	Bioinformática
----	----------------------------	--	----------------



ANEXO Nº 3:

PLAN DE ESTUDIOS DEL PROGRAMA DII

El estudiante de doctorado debe aprobar un programa de estudios de, al menos, 60 SCT en asignaturas de nivel de postgrado, y una actividad de tesis compuesta por 180 SCT, correspondientes a los Seminarios de Tesis I y II, en un período total de 8 semestres académicos.

AÑO 1				AÑO 2		AÑO 3 y 4	
1ER. SEMESTRE		2DO. SEMESTRE					
INF-400	7SCT	INF-400	7SCT	INF-591	INF-591	INF-592	INF-592
INF-400	7SCT	INF-400	7SCT	SEMINARIO DE TESIS I		SEMINARIO DE TESIS II	
INF-500	8SCT	INF-500	8SCT				
INF-500	8SCT	INF-500	8SCT				
30 SCT		30 SCT		60 SCT		120 SCT	
60 SCT				180 SCR TRABAJO DE TESIS DOCTORAL			

Observaciones:

- El tiempo normal de permanencia en el Programa, para un estudiante regular, es de 4 años.
- La permanencia máxima en el Programa, para un estudiante regular es de 6 años.
- Toda asignatura INF-5XX debe culminar con un informe técnico de investigación.
- Este programa de estudios supone una carga mínima de 60SCT en el primer año del DII.
- Durante el segundo año, la carga mínima es de 60 SCT, correspondiente al Seminario de Tesis I.
- El Tercer y cuarto año, la carga mínima es de 120 SCT, también correspondiente a 60 SCT anuales (Seminario de Tesis II).



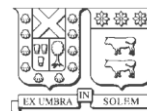
ANEXO Nº 4:

ASIGNATURAS DEL PROGRAMA DII

SIGLA	ASIGNATURA	PROFESOR
Asignaturas Nivel 400 (4 créditos)		
INF-406	Teoría y Aplicaciones de Wavelets	L. Salinas
INF-413	Calidad y Productividad de Software	M. Visconti
INF-414	Arquitectura de Software	H. Astudillo
INF-417	Análisis de Programas	J. Simmonds
INF-424	Pruebas de Software	M. Visconti
INF-427	Interfaces Hombre-Máquina	L. Dombrowskaia
INF-435	Programación de Sistemas	H. von Brand
INF-440	Sistemas Distribuidos	R. Monge
INF-451	Computación Gráfica	H. Hoffmann
INF-472	Optimización Combinatoria	C. Castro
INF-474	Tópicos Avanzados en Inteligencia Artificial	M.C. Riff
INF-475	Modelamiento Estocástico y Simulación	H. Allende
INF-477	Redes Neuronales Artificiales	H. Allende
INF-478	Análisis Inteligente de Datos	H. Allende
INF-480	Redes Complejas	A. Moreira
INF-481	Tecnologías de Búsqueda Avanzada en la Web	M. Mendoza
INF-490	TEI: Computación Peer to Peer	X. Bonnaire
INF-490	TEI: Arquitecturas Empresariales	H. Astudillo
INF-490	TEI: Seguridad de Sistemas Distribuidos	E. Fernández
INF-490	TEI: Seminario de Sistemas Distribuidos	R. Monge
INF-490	TEI: Astroinformática	M. Solar
INF-492	TEI: Ingeniería de Software Experimental	M. Visconti
INF-493	TEI: Mejoramiento de Procesos de Software	M. Visconti

Nota:

TEI: Temas Especiales en Informática -asignaturas ofrecidas por los distintos profesores.



SIGLA	ASIGNATURA	PROFESOR
Asignaturas Nivel 500 (5 créditos)		
INF-520	Compresión de Texto	D. Arroyuelo
INF-524	Evaluación de Arquitectura de Software	H. Astudillo
INF-528	Computación Autónoma	H. Astudillo
INF-534	Programación Paralela Aplicada Avanzada	X. Bonnaire
INF-560	Modelos Computacionales en Series de Tiempo	H. Allende
INF-562	Métodos Computacionales en Teoría de Funciones	L. Salinas
INF-565	Metodología de la Investigación	H. Allende/ H. Astudillo/P. Invitado
INF-568	Simulación Mediante Mallas Geométricas	C. Lobos
INF-569	Métodos Cuantitativos en el Procesamiento Computacional de Imágenes	L. Salinas
INF-571	Programación con restricciones	C. Castro
INF-572	Computación Evolutiva	M. C. Riff
INF-575	Lógica Borrosa	C. Moraga
INF-578	Máquinas de Aprendizaje Computacional	H. Allende
INF-579	Agentes Inteligentes y Autónomos	E. Canessa
INF-584	Sistemas Complejos Discretos	A. Moreira
INF-585	Bases de Datos Documentales	M. Mendoza
INF-586	Procesamiento Digital de Imágenes usando R	A. Frery
INF-581	Seminario de Especialidad I	Director de Tesis
INF-581	Seminario de Especialidad I: Bioinformática	A. Zamyatnin
INF-582	Seminario de Especialidad II	Director de Tesis
INF-582	Seminario de Especialidad II : Computación Cuántica	C. Moraga
Seminarios de Tesis Doctoral:		
INF-591	Seminario de Tesis I (34 créditos)	Director de Tesis Doctoral
INF-592	Seminario de Tesis II (68 créditos)	Director de Tesis Doctoral

Nota:

INF-581: Corresponde a temas de especialidad, dictados por el tutor a sus estudiantes.

INF-582: Corresponde a temas de especialidad, dictados por el tutor a sus estudiantes.



ANEXO Nº 5:

INFRAESTRUCTURA Y EQUIPAMIENTO CON EL QUE CUENTA

EL PROGRAMA DII

Espacios físicos de que disponen los profesores y estudiantes para las actividades del Programa.

El Departamento de Informática ocupa una superficie de 3.120 m², en el Edificio F de la casa central de la UTFSM, entre los pisos 0 a 3º. A grandes rasgos, esta superficie comprende las siguientes áreas:

- 1) 34 oficinas (profesores, administrativos, secretarías, entre otros).
- 2) 8 Laboratorios (1 de uso exclusivo para Doctorado).
- 6) 2 Auditorios debidamente equipados, con internet (fija más wifi) y sistema de videoconferencia.
- 7) 10 Espacios comunes (cocina, cafetería, circulación, terraza, entre otros).
- 8) 10 Salas que prestan diversos servicios (salas de servidores, 1 UCSCI, Centro de Alumnos, Sala de Control, Sala de Estar, Sala de Clúster, entre otros).
- 12) 11 Baños (estudiantes, profesores y funcionarios).

Laboratorios y otras instalaciones disponibles para uso del Programa, con su respectivo equipamiento

El Departamento de Informática tiene, actualmente, 8 Laboratorios y cuenta con el siguiente equipamiento:

1. Laboratorio de Computación “LABCOMP”:

Este laboratorio es de uso público, tiene un número de 32 PCs, 1 pizarra, 1 proyector, 1 telón, 1 sala para ayudantes.

2. Laboratorio Aula Tecnológica:

Laboratorio de uso público, tiene un número de 20 PCs, 1 pizarra, 1 proyector, 1 telón, cuenta con sistema de videoconferencias y sistema de audio.

3. Laboratorio Hardware:

Este laboratorio es de uso específico para talleres y capacitación. Cuenta con un total de 10 PCs, 1 proyector y 1 telón.

4. Laboratorio “ADA LOVELACE”:

Este laboratorio es de uso específico para proyectos. Cuenta con 12 PCs, 1 mesa de reuniones, 1 pizarra, 1 proyector, 1 telón, servicio de cafetería y área de descanso.



5. Laboratorio “CSRG” (ex –distribuidos):

Laboratorio de uso específico para proyectos. Cuenta con un total de 8 PCs, mesa de reuniones, 1 pizarra, servicio de cafetería y área de descanso.

6. Laboratorio Integración Tecnológica “LabIT”:

Laboratorio de uso público, cuenta con un número de 16 PCs, mesa de reuniones, 1 pizarra, 1 proyector, 1 telón, 1 sala para ayudantes.

7. Laboratorio de Programación Avanzada “LABPRO”:

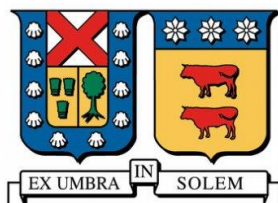
Laboratorio de uso público, cuenta con 38 PCs, mesa de reuniones, 1 pizarra, una sala para ayudantes y un área de descanso.

8. Laboratorio Computación Avanzada “Doctorado”:

Laboratorio de uso específico del DII, cuenta con un total de 17 PCs, modelo MAC de 21, 5 pulgadas, 2 estaciones de trabajo con capacidad para 4 doctorandos cada una. Existen 8 cubículos individuales, para profesores visitantes y estudiantes del programa. Una oficina de la Secretaría de Investigación y Postgrado también equipada con tecnología MAC, además de su propia impresora multifuncional. Este laboratorio cuenta con mesa de reuniones para 10 personas, 1 impresora multifuncional para uso de los estudiantes, 1 pizarra, 1 proyector, 1 telón, área de cafetería y de descanso.



ANEXO Nº 6



DECLARACION SIMPLE DEL ESTUDIANTE

En Valparaíso, a “Fecha”, Yo “Nombre del estudiante”, RUT “Nº Cédula de Identidad” “Nacionalidad”, con domicilio en “Domicilio particular”, “comuna”, cuya tesis para optar al Grado de Doctor en Ingeniería Informática de la Universidad Técnica Federico Santa María se denomina “título de la tesis”, formulo la siguiente declaración.

DECLARO QUE:

- El contenido de este trabajo de tesis es **original e independiente**, y no ha sido presentado en otra Universidad o Programa de Estudios, tanto a nivel nacional como internacional. Se excluyen de esta situación, los temas de tesis que se hayan entregado en el marco de un convenio actualmente vigente, que mantenga el Programa con otra casa de estudios.
- Dicho tema de tesis, es **exclusivo** para optar al grado de Doctorado en Ingeniería Informática de la Universidad Técnica Federico Santa María.
- Las investigaciones y resultados obtenidos e incluidos en dicho trabajo, son total y verídicamente **reproducibles**.
- Una vez presentado mi trabajo de tesis y firmada esta declaración simple, no podré retractarme, ni solicitar la detención del proceso de evaluación.

Me comprometo, al momento de suscribir esta declaración, a que todo lo señalado anteriormente se ajusta a la realidad.

En el evento que el Comité Académico del Doctorado en Ingeniería Informática estime que el contenido de esta declaración no se ha cumplido fielmente, estoy de acuerdo en que se me aplique la normativa que la Universidad contemple en este caso y que se encuentre actualmente vigente.

“Nombre”

“Firma”



ANEXO Nº 7

PROGRAMAS DE ASIGNATURAS DEL PROGRAMA DII

	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Teoría y aplicaciones de las "wavelets"	Sigla	INF-406
Créditos	4	Conocimientos previos: ILI-285 e ILI-286: Computación Científica I y II. Autorización del Profesor.	

Descripción

Asignatura teórico-práctica que cubre los aspectos fundamentales de la teoría y las aplicaciones de los "wavelets". Contiene elementos del análisis funcional, de la teoría de la aproximación y del análisis de Fourier, bancos de filtros, transformaciones de "wavelet" continua y discreta, análisis de multi-resolución, "wavelets" ortonormales, aplicaciones a la compresión de información, al procesamiento de imágenes y a problemas de valores de borde

Objetivos

Capacitar al estudiante en la comprensión de los temas fundamentales de la teoría de los "wavelets" con especial énfasis en sus aplicaciones al procesamiento de imágenes, compresión de "data" y los aspectos computacionales involucrados. Conducir al estudiante hacia temas actuales de investigación en el área y capacitarlo para comprender esa problemática. Capacitar al estudiante en la comprensión de los temas fundamentales del procesamiento de señales con especial énfasis en aplicaciones y en los aspectos computacionales.

Contenidos

1. Análisis de Fourier discreto.
2. Filtros y bancos de filtros.
3. Las transformaciones de "wavelet" continua y discreta.
4. "Wavelets" continuas, transformada breve de Fourier (STFT) y "frames".
5. Análisis de multiescala.
6. "Wavelets" ortonormales con soporte compacto.
7. Algoritmos y complejidad.
8. Aplicaciones: compresión de "data", procesamiento de señales e imágenes,



problemas de valores de borde, modelado estadístico y ``wavelets'', etc.

Metodología

Exposiciones del profesor y, de vez en cuando, de los alumnos en temas específicos. Discusión crítica de los temas tratados con miras a proyectarlos como temas de investigación. Estudio y trabajo personal (textos y artículos).

Evaluación

Examen final escrito.
Exposiciones (1 o 2) de artículos relevantes.

Bibliografía

- [1] St. Mallat. *A wavelet tor of signal processing*. Academic Press – Elsevier, 1999.
- [2] Ch. Blatter. *Wavelets, eine Einführung*. Vieweg, Braunschweig-Wiesbaden, 1998.
- [3] G. Strang, T. Nguyen. *Wavelets and filter banks*. Wellesley_Cambridge Press, Wellesley, MA, 1997.
- [4] M. Vetterli, J. Kovacevic. *Wavelets and subband coding*. Prentice Hall, Upper Saddle River, NJ, 1995.
- [5] B. Vidakovic. *Statistical modeling by wavelets*. Prentice John Wiley & Sons, Inc., New York, 1999.
- [6] I. Daubechies, *Ten lectures on wavelets*. SIAM, Philadelphia, PA, 1992.
- [7] A.K. Louis, P. Maass, A. Rieder. *Wavelets, Theorie und Anwendungen*. B.G. Teubner, Stuttgart, 1994.
- [8] Y. Meyer. *Ondelettes et operateurs*. Hermann, Paris, 1990

Elaborado:	Luis Salinas	Observaciones: Actualización: Diciembre 2012.
Aprobado:	Depto. de Informática	
Fecha:	10/01/2003	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Calidad y Productividad de Software	Sigla	INF-413
Créditos	4	Conocimientos previos: Ingeniería de Software	

Descripción

En este curso se presentan los conceptos clave de medición en Ingeniería de software, particularmente en las áreas de gestión de proyectos y aseguramiento de calidad de software. Se aplican diversas técnicas y modelos del estado del arte para el dimensionamiento de software, la estimación de esfuerzo, tiempos y costos, y la gestión cuantitativa de la calidad en la producción de software. Finalmente, se analizan los fundamentos básicos del mejoramiento de procesos de software.

Objetivos

- Dimensionar software mediante el uso de métricas.
- Aplicar técnicas y modelos de estimación de productividad de software.
- Aplicar técnicas y modelos para la gestión de calidad de software.
- Conocer los fundamentos del mejoramiento de procesos de software

Contenidos

1. Rol de las mediciones en software: mediciones de producto, proceso, recursos, paradigma Goal-Question-Metric.
2. Métricas de software: clásicas, puntos de función, orientadas a objeto.
3. Productividad de software: factores, modelos.
4. Estimación de esfuerzo: modelo Putnam, COCOMO, COCOMO II.
5. Calidad de software: técnicas de aseguramiento de calidad, mediciones de calidad, gestión cuantitativa de calidad.
6. Mejoramiento de procesos de software: enfoques, modelos, estado de la práctica.

Metodología

Clases expositivas, presentación y discusión de casos, trabajos prácticos, investigación independiente.

Evaluación

Trabajos prácticos (20%), Trabajo independiente de investigación (20%), Certamen: 20%, Examen final: 40%



Bibliografía

- [1] N. Fenton, S.L. Pfleegler, "Software Metrics, A Rigorous and Practical Approach", IEEE CS, Press, 2da Edición, 1998.
- [2] D. Galin, "Software Quality Assurance", Pearson Education, 2004.
- [3] J. Persse, "Process Improvement Essentials", O'Reilly, 2006.
- [4] S. McConnell, "Software Estimation", Microsoft Press, 2006.
- [5] IEEE Software.
- [6] Software Quality Journal.

Elaborado:	Marcello Visconti	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como curso de especialidad. Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Arquitectura de Software	Sigla	INF-414
Créditos	4	Conocimientos previos: ILI-225 Ingeniería de Software	

Descripción

El diseño de sistemas de software complejos requiere competencias de concepción, evaluación y construcción diferentes de las requeridas por diseño de aplicativos individuales. El énfasis radica en la satisfacción de propiedades sistémicas ("requisitos extra funcionales") y el uso de tecnologías para sistemas distribuidos. Este curso reporta técnicas, modelos y criterios para describir, evaluar y desarrollar sistemas de software complejos. Los elementos utilizados incluyen ejemplos de documentación de proyectos reales, casos de estudio, talleres grupales de evaluación, y lecturas complementarias.

Objetivos

Al aprobar el curso el alumno podrá:

- comprender arquitecturas de software como guías para la construcción de sistemas
- evaluar, comparar y mejorar especificaciones de arquitectura y las estructuras que ellas denoten, con criterios de calidad intrínseca (técnicos) y extrínsecos (objetivos)
- elaborar y describir arquitecturas de software a partir de especificaciones funcionales y propiedades sistémicas
- distinguir los problemas de arquitectura de los que no lo son, y explicar articuladamente las nociones básicas de la disciplina
- proponer arquitecturas para situaciones concretas

Contenido

1. Arquitectura en el proceso de desarrollo de software.
2. Descripción de arquitecturas: vistas, notaciones, perfiles UML, estilos de arquitectura.
3. Evaluación de arquitecturas: calidad de la especificación, el sistema y el proceso.
4. Reuso de arquitectura: arquitecturas de referencia, componentes, frameworks, patrones de arquitectura, líneas de productos.
5. Recuperación de arquitecturas.
6. Bases tecnológicas: middleware, transparencias de sistemas distribuidos, escalabilidad, disponibilidad.
7. Roles del arquitecto y estructura organizacional.



Metodología

Clases expositivas, proyectos parcelados durante el semestre, informes periódicos sobre lecturas, y presentaciones intermedias de proyectos. Se apunta a conocer el estado del arte en la elaboración, descripción y evaluación de arquitecturas de software, y practicarlo en un contexto de proyectos monitoreados. Se espera participación activa.

Evaluación

Examen escrito

Bibliografía

- [1] "Software Architecture in Practice (3rd Ed.)" Len Bass, Paul Clements, Rick Kazman: Addison-Wesley Professional (2012).
- [2] "Essential Software Architecture (2nd Ed.)" Ian Gorton: Springer (2011).
- [3] "Documenting Software Architectures: Views and Beyond (2nd Ed.)" Paul Clements, Felix Bachmann, Len Bass and David Garlan: Addison-Wesley Professional (2010).
- [4] "Evaluating Software Architectures: Methods & Case Studies," Paul Clements, Rick Kazman, Mark Klein: Addison-Wesley (2001).
- [5] "Pattern-Oriented Software Architecture, Vol.5: On Patterns and Pattern Languages." Frank Buschmann, Kevlin Henney, Douglas C. Schmidt: Wiley (2007)
- [6] "Just Enough Software Architecture: A Risk-Driven Approach". George Fairbanks: Marshall & Brainerd (2010).
- [7] Artículos y manuales contingentes al proyecto.
- [8] Referencias en Web

Elaborado:	Hernán Astudillo	Observaciones: Corresponde a asignatura de la especialidad Desarrollo de Software.
Aprobado:	Depto. de Informática	
Fecha:	03/05/2004	Actualización: Diciembre 2012



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Análisis de Programas	Sigla	INF-417
Créditos	4	Conocimientos previos: Fundamentos de Informática Teórica, Teoría de Autómatas	

Descripción

Esta asignatura forma parte del área de Desarrollo de Software. En ella, los estudiantes aprenderán conceptos y fundamentos de análisis de programas, y conocerán y aplicarán técnicas para el análisis y especificación formal de programas. También utilizarán herramientas existentes para poner en práctica sus conocimientos.

Objetivos

- Explicar las técnicas para el análisis de programas tratadas en clases
- Especificar en forma formal los requerimientos funcionales de un programa
- Analizar programas usando las técnicas tratadas en clase
- Elaborar una evaluación de la calidad de un programa, en base a las técnicas aplicadas

Contenidos

1. La crisis del Software, verificación y validación de Programas
2. Especificación formal de propiedades de Programas
3. Creación y Análisis de Modelos de Programas
4. Técnicas de análisis estático de programas
5. Validación dinámica de programas
6. Herramientas informáticas para el análisis de programas

Metodología

Clases expositivas con apoyo de medios audiovisuales

Aprendizaje basado en el análisis y discusión de casos

Aprendizaje basado en el desarrollo de proyecto (análisis de un programa usando varias técnicas)

Estudio independiente y exposiciones de estudiantes acerca de temas específicos

En las tareas, los estudiantes deberán escribir una especificación formal de un programa y analizarlo usando las técnicas aprendidas en la clase



Evaluación

Certámenes: 60%

Tareas: 40%

Bibliografía

- [1] F. Nielson, H. Nielson, C. Hankin, "Principles of Program Analysis", Springer, 2005
- [2] E. M. Clarke, O. Grumberg, D. Peled, "Model Checking", MIT Press, 2000
- [3] M. Pezze, M. Young, "Software Testing and Analysis", John Wiley, 2007.

Elaborado:	Jocelyn Simmonds	Observaciones: Se eliminó el proyecto grupal, aumentando la ponderación de las tareas
Aprobado:	Depto. de Informática	
Fecha:	07-07-2011	Actualización: Diciembre 2012



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Pruebas de Software	Sigla	INF-424
Créditos	4	Conocimientos previos: Ingeniería de Software	

Descripción

En este curso se presentan los conceptos fundamentales de las pruebas de software como técnica fundamental de aseguramiento de calidad. Se aplican técnicas para diseñar casos de prueba, y para planificar, especificar, ejecutar y evaluar las pruebas de software. Se utilizan técnicas de cobertura para determinar la calidad de las pruebas. Se utilizan técnicas y herramientas avanzadas para mejorar la efectividad de las pruebas de software. Se analizan los elementos críticos para la gestión del proceso de pruebas de software. Finalmente se discute el estado del arte en pruebas de software.

Objetivos

- Analizar los conceptos fundamentales de pruebas de software en el contexto del aseguramiento de calidad del software.
- Diseñar casos de prueba, planes de prueba y especificaciones de prueba utilizando técnicas apropiadas.
- Planificar, especificar, ejecutar y evaluar pruebas de software.
- Utilizar herramientas para mejorar la efectividad de las pruebas de software.
- Analizar los elementos críticos para la gestión del proceso de pruebas de software.

Contenidos

1. *Introducción:* software quality assurance, conceptos básicos de pruebas, temas esenciales, principios, aspectos psicológicos y económicos, proceso, estado del arte v/s práctica, taxonomía de errores, debugging.
2. *Fundamentos:* pruebas: definiciones, objetivos, casos de prueba, diseño de casos de prueba (black-box, white-box), cobertura de pruebas white-box, estrategias, métodos particulares de pruebas, testability, cleanroom, plan, especificación, ejecución y evaluación de pruebas.
3. *Gestión:* enfoques organizacionales, prácticas, tendencias, desafíos, mejoramiento de procesos, costos y beneficios, mediciones, herramientas prácticas en el mercado, estándares, documentación de pruebas.
4. *Temas avanzados de pruebas:* avances en pruebas de mutación, pruebas de regresión.



Metodología

Clases expositivas, trabajos prácticos e investigación independiente.

Evaluación

Trabajos prácticos: 20%

Trabajo independiente de investigación: 20%

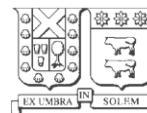
Certamen: 20%

Examen final: 40%

Bibliografía

- [1] Boris Beizer, *"Software testing techniques"*, Van Nostrand Reinhold Company Inc., 2nd Edition 1990.
- [2] Edward Ki, *"Software testing in the real world, improving the process"*, Addison Wesley, 1995.
- [3] Cem Kaner et al, *"Lessons learned in software testing"*, John Wiley, 2002.
- [4] Mauro Pezze, Michal Young. *"Software Testing and Analysis: Process, Principles and Techniques"*, John Wiley, 2007.
- [5] Rex Black. *"Pragmatic Software Testing: Becoming an Effective and Efficient Test Professional"*, John Wiley, 2007.

Elaborado:	Marcello Visconti	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como curso de especialidad.
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	Actualización: Diciembre 2012



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Interfaces Hombre- Máquina	Sigla	INF-427
Créditos	4	Conocimientos previos: Conocimientos básicos de ingeniería de software y estadística	

Descripción

Este curso introduce al alumno los conceptos generales de la compleja interacción entre personas y computadores. El diseño de interfaces interactivas se concibe bajo el prisma de una metodología centrada en el usuario, que incluye procesos de análisis de requerimientos, diseño de los prototipos, revisión y mantención de interfaces de sistemas interactivos. Además, se provee la guía para el diseño de diferentes estilos de interacción, entre las cuales se destacan Web, redes sociales y aplicaciones móviles. Los aspectos teóricos se complementarán con ejercicios prácticos de diseño.

Objetivos

Al aprobar el módulo el alumno debiera ser capaz de:

- Evaluar interfaces usuarias de software usando técnicas de evaluación heurística y de observación de usuarios.
- Conducir simples experimentos formales para evaluar las hipótesis de facilidad de uso
- Aplicar el diseño centrado en el usuario y los principios de ingeniería de facilidad de uso al diseño de una variedad de interfaces usuarias de software
- Implementar los componentes de interfaz usuaria usando entornos de rápida generación de prototipos

Contenidos

1. Factores Humanos del Software Interactivo
2. Teorías, Principios y Pautas
3. Procesos de Diseño
4. Test y Estudio de Facilidad de Uso
5. Manipulación Directa y tacto
6. Menú y Formularios
7. Diseño: Función y Estilo
8. Web, Web 2.0
9. Aplicaciones móviles



Metodología

Clases expositivas con apoyo de medios audiovisuales, resolución de casos y problemas, trabajos de aplicación

Evaluación

Dos certámenes, trabajo en grupo, trabajo individual, presentación. La nota final se calcula como el promedio ponderado de las notas.

Bibliografía

- [1] Shneiderman B. and Plaisant C., Designing the User Interface: Strategies for Effective Human-Computer Interaction, Cuarta Edición, Addison-Wesley, 2004
- [2] Tidwell J., Designing Interfaces: Patterns for Effective Interaction Design, O'Reily, 2005
- [3] Nielsen Norman Group, Report on Usability of Mobile Websites & Applications, 2010 <http://www.nngroup.com/reports/mobile/>
- [4] U.S. Department of Health and Human Services, Research-Based Web Design & Usability Guidelines, 2006
- [5] Nielsen J., Tahir M., Homepage usability: 50 websites deconstructed, New Riders Publishing, Virginia, USA, 2002

Elaborado:	Liubov Dombrovskaia	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como curso de especialidad. Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Programación de Sistemas	Sigla	INF-427
Créditos	4	Conocimientos previos: ILI-246, ILI-256	

Descripción

Este curso trata del desarrollo de aplicaciones que tienen íntima relación con el sistema operativo o la máquina, como lo son servicios de red de alto rendimiento. Incluye aspectos de programación en tiempo real.

Objetivos

- Programar aplicaciones con íntima relación con el sistema operativo o la máquina

Contenidos

1. Estándares: APIs, ABIs, ISO C, POSIX
2. Llamadas al sistema: Archivos, procesos, hebras, uso de red
3. Herramientas de desarrollo: Ambientes de programación, control de versiones, pruebas unitarias, análisis de código, *debugging*
4. Proyectos individuales o grupales

Metodología

Clases expositivas y desarrollo de pequeños proyectos para familiarizarse con las herramientas y temas, seguidos por un proyecto mayor (individual o grupal según envergadura) La primera parte del curso se desarrolla con dos bloques de exposición semanales, luego sólo reuniones semanales o quincenales de coordinación e informe de avance.

Evaluación

Certamen sobre los contenidos iniciales (puntos 1 a 3 de los contenidos), 25%; evaluación de los mini proyectos, 20%; desarrollo y presentación final del proyecto del semestre, 55%.



Bibliografía

- [1] Robert Love "Linux System Programming", O'Reilly Media (2007)
- [2] W. Richard Stevens "Advanced Programming in the Unix Environment", Addison-Wesley (2ª edición, 2005)
- [3] Manuales y demás documentación relevante según proyectos

Elaborado:	Horst von Brand	Observaciones: Actualización: Diciembre 2012.
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Sistemas Distribuidos	Sigla	INF-440
Créditos	4	Conocimientos previos: Conceptos de sistemas operativos y redes de computadores, programación de sistemas y lógica matemática.	

Descripción

La asignatura introduce al alumno en los problemas propios de los sistemas distribuidos de computación, donde se estudian técnicas y métodos que pueden ser aplicados en el diseño sistemas y servicios informáticos distribuidos, de manera abordar y resolver satisfactoriamente estos problemas.

Contenido: Caracterización de los sistemas distribuidos. Arquitecturas de Sistemas distribuidos. Programación y comunicación distribuida. Modelos de computación distribuida. Algoritmos distribuidos básicos. Tolerancia a falla y alta disponibilidad. Sistemas de base de datos distribuidos. Transacciones distribuidas. Replicación de datos. Seguridad de información en ambientes distribuidos. Tecnología de middleware

Objetivos

Al finalizar el curso los alumnos serán capaces de:

- Explicar los conceptos fundamentales y paradigmas asociados a computación distribuida.
- Describir los diferentes estilos de arquitecturas de sistemas distribuidos existentes y las técnicas de diseño que se aplican para implementar servicios informáticos distribuidos.
- Entender los algoritmos distribuidos básicos y cómo éstos pueden ser combinados para resolver diferentes tipos de problemas.
- Decidir qué técnicas y métodos son los más adecuados para la resolución de problemas en el área de sistemas distribuidos.



Contenidos

1. Introducción General a los Sistemas Distribuidos (2)
Motivación. Definición y características de un sistema distribuido. Ventajas y desventajas. Sistemas centralizados vs. distribuidos. Técnicas de distribución. Modelos y arquitecturas distribuidas. Sinopsis del curso.
2. Programación y Comunicación Distribuida (4)
Paradigmas de programación. Comunicación orientada a mensajes. Sockets en TCP/IP. Multicasting. Protocolos de comunicación de Middleware. Invocación remota de procedimientos. Middleware orientado mensajería (MoM). Data Streaming.
3. Computación Distribuida y Algoritmos Distribuidos (5)
Modelos de Sistemas Distribuidos. Tiempo, causalidad y ordenamiento de eventos. Relojes Lógicos. Sincronización de relojes. Observaciones válidas. Historias causales y relojes de vector. Cortes, estados globales y consistencia. Algoritmos distribuidos básicos. Elección distribuida. Instantánea distribuida. Detección de término. Detección de deadlock. Exclusión mutua distribuida.
4. Tolerancia a Fallas (4)
Conceptos y enfoques básicos sobre tolerancia a fallas y alta disponibilidad. Acuerdo bizantino. Sincronización de relojes. Memoria estable. Entrega fiable de mensajes. Broadcast fiable, atómico y causal. Recuperación y estado consistente. Procesos tolerantes a fallas.
5. Base de Datos y Transacciones Distribuidas (5)
Sistemas de Bases de Datos Distribuidas. Fragmentación y replicación de base de datos. Consultas distribuidas. Transacciones distribuidas. Control de concurrencia y serialización. Recuperación de errores. Protocolos de compromiso. Replicación de datos.
6. Infraestructura y servicios distribuidos (5)
Servicios de directorio y de nombres. Seguridad computacional e Infraestructura de Clave Pública. Sistemas de Archivos Distribuidos. Tecnología de middleware.

Metodología

Clases lectivas con dos sesiones a la semana para presentación y discusión de la materia, cuya comprensión es evaluada en dos partes (2 certámenes).

Un trabajo de investigación donde el alumno aplique los conceptos estudiados en el curso, que al término será presentado a los demás alumnos.



Evaluación

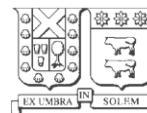
2 Certámenes: 67%

Trabajo de investigación: 33%

Bibliografía

- [1] Coulouris, et.al. "*Distributed Systems: Concepts and Design*" 5th. Edition, Addison Wesley, 2011.
- [2] A. Tanenbaum, M. van Steen, "Distributed Systems: Principles and Paradigms", 2nd. Edition, Prentice Hall, 2006.
- [3] M. Singhla & N.G. Shivaratri, "*Advanced Concepts in Operating Systems*", McGraw-Hill, 1994.

Elaborado:	Raúl Monge	Observaciones: Actualización: Diciembre 2012.
Aprobado:	Depto. de Informática	
Fecha:	20/06/2000	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Computación Gráfica	Sigla	INF-451
Créditos	4	Conocimientos previos: Programación, Estructuras de Datos y Arquitectura de Computadores	

Descripción

En esta asignatura se introduce a la Computación Gráfica. Se define un sistema computacional con gráfica y se programan aplicaciones a través de una librería con funciones gráficas como OpenGL. Temas que se estudian son: Interacción hombre-computador, transformaciones geométricas, proyecciones y modelación. Para estos temas se programan ejemplos en un lenguaje de programación como C/C++.

Objetivos

- Manejar los conceptos fundamentales de la computación gráfica en hardware, software y aplicaciones.
- Aplicar técnicas de programación en C/C++ en aplicaciones gráficas.
- Implantar estructuras de datos para aplicaciones con gráfica.
- Desarrollar programas utilizando las librerías gráficas OpenGL y GLUT.

Contenidos

1. Sistemas Gráficos y Modelos.
2. Programación de Aplicaciones Gráficas.
3. Entrada e Interacción.
4. Objetos Geométricos y Transformaciones.
5. Viewing.
6. Shading.
7. Implementación.
8. Trabajo con Modelos.

Metodología

Cada semana se realiza una clase. Los alumnos deben realizar 2 tareas en el semestre. Además trabaja cada alumno en un tema específico de la computación gráfica lo que implica en el semestre por lo menos 3 presentaciones sobre el tema: Presentación del tema, presentación de avance y presentación final. Al final se entrega un informe escrito. Los trabajos específicos pueden ser estudios bibliográficos o desarrollos de aplicaciones.



Evaluación

Presentaciones 20%
Trabajo Escrito 20%
Tareas 50%
Nota del Profesor 10%

Bibliografía

- [1] Edward Angel, Dave Shreiner: *"Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL™"*, Addison-Wesley, 6° Ed. 2012 (Texto Guía).
- [2] Robert Whitrow: *"OpenGL Graphics through Applications"*, Springer 2008.
- [3] Peter Shirley: *"Fundamentals of Computer Graphics"*, A.K. Peters 2° Ed. 2005.
- [4] Edward Angel: *"OpenGL: A Primer"*, Addison-Wesley 3° Ed. 2007.
- [5] Revista: IEEE Computer Graphics and Applications (ISSN 0272-1716).
- [6] Revista: ACM SIGGRAPH Computer Graphics (ISSN 0097-8930).
- [7] Revista: Computers & Graphics – An International Journal of Systems & Applications in Computer Graphics (ISSN 0097-8493)

Elaborado:	Hubert Hoffmann	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como asignatura de especialidad o electiva. Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Optimización Combinatoria	Sigla	INF-472
Créditos	4	Conocimientos previos: Programación Lineal	

Descripción

El curso presenta los conceptos fundamentales de la programación entera. Se estudia los modelos clásicos donde la programación entera es aplicada y los fundamentos de los métodos de resolución de los modelos de optimización combinatoria. Los conceptos teóricos son complementados con el uso de herramientas computacionales para el modelamiento y la resolución de problemas de optimización combinatoria.

Objetivos

- Formular modelos matemáticos deterministas para la optimización de operaciones representables por funciones sobre variables discretas.
- Conocer técnicas de resolución de modelos de optimización combinatoria provenientes de la Investigación de Operaciones.

Contenidos

1. Modelos de optimización combinatoria: conceptos básicos, variables enteras, variables binarias, formulación de modelos, modelos clásicos.
2. Resolución de modelos de optimización combinatoria: enumeración exhaustiva, técnicas de ramificación y acotamiento, técnicas de enumeración implícita, planos de corte.
3. Resolución utilizando herramientas computacionales.
4. Lenguajes para la formulación de modelos de optimización combinatoria.

Metodología

La asignatura contempla la realización de clases de exposición de los elementos teóricos, la realización de tareas individuales de aplicación de los conceptos vistos en clases y el estudio de casos y técnicas propuestas en artículos que presentan el desarrollo actual de la materia.

Evaluación

La asignatura se evalúa en base a certámenes (60%), análisis de artículos (20%) y tareas (20%).



Bibliografía

- [1] Combinatorial Optimization: Theory and Algorithms, Bernhard Korte & Jens Vygen, Springer, 2012.
- [2] Combinatorial Optimization, Rita Malhotra, C. S. Lalitha, Delhi Pankaj Gupta & Aparna Mehra, Narosa, 2006.
- [3] A First Course in Combinatorial Optimization, Jon Lee, Cambridge University Press, 2004

Elaborado:	Carlos Castro	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como curso de especialidad.
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	
		Actualización: Diciembre 2012



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Inteligencia Artificial Avanzada		Sigla	INF-474
Créditos	4	Conocimientos previos: Programación en C., Inteligencia Artificial		

Descripción

Este curso es una revisión de las nuevas tendencias y avances en los métodos de búsqueda basados en heurísticas más conocidos. Tanto los problemas de optimización combinatoria y los problemas de optimización continua serán tratados con mayor énfasis en la combinatoria. Se presentarán las principales técnicas discutiéndolas en forma crítica y sus variaciones. Se utilizarán papers clave incluyendo aplicaciones. Los estudiantes conocerán cómo y por qué estas técnicas funcionan, cuándo aplicarlas, las ventajas respecto de otras técnicas más tradicionales.

Objetivos

- Presentar al estudiante un panorama de las nuevas tendencias y avances en inteligencia artificial.
- Mostrar al alumno el uso de la inteligencia artificial en la resolución de problemas del mundo real.
- Desarrollar una visión crítica respecto a nuevas propuestas en esta área de investigación

Contenidos

1. Repaso de técnicas de inteligencia artificial para la resolución de problemas
2. Estado del arte en problemas clásicos y nuevas aproximaciones para su resolución
3. Formas de evaluar avances en métodos provenientes de la inteligencia artificial
4. Revisión de la literatura reciente proveniente de conferencias y publicaciones en revistas

Metodología

Clases expositivas, seminarios, reuniones y presentación de avances del proyecto



Evaluación

Cada tarea tiene una ponderación de un 10%, revisión de los papers un 10% y el proyecto un 40%.

Bibliografía

- [1] Artificial Intelligence: A Modern Approach, D. Russel and P. Norvig, 3rd. Edition, Prentice Hall, 2010.
- [2] G. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Addison Wesley, 2008.
- [3] Reeves Colin, "Modern heuristic technique for combinatorial problems", John Wiley & Sons, 1993.
- [4] Artificial Intelligence in the 21st Century, S. Lucci and D. Kopec, Mercury Learning and Information, 2012.
- [5] Adicional: Journal of Knowledge Intelligent Systems, Journal of Engineering Applications of Artificial Intelligence, IEEE Computational Intelligence

Elaborado:	Maria Cristina Riff	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como curso de especialidad.
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	Actualización: Diciembre 2012



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Modelamiento Estocástico y Simulación	Sigla	INF-475
Créditos	4	Conocimientos previos: Lenguajes de programación y Estadística computacional	

Descripción

Este curso se introducen los fundamentos de los procesos estocásticos, y sus aplicaciones a la modelización de sistemas complejos. El diseño de un sistema computacional exige no sólo satisfacer ciertas funciones, sino también determinar de forma cuantitativa el comportamiento y la eficacia del mismo. Para ello, típicamente debemos construir modelos que suelen incluir algún elemento estocástico. En el curso se introducen los principales modelos estocásticos, así como su utilidad para la toma de decisiones. El análisis de sistemas complejos suele conducir a problemas de optimización estocásticos de difícil solución. Una alternativa puede ser el empleo de la simulación, que esencialmente, consiste en la experimentación computacional de un modelo estocástico, que describe el comportamiento dinámico del sistema bajo estudio.

Objetivos

- Comprender los fundamentos de los procesos estocásticos y sus aplicaciones.
- Conocer diversos tipos de procesos estocásticos.
- Conocer las diversas etapas en el desarrollo de modelos de simulación.
- Conocer y comprender los principales algoritmos para generar números, variables aleatorias y procesos estocásticos.
- Conocer métodos para efectuar análisis de salidas de un modelo de simulación.
- Conocer métodos de validación de modelos de simulación.



Contenidos

1. *Introducción a los procesos estocásticos*: teorema de Kolmogorov, clasificación de procesos estocásticos, estacionarios y no estacionarios, procesos ergódicos, procesos de Markov, procesos de Poisson, aplicaciones a teoría de colas, sistemas de colas exponencial y no exponencial, y aplicación a modelos de pronósticos.
2. *Análisis Espectral de un proceso estocástico*: función de autocorrelación, función de densidad espectral, teorema de Wiener y Kintchine, análisis espectral de procesos simples y bidimensionales, y aplicación a procesamiento de señales.
3. *Introducción a modelos de simulación*: generalidades del modelado, simulación y método de Montecarlo, ejemplos ilustrativos en sistemas de colas.
4. *Generación de números aleatorios*: métodos congruenciales, mixtos, multiplicativos, aditivos y mezclas, métodos de registros desfasados, propiedades Test de aleatoriedad: Test de bondad de ajuste.
5. *Generación de variables aleatorias*: método de la Transformación Inversa, método de la Composición, Técnicas de aceptación y rechazo, generación de variables continuas, discretas, generación de variables correlacionadas, generación de procesos estocásticos.
6. *Análisis de salida de modelos de simulación*: medidas de desempeño, contrastes, intervalos de confianza, métodos de comparación.
7. *Técnicas de reducción de variancia*.
Validación de modelos de simulación: validación de datos, validación de supuestos, validación experimental, procedimientos estadísticos de validación

Metodología

Clases expositivas, tareas individuales, presentación de trabajo en forma de seminario

Evaluación

Tareas (30%), Examen (30%) y Proyecto (40%).

Bibliografía

- [1] Ross, S.M., "*Simulation*", Ed. Academic Press, 5 Edition, 2012.
- [2] Feldman, R.M. and Valdez-Flores, C., "*Applied Probability and Stochastic Process*", Ed. Springer, 2 Edition, 2010.
- [3] Nelson, B.L., "*Stochastic Modeling: Analysis and Simulation*", Ed. Dover, 2010.



- [4] Stroock, D.W., “*An Introduction to Markov Processes*”, Ed. Springer, 2005.
- [5] Law, A.M., “*Simulation Modeling and Analysis*”, Ed. McGraw-Hill, 4 Edition, 2006.
- [6] Kalos, M.H. and Whitlock, P.A., “*Monte Carlo Methods*”, Ed. Wiley-VCH, 2 Edition, 2008.
- [7] Robert, C.P. and Casella, G., “*Introducing Monte Carlo Methods with R*”, Ed. Springer Verlag, 1 Edition, 2009.
- [8] Revistas : Simulation, Journal of forecasting, ACM transaction on modeling
Computer Simulation, and Statistics and Computing.

Elaborado:	Héctor Allende	Observaciones: Abierto a estudiantes de Ingeniería Informática e Ingeniería Civil Informática como curso de especialidad.
Aprobado:	Depto. de Informática	
Fecha:	1/8/2012	
		Actualización: Diciembre 2012



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Redes Neuronales Artificiales		Sigla	INF-477
Créditos	4	Conocimientos previos: Lenguajes de programación, cálculo numérico y estadística computacional		

Descripción

Este curso introduce los fundamentos de las redes neuronales artificiales (*Artificial Neural Network*, ANN) y sus aplicaciones a problemas de Regresión, Clasificación y , Predicción y reconocimiento de patrones en general . En el curso se introducen algunos conceptos básicos de las ANN, para diferentes arquitecturas de redes aplicadas tanto a patrones estáticos como a patrones dinámicos. Se analizan diferentes aplicaciones de las ANN, así como sus posibilidades y limitaciones. Finalmente se estudian las redes neuronales recurrentes analizando los problemas de estabilidad, así como la efectividad y aplicación de algunos algoritmos constructivos.

Objetivos

- Conocer las diferentes arquitecturas de las ANN.
- Conocer diversos tipos de funciones de activación de las ANN.
- Conocer y comprender los principales algoritmos para generar ajustar los pesos de las ANN.
- Conocer los principios básicos de reconocimiento de patrones.
- Conocer los principales avances de las ANN, para los problemas de clasificación y reconocimiento de patrones.

Contenidos

1. *Introducción a las ANN:* neuronas simples y redes, las neuronas como funciones.
2. *Redes de una Capa:* función lineal discriminante, separabilidad lineal, discriminante lineal generalizado, técnicas de mínimos cuadrados, perceptrón, discriminante lineal de Fisher.
3. *Redes de Multicapa:* redes Feed-forward, unidades Threshold y sigmoidal, espacio de pesos simétricos, redes de alto orden, regresión Pursuit, teorema de Kolmogorov.
4. *Redes recurrentes:* la máquina de Boltzman, Back- propagation recurrente, aprendizaje secuencial, aprendizaje reforzado, aplicaciones a modelos de pronósticos.



5. *Funciones de Error*: diversas funciones de error; Gauss, Minkowski, Varianza, modelos de distribución condicional, problemas de clasificación usando suma de cuadrados, entropía.
6. *Optimización de Parámetros*: diversas técnicas de optimización de parámetros; método del gradiente, gradiente conjugado, método de Newton y método de Levenberg Marquardt.
7. *Tópicos Avanzados en Redes Neuronales*: redes probabilísticas, optimización del aprendizaje, técnicas Bayesianas.
8. *Aplicaciones*: procesamiento de imágenes, selección de características, predicción de series de tiempo.

Metodología

Clases expositivas, tareas individuales, presentación de trabajo en forma de seminario.

Evaluación

Examen (30%), tareas (30%) y presentación de monografía (40%).

Bibliografía

- [1] Bishop, *"Neural Networks for Pattern Recognition"*, Ed. Clarendon Press Oxford, 1995.
- [2] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
- [3] C. Sammut, G.I. Webb. *Encyclopedia of Machine Learning* Springer, 2011.
- [4] C. Bishop *Pattern recognition and Machine Learning*, Springer, 2006.
- [5] T. Hastie, R. Tibshirani and J. Friedman. *The Elements of statistical Learning*, Ed Springer 2001.
- [6] R.M. Hristev, *"Artificial Neuronal Networks"*, Ed. The book ANN, 2000.
- [7] T.L. Fine, *"Feedforward Neural Network Methodology"*, Ed. Springer Verlag, 1999.
- [8] Revistas: IEEE Trans. on Neuronal Networks, Neuronal computing, Pattern Recognition, Journal of Machine Learning, Neuronal computing, Pattern Recognition, Journal of Intelligent Data Analysis, Engineering Optimization.

Elaborado:	Héctor Allende	Observaciones: Abierto a estudiantes de Postgrado Ingeniería de otras especialidades. Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	1/8/2012	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Análisis Inteligente de Datos	Sigla	INF-478
Créditos	4	Conocimientos previos: ILI-280 Estadística computacional	

Descripción

Este curso pretende introducir a los alumnos en el análisis de datos basado en sistemas inteligentes y abordar algunas sus aplicaciones. Se estudiarán los fundamentos teóricos de los problemas de clasificación; agrupamiento y pronóstico en grandes bases de datos, tanto desde el punto de vista estadístico, como computacional; además, se abordarán los fundamentos probabilísticos de los modelos clásicos paramétricos y no paramétricos y aquellos basados en máquinas de aprendizaje. Se estudiarán las metodologías más relevantes para construir modelos de clasificación, agrupamiento y pronóstico, poniendo énfasis en aspectos computacionales y se abordarán algunas técnicas de validación.

Objetivos

Al aprobar la asignatura el alumno será capaz de conocer, comprender y aplicar los principios de análisis automático de datos utilizando técnicas avanzadas de inteligencia computacional, de manera tal que, el descubrimiento de nuevo patrones, sirva como sistema de apoyo a la toma de decisiones.

Contenidos

1. Introducción a la Minería de datos (Data Mining) y su relación con Análisis inteligente de datos (IDA).
2. Conceptos estadísticos del proceso de aprendizaje: Clasificación, Regresión y Estimación de densidades de probabilidad, selección de modelos, la maldición de la dimensionalidad.
3. Distribuciones muestrales, remuestreo; Bootstrap, Boosting Adaboosts Tipos de Errores y Error de predicción.
4. Métodos lineales de clasificación : Función discriminante , clasificadores Bayesianos; El paradigma Bayesiano; Inferencia Bayesiana; Modelos Bayesianos de clasificación y Redes Bayesianas.
5. Técnicas de Inteligencia Computacional aplicados al análisis de datos: Redes Neuronales Artificiales, aprendizaje supervisado, Redes Feedforward , Error Backpropagation Regularización en redes neuronales.
6. Técnicas de Inteligencia Computacional aplicados al análisis de datos: Redes de kohonen , aprendizaje no supervisado, clustering.



7. Lógica Multivaluada : Conjuntos fuzzy y lógica fuzzy, Modelos de datos, Árboles de decisión fuzzy, redes neuro-fuzzy.
8. Métodos Estadísticos Multivariados: Análisis de Componentes Principales , Kernel PCA y Métodos de Kernel: El Perceptron , Máquinas de vector soporte, Kernel.
9. Series de Tiempo: Sistemas lineales y no lineales predicción en Series de tiempo, métodos bootstrap en series de tiempo.
10. Búsqueda de Conocimiento en Base de Datos: Modelos Predictivos, Inducción de Reglas, Detección de datos atípicos y Visualización.
11. Análisis de Datos Reales: construcción y verificación de modelos, Diseño de Experimentos computacionales, conjuntos de validación, validación cruzada Simulación y Preparación y análisis de Experimentos y análisis de resultados

Metodología

El curso se desarrollará con clases expositivas de 90 minutos. El alumno deberá realizar tareas individuales para complementar su aprendizaje. Durante el desarrollo del curso el alumno deberá seleccionar lecturas relevantes de artículos y elaborar un proyecto de investigación aplicada en algún tópico de su elección y presentar su propuesta en forma de seminario.. El proyecto será evaluado mediante una presentación oral y un reporte escrito. El examen final del curso estará basado en los tópicos tratados en clases por el profesor

Evaluación

Tareas	30%
Proyecto	40%
Examen	30%

Bibliografía

- [1] Michael Berthold and David J. Hand. "Intelligent Data Analysis". Ed. Spinger. 2003
- [2] Christoph M. Bishop. Pattern Recognition and Machine Learning. Ed. Springer, 2006.
- [3] "Principles of Data Mining". D. Hand, H. Mannila and P. Smyth. "Principles of Data Mining". The MIT Press. 2001
- [4] B.D. Ripley. "Pattern Recognition and Neural Networks". Cambridge University Press. 2004
- [5] R. Duda, P. Hart and D. Stork. "Pattern Classification". John Wiley & Sons Inc. 2000.
- [6] Revistas: Journal of Machine Learning, IEEE Trans. on Neuronal Networks, Neuronal computing, Pattern Recognition, Journal of Intelligent Data Analysis.etc.



--

Elaborado:	Héctor Allende	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	19/05/2010	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Redes Complejas	Sigla	INF-480
Créditos	4	Conocimientos previos: Estructuras de Datos, Fundamentos de la Informática, Estadística Computacional.	

Descripción

Este curso es una introducción al reciente campo interdisciplinario de las redes complejas, con énfasis en redes sociales y tecnológicas. Esta área, que trata con grafos de gran tamaño y estructura no aleatoria, ha emergido con fuerza desde fines de los años 90, y ha contribuido a entender una amplia variedad de fenómenos, desde el crecimiento de la Web hasta la propagación de epidemias o rumores, o la formación de comunidades. A la vez ha puesto de manifiesto la existencia de patrones comunes en la conectividad de sistemas tan distintos como la regulación génica, el cerebro humano, las conexiones entre conceptos, o las redes sociales online. El curso cubre los principales modelos, propiedades genéricas, herramientas de análisis y los ejemplos más importantes de redes específicas.

Objetivos

Al aprobar la asignatura el alumno tendrá una perspectiva general del estudio de las redes complejas, su relevancia en distintos ámbitos (con énfasis en las redes más cercanas al quehacer informático), y las herramientas prácticas e intelectuales para su análisis. Además habrá trabajado con datos concretos de redes y estará preparado para abordar proyectos en que este tipo de información sea relevante. En varios temas específicos habrá tenido acceso a literatura de investigación de punta, o estará en condiciones de abordarla.

Contenidos

1. Introducción a las redes de gran tamaño; ejemplos. Modelos clásicos (Erdős-Renyi). Propiedades principales: distancias, clusterización, distribución de grados. Pajek, Network Workbench, Gephi y otras herramientas de análisis y visualización.
2. Modelos principales de redes complejas: Watts-Strogatz, Barabási-Albert. Robustez estructural y dinámica bajo falla o ataque. Otras propiedades topológicas de redes. Otros modelos de redes complejas, estáticos y generativos.
3. Medidas de centralidad en redes: análisis de roles, análisis de enlaces; aplicación en redes sociales y en el grafo de la Web. Asortatividad; "club de ricos". Identificación de comunidades. Propiedades de navegabilidad y



búsqueda descentralizada.

4. Patrones en redes. Introducción a la representación gráfica y a la minería de datos de redes. Muestreo de redes e inferencia de propiedades estructurales. Interacción de características topológicas locales y globales.
5. Redes dinámicas y dinámicas en redes. Ejemplos de estudios en redes sociales, tecnológicas y biológicas. Validación de modelos dinámicos. Fenómenos de difusión y contagio; estrategias de control. Auto-organización y fenómenos emergentes.
6. Aspectos específicos de redes importantes: el grafo de la Web; redes sociales online; internet y otras redes tecnológicas; redes semánticas; colaboración científica; redes biológicas.

Metodología

El curso se desarrollará en clases expositivas de 90 minutos de duración. El alumno deberá elaborar tareas en forma individual para complementar su aprendizaje; principalmente se analizarán datos de redes. Adicionalmente, se asignarán lecturas relevantes a los contenidos del curso, y se evaluarán mediante tareas y/o presentación en clases. El examen final del curso será basado en una prueba escrita..

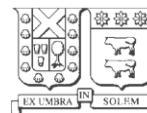
Evaluación

Tareas y presentaciones 60%
Interrogaciones y examen 40%

Bibliografía

- [1] Brandes, U., Erlebach, T. (Eds.), *Network Analysis: Methodological Foundations* (LNCS 3418), Springer-Verlag, 2005.
- [2] Bornholdt, S., Schuster, H.G. (Eds.), *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley-VCH, 2003.
- [3] Caldarelli, G., Vespignani, A. (Eds.), *Large Scale Structure And Dynamics Of Complex Networks*. Word Scientific, 2007.
- [4] Cook, D., Holder, L. (Eds.), *Mining Graph Data*. John Wiley & Sons, 2006.
- [5] Barrat, A., Barthélemy, M., Vespignani, A., *Dynamical Processes on Complex Networks*. Cambridge University Press, 2008.
- [6] Newman, M., *Networks: An Introduction*. Oxford University Press, 2010.

Elaborado:	A. Moreira	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01-07-2010	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Tecnologías de búsqueda Avanzada en la Web	Sigla	INF-481
Créditos	4	Conocimientos previos: estructuras de datos y algoritmos, bases de datos, probabilidad y estadística.	

Descripción

La asignatura cubre los fundamentos de recuperación de información que permiten comprender y desarrollar tecnologías de búsqueda de información en la Web. Cubre también temas más avanzados que permiten que el estudiante comprenda el estado del arte de esta área, entre ellas búsqueda en la Web social, búsqueda vertical y búsqueda por facetas.

Objetivos

Al aprobar la asignatura el estudiante será capaz de:

- Comprender los fundamentos de recuperación de información en texto.
- Aplicar modelos de recuperación de información en sistemas de búsqueda en la Web.
- Comprender los principales desafíos y limitaciones de los métodos existentes para realizar búsquedas en la Web social.
- Comprender y conocer tendencias en búsqueda en la Web.
- Analizar nuevas tecnologías y métodos de búsqueda, entre ellas búsqueda en la Web social.

Contenidos

1. Fundamentos de extracción de información en texto: Leyes del texto, técnicas de extracción de información desde texto, estructuras de datos para texto, evaluación de recuperación de información en texto, técnicas de procesamiento de lenguaje natural.
2. Modelos de recuperación de información en texto: Modelo Booleano, modelo Tf-Idf, modelo BM25.
3. Búsqueda en la Web: Caracterización de la Web, extracción de información en la Web (*crawling*), análisis de hiper-enlaces, arquitecturas de motores de búsqueda.
4. Búsqueda en la Web 2.0: Caracterización de la Web 2.0, análisis de redes sociales en la Web.



Metodología

1. Clases expositivas con apoyo de medios audiovisuales.
2. Desarrollo de ejercicios en clases que permitirán ilustrar los conceptos del área.
3. Actividades grupales: Tareas de procesamiento de texto en grupo que permitirán trabajar en torno al desarrollo de un sistema de búsqueda en la Web.
4. Actividades individuales: Pruebas escritas cortas.

Evaluación

1. 5 pruebas escritas cortas
2. 3 tareas

Bibliografía

[1] Manning, Ch., Raghavan, P., Schütze, H. "Introduction to Information Retrieval", Cambridge University Press, 2008.

Elaborado:	Marcelo Mendoza	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	07-07-2011	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Tema Especial en Informática: Computación Peer to Peer	Sigla	INF-490
Créditos	4	Conocimientos previos: Sistemas Distribuidos, Redes, Sistemas Operativos	

Descripción

Este curso es un curso avanzado con énfasis en la propia exploración e investigación. La evaluación se realizará con dos certámenes y un proyecto. El proyecto está orientado a diseñar e implementar un servicio a gran escala usando técnicas Peer to Peer.

Objetivos

- . Presentar al estudiante los algoritmos que existen para manejar sistemas distribuidos de gran escala.
- . Mostrar al alumno la forma de manejar los aspectos relacionados con la tolerancia a falla, disponibilidad del sistema y de seguridad.
- . Presentar aplicaciones existentes en diversos ámbitos tales como: sistemas de bases de datos, sistemas de archivos, correo electrónico, e-business, computación móvil.

Contenidos

1. Problemática de los Sistemas a Gran Escala: historia, problemas de identificación, de partición, latencia, fallas bizantinas.
2. Funciones Hashing: aspectos teóricos y prácticos. Propiedades matemáticas de las funciones de hash. Ejemplos MD5, SHA-1, SHA-256.
3. Técnica Peer to Peer: conceptos, definición y aplicaciones.
4. Overlays Peer-to-Peer estructurados (PASTRY, CHORD, TAPESTRY, KADEMLIA) sus semejanzas y diferencias conceptuales. Propiedades estructurales de estos sistemas.
5. Ruteo en Peer to Peer: conceptos y ejemplos de ruteo con prefijo, ruteo de saltos exponenciales. Mantención de tablas de ruteo.
6. Tolerancia a Falla: problemas, algoritmos, réplicas. Ejemplos en PASTRY y CHORD.
7. Seguridad: Manejo de fallas bizantinas, métodos de autenticación de usuarios, encriptación convergente.
8. Overlays Peer-to-Peer no estructurados (GNUTELLA). Propiedades.



Algoritmos de búsqueda (K-Random Walk).

9. Manejo de nodos maliciosos (Sistemas de Reputación, Accountability). Algoritmos de Membership para la creación de comunidades de nodos confiables.
10. Estimación de la cantidad de nodos en los sistemas estructurados y no estructurados.
11. Conclusiones, investigación actual y aplicaciones

Metodología

El curso es de tipo presencial apoyado por documentación disponible en página WEB . Incluye análisis de artículos recientes y relevantes en el tema. El curso incluye una presentación y discusión de los problemas de investigación más relevantes en el tema, así que propuestas de solución.

Evaluación

La evaluación global corresponde a un 50% del promedio de los dos certámenes, y un 50% correspondiente a un trabajo de investigación que incluye el análisis de artículos en el tema.

Bibliografía

- [1]SCRIBE: The Design of a Large-Scale Event Notification Infrastructure, Antony I. T. Rowstron, Anne-Marie Kermarrec, Miguel Castro, Peter Druschel, NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication, November 2001.
- [2]Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems, Antony I. T. Rowstron, Peter Druschel, Middleware '01: Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, November 2001.
- [3]Re-Chord: a self-stabilizing chord overlay network, Sebastian Kniesburges, Andreas Koutsopoulos, Christian Scheideler, SPAA '11: Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures, June 2011.
- [4]Chord: a scalable peer-to-peer lookup protocol for internet applications, Ion Stoica, Robert Morris, David Liben-Nowell, David R. Karger, M. Frans Kaashoek, Frank Dabek, Hari Balakrishnan, IEEE/ACM Transactions on Networking (TON).
- [5]Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, Petar Maymounkov, David Mazières, IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems, March 2002.
- [6]Fixed Interval Nodes Estimation: An accurate and low cost algorithm to estimate the number of nodes in Distributed Hash Tables, Xavier Bonnaire, Information Sciences, Volume 218, Publisher: Elsevier Science



Inc, January 2013.

- [7] A critical analysis of latest advances in building trusted P2P networks using reputation systems, Xavier Bonnaire, Erika Rosas, WISE'07: Proceedings of the 2007 international conference on Web information systems engineering (LNCS), December 2007.
- [8] WTR: a reputation metric for distributed hash tables based on a risk and credibility factor, Xavier Bonnaire, Erika Rosas, Journal of Computer Science and Technology, Volume 24 Issue 5, Springer, September 2009.
- [9] Erika Rosas, Olivier Marin and Xavier Bonnaire, CORPS: Building a Community of Reputable PeerS in Distributed Hash Tables, The Computer Journal, Vol. 54, N°10, pp 1721- 1735.

Elaborado:	Xavier Bonnaire	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01/05/05	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema Especial en Informática: Arquitecturas Empresariales		Sigla	INF-490
Créditos	4	Conocimientos previos: Conocimientos básicos sobre la ingeniería de procesos, Conocimientos básicos sobre administración de empresas, Analizar, modelar, diseñar procesos y sistemas, Comprensión lectora en inglés		

Descripción

Esta asignatura forma parte del plan de estudios de la carrera Ingeniería Civil Informática y forma parte del área de conocimientos de Desarrollo de Software. Es parte del plan de estudios de Licenciatura para las carreras de Ingeniería Civil. En esta asignatura los estudiantes conocerán y aplicarán los conceptos fundamentales de BPM en el contexto de una estructura empresarial, conociendo los modelos fundamentales y la integración de estos.

Objetivos

Al aprobar la asignatura el alumno será capaz de:

- Identificar y explicar los conceptos fundamentales de Enterprise Architecture (EA)
- Definir, describir y evaluar una EA para una organización específica.
- Explotar notaciones y estándares de BPM como requisitos para una EA.
- Usar herramientas de mercado para configurar una EA.
- Describir Arquitecturas y Modelos de Referencia EA para algunos rubros.
- Describir Frameworks y Modelo de Referencia para Gobernabilidad de SOA.

Contenidos

1. Introducción: Trasfondo histórico; conceptos; objetivos y beneficios de una EA.
2. Vistas y niveles de una EA
3. Áreas de aplicación en una EA
4. La relación EA & BPM & SOA
5. De EA a EAM
6. Enterprise frameworks; ARIS framework, Zachman framework, TOGAF. FEAF
7. Definición de una EA usando P-BPM: Planificación Estratégica de Proyectos BPM para una Organización.



8. Metamodelo estructural de EA: Elementos estructurales; vistas y sus relaciones; métricas de calidad y control.
9. Herramientas y Plataformas EA (ARIS, Idungu, comparación con otras herramientas)
10. Arquitecturas de Referencia: ETOM, SCOR, IAA.
11. Gobernabilidad SOA

Metodología

Clases expositivas con apoyo de medios visuales
Aprendizaje basado en prácticas de laboratorio
Técnicas de análisis sobre casos
Discusión en clases sobre experiencias de las prácticas realizadas

Evaluación

2 certámenes, realización de presentaciones, lectura complementarias

Bibliografía

- [1] Marc Lankhorst et al. (2009), Enterprise Modelling, Communication and Analysis, Architecture at Work, Springer, Second Edition (+)
- [2] Enterprise Architecture as Strategy, Jeanne Ross et al, Harvard Business School Press, 2006
- [3] The 3rd Workshop on Trends in Enterprise Architecture Research (TEAR 2008) Sydney, Australia, December 1
- [4] The Open Group: The Open Group Architecture Framework (TOGAF), version 8 Enterprise Edition. The Open Group (2005)
- [5] Schekkerman, J.: Extended Enterprise Architecture Framework Essentials Guide. Institute For Enterprise Architecture Developments (2006)

Elaborado:	Hernán Astudillo	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	15/11/2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema Especial en Informática: Seminario de Sistemas Distribuidos		Sigla	INF-490
Créditos	4	Conocimientos previos: Sistemas Distribuidos (INF-440).		

Descripción

Esta asignatura tiene como propósito explorar y profundizar en materias específicas relacionados con el curso previo de introducción general a Sistemas Distribuidos. Cada año, según los temas que estén más vigentes y los intereses de los alumnos, se trabaja en una temática específica.

Objetivos

Al finalizar el curso los alumnos serán capaces de:

- Conocer en profundidad un tema específico relacionado con Sistemas Distribuidos, conociendo los trabajos más relevantes relacionados con el tema.
- Realizar una investigación bibliográfica.

Contenido

Variable según el año. En general, el profesor realiza una presentación a los temas y entrega conocimientos básicos que ayuden a la comprensión de los temas tratados, mientras los alumnos en paralelo trabajan en una investigación bibliográfica respecto a un tema específico elegido.

Metodología

En la primera parte del curso el profesor presentará la problemática global del tema del seminario, debiendo el alumno elegir y definir un tema de investigación, que en la segunda parte del curso, deberá exponer ante sus compañeros. El curso termina con un informe técnico sobre la investigación realizada, que será usado para evaluar mediante un certamen la comprensión que tiene sobre los temas desarrollados por sus demás compañeros.

Evaluación

Trabajo de investigación: 80%
Certamen sobre todos los trabajos desarrollados: 20%



Bibliografía

Se define según la ocasión. Principalmente se usan publicaciones de revistas.

Elaborado:	Raúl Monge	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	Junio 2002	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema Especial en Informática: Seguridad de Sistemas Distribuidos		Sigla	INF-490
Créditos	4	Conocimientos previos: Un curso introductorio de seguridad, conocimientos básicos de UML. Los slides estarán en inglés.		

Descripción

La mayoría de los sistemas de información modernos son distribuidos. En este curso se considera los requerimientos de seguridad para este tipo de sistemas, incluyendo sus ataques y sus posibles defensas. Se aplica Unified Modeling Language (UML) y patrones para describir arquitecturas y mecanismos específicos.

En particular, se abordan sistemas como servicios de red (web services), computación en nube (Cloud computing), y sistemas inalámbricos. Además, se estudia una metodología para construir y evaluar sistemas seguros. Un objetivo importante del curso es proporcionar una perspectiva de cómo la seguridad se coordina con los aspectos funcionales de una aplicación. Otro objetivo es comprender la estructura e implementación de los mecanismos para poder construir y evaluar sistemas seguros.

Objetivos

Los objetivos de aprendizaje son:

- 1) Conocer y comprender los requerimientos de seguridad que tiene los sistemas de información distribuidos
- 2) Aplicar UML y patrones para describir aspectos de seguridad en arquitectura de sistemas y de software
- 3) Conocer una metodología para construir y evaluar sistemas seguros, considerando su estructura y los mecanismos integrados
- 4) Comprender cómo la seguridad se relaciona con los aspectos funcionales de un aplicación



Contenidos

1. Motivación y perspectiva.
2. Patrones de seguridad. Patrones de abuso. Arquitecturas de referencia. Revista deUML.
3. Metodologías para aplicaciones seguras distribuidas
4. Arquitecturas distribuidas. Estilos y patrones: Broker, pipes & filters, blackboard, agentes. P2P. Middleware. SOA.
5. SOA y Web services: arquitecturas, ataques, y estándares. Patrones de seguridad para web services. Identidad
6. Seguridad en nubes (cloud computing): amenazas y defensas.
7. Seguridad de sistemas inalámbricos. Estándares de seguridad. Seguridad de redes ad hoc y sensores.
8. Sistemas ciberfísicos. Smart grid (red inteligente). Internet of things.
9. Aplicaciones complejas distribuidas: sistemas médicos, financieros, de transporte

Metodología

Clases con tareas de diseño e investigación

Evaluación

Tres tareas (30%) y examen o proyecto final (70%)
Todos se hacen en la casa.

Bibliografía

- [1] E.B.Fernández, E.Gudes, and M. Olivier, The design of secure systems, bajo contrato con Addison-Wesley. (Borrador como notas de clases)
- [2] E.B.Fernandez, "Security patterns in practice: Building secure architectures using software patterns", libro a aparecer en Wiley Series on Software Design Patterns. Mayo 2013. (Borrador como notas de clases)
- [3] E.B.Fernández, O.Ajaj, I.Buckley, N.Delessy-Gassant, K.Hashizume, M.M. Larrondo-Petrie, "A Survey of Patterns for Web Services Security and Reliability Standards". Future Internet 2012, 4, 430-450. <http://www.mdpi.com/1999-5903/4/2/430/>
- [4] Keiko Hashizume, David G. Rosado, Eduardo Fernández-Medina, Eduardo B. Fernández, "An Analysis of Security issues for Cloud Computing", accepted for the Journal of Internet Services and Applications, Springer. (SCOPUS)
- [5] Keiko Hashizume, Eduardo B. Fernández, and María M. Larrondo-Petrie, "A Reference Architecture for Cloud Computing", enviada para



publicación.

- [6] A.V. Uzunov, E.B. Fernández & K. Falkner (2012), "Securing distributed systems using patterns: A survey", Computers & Security, 31(5), 681 - 703. (ISI, IF= 0.868) doi:10.1016/j.cose.2012.04.005
- [7] A.V. Uzunov, E.B. Fernandez & K. Falkner , "Engineering Security into Distributed Systems: A Survey of Methodologies", accepted for the Journal of Universal Computer Science (ISI)
- [8] [Sta12] W. Stallings and L. Brown, Computer security: Principles and practice (2nd Ed.), Pearson 2012.

Elaborado:	Eduardo Fernández	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	26-12-2012	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema Especial en Informática: Astro-Informática		Sigla	INF-490
Créditos	4	Conocimientos previos: ILI-236, FIS-120		

Descripción

Corresponde a un curso electivo del área de Sistemas Computación, donde los alumnos aprenderán temas fundamentales para comprender astronomía, y cuáles son sus implicancias dentro de la astro-ingeniería, y en particular en la Ingeniería Informática. El objetivo general de este curso es dar al alumno suficiente conocimiento para saber cuáles son los diversos tipos de telescopios, la tecnología desarrollada para realizar observaciones astronómicas, los sistemas computacionales envueltos, y las características principales y más relevantes de los distintos proyectos de astronomía desde una perspectiva de la Ingeniería Informática.

Objetivos

- Explicar movimientos terrestres y celestes.
- Realizar observaciones en telescopios virtuales (ópticos y radio).
- Conocer estándares de Observatorios Virtuales (OV).
- Explicar distintos tipos de observaciones astronómicas (óptica, radio, alta energía) y su relación con la informática.
- Crear implementaciones básicas de observatorios virtuales (IVOA).
- Conocer el estado actual de la astro-ingeniería, y cuáles son los actuales y próximos desafíos en el área (data mining sobre OV)

Contenidos

1. Historia y fundamentos teóricos de la astronomía y sus instrumentos.
2. Observatorios Virtuales (OV) y diferentes tipos de Telescopios.
3. Algoritmos para búsqueda en OV
4. Algoritmos para procesamiento sobre OV (CASA)
5. Estado de la Práctica
6. Nuevos desafíos

Metodología

1. Clases expositivas con apoyo de medios audiovisuales.
2. Utilización de telescopios virtuales para realizar observaciones con datos de telescopios almacenados.
3. Visitas a observatorios de mediana envergadura para realizar observaciones y conocer el software que utilizan en él.



4. Familiarización y exposición de un tema de actualidad en astro-ingeniería y su relación con la ingeniería informática.
5. Taller de desarrollo de aplicaciones en observatorios virtuales

Evaluación

C1 = Nota Certamen 1; C2 = Nota Certamen 2; TA = Nota Talleres; E = Nota Exposición; P = Nota Participación.

Calificación Nota Final $C1*0.25 + C2*0.25 + TA*0.30 + E*0.10 + P*0.10$

Bibliografía

- [1] ACS Workshop 2007, 2008, 2009, 2010, UTFSM, ESO, NRAO, NAOJ, JAO. <http://acsworkshop.inf.utfsm.cl/>
- [2] ALMA Common Software 8.0 -- Documentation, ALMA Project <http://www.eso.org/projects/alma/develop/acs/OnlineDocs/index.html>
- [3] ACS C++ Component/Container Framework Tutorial, by Grega Milcinski, Matej Sekoranj, Bernhard Lopez, David Fugate, Alessandro Caproni.
- [4] ALMA Common Software and Python, by David Fugate.
- [5] ACS Java Component Programming Tutorial, by Heiko Sommer.
- [6] Astronomical Algorithms, Second Edition, Jean Meeus, Willmann-Bell Inc. Publishers, 2005.
- [7] A. Richard Thompson, J.M. Moran, G. W. Swenson, Jr. Interferometry and Synthesis in Radio Astronomy, second edition (2001).
- [8] Telescope Control, Second Edition, Mark Trueblood and Russel Merle Genet, 1997.
- [9] Telescope Control Systems, Krewalk et al., United State Patent 4682091, 1987.
- [10] The SAO/NASA Astrophysics Data System, <http://www.adsabs.harvard.edu/>
- [11] VLT Common Software 2009 -- Documentation Kit, European Southern Observatory.

<http://www.eso.org/projects/vlt/sw-dev/wwwdoc/VLT2009/dockit.html>

Elaborado:	Mauricio Solar	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	24-08-2012	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema Especial en Informática: Ingeniería de Software Experimental	Sigla	INF-492
Créditos	4	Conocimientos previos: Ingeniería de Software	

Descripción

Experimentación en Ingeniería de Software concierne el uso del diseño y análisis experimental para validar ideas y creencias, en un campo ampliamente dominado por suposiciones y especulaciones, orientado en forma práctica para los ingenieros de software. ¿Son válidas nuestras suposiciones? ¿Qué afirmaciones de la comunidad de desarrollo de software son válidas? ¿Bajo qué circunstancias son válidas? Responder estas preguntas es crítico para otorgar mayor certeza a las ideas en que se fundamenta la Ingeniería de Software. Durante la construcción de software no se utilizan, habitualmente, técnicas formales de experimentación. Este hecho contrasta con las prácticas comunes de otras ingenierías y campos científicos, en las cuales es obligatorio realizar una rigurosa experimentación que apoye las investigaciones realizadas. En este curso se discute el uso de Análisis y Diseño de Experimentos en Ingeniería de Software, estableciendo las bases teóricas para la efectiva realización de experimentos.

Objetivos

Al finalizar el curso el alumno:

- Comprenderá la importancia de validar empíricamente las técnicas usadas en el desarrollo de software
- Sabrá aplicar diversas técnicas de experimentación en Ingeniería de Software
- Conocerá algunas experiencias reales de experimentación en Ingeniería de Software
- Conocerá líneas de investigación actuales y futuras en Ingeniería de Software Experimental

Contenidos

1. Introducción a la experimentación en Ingeniería de Software
2. Principales tipos de diseños y análisis de experimentos
3. Estudios experimentales reales realizados en Ingeniería de Software (diseño, ejecución, análisis crítico)
4. Estado del arte en Ingeniería de Software Experimental (líneas de investigación actuales y futuras, desafíos de la disciplina)



Metodología

1. Clases tipo
2. seminario de 90 minutos, combinando exposición con discusión grupal
3. Lectura sistemática guiada de artículos fundamentales y capítulos de libros, elaboración de comentarios semanales
4. Selección de un tópico de profundización, selección de bibliografía relevante, elaboración de informes periódicos quincenales e informe final
5. Diseño, ejecución y análisis crítico de estudios experimentales en Ingeniería de Software

Evaluación

1. Informes periódicos/comentarios semanales/quincenales: 30%
2. Informe final: 30%
3. Presentación oral: 30%
4. Participación: 10%

Bibliografía

- [1] Experimentation in Software Engineering. Claes Wohlin, Per Runeson, Martin Höst, Magnus Ohlsson, Björn Regnell, Anders Westlén. Springer, 2012
- [2] Basics of Software Engineering Experimentation. Natalia Juristo, Ana María Moreno. Kluwer, 2001
- [3] Experimentation in Software Engineering: An Introduction. Claes Wohlin, Per Runeson, Martin Höst. Springer, 1999
- [4] Empirical Software Engineering Issues. Vic Basili et al (editors). Lecture Notes in Computer Science 4336. Springer-Verlag, 2007

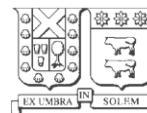
Conferencias

- [1] Empirical Software Engineering and Measurement (ESEM)

Revistas

- [1] Journal of Empirical Software Engineering (ESE)
- [2] IEEE Software

Elaborado:	Marcello Visconti	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	20/07/2007	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema Especial en Informática: Mejoramiento de Procesos de Software	Sigla	INF-493
Créditos	4	Conocimientos previos: Ingeniería de software, calidad de software, Productividad de software	

Descripción

En este curso se analizan a fondo las técnicas para diagnosticar, planificar y ejecutar acciones orientadas a mejorar los procesos de producción de software, partiendo de la premisa que se pueden obtener mejores productos de software como consecuencia de procesos más maduros, i.e. que incorporan adecuadamente las prácticas clave de la producción de software. Se discutirá conceptos básicos, visiones, métodos, modelos (ej. CMMI, normas ISO, etc.), herramientas y experiencias prácticas de mejoramiento de procesos de software, y se analizará el estado del arte en estas materias.

Objetivos

- Conocer, aplicar y adaptar técnicas de diagnóstico, planificación y acción para el mejoramiento de procesos de software.
- Definir planes de acción y métricas para el mejoramiento de procesos de software.
- Conocer y contrastar enfoques modernos alternativos para el mejoramiento de procesos de software.
- Desarrollar un análisis crítico de la problemática de mejoramiento de procesos de software

Contenidos

1. Conceptos básicos de mejoramiento de procesos de software.
2. Modelo IDEAL, modelos de madurez, meta modelos.
3. Métodos de evaluación de procesos: Constelaciones CMMI, SCAMPI, otros
4. Nuevos desarrollos: CMMI-Acq, CMMI-Svc, ISO, PSP-TSP, CM Agilismo, Six Sigma, otros.
5. Experiencias prácticas de mejoramiento de procesos de software.

Metodología

Clases expositivas, presentación y discusión de casos, trabajos prácticos, investigación independiente.



Evaluación

1. Participación en presentaciones y discusiones: 10%
2. Trabajos prácticos: 15%
3. Investigación independiente y presentación sobre un tema asignado: 10 %
4. Desarrollo y presentación de un artículo científico propio sobre un tema avanzado: 25%
5. Exámen final: 40%

Bibliografía

- [1] J. Persee, "Software Improvement Essentials", O'Reilly, 2006.
- [2] S. García, R. Turner, "CMMI Survival Guide", Addison-Wesley, 2007.
- [3] H. Oktaba, M. Piattini, "Software Process Improvement for Small and Medium Enterprises: Techniques and Case Studies", Premier Reference Sou 2008.
- [4] R. Grady, "Successful Software Process Improvement", Prentice Hall, 1997.
- [5] CrossTalk, The Journal of Defense Software Engineering
- [6] IEEE Software
- [7] Software Quality Journal

Elaborado:	Marcello Visconti	Observaciones: Actualización: Diciembre 2012.
Aprobado:	Depto. de Informática	
Fecha:	10/03/2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Compresión de Texto	Sigla	INF-520
Créditos	5	Conocimientos previos: Estructuras de datos	

Descripción

Al finalizar este curso el alumno conocerá los fundamentos de teoría de la información, será capaz de reconocer qué tipos de textos pueden ser comprimidos, conocerá las distintas familias de algoritmos de compresión, conocerá a fondo las principales herramientas de compresión de texto disponibles en la actualidad, y será capaz de modificar las herramientas de compresión actuales para agregarles mejoras o proponer nuevas variantes.

Objetivos

- Entregar los fundamentos de teoría de la información relacionados a compresión de secuencias de texto sin pérdida.
- Estudiar las familias principales de compresores de texto.
- Estudiar las principales herramientas de compresión de texto del estado del arte

Contenidos

1. Teoría de la Información: Complejidad de Kolmogorov y sus limitaciones. Entropía, entropía relativa e información mutua. Teorema de Shannon. Desigualdad de Jensen y otras de teoría de la información. Propiedad de equipartición asintótica, entropía empírica. Procesos estocásticos, cadenas de Markov y su entropía. Procesos ergódicos. Compresión universal, convergencia y redundancia.
2. Compresión Estadística: Modelamiento y codificación. Modelamiento estático, semiestático y dinámico. Códigos y sus propiedades. Desigualdad de Kraft. Códigos óptimos y redundancia. Códigos de Fano y de Shannon. Códigos de Huffman. Optimalidad. Codificación aritmética. Códigos dinámicos o adaptativos: Huffman y aritméticos. Modelamiento de lenguaje natural. Códigos multiarios. Códigos densos y su uso para bases de datos de texto. Compresión PPM.
3. Compresión Basada en Diccionarios: Compresión por sustitución de substrings. Compresión Lempel-Ziv: LZ77, LZ78, LZW y variantes. Universalidad y convergencia. Compresión basada en gramáticas. Re-Pair, Sequitur. Compresión de texto estructurado.
4. Compresión basada en Transformación: Transformación de Burrows-Wheeler. Move-to-front y run-length encoding. Compresión de árboles rotulados.



5. Herramientas de Compresión: gzip, bzip, lzma, ppm-d, google snappy. Comparación práctica entre ellas.

Metodología

El curso se desarrollará mayoritariamente mediante clases magistrales.

Evaluación

Individual, mediante dos certámenes y la entrega de un informe de investigación y una tarea (implementación + evaluación de un método de compresión, o modificación de una herramienta de compresión existente).

Bibliografía

- [1] T. Cover and J. Thomas. Elements of Information Theory. Wiley, 2nd ed, 2006.
- [2] T. Bell, J. Cleary, and I. Witten. Text Compression. Prentice-Hall, 1990.
- [3] I. Witten, A. Moffat, and T. Bell. Managing Gigabytes. Morgan Kauffmann, 2nd ed, 1999.
- [4] D. Salomon. Data Compression: The Complete Reference. Springer, 4th ed, 2007.

Revistas

- [1] Jeffrey Scott Vitter: Design and analysis of dynamic Huffman codes. J. ACM 34(4): 825-845 (1987)
- [2] Ian H. Witten, Radford M. Neal, John G. Cleary: Arithmetic Coding for Data Compression. Commun. ACM 30(6): 520-540 (1987)
- [3] Jacob Ziv, Abraham Lempel: A Universal Algorithm for Sequential Data Compression. IEEE Transactions on Information Theory 23(3): 337-343 (1977)
- [4] Jacob Ziv, Abraham Lempel: Compression of Individual Sequences via Variable-Rate Coding. IEEE Transactions on Information Theory 24(5): 530-536 (1978)
- [5] Burrows M and Wheeler D (1994), *A block sorting lossless data compression algorithm*, Technical Report 124, Digital Equipment Corporation
- [6] Paolo Ferragina, Fabrizio Luccio, Giovanni Manzini, S. Muthukrishnan: Compressing and indexing labeled trees, with applications. J. ACM 57(1): (2009)

Elaborado:	Diego Arroyuelo	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	16-01-2012	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Evaluación de Arquitectura de Software	Sigla	INF-524
Créditos	5	Conocimientos previos:	

Descripción

Exploración sistemática de literatura científica reciente relativa a la medición, comparación y evaluación de arquitecturas de software.

Objetivos

Al aprobar el curso el alumno podrá:

- [7] evaluar, comparar y mejorar especificaciones de arquitectura y las estructuras que ellas denoten, con criterios de calidad intrínseca (técnicos) y extrínsecos (objetivos)
- [8] elaborar y describir modelos de calidad para arquitecturas de software
- [9] distinguir los problemas de arquitectura de software de los que no lo son, y explicar articuladamente las nociones básicas de la disciplina
- [10] proponer formas de evaluación de arquitecturas para situaciones concretas

Contenidos

- 9. Modelos de calidad para arquitecturas de software
- 10. Evaluación de arquitecturas de software
- 11. Comparación de arquitecturas de software

Metodología

- Lectura sistemática de artículos de la literatura científica, reciente y/o fundamental.
- Talleres de técnicas específicas (e.g. árboles de utilidad).

Evaluación

- Evaluación de los talleres de técnicas específicas.
- Artículo publicable en conferencia internacional (indexada o no).

Bibliografía

- [1] *Evaluating Software Architectures: Methods and Case Studies*. P.Clements, R.Kazman, M.Klein. Addison-Wesley Professional (2001). ISBN-10: 020170482X.
- [2] "A Basis for Analyzing Software Architecture Analysis Methods". Kazman, Bass, Klein, Lattanze, Northrop. *Software Quality Journal*, 13, 329–355, 2005
- [3] *ISO/IEC 25010, Software product Quality – Requirements and Evaluation (SQuaRE) — Quality Model*. ISO – International Standards organization, Geneva (2006).
- [4] "Five Ontological Levels to Describe and Evaluate Software Architectures". H.Astudillo. Revista de la Facultad de Ingeniería de la Universidad de Tarapacá



(2004).

- [5] "Standard quality model to design software architecture". F.Losavio. Journal of Object Technology, 1(4), 2002, pp.165-178.
- [6] "Tools and Methods for Evaluating the Architecture". Software Engineering Institute, Carnegie Mellon University.
<http://www.sei.cmu.edu/architecture/tools/evaluate/>. Last visited: 2012-01-02

Revistas

- [1] Proceedings de WICSA, QoSA, ECSA, FSE, ICSE
[2] Lecture Notes in Computer Science (LNCS), Springer

Elaborado:	20 Julio 2006	Observaciones: Actualizado Enero de 2013.
Aprobado:	05-09-2007	
Fecha:		



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Computación Autónoma	Sigla	INF-528
Créditos	5	Conocimientos previos:	

Descripción

La computación autónoma (CA) fue propuesta en 2001 por investigadores arguyendo que sistemas computacionales complejos, imitando al cuerpo humano, necesitan un sistema nervioso autónomo que administra tareas corporales sin intervención consciente. Tales sistemas deberían controlar independientemente su mantención regular y optimización, y exhibir 4 propiedades: auto-configuración, auto-optimización, auto-sanación, y auto-protección.

Objetivos



Al aprobar el curso el alumno podrá:

- [11] comprender y describir los conceptos clave de CA
- [12] describir y comparar enfoques alternativos a los problemas de CA
- [13] identificar publicaciones y grupos de investigación del área, y los problemas de investigación abiertos en CA
- [14] proponer estudios que generen conocimiento novedoso y validable sobre CA

Contenidos

1. Introducción a la Computación Autonomica: Orígenes, evolución y dirección
2. Arquitectura de los Sistemas Autonomicos
3. Propiedades de los Sistemas Autonomicos
4. Arquitecturas de Aplicaciones con propiedad de auto-sanación
5. Arquitecturas de Aplicaciones Auto-Adaptivas
6. Patrones de diseños inspirados en la biología para los sistemas autonomicos.
7. Plataforma de desarrollo de los Sistemas Autonomicos

Metodología

Lectura, presentación y resumen de artículos en la literatura de CA.

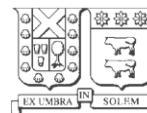
Evaluación

El alumno deberá preparar una publicación final sobre CA, e.g. exploración de un problema abierto, estudio de enfoques alternativos, o cualquier propuesta que sea considerado interesante por la comunidad (a juzgar por su publicabilidad).

Bibliografía

Artículos

- [1] "A survey of Autonomic Computing: Degrees, Models, and Applications". M. Huebscher, J. McCann: ACM Computing Surveys, 40(3) (2008).
- [2] "The Vision of Autonomic Computing", Jeffrey O'Kephart, David M. Chess: IEEE Computer, 36(1) (2003), pp. 4150.
- [3] "An Architecture based Approach to Self-Adaptive Software." Oreizy, P. and Taylor, R.N.: IEEE Intelligent Systems, 14 (3), p.5462, May/June, 1999.
- [4] "Toward an Autonomic Service Management Framework: A Holistic Vision of SOA, AON, and Autonomic Computing", Yu Cheng; Leon-Garcia, A.; Foster, I.: IEEE Communications Magazine, 46(5), May 2008, pp.138–146
- [5] "Guest Editors' Introduction: Autonomic Computing", Menasce, D.A.; Kephart, J.O.: IEEE Internet Computing, 11(1), Jan-Feb 2007 Pp: 18–21
- [6] "Architectural Constraints in the Model-Driven Development of Self-Adaptive Applications", Khan, M.U.; Reichle, R.; Geihs, K.; IEEE



Distributed Systems Online, 9(7), July 2008, pp. 1-10

- [7] "Towards architecture based selfhealing systems", E. M. Dashofy, A. van der Hoek, R. N. Taylor: First workshop on Self-healing (2002), pp.2126.
- [8] "A Reconfiguration Framework for Self-Healing Software," Jeongmin Park, Giljong Yoo, and Eunseok Lee: International Conference on Convergence and Hybrid Information Technology (2008)
- [9] "Research challenges of autonomic computing", Jeffrey O. Kephart: ICSE 2005 (27th ACM international conference on Software engineering). (2005)
- [10] "Bits of History, Challenges for the Future and Autonomic Computing Technology", Hausi A. Muller: WCRE 2006 (13th Working Conference on Reverse Engineering) (2006) pp: 9-18

Libros

- [1] *Autonomic Computing: Concepts, Infrastructure and Applications*. Manish Parashar, Salim Hariri: CRC Press, Taylor & Francis Group (2007).
- [2] *Autonomic Computing* (On Demand Series), Richard Murch: IBM Press (2004)

Revistas

- [1] Bibliotecas digitales de IEEE, ACM, Springer y ScienceDirect.
Proceedings of the IEEE Autonomous and Autonomic Systems (ICAS) Conference

Elaborado:	Hernán Astudillo	Observaciones:
Aprobado:	Depto. de Informática	Actualización: Diciembre 2012
Fecha:	01/03/2009	
 UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática		

Asignatura	Tema Especial en Informática: Programación Paralela Aplicada Avanzada	Sigla	INF-534
Créditos	5	Conocimientos previos: Sistemas Distribuidos, Redes, Sistemas Operativos	

Descripción

Este curso es un curso dedicado a la programación paralela con las últimas arquitecturas paralelas existentes (procesadores multi-core, clusters, grids). El objetivo del curso es ser capaz de programar un algoritmo en paralelo en el marco de una tesis de Magister o de Doctorado, así que escribir un short paper para presentar los resultados obtenidos.

Objetivos



- . Presentar las arquitecturas modernas para la programación paralela.
- . Presentar los problemas de paralelización y los métodos para resolverlos.
- . Saber detectar y solucionar los problemas de sincronización y de secciones críticas. Saber programar un algoritmo usando los Threads POSIX con un procesador multi-core..
- . Saber programar con el interfaz de programación paralela MPI (Message Passing para redes locales y clusters).
- . Programación de un algoritmo paralelo (a elección) y presentación de los resultados en un short paper.

Contenidos

Introducción

1. Los computadores paralelos obsoletos.
2. Las arquitecturas modernas para la programación paralela (Multicore, Clusters, Grids).
3. Objetivos de la programación paralela

Paralelización

1. Noción de Speedup, Leyes de Amdahl y Gustafson.
2. Técnicas de paralelización (decomposición).
3. Paralelización automática.
4. Pensar Paralelo (mejorar el grado de paralelismo).
5. Modelos de Memoria Compartida, Message Passing. Lo que se puede paralelizar.
6. Lo que no se puede paralelizar.

Gestión de recursos compartidos

1. Accesos concurrentes (Inconsistencia de los datos).
2. Ejemplos.
3. Secciones críticas y exclusión mutua
4. Semáforos y secuencialización



Sincronización.

1. ¿Por qué sincronizar?
2. Barriers, Rendez-Vous
3. Uso de los semáforos
4. Modelo Productor / Consumidor (1-1, 1-C, P-C)

Programación Multi-Threads con LINUX / UNIX

1. Definición de un Thread (diferencias con el multi-processos)
2. Los Threads POSIX (biblioteca de programación)
3. Modelos de programación con Threads (Pool de Threads, Master/Slave, Modelo híbrido)

Programación con Message Passing Interface (Redes, Clusters, ...).

1. Modelo del Message Passing. Diferencia con la memoria compartida. Biblioteca de programación. MPI (Message Passing Interface).

Herramientas de programación

1. IDE.
2. Debuggers (ddd, Valgrin, Dbmalloc).
3. Profilers (Gcc, Gprof).

Metodología

El curso es de tipo presencial apoyado por documentación disponible en página WEB . Incluye análisis de artículos recientes y relevantes en el tema. El curso incluye una presentación y discusión de los problemas de investigación más relevantes en el tema, así que propuestas de solución.

Evaluación

La evaluación global corresponde a un 50% del promedio de los dos certámenes, y un 50% correspondiente a un proyecto de investigación. El proyecto incluye:

1. La programación de un algoritmo paralelo con un tema a elección.
2. La escritura de un short-paper para presentar y analizar los resultados (en inglés).
3. Una presentación oral del short-paper de tipo conferencia internacional (en inglés).

Bibliografía



- [1] Pthreads programming, Bradford Nichols, Dick Buttlar, Jacqueline Proulx Farrell, Publisher: O'Reilly & Associates, Inc., September 1996.
- [2] Pthreads Performance Characteristics on Shared Cache CMP, Private Cache CMP and SMP, Ian K. T. Tan, Ian Chai, Poo Kuan Hoong, ICCEA '10: Proceedings of the 2010 Second International Conference on Computer Engineering and Applications - Volume 01, March 2010.
- [3] Using MPI (2nd ed.): portable parallel programming with the message-passing interface, William Gropp, Ewing Lusk, Anthony Skjellum, Publisher: MIT Press, December 1999 .
- [4] Recent Advances in the Message Passing Interface, Yiannis Cotronis, Anthony Danalis, Dimitris Nikolopoulos, Jack Dongarra, 18th European MPI Users' Group Meeting, EuroMPI, Santorini, Greece, September 18-21, 2011.
- [5] Task synchronization and allocation for many-core real-time systems, Pi-Cheng Hsiu, Der-Nien Lee, Tei-Wei Kuo, EMSOFT '11: Proceedings of the ninth ACM international conference on Embedded software,, October 2011.
- [6] The Little Book of Semaphores, Allen B. Downey, Second Edition, Ver. 2.1.12, 2007. <http://www.greenteapress.com/semaphores/>.
- [7] Valgrind: Programming Tool, Memory Debugger, Memory Leak, Performance Analysis, Debugging, Julian Seward, GNU General Public License, Lambert M. Surhone, Miriam T. Timpledon, Susan F. Marseken, Publisher: Betascript Publishing, March 2010 .

Elaborado:	Xavier Bonnaire	Observaciones: Actualización: Diciembre 2012.
Aprobado:	Depto. de Informática	
Fecha:	01/05/05	





Asignatura	Modelos computacionales en Series de Tiempo	Sigla	INF-560
Créditos	5	Conocimientos previos: Lenguajes de programación, Computación Científica, Estadística Computacional e Introducción a las máquinas de aprendizaje.	

Descripción

Este curso pretende introducir a los alumnos en el análisis de series de tiempo y sus aplicaciones. Se estudiarán los fundamentos teóricos del problema de pronóstico de series de tiempo, tanto desde el punto de vista estadístico, como computacional; además, se abordarán los fundamentos probabilísticos de los modelos clásicos de series de tiempo y aquellos basados en máquinas de aprendizaje. Serán tratadas las metodologías más relevantes para construir modelos de series de tiempo poniendo énfasis en aspectos computacionales y se abordarán diversas metodologías de validación.

Objetivos

Al término del curso el alumno debe ser capaz de:

- Comprender los fundamentos teóricos del análisis de series de tiempo.
- Conocer las metodologías clásicas de análisis de series de tiempo y sus limitaciones.
- Aplicar los modelos clásicos a problemas de series no estacionarias y de alta frecuencia.
- Aplicar las metodologías de análisis de series de tiempo basados en máquinas de aprendizaje en diversos problemas reales.
- Diseñar modelos de series de tiempo y validarlos empíricamente.

Contenidos

1. Características de una serie de tiempo.
 - Fundamentos teóricos.
 - Procesos estocásticos.
 - Teorema de descomposición de Wold.
 - Modelos estadísticos de series de tiempo.
 - Series de tiempo estacionarias y no estacionarias.
2. Inteligencia computacional
 - Introducción.
 - Computación flexible.



- Algoritmos probabilísticos.
- Computación Evolutiva.
- Métodos Híbridos.
- Lógica Fuzzy.
- Aplicaciones.

3. Modelos clásicos.

- Introducción.
- Análisis exploratorio de datos.
- Transformaciones en series de tiempo.
- Modelos ARMA y ARIMA.
- Ecuaciones en diferencias.
- Modelos integrados para datos no estacionarios.
- Identificación, Estimación y Pronóstico.
- Construcción de modelos.
- Validación de modelos.
- Otros modelos ARFIMA, GARCH (opcional).
- Aplicaciones.

4. Máquinas de aprendizaje en Series de Tiempo

- Introducción.
- Redes Neuronales de multicapa.
- Máquinas de soporte vectorial.
- Modelos NARMA.
- Pronóstico de Series Multivariadas.
- Series de Tiempo Caóticas.

5. Temas opcionales.

- Modelos de espacios de estados.
- Modelos estadísticos en el dominio de la frecuencia.
- Series de tiempo de alta frecuencia.
- Tópicos de finanzas computacionales.
- Series de tiempo multidimensionales.
- Pronóstico de series de tiempo mediante Lógica Fuzzy.
- Modelos Neuro-Fuzzy en series de tiempo.
- Predicción de series georeferenciadas (Kriging).
- Análisis de correlación de procesos espaciales.

Metodología

El curso se desarrollará con clases expositivas de 90 minutos. El alumno deberá



realizar tareas individuales para complementar su aprendizaje.

Durante el desarrollo del curso el alumno deberá seleccionar lecturas relevantes de artículos y elaborar un proyecto de investigación en algún tópico de su elección en forma de seminario y discusión del estado del arte. Este proyecto será evaluado mediante una presentación oral y un reporte escrito.

El examen final del curso estará basado en los tópicos tratados en clases por los profesores.

Evaluación

Tareas	30%
Proyecto	40%
Examen	30%

Bibliografía

[1]C. Bishop. Pattern Recognition and Machine Learning, Springer Verlag 2006.

[2]R. Shumway and D. Stoffer. Time Series Analysis and Its Applications with R Examples, 3rd edition, Springer 2011.

[3]A. Palit and D. Popovic. Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications, 1st edition. Springer Verlag 2005.

[4]G. Box, G. Jenkins and G. Reinsel. Time Series Analysis: Forecasting and Control, 4th edition, Wiley, 2008.

[5]R. Tsay. Analysis of Financial Time Series, Wiley, 3rd edition, 2010.

[6]

Elaborado:	Héctor Allende	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	02/07/2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Métodos Computacionales en Teoría de Funciones	Sigla	INF-562
Créditos	5	Conocimientos previos: Contenidos asignaturas ILI-285 e ILI-286: Computación Científica I y II. Autorización del Profesor.	

Descripción

Durante la última década, la ciencia de la computación y la teoría de la aproximación han llegado a integrarse en un área de la investigación que subsume aspectos relevantes de varias disciplinas, tales como las redes neuronales y sus derivados, el diseño asistido por computador, el procesamiento de imágenes, las funciones especiales de la Física-Matemática, las transformaciones conformes, las "wavelets", los fractales, y otros temas de la ciencia, la técnica y la matemática aplicada. El objetivo de este curso es discutir, y aplicar a algunas situaciones concretas, los resultados más recientes de esta activa área de la investigación en la frontera entre la ciencia de la computación y la teoría de la aproximación.

Objetivos

Aplicar métodos y estrategias de diseño de algoritmos a problemas en el ámbito de la aproximación real y compleja.

Aplicar esos métodos y algoritmos a problemas de la ciencia y la técnica.

Determinar la complejidad promedio y los peores casos de esos algoritmos.

Contenidos

1. Elementos de la teoría de funciones y de la aproximación.
2. Métodos computacionales en la teoría de funciones y de la aproximación
3. Métodos computacionales en la teoría de la convolución.
4. Transformaciones conformes numéricas.
5. Análisis de los algoritmos estudiados.

Metodología

Sesiones en la modalidad de seminario, con clases expositivas periódicas, exposiciones de artículos y problemas específicos por parte de los participantes, desarrollo de ejemplos y tareas (tanto teóricas como de programación).

Discusiones técnicas y preparación de artículos y monografías.



Dedicación a la asignatura

Exposición del profesor	20 Hrs.
Estudio Personal	50 Hrs.
Confección de Informes	20 Hrs.
Resolución de Problemas	60 Hrs.
Total	150 Hrs.

Evaluación

Disertaciones individuales de los alumnos acerca de temas escogidos de los contenidos temáticos.

Redacción de artículos y monografías

Presentación y defensa de los resultados resumidos

Calificación

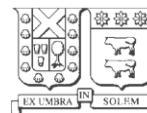
Promedio de las notas obtenidas en los instrumento que se apliquen

Bibliografía

- [1] St. Ruscheweyh et al., *Computational Methods and Function Theory*. World Scientific, Singapore, 2009/2005/2001/1997/1994.
- [2] St. Ruscheweyh, L. Salinas. *New Pólya-Schoenberg type theorems*. Journal of Mathematical Analysis and Applications, 363 (2) (2010), pp. 481-496.
- [3] St. Ruscheweyh, L. Salinas. *Universally Prestarlike Functions as Convolution Multipliers*. Mathematische Zeitschrift, 263 (3) (2009) , pp. 607-617.
- [4] St. Ruscheweyh, L. Salinas, T. Sugawa. *Completely Monotone Sequences and Universally Prestarlike Functions*. Israel Journal of Mathematics, Volume 171, Number 1 (2009), pp. 285-304.
- [5] N. Papamichael, N. Stylianopoulos. *Numerical Conformal Mapping*. World scientific, Singapore, 2010.

Elaborado:	Luis Salinas	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	03/08/2010	

	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
--	--



Asignatura	Metodología de la Investigación.	Sigla	INF-565
Créditos	5	Conocimientos previos:	

Descripción

Este curso introduce a los alumnos en la metodología de la investigación científica en Ingeniería Informática y les permite desarrollar algunas competencias necesarias en la formulación de proyectos de investigación y en la escritura de artículos y tesis.

Objetivos

Al finalizar este curso, los alumnos:

- Conocerán los diferentes tipos de investigación teórica, aplicada y experimental.
- Conocerán las principales visiones de la filosofía de la ciencia.
- Serán capaces de diseñar experimentos computacionales.
- Serán capaces de preparar artículos de investigación, reportes técnicos y tesis.
- Serán capaces de formular proyectos de investigación.

Contenidos

1. Epistemología de la ciencia.

- Introducción: tipos de ciencias (natural, artificial, exactas, etc.).
- Sociología de la ciencia: disciplinas y comunidades, praxis científica, rol de la publicación.
- Filosofía de la ciencia.
 - i. Empirismo.
 - ii. Positivismo y demarcación.
 - iii. Popper y falsacionismo.
 - iv. Kuhn y paradigmas.
 - v. Lakatos y programas de investigación.
 - vi. Feyerabend y la argumentación contra el método.
 - vii. Perspectiva bayesiana de la ciencia.
- Inferencia, inducción, abducción.



2. La investigación científica.

- La investigación: definición, características, formas y tipos.
- Método científico tradicional: elementos, etapas, características.
- La ética en la investigación científica.
- Investigación empírica en ciencias de la computación.
 - i. El proceso de la investigación empírica.
 - ii. Tipos de investigación empírica: métricas, surveys, estudios de casos, cuasi-experimentos, experimentos y simulación.
- Investigación científica como empresa cooperativa.
 - i. Gestión de proyectos de investigación.
- Formulación de proyectos: formulación, preguntas de la investigación y justificación de la investigación.
 - ii. Definición del problema, hipótesis de trabajo, objetivos, metodología y plan de trabajo, divulgación de los resultados.
 - iii. Elaboración del marco teórico: revisión de la literatura, tipos de fuentes bibliográficas y recursos para la búsqueda.
 - iv. Experimentación: unidad experimental, factor, tratamientos, bloqueos, variables dependientes, variables independientes, variables de bloqueo, sujetos, hipótesis, validación y replicación.

3. Experimentos computacionales.

- Principios del diseño experimental.
- Experimentos en informática: diseño, ejecución, interpretación, reporte.
- Análisis de datos.
 - i. Análisis estadístico de datos.
 - ii. Análisis descriptivo.
 - iii. Hipótesis estadística.
 - iv. Análisis paramétrico y no paramétrico.
 - v. Métodos de reconocimiento de patrones.
- Relación entre hipótesis y evidencia: el problema de la validación de la hipótesis.
- Diseño de experimentos en subdisciplinas de la Informática
 - i. Algoritmos Probabilísticos.
 - ii. Ingeniería de Software.
- Investigación reproducible



4. Elaboración de informes científicos.

- Resultados intermedios del proceso científico: reporte técnico, artículo (workshops, conferencia, revista), tesis.
- Desarrollo de la argumentación.
- Calidad de un documento: claridad, precisión, fluidez, objetividad, comunicación eficiente.

Metodología:

El Curso se desarrollará con clases expositivas de 90 minutos. Durante el desarrollo del curso se entregará lecturas relevantes de artículos para complementar el aprendizaje de los alumnos. Éstos también deberán elaborar un proyecto de investigación y un artículo en algún tópico de su interés. El examen final del curso será basado en una presentación oral y un reporte escrito de su tema de investigación.

Evaluación:

- | | |
|---|-----|
| • Lectura y análisis de artículos de investigación. | 25% |
| • Formulación de un proyecto de investigación. | 35% |
| • Escritura de un artículo científico. | 40% |

Bibliografía:

- [1] T. Bartz-Beielstein "Experimental Research in Evolutionary Computation" Ed. Springer 2010.
- [2] S. Bem, H. L. de Jong: *Theoretical Issues in Psychology: An Introduction (2nd ed.)*. Sage Publications (2006).
- [3] Kai-Tai Fang, Runze Li and A. Sudjianto, T. "Design and Modeling for Computer Experiment" Ed. Chapman Hall/CRC Springer 2006.
- [4] N. Juristo, A. M. Moreno: *Basics of Software Engineering Experimentation*. Kluwe Academic Publishers (2001).
- [5] E. M. Phillips, D. S. Pugh: *How to get a PhD, a handbook for students and their supervisors*. Open University Press (2010).
- [6] A. F. Chalmers, "What is this thing called Science" University of Queensland Press, revised edition (1999).
- [7] P. P. Balestrassi, E. Popova, A. P. Paiva, J. W. Marangon Lima: *Design of experiments on neural network's training for nonlinear time series forecasting*.
- [8] K. Hinkelmann, O. Kempthorne: *Design and Analysis of Experiments, Volume I: Introduction to Experimental Design*. John Wiley & Sons (1994).



Elaborado:	Héctor Allende; Hernán Astudillo; Alejandro Frery	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	11/06/2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Simulación mediante Mallas Geométricas	Sigla	INF-568
Créditos	5	Conocimientos previos: MAT-023	

Descripción

Esta asignatura forma parte de los cursos electivos de los programas de posgrado científico del Departamento de Informática y forma parte del área de conocimientos Ingeniería Aplicada. Esta asignatura está orientada a alumnos de Magíster y Doctorado en Ciencias de la Ingeniería Informática. En esta asignatura los estudiantes aprenderán como un conjunto de Ecuaciones Diferenciales Parciales (EDPs), que representan un comportamiento físico, se simula mediante Elementos, Volúmenes o Diferencias Finitas en un computador. El énfasis de la asignatura está en la generación de mallas geométricas, sobre todo, para el método de Elementos Finitos. De igual forma se entregan conocimientos básicos para los otros métodos numéricos mencionados

Objetivos

Entender cómo se modela un objeto en un Computador.

- Conocer las diferencias entre los distintos métodos de aproximación numérica.
- Conocer para que sirve una malla geométrica.
- Programar las técnicas más conocidas

Contenidos

- 1.Representación discreta de un objeto continuo en un computador.
- 2.Modelando una interacción sobre una geometría simple.
- 3.Métodos de generación de mallas de Superficie y Volumen.
- 4.Definición de validez de un elemento.
- 5.Definición de calidad de un elemento.
- 6.Métodos de reparación y mejoramiento.

Metodología

- Clases expositivas, - Aprendizaje basado en la comprensión e implementación de algoritmos - Aprendizaje basado en estudio de autores - Estudio independiente y exposiciones de estudiantes acerca de temas - específicos - Al finalizar el semestre, los estudiantes deberán presentar un proyecto de fin de curso en el cual deberán leer, comprender, programar y exponer una publicación científica aprobada por el profesor.



Evaluación

Nota final = $NC \cdot 0.6 + NP \cdot 0.4$ En donde, NC es el promedio de dos certámenes y NP, es la nota del proyecto. Tanto NC como NP deben ser aprobados por separado.

Bibliografía

- [1] Frey, P. & George, P. Mesh Generation: Applications to Finite Elements Hermes, Paris, 2000.
- [2] Oden J., Tinsley R. & Junuthula N. An introduction to the mathematical theory of finite elements, New York : John Wiley, 1976.

Revistas

- [1] Ito, Y., Shih, A., Soni, B.: Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. International Journal for Numerical Methods in Engineering 77(13), 1809–1833 (2009).
- [2] Bucki M., **Lobos C.**, Payan Y., *A Fast and Robust Patient Specific Finite Element Mesh Registration Technique: Application to 60 Clinical Cases*. Medical Image Analysis 14(3), 303–317 (2010).

Elaborado:	Claudio Lobos	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	02/07/2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Métodos Cuantitativos en el Procesamiento Computacional de Imágenes (LSC)	Sigla	INF-569
Créditos	5	Conocimientos previos: ILI-286 Computación Científica 2 + autorización del Profesor	

Descripción

Asignatura centrada en los métodos matemáticos y computacionales aplicados al procesamiento de señales y, especialmente, imágenes. La asignatura introduce los elementos computacionales básicos del procesamiento de señales e imágenes, tales como los modelos vectoriales, matriciales y de campos tensoriales, las técnicas de filtrado y convolución, ciertos tipos de transformadas usuales en este contexto (Fourier, Hilbert, wavelets), y aplica estos formalismos a la solución de problemas de visualización, compresión, y procesamiento de señales e imágenes en general

Objetivos

- Identificar problemas en el ámbito del procesamiento de señales e imágenes, susceptibles de ser representados adecuadamente mediante modelos computacionales con base en campos vectoriales, matriciales y tensoriales.
- Construir modelos computacionales, como los mencionados, para abordar problemas y situaciones en el procesamiento de señales e imágenes donde ello es pertinente.
- Seleccionar algoritmos, o eventualmente diseñar unos nuevos, que permitan resolver problemas en el ámbito del procesamiento de señales e imágenes mediante técnicas computacionales con base en el análisis lineal.
- Contribuir a mejorar las competencias de los estudiantes en el ámbito del modelado y la resolución de problemas del procesamiento de señales e imágenes y, de este modo, contribuir también a incrementar la experiencia de los estudiantes en la redacción de documentos científicos

Contenidos



1. Elementos del análisis tensorial.
2. Campos tensoriales y procesamiento de imágenes.
3. Filtros y convolución.
4. Reconstrucción discreta de imágenes.
5. Técnicas algebraicas de reconstrucción de imágenes.
6. La transformada de Radon.
7. Imágenes obtenidas mediante resonancia magnética.

Metodología

1. Clases expositivas con apoyo de medios visuales.
2. Exposiciones frecuentes de los participantes.
3. Aprendizaje con base en el desarrollo de tareas y de laboratorio.
4. Resolución de problemas y casos.
5. Discusiones

Evaluación

Seminarios y exposiciones de los estudiantes

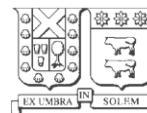
Desarrollo de artículos científicos

Calificación: Nota final = Promedio de todas las notas obtenidas por cada alumno en la asignatura.

Bibliografía

- [1] K. Bredies, D. Lorenz, *Mathematische Bildverarbeitung*. Vieweg + Teubner, Wiesbaden, 2011.
- [2] G. Aubert, P. Kornprobst. *Mathematical problems in image processing*. Springer Science, New York, 2010.
- [3] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmaier, F. Lenzen. *Variational methods in imaging*. Springer Science, New York, 2010.
- [4] T.G. Freeman, *The mathematics of medical imaging*. Springer, New York, 2010.
- [5] A. Mitiche, I.B. Ayed. *Variational and level set methods in image segmentation*. Springer, Berlin-Heidelberg, 2010.
- [6] G. Dougherty, *Digital Image Processing for Medical Applications*. Cambridge U. Press, New York, 2009.
- [7] G.T. Herman. *Fundamentals of computerized tomography*. Springer, London, 2009.

Elaborado:	Luis Salinas	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	20-08-2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Programación con Restricciones		Sigla	INF-571
Créditos	5	Conocimientos previos:	Optimización Combinatoria	

Descripción

El curso presenta los conceptos fundamentales de la programación con restricciones. Se estudia los problemas clásicos donde el modelamiento basado en la satisfacción de restricciones es aplicado. Se estudia los fundamentos de los métodos de resolución de problemas de satisfacción de restricciones. Los conceptos teóricos son complementados con el uso de herramientas computacionales para el modelamiento y la resolución de problemas de satisfacción de restricciones.

Objetivos

- Formular modelos basados en la satisfacción de restricciones y en la optimización con satisfacción de restricciones.
- Conocer las técnicas de resolución de problemas de satisfacción de restricciones.
- Conocer lenguajes para la programación con restricciones.

Contenidos

1. Problemas de satisfacción de restricciones (CSP): conceptos básicos, modelamiento de problemas basado en la satisfacción de restricciones
2. Técnicas de resolución de CSPs: enumeración exhaustiva, consistencia local, técnicas híbridas.
3. Problemas de optimización con satisfacción de restricciones (CSOP): conceptos básicos, modelamiento de problemas basado en la optimización con satisfacción de restricciones.
4. Técnicas de ramificación y acotamiento para la resolución de CSOPs: técnicas basadas en backtracking, técnicas basadas en la ramificación y acotamiento.
5. Lenguajes para la programación con restricciones: modelamiento y resolución de problemas industriales utilizando los diversos lenguajes.



Metodología

La asignatura contempla la realización de clases de exposición de los elementos teóricos, la realización de tareas individuales de aplicación de los conceptos vistos en clases y el estudio de casos y técnicas propuestas en artículos que presentan el desarrollo actual de la materia.

Evaluación

La asignatura se evalúa en base a certámenes (60%), análisis de artículos (20%) y tareas (20%).

Bibliografía

- [1] Principles of Constraint Programming, Krzysztof Apt, Cambridge University Press, 2009.
- [2] Handbook of Constraint Programming, Francesca Rossi, Peter van Beek & Toby Walsh, Elsevier, 2006.
- [3] Constraint Processing, Rina Dechter, Morgan Kaufmann Publishers, 2003.

Elaborado:	Carlos Castro	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	1/8/2003	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Computación Evolutiva		Sigla	INF-572
Créditos	5	Conocimientos previos: Inteligencia Artificial, programación en C (C++) para la implementación de las diversas estrategias.		

Descripción

En este curso se profundizan los conceptos de algoritmos de búsqueda local, algoritmos genéticos, algoritmos evolutivos, meméticos. Se hace énfasis en las aplicaciones y en los problemas de optimización combinatoria con restricciones.

Objetivos

- Presentar al estudiante las técnicas basadas en la evolución como alternativas en la resolución de problemas de optimización y de satisfacción de restricciones.
- Presentar al estudiante los requerimientos formales de la aplicación de estas técnicas, tanto para su diseño e implementación, así como para su evaluación.
- Mostrar al alumno el uso de la computación evolutiva en la resolución de problemas del mundo real.

Contenidos

1. Optimización Combinatoria: historia, problemas clásicos de optimización combinatoria con satisfacción de restricciones (CSOP), algoritmos para resolver problemas de satisfacción de restricciones (CSP), búsqueda del óptimo (técnicas basadas en Hill Climbing, Greedy Search, Tabu Search, Simulated Annealing).
2. Algoritmos Genéticos (AG): historia, y componentes de un AG (criterios para definir funciones de evaluación, criterios para definir operadores, noción de auto-adaptabilidad y tuning de parámetros). Algoritmos Híbridos, Co-evolución
3. Estrategias Evolutivas, el método de Schwefel
4. Programación Evolutiva, el método de Fogel & Fogel.
5. Swarm Intelligence
6. Investigaciones relacionadas en Computación Emergente
7. Conclusiones, investigación actual.



Metodología

Clases expositivas, seminarios, reuniones y presentaciones de avances del proyecto

Evaluación


El ramo implica la realización de un proyecto, con dos informes (20% y 30%), una presentación y una defensa (50%).

Bibliografía

- [1] A Field Guide to Genetic Programming, R. Poli, W.B.Langdon, and N. Freitag, 2008
- [2] Introduction to Evolutionary Computing, A. Eiben and J.E. Smith, Springer, 2010
- [3] Bio-Inspired Artificial Intelligence, D. Floreano and C. Mattiussi, MIT Press, 2008
- [4] Emergent Computing Methods in Engineering Design, D.E. Grierson and P.Hajela, 2010.
- [5] Handbook of evolutionary Computation", T Bäck, Z. Michalewicz and D. Fogel, Oxford University Press, 1997.
- [6] How to Solve It: Modern Heuristics, Z. Michalewicz and D. Fogel, Springer Ed., 1999
- [7] Swarm Intelligence, R. Eberhart, Y. Shui, J. Kennedy, Morgan & Kuffmann, 2001.
- [8] Artículos en IEEE Transactions on Evolutionary Computation, Journal of Heuristics, Applied Soft Computing Complex Systems. Y Conferencias GECCO, IEEE CEC, PPSN.

Elaborado:	María Cristina Riff	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Lógica Borrosa	Sigla	INF-575
Créditos	5	Conocimientos previos: Lenguajes de programación, Cálculo Numérico, Estadística computacional, Álgebra	

Descripción

En este curso se dan los fundamentos formales de la Lógica Borrosa, a partir del contexto de la lógica matemática clásica, particularmente el cálculo de predicados. Los (sub)conjuntos borrosos emergen vinculados al lenguaje natural con una intencionalidad semántica: por una parte representan el uso del lenguaje (en un contexto dado) y por otra, permiten el manejo formal de conocimientos vagos. Se discutirán distintos operadores para el cálculo con predicados borrosos, se presentarán teoremas de caracterización y mecanismos de generalización. El curso terminará con el análisis de inferencias en lógica borrosa.

Objetivos

Proporcionar un marco teórico formal para poder trabajar con lógica borrosa y poder desarrollar aplicaciones en p.ej. control automático borroso o sistemas expertos

Contenidos

1. Introducción; tipos de predicados
2. Funciones de E en $[0,1]$
3. t-normas, t-conormas y negaciones
4. Conjuntos borrosos
5. Consistencia lógica e inconsistencia en F (E)
6. Modificadores lingüísticos
7. Medidas de posibilidad
8. Funciones de Agregación
9. Principios de Extensión
10. Cortes alfa y números borrosos
11. Implicaciones borrosas
12. Regla Composicional de Inferencia
13. Cuantificadores borrosos



Metodología

Clases expositivas apoyadas con presentaciones en Powerpoint. Discusiones en grupo.

Evaluación

Tareas (25%); Trabajo final (75%)

Bibliografía

- [1] Barro S., Bugarín A., Moraga C., Trillas E.: *Computación fuzzy*. En: "Fronteras de la Computación" (S. Barro, A. Bugarín, Eds.) Fundación Dintel – Díaz Santos, Madrid, ISBN 84-7978-517-9, 2002
- [2] Klir G.J., St.Clair U.H., Yuan B.: *Fuzzy Set Theory*, Prentice Hall, 1997
- [3] Mesiar R.: *Triangular Norms: an Overview*, In: *Computational Intelligence in Theory and Practice* (B. Reusch, K.-H. Temme Eds.) Physica Verlag, Heidelberg, 2001
- [4] Moraga C., Trillas E., Guadarrama S.: Multiple-valued Logic and Artificial Intelligence. Fundamentals of Fuzzy Control Revisited. *Journal of Artificial Intelligence*, **20** (3-4) 169-197, 2003 (*)
- [5] Moraga C.: Introduction to Fuzzy Logic. *Facta Universitatis, Serie E.E.*, (University of Niš, Serbia), **18** (2), 2005 (*)
- [6] Nguyen H.T., Walker E.A.: *A first Course in Fuzzy Logic*. CRC Press, Boca Raton FLA, 1997
- [7] Tanaka K.: *An Introduction to Fuzzy Logic for Practical Applications*, Springer, Berlin, 1997
- [8] Trillas E., Cubillo S.: *Lecciones de Lógica Borrosa*. Facultad de Informática, Universidad Politécnica de Madrid, 1998

Elaborado:	C. Moraga	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	16-03-2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Máquinas del Aprendizaje Computacional	Sigla	INF- 578
Créditos	5	Conocimientos previos: ILI-280; MII-475	

Descripción

En este curso se presenta un tratamiento sistemático de los fundamentos del aprendizaje estadístico inductivo, deductivo y transductivo, abordando temas como el riesgo de generalización, minimización del riesgo estructural y los dilemas sesgo-varianza y estabilidad-flexibilidad. También se estudian diversas metodologías y técnicas para el diseño y construcción de algoritmos de aprendizaje supervisado, tales como las Redes Neuronales Artificiales (ANN), las Máquinas de Vectores de Soporte (SVM), y Algoritmos de Aprendizaje No-Supervisado tales como las Redes de Kohonen o Mapas Auto-Organizativos (SOM) y el Algoritmo Vecino más Cercano (NN) con sus variantes más relevantes, incluyendo también el razonamiento aproximado basado en diversos métodos de agregación de máquinas (*ensembles*). Todo lo anterior inmerso en el contexto de problemas de reconocimiento de formas, tales como clasificación, regresión, agrupamiento y pronóstico. Además se discuten algunos temas de reciente interés tales como el aprendizaje incremental, el aprendizaje distribuido, y la búsqueda de relaciones entre el aprendizaje de máquinas y el aprendizaje humano.

Objetivos:

Al finalizar el curso el alumno estará capacitado para:

- Conocer los fundamentos estadísticos y computacionales de las Máquinas de Aprendizaje.
- Diseñar y aplicar Máquinas de Aprendizaje a problemas de reconocimiento de formas: Clasificación, Asociación, Pronóstico, entre otras.
- Conocer y aplicar diversos algoritmos de Máquinas de Aprendizaje.
- Conocer los principales avances de las Artificial Neural Networks (ANN), para los problemas de clasificación y reconocimiento de patrones.

Contenidos:

1. Conceptos estadísticos en el aprendizaje: problemas de Inducción e Inferencia Estadística. Modelos de inferencia Paramétrica y no Paramétrica. Generalización de la teoría de Glivenko – Cantelli –



Kolmogorov.

2. Teoría de aprendizaje y generalización: el Problema de minimización del riesgo en datos empíricos y el dilema Sesgo-Varianza. El Problema de reconocimiento de formas. El problema de estimación en regresión generalizada. El Problema de estimación de la densidad, Teorema de Glivenko y Cantelli, tipos de convergencia, Ley de Kolmogorov – Smirnov y Ley de los Algoritmos Iterados.
3. Estimación de la densidad de probabilidad y el problema de aprendizaje: convergencia y condiciones de convergencia para medidas de probabilidades desconocidas.
4. Riesgo empírico y principios de minimización: Teorema de Key de la Teoría de Aprendizaje. Cotas y funciones de pérdidas.
5. El Perceptrón y sus generalizaciones: El Perceptrón de Rosenblatt. El método estocástico de aproximación.
6. Elementos de teoría de optimización teorema de Fermat: la regla de multiplicadores de Lagrange y la teoría de Kühn – Tucker.
7. Redes Neuronales en regresión generalizada.
8. La Máquina de Vector Soporte (SVM, Support Vector Machine): aproximación por hiperplanos, las propiedades estadísticas de los hiperplanos óptimos, generalización a espacios de alta dimensión. Selección de SVM. SVM en reconocimiento de formas. SVM en problemas de clasificación múltiple. Aplicaciones.

Metodología de Trabajo:

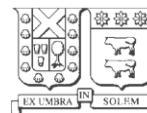
El Curso se desarrollará con Clases expositivas de 90 minutos de exposición. El alumno deberá rendir interrogaciones y tareas individuales para complementar su aprendizaje. Durante el desarrollo del curso el alumno deberá seleccionar lecturas relevantes de artículos y elaborar un proyecto de investigación en algún tópico de su elección en forma de seminario y discusión del estado del arte. El examen final del curso estará basado en una presentación oral y un reporte escrito de su tema de investigación.

Evaluación:

Tareas e interrogaciones	30%
Proyecto	40%
Examen	30%

Bibliografía:

- [1] Vladimir N. Vapnik. Statistical Learning Theory. Wiley, New York, 1998.
- [2] B. Schölkopf, A.J. Smola. Learning with Kernels: Support Vector



Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning). MIT Press, 2001.


- [3] E. Alpaydin, Introduction to Machine Learning, Second edition. The MIT Press, 2009.
- [4] T. M. Mitchell, Machine Learning, 1.^a ed. McGraw-Hill Science/Engineering/Math, 1997.
- [5] C. M. Bishop, Pattern Recognition and Machine Learning, 1st ed. 2006. Corr. 2nd printing 2011. Springer, 2007.
- [6] B. D. Ripley, "Pattern recognition and Neural Network", .Ed. Cambridge University Press, 1996.
- [7] T.L. Fine, "Feedforward Neural Network Methodology", Ed. Springer Verlag, 1999.
- [8] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification, Wiley, 2001.
- [9] **Revistas:** Journal of Machine Learning, IEEE Trans. on Neuronal Networks, Neuronal computing, Pattern Recognition, Journal of Intelligent Data Analysis, entre otras.

Material complementario en línea:

- [10] T. Poggio, L. Rossaco, C. Fronger, y G. Canas, «Statistical Learning Theory and Applications, Spring 2012». [Online]. Available: <http://www.mit.edu/~9.520/>. [Accessed: 27-dic-2012].
- [11] A. Ng, «Machine Learning | Coursera». [Online]. Available: <https://www.coursera.org/course/ml>. [Accessed: 27-dic-2012].
- [12] T. Jaakkola, «Machine Learning MIT OpenCourseware», MIT OpenCourseWare. [Online]. Available: <http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-867-machine-learning-fall-2006/>. [Accessed: 27-dic-2012].
- [13] P. Domingos, «Machine Learning | Coursera». [Online]. Available: <https://www.coursera.org/course/machlearning>. [Accessed: 27-dic-2012].

Elaborado:	Héctor Allende	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01-08-2006	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Agentes Inteligentes y Autónomos		Sigla	INF-579
Créditos	5	Conocimientos previos: redes neuronales artificiales, algoritmos genéticos, reconocimiento de patrones, análisis y diseño orientado al objeto		

Descripción

Este curso presenta una visión de los distintos tipos de agentes computacionales y sus aplicaciones a los problemas de análisis de sistemas complejos, planificación y resolución de problemas en forma distribuida, algoritmos de control distribuidos y programación concurrente de aplicaciones de inteligencia artificial. Se analizan agentes inteligentes y autónomos y su incorporación a sistemas multiagentes distribuidos en diversos ámbitos de la computación, exponiendo las diferentes arquitecturas y protocolos de interacción existentes.

Objetivos

Al término del curso el alumno deberá ser capaz de:

- Conocer los conceptos fundamentales del ámbito de agentes inteligentes y autónomos.
- Conocer y aplicar sistemas multiagentes para resolver varios problemas en análisis de sistemas complejos, planificación distribuida y programación concurrente.
- Seleccionar el modelo, arquitectura y protocolo de interacción más adecuado para la resolución de un problema dado.
- Especificar y diseñar la solución basado en algunos formalismos.

Contenidos :

1. Introducción a los agentes inteligentes
 - i. Agentes inteligentes.
 - ii. Agentes autónomos.
 - iii. Sistemas multiagentes.
 - iv. Arquitecturas abstractas para agentes.
 - v. Arquitecturas concretas para agentes.



2. Agentes con razonamiento deductivo
 - i. Agentes deliberantes.
 - ii. Programación orientada a agentes.
 - iii. MetaM Concurrente.
3. Agentes con razonamiento práctico
 - i. Razonamiento práctico.
 - ii. Razonamiento objetivo-medio.
4. El sistema de razonamiento procedimental.
Agentes reactivos e híbridos
 - i. Arquitectura de subsumisión.
 - ii. Limitaciones de agentes reactivos.
 - iii. Agentes híbridos.
5. Sistemas multiagentes y sociedades de agentes
 - i. Comunicación entre agentes.
 - ii. Protocolos de interacción entre agentes.
 - iii. Sociedades de agentes.
 - iv. Arquitecturas para sistemas multiagentes.
6. Análisis de sistemas complejos usando agentes
 - i. Sistemas CS, CAS, MAS.
 - ii. Propiedades emergentes, sistemas autoorganizados.
 - iii. Organización entre agentes.
7. Especificación y Diseño usando LORA
 - i. Modelo creencia-deseo-intención.
 - ii. Estados mentales colectivos.
 - iii. Comunicación y cooperación.
 - iv. Arquitecturas.
8. Aplicaciones de sistemas multiagentes
 - i. Planificación distribuida.
 - ii. Control en tiempo real.
 - iii. Trabajo Cooperativo soportado por computador y de grupo (CSCW y Groupware).
 - iv. Modelos distribuidos para toma de decisiones.



Metodología

Clases expositivas, investigaciones individuales, presentación de trabajos en forma de seminario.

Evaluación

Presentación de trabajos (30%), monografía final (70%).

Bibliografía:

- [1] Fundamentals of Multiagent Systems, José M. Vidal. 2010. Disponible en <http://multiagent.com/2010/02/multiagent-systems-textbook.html>.
- [2] An Introduction to MultiAgent Systems, Michael Wooldridge. John Wiley & Sons, 2nd Ed, 2009. (ISBN 0470519460).
- [3] Agents and Artificial Intelligence, Third International Conference, ICAART 2011, Rome, Italy, January 28-30, 2011. Revised Selected Papers, Filipe, Joaquim; Fred, Ana (Eds.), : Communications in Computer and Information Science, Vol. 271, Springer-Verlag, ISBN 978-3-642-29965-0, 2013.
- [4] Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Yoav Shoham and Kevin Leyton-Brown. Cambridge University Press, 2009.
- [5] Agent-Directed Simulation and Systems Engineering, Levent Yilmaz (Editor), Tuncer Ören, Eds. Wiley Series in Systems Engineering and Management, 2009. (ISBN 3527407812)
- [6] Developing Multi-Agent Systems with JADE, Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. Wiley Series in Agent Technology, 2007.
- [7] Revistas : papers representativos del estado del arte publicados en Journal of Intelligent Information Systems, Decision Support Systems, IEEE Transactions on Robotics and Automation, Autonomous Agents and Multi-Agent Systems, Computational and Mathematical Organization Theory, Journal of Artificial Societies and Social Simulation (JASSS).

Elaborado:	Enrique Canessa	Observaciones: Este curso está enfocado a estudiantes de Doctorado y se basa en un alto compromiso de los estudiantes por aportar con su trabajo al curso y participar activamente en las clases. Por ello, el curso toma la forma de un seminario de postgrado al igual que la evaluación de cada alumno que se realizará. Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	Junio 2004	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Sistemas Complejos Discretos	Sigla	INF-584
Créditos	5	Conocimientos previos: Teoría de grafos, lenguajes formales, autómatas	

Descripción

Este curso presenta una panorámica de los sistemas complejos discretos, desde los sistemas más homogéneos (autómatas celulares), donde es posible obtener más resultados teóricos exactos, hasta las redes heterogéneas y masivas, con las que debemos interactuar sin jamás llegar a conocerlas por completo; en el camino se estudian sistemas dinámicos definidos sobre grafos pequeños (en particular, redes booleanas), y se ilustran repetidamente los fenómenos emergentes y la dificultad de análisis que caracterizan a los sistemas complejos, ya desde sus instancias supuestamente más simples. Se pone énfasis en las aplicaciones al campo bioinformático (pasando revista también a otras áreas de aplicación), aunque las herramientas algorítmicas y la perspectiva amplia sobre el tipo de fenómenos son aplicables en una gran cantidad de áreas.

Objetivos

Al aprobar la asignatura el alumno tendrá una perspectiva teórica sólida y amplia de los sistemas discretos masivos, presentes hoy en día en casi toda área del quehacer; estará familiarizado con los fenómenos dinámicos y problemas analíticos que aparecen como consecuencia de la complejidad de las interacciones y naturaleza de los agentes que interactúan, y conocerá las formas actuales de abordar los principales problemas que se presentan. Habrá trabajado con algoritmos y habrá debido enfrentar desafíos teóricos y prácticos en sistemas con diversos niveles de complejidad estructural y dinámica. Podrá conectar estos conocimientos con diversos sistemas de importancia social, científica y tecnológica, y en prácticamente todos los temas habrá conocido investigación de punta (y muchas preguntas aún por contestar).

Contenidos

1. Autómatas celulares. Ejemplos. Estudio teórico. Subshifts y lenguajes formales asociados. Universalidad, reversibilidad, problemas duros y problemas indecidibles. Clasificaciones fenomenológicas y analíticas. Complejidad comunicacional, flujo de información, física emergente del procesamiento de información. Relación con modelos continuos y con modelos basados en agentes; autómatas conservativos y modelos de tráfico. Usos: modelos físicos, biológicos, sociales, epidemiológicos; procesamiento de imágenes, criptografía, protocolos de sincronización. Autómatas celulares



estocásticos. Autómatas cuánticos (QCA). Fenómenos emergentes; estructuras replicativas. Pilas de arena, criticalidad auto-organizada.

2. Conceptos generales de sistemas complejos discretos. Nociones de sistemas dinámicos, atractores, sensibilidad, no-linearidad, dinámicas caóticas. Fenómenos emergentes, nociones de complejidad.
3. Redes de autómatas: dinámicas sobre grafos de tamaño moderado, con énfasis en redes booleanas. Redes neuronales discretas, circuitos lógicos; modos de iteración. Convergencia, funcionales de Lyapunov. Robustez dinámica frente a ruido y cambios en modos de iteración. Dependencia de dinámica respecto a la topología: modelos de evolución y cooperación en grafos; juegos iterativos. Problemas algorítmicos asociados a redes booleanas; estudio especial de la aplicación a redes de regulación genética. El problema inverso de determinación de redes a partir de información parcial. Redes booleanas aleatorias (RBN) como modelos de dinámicas biológicas.
4. Modelos de grafos aleatorios para sistemas de gran tamaño. Modelos clásicos (Erdős-Renyi). Estadísticas sobre redes: conexidad, patrones locales, interacción de características estructurales locales y globales.
5. Redes complejas. Topologías principales: scale-free, small worlds. Propiedades: autosimilaridad, clusterización, diámetro pequeño. Algoritmos de construcción de redes; mecanismos generativos espontáneos. Inferencia de propiedades estructurales en grandes redes; ley de potencia de grados. Navegabilidad y búsquedas descentralizadas en redes complejas. Representación gráfica. Detección de clusters y comunidades. Minería de datos de redes. Análisis de roles, análisis de enlaces. Robustez estructural ante fallas o ataques; mecanismos de evolución estructural. Sistemas dinámicos en redes complejas: autoorganización, fenómenos de contagio, estrategias de control dinámico y estructural. Estabilidad dinámica, acoplamiento de oscilaciones, correlaciones y flujo de información en redes. La aplicación de las ideas será con énfasis en redes complejas biológicas (nerviosas, metabólicas, regulatorias, interacciones de proteínas) pero con atención a las particularidades de redes complejas sociales, semánticas y tecnológicas (como la Web).

Metodología

El curso se desarrollará en clases expositivas de 90 minutos de duración. El alumno deberá rendir interrogaciones y elaborar tareas en forma individual para complementar su aprendizaje. Adicionalmente, se asignarán lecturas relevantes a los contenidos del curso. Durante el desarrollo del curso cada alumno deberá elaborar un proyecto de investigación en el tópico de su elección. Los alumnos deberán presentar una propuesta y estados de avance de este proyecto. El examen final del curso será basado en una prueba escrita y un reporte final escrito de su trabajo de investigación.



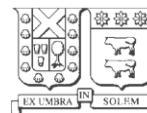
Evaluación

Tareas e interrogaciones	30%
Proyecto	30%
Examen	40%

Bibliografía

- [1] Boccara, N. Modeling Complex Systems. Springer Verlag, 2004.
- [2] Wolfram, S., Ed. Theory and applications of cellular automata. World Scientific Publication, 1986.
- [3] Brandes, U., Erlebach, T., Eds. Network Analysis: Methodological Foundations (LNCS 3418), Springer-Verlag, 2005.
- [4] Bornholdt, S., Ed. Handbook of Graphs and Networks: From the Genome to the Internet. Wiley-VCH, 2003.
- [5] Junker, B., Schreiber, F., Eds. Analysis of Biological Networks. John Wiley & Sons, 2008.

Elaborado:	A. Moreira	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	Julio 2008	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Bases de Datos Documentales	Sigla	INF-585
Créditos	5	Conocimientos previos:	

Descripción

La asignatura se dicta a estudiantes de postgrado y aborda los principales modelos de recuperación de información en bases de datos documentales, las estructuras de datos utilizadas con mayor frecuencia y los conceptos de indexamiento de documentos. También cubre los temas de investigación de mayor actualidad en el área, con foco en los métodos del estado del arte y sus aplicaciones prácticas.

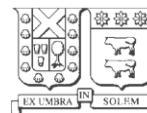
Objetivos:

Al finalizar el curso el alumno estará capacitado para:

- Conocer los conceptos fundamentales de recuperación de información
- Conocer los modelos de recuperación de información más usados en bases de datos documentales
- Conocer las estructuras de datos usadas con mayor frecuencia en la implementación de sistemas de bases de datos documentales
- Conocer las nociones fundamentales para búsqueda e indexamiento de documentos en la web.
- Conocer los principales avances y temas de investigación en recuperación de información.

Contenidos:

1. Propiedades y operaciones en texto: crecimiento, dispersión, distribuciones, metadatos v/s texto, formatos, nociones de teoría de información, preprocesamiento de texto, stemming, tesauros.
2. Modelos de recuperación de información: modelo booleano, vectorial, probabilístico, fuzzy, booleano extendido, vectorial generalizado, LSI, ANN, Bayesiano.
3. Evaluación de modelos de recuperación de información - métricas y metodologías: precisión, cubrimiento, F-medida.
4. Estructuras de datos para texto: archivos invertidos, árboles de sufijos, arreglos de sufijos, archivos de firma.
5. Recuperación de información en la web: estructura de buscadores (crawlers, índices, ranking en la Web). Uso de hyperlinks, texto, logs: algoritmos HITS, PageRank, Bayesiano, clustering de consultas, web



mining.

6. Temas avanzados: motores de búsqueda 3G, recuperación de información en música, recuperación de información en video, web semántica.

Metodología

El Curso se desarrollará con clases expositivas de 90 minutos de exposición. El alumno deberá rendir una evaluación escrita a mitad del semestre, realizar una presentación de un artículo de su interés sobre algún tema avanzado de investigación y desarrollar un mini-proyecto con aporte creativo. El examen final del curso será basado en una presentación oral y en un informe escrito de su mini-proyecto.

Evaluación:

Certamen	30%
Presentación	30%
Examen	40%

Bibliografía Obligatoria

- [1] Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley, 1999.

Bibliografía Complementaria

- [1] Witten, I., Moffat, A. y Bell, T. Managing Gigabytes, Morgan Kauffman, 1999 (segunda edición).
[2] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, Wiley, 2001
[3] Agosti, M. y Smeaton, A. (editores) Information Retrieval and Hypertext, Kluwer, 1996.
[4] Revistas: Journal of the American Society for Information Science and Technology, Journal of Information Retrieval, Pattern Recognition, Journal of Intelligent Data Analysis, etc.

Elaborado:	Marcelo Mendoza	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	23/04/08	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Procesamiento Digital de Imágenes usando R	Sigla	INF-586
Créditos	5	Conocimientos previos:	

Descripción

Este curso presenta los conceptos, notaciones y algoritmos básicos de procesamiento de imágenes digitales. La plataforma de trabajo es el sistema R, un lenguaje de programación matricial con excelentes propiedades numéricas y gráficas. R está disponible gratuitamente en <http://www.r-project.org>

Objetivos

Conocer los grandes problemas de procesamiento de imágenes digitales, desde la captura de los datos hasta su salida para uso en técnicas de análisis de imágenes y de visión computacional. Cada problema será visto con una notación consistente, con la presentación de los principales algoritmos y de su implementación en R.

Contenidos

1. Introducción a imágenes y a R: procesamiento de imágenes y la plataforma R
2. Imágenes: definiciones formales y notación
3. Formatos de imágenes y lectura de datos
4. Operaciones puntuales
5. Filtrado lineal
6. Filtrado no lineal
7. Problemas abiertos

Metodología

Clases expositivas y trabajos prácticos en laboratorio.

Evaluación

Proyectos con aspectos teóricos y prácticos



Bibliografía

- [1] Velho, L., Miranda, J., and Frery, A. C. (2008). *Image Processing for Computer Graphics and Vision*. Springer, London, 2 edition.
- [2] de Mello, C. A. B.; Oliveira, A. L. I. & dos Santos, W. P. (Eds.) *Digital Document Analysis and Processing*. Nova Publishers, 2012.
- [3] Jain, A. K. (1989). *Fundamentals of Digital Image Processing*. Prentice-Hall International Editions, Englewood Cliffs, NJ.
- [4] Spector, P. (2008). *Data Manipulation with R. Use R!* Springer.
- [5] Sarkar, D. (2008). *Lattice: Multivariate Data Visualization with R. Use R!* Springer.
- [6] Murrell, P. (2006). *R graphics*. Chapman & Hall.

Revistas

- [1] IEEE Transactions on Image Processing
- [2] IEEE Transactions on Geoscience and Remote Sensing
- [3] IEEE Geoscience and Remote Sensing Letters
- [4] Image and Vision Computing

Elaborado:	Alejandro Frery	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	20/04/2009	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Seminario de Especialidad I	Sigla	INF-581
Créditos	5	Conocimientos previos: Variable según tema de especialidad	

Descripción

En este curso se cubre un tema específico de Especialidad a definir por el profesor del programa en un semestre particular.

Objetivos

Profundizar en un tema específico de especialidad, explorando las fronteras del conocimiento.

Contenidos

Variable

Metodología

Presentación y discusión de trabajos relacionados con el tema de especialidad, en forma de seminario.

Evaluación

Variable.

Bibliografía

Variable

Elaborado:	Director de Tesis	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01/2002	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Seminario de Especialidad I: Bioinformática	Sigla	INF-581
Créditos	5	Conocimientos previos: Variable según tema de especialidad	

Descripción

En este curso se cubre un tema específico de Especialidad a definir por el profesor del programa en un semestre particular.

Objetivos

Profundizar en un tema específico de especialidad, explorando las fronteras del conocimiento.

Contenidos

Variable

Metodología

Presentación y discusión de trabajos relacionados con el tema de especialidad, en forma de seminario.

Evaluación

Variable.

Bibliografía

Variable

Elaborado:	A. Zamyatnin	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01/2010	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Seminario de Especialidad II	Sigla	DII-582
Créditos	5	Conocimientos previos: Variable según tema de especialidad	

Descripción

En este curso se cubre un tema específico de Especialidad a definir por el profesor del programa en un semestre particular.

Objetivos

Profundizar en un tema específico de especialidad, explorando las fronteras del conocimiento.

Contenidos

Variable

Metodología

Presentación y discusión de trabajos relacionados con el tema de especialidad, en forma de seminario.

Evaluación

Variable.

Bibliografía

Variable

Elaborado:	Director de Tesis	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01/2002	



UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
Escuela de Graduados / Departamento de Informática

Asignatura	Tema de Especialidad II : Computación Cuántica		Sigla	INF-582
Créditos	5	Conocimientos previos: Necesarios: Álgebra Lineal, Probabilidades y Estadística Deseables: Diseño de Circuitos Lógicos, Informática Teórica		

Descripción

El curso está destinado a estudiantes de Informática, sin conocimientos profundos de Física Teórica. El mundo cuántico será modelado como un espacio de Hilbert sobre los complejos. Operaciones elementales sobre vectores cuánticos serán realizadas mediante transformaciones unitarias y visualizadas mediante la Esfera de Bloch. La meta del curso es discutir el diseño de circuitos computacionales y algoritmos en este mundo "extraño" y aun poco conocido. No es meta del curso discutir algunos de los fascinantes y candentes problemas de la mecánica cuántica.

Objetivos :

Al término del curso el alumno deberá ser capaz de:

- Conocer los conceptos fundamentales de la Computación Cuántica.
- Conocer y aplicar un modelo basado en espacios de Hilbert para el estudio de Computación Cuántica
- Diseñar circuitos computacionales reversibles / cuánticos
- Especificar, analizar y diseñar algoritmos para un entorno de Computación Cuántica

Contenidos

- Por qué ocuparse de computación cuántica?
- Computación reversible
- El bit cuántico o „qubit“
- La notación de Dirac
- Estados básicos, superposición de estados, acoplamiento de estados
- Medición de observables
- La Esfera de Bloch



- Matrices de Pauli, transformación de Hadamard
- Modelo computacional cuántico
- Compuertas cuánticas básicas: NO, NO-controlado, Toffoli, Deutsch
- Compuertas universales, Arquitecturas del grupo Barenco y colegas.
- Como realizar un mundo discreto confiable en un universo continuo?
- Ejemplos computacionales
- Complejidad clásica y complejidad cuántica
- Algoritmos cuánticos, aspectos de diseño, análisis de los mas famosos
- (Si el tiempo lo permite) Corrección automática de errores
- (Si el tiempo lo permite) Lógica cuántica

Metodología:

El Curso se desarrollará con Clases expositivas de 90 minutos, con presentaciones powerpoint. El alumno deberá rendir tareas individuales para complementar su aprendizaje. Se contempla además un proyecto final, desarrollado individualmente. El proyecto final del curso será evaluado en base a un reportaje escrito sobre el tema de investigación acordado con el Profesor.

Evaluación:

Tareas	30%
Proyecto Final	70%

Bibliografía:

El curso comprende una selección de temas con una orientación similar a la del libro "*Quantum Computation and Quantum Information*," Cambridge University Press, 2000, de M.A. Nielsen e I.L. Chuang. Los participantes recibirán Apuntes de Clase especialmente elaborados para el curso, como también apuntes de clase utilizados en CalTech y en la Universidad de Indiana además de una colección de artículos que amplían los aspectos presentados en el curso (Habrà un CD con todo este material, que puede ser copiado por los participantes para su uso personal.)

Adicionalmente, los participantes tendrán acceso a trabajos presentados en los recientes Workshops Internacionales realizados en Europa, sobre Computación Reversible / Computación Cuántica



Elaborado:	Claudio Moraga	Observaciones: este curso está enfocado a estudiantes de Doctorado en Ingeniería, especialmente (pero no exclusivamente) para los estudiantes del Doctorado en Informática, y se basa en un alto compromiso de los estudiantes por aportar con su trabajo al curso y participar activamente en las clases. Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	02-08-2005	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Seminario de Tesis I	Sigla	INF-591
Créditos	34	Conocimientos previos: Examen de calificación aprobado	

Descripción

Seminario destinado a estudiar en la literatura especializada, los trabajos relacionados con el tema de tesis del alumno.

Objetivos

Profundizar en el tema específico de la tesis, explorando las fronteras del conocimiento y proponiendo métodos alternativos.

Contenidos

Variable

Metodología

Presentación y discusión de trabajos en forma de seminario y presentación en congresos de especialistas

Evaluación

Obtiene una nota al aprobar el tema de tesis en su examen de calificación.

Bibliografía

Variable

Elaborado:	Director de Tesis Doctoral	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01/2002	



	UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA Escuela de Graduados / Departamento de Informática
---	--

Asignatura	Seminario de Tesis II	Sigla	INF-592
Créditos	68	Conocimientos previos: Examen de calificación aprobado	

Descripción

Seminario destinado a estudiar en la literatura especializada, los trabajos relacionados con el tema de tesis del alumno.

Objetivos

Profundizar en el tema específico de la tesis, explorando las fronteras del conocimiento y proponiendo métodos alternativos.

Contenidos

Variable

Metodología

Presentación y discusión de trabajos en forma de seminario y presentación en congresos de especialistas

Evaluación

Obtiene una nota al finalizar su trabajo de tesis y entregar el escrito para revisión de la comisión de examen de doctoral

Bibliografía

Variable

Elaborado:	Director de Tesis Doctoral	Observaciones: Actualización: Diciembre 2012
Aprobado:	Depto. de Informática	
Fecha:	01/2002	