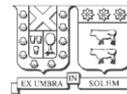




REGLAMENTO INTERNO DEL PROGRAMA:

“Magíster en Ciencias de la Ingeniería Informática”



Aprobado por CCDIP de fecha junio 2015.

HISTORIAL DE APROBACIONES DE CCDIP	
Fecha	Descripción
20/03/2014	Aprobación original de Arts. 1 a 43 y T1 y T2, y de Anexos N° 1 a 6.
18/06/2015	Actualización de Anexos N° 1, 2 y 4.

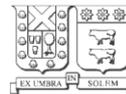
Dada la naturaleza del trabajo académico y en pos de un mejoramiento continuo, el presente reglamento será revisado y sancionado por el CCDIP anualmente. Si se registraren cambios esenciales, éstos aplicarán solamente a nuevas cohortes de estudiantes.

INTRODUCCIÓN

- Art. 1** El programa de **Magíster en Ciencias de la Ingeniería Informática** de la Universidad Técnica Federico Santa María (UTFSM), en adelante MII, fue creado con fecha 18 de Enero de 1992, por acuerdo de la Junta Directiva de la UTFSM en su sesión N° 20.
- Art. 2** El MII se desarrollará de acuerdo a las políticas de Postgrado de la UTFSM y se regirá por el Reglamento General N°47 de los Estudios de Postgrado (en adelante RGEP), y por el presente Reglamento.
- Art. 3** Las normas de este Reglamento complementan el RGEP y el Reglamento de Graduación para Grados de Doctor y Magíster en todas aquellas materias no contempladas en éstos, o en aquellas que expresamente se requiera una regulación más específica.

TÍTULO I: DISPOSICIONES GENERALES.

- Art.4** **Objetivos del Programa:** El objetivo del MII es formar especialistas con 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 de grado, de modo que, a través de dicha investigación, contribuya al desarrollo científico-tecnológico de la Ingeniería Informática.
- Art.5** **Áreas de especialización del Programa:** El MII ofrece la especialización en:
- a) Computación Científica e Inteligencia Computacional
 - b) Informática Teórica
 - c) Arquitectura e Ingeniería de Software
 - d) Sistemas de Computación y Bases de Datos



- Art. 6 Perfil del graduado:** El graduado del MII debe ser capaz de:
- a) Investigar y/o innovar en una línea de especialización de la Ingeniería Informática, estudiando el estado del arte asociado y proponer una solución que contribuya al cuerpo de conocimiento de la Ingeniería Informática.
 - b) Evaluar la calidad o el desempeño de la solución propuesta a un problema en estudio.
 - c) Resumir y difundir los resultados obtenidos durante la investigación y divulgarlos en revistas y/o conferencias internacionales de su especialidad.
- Art. 7** El MII consta de un total de 100 créditos SCT (Sistema de Créditos Transferibles) y tiene una duración normal de cuatro semestres académicos para un estudiante en un régimen de estudios de dedicación completa.
- Art. 8** El estudiante deberá tener una permanencia activa mínima en el MII equivalente a 60 SCT en la Institución (1 año) en régimen de jornada completa. La permanencia máxima no podrá exceder los 3 años.

TITULO II: ADMINISTRACIÓN DEL PROGRAMA.

- Art. 9** La tuición académica del programa MII dependerá, de manera exclusiva, del Departamento de Informática (en adelante DI) de la UTFSM.
- Art. 10** El MII será dirigido por el Comité de Programa del MII (en adelante CP), constituido por al menos 4 académicos del Cuerpo de Directores de Tesis del Programa MII (en adelante CDTP), definido en el Título III. El CP es presidido por uno de sus miembros, quien se desempeñará como el Director del MII.
- Art. 11.** Los miembros del CP, incluido quien ejerza el cargo de Director del MII, serán designados por el Consejo del DI a proposición del Director del DI (nómina del CP en Anexo N° 1).
- Art. 12** Le corresponde al CP, además de las funciones establecidas en el Art. 16 del RGEF, pronunciarse sobre las siguientes materias del MII:
- a) Programación de cursos y profesores encargados, sin perjuicio de las atribuciones del Consejo del DI.
 - b) Aprobación en primera instancia de programas de nuevas asignaturas o cualquier otra modificación del plan de estudios.
 - c) Actualización periódica del cuerpo de profesores y directores de tesis, de acuerdo a los criterios establecidos en este Reglamento, sin perjuicio de las atribuciones del Consejo del DI.
 - d) Aplicación de los mecanismos de evaluación del MII establecidos.
 - e) Participación en las actualizaciones de los planes de desarrollo del DI.
 - f) Exposición ante el cuerpo académico del MII de situaciones de conflicto académico o disciplinario que se presentaren, para una adecuada resolución.



- g) Supervisión del buen funcionamiento de la administración del MII y la calidad de éste, estableciendo políticas de mejora si fuere necesario.

Otras competencias o actos, de índole académico, necesarios para la buena marcha del MII, corresponden al Director del MII, quien además actúa como la autoridad ejecutiva del MII.

Art. 13 Las sesiones del CP son convocadas por el Director del MII. El CP sólo puede sesionar con la concurrencia de más de la mitad de sus miembros. En caso de que alguna materia deba decidirse por votación, se aplica el criterio de mayoría absoluta. De producirse un empate en la votación, el Director del MII es quien dirime. El CP deberá sesionar como mínimo dos veces en cada semestre académico, y sus acuerdos deberán quedar consignados en actas.

Art.14 El Director del MII designa entre los miembros del CP quien ha de subrogarlo en su ausencia.

TITULO III: DE LOS PROFESORES DEL PROGRAMA.

Art. 15 Podrán pertenecer al Cuerpo de Profesores estables del MII sólo académicos jornada completa del DI, que tengan el grado de Doctor y que pertenezcan preferentemente a las jerarquías de profesor adjunto o titular.

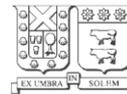
Art. 16 Excepcionalmente, y por acuerdo del CP, podrán ser parte del MII, académicos del DI que no cumplan con lo establecido en el Art. 15, quienes participan del MII en calidad de Profesores colaboradores. Además podrán participar del MII, y por acuerdo del CP, Profesores visitantes que tengan reconocida trayectoria en una línea de investigación relacionada con un tema específico de interés para el MII.

Art. 17 Los profesores del MII miembros del CDTP conforman el Claustro de Profesores del MII. Podrán pertenecer al CDTP académicos del Cuerpo de Profesores estables del MII que cumplan con las siguientes exigencias de productividad:

- a) Índice mínimo de productividad promedio anual equivalente a 0,8 en los últimos 5 años, con al menos 3 publicaciones en Journals ISI.
- b) Se considera para el cálculo del índice de productividad las publicaciones en Journals ISI (ponderación 1,0) y Proceedings indexados ISI o Scopus (ponderación 0,5).

Nómina de profesores del MII en Anexo N° 2.

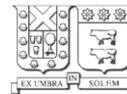
Art. 18 Excepcionalmente, y por acuerdo del CP, podrá ser co-director de tesis un académico que no pertenezca al CDTP, pero que tenga reconocida trayectoria en una línea de investigación relacionada con el tema de tesis específico. En tal caso el CP designará un Director de Tesis entre los miembros del CDTP, quien



tendrá la responsabilidad de supervisar que el trabajo se ajuste a los estándares exigidos por el Programa MII.

TITULO IV: DE LA ADMISION.

- Art. 19** Las postulaciones se canalizan a través de la Dirección de Postgrado (en adelante DP), y deberán seguir los procedimientos y formalidades establecidos en el RGEP. El requisito básico para postular al MII es tener grado de Licenciado en Ciencias de la Ingeniería Informática o un título profesional cuyo nivel, contenido y duración de estudios sean equivalentes a los necesarios para obtener el grado de Licenciado correspondiente.
- Art.20** Una solicitud de admisión será analizada en el CP sólo cuando el postulante haya hecho llegar todos los antecedentes requeridos, establecidos en el procedimiento de admisión de la DP.
- Art. 21** Una vez recibidos los antecedentes completos del postulante por parte de la DP, el Director del MII los pondrá a disposición del CP, el que resolverá por mayoría absoluta sobre la aceptación o rechazo de la postulación.
- Art. 22** Un postulante podrá ser 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. Para tomar la decisión, el CP deberá considerar las calificaciones de pregrado, carta de interés del postulante, cartas de recomendación, publicaciones en revistas y conferencias y otras consideraciones académicas.
- Art. 23** El CP o el Director del MII podrán exigir que el postulante entregue antecedentes adicionales y/o que participe en una entrevista presencial o remota, para evaluar aspectos relevantes que permitan decidir con mayor información sobre la solicitud de admisión.
- Art.24** El CP podrá aceptar condicionalmente al postulante, quien deberá primero aprobar una etapa de nivelación antes de ser aceptado definitivamente en el MII.
- Art. 25** Una vez que el CP resuelva sobre la aceptación o rechazo del postulante al MII, el Director del MII informará dicha decisión a la DP. En caso que el postulante sea admitido en el MII, el Director del MII designará un Tutor, lo que también deberá ser informado a la DP.
- Art.26** El Tutor deberá proponer un programa de estudios específico para el estudiante recién admitido, y cuando corresponda, podrá incluir una homologación y/o convalidación de un máximo de 30 SCT en asignaturas del Programa de Estudios. Será el CP quien aprobará definitivamente este programa de estudios específico, lo que deberá ser informado por el Director del MII a la DP.



TITULO V: SOBRE EL DESARROLLO DEL PROGRAMA.

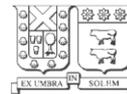
- Art. 27** El Plan de Estudios del MII (se adjunta en el Anexo N° 3 de este reglamento) contempla un total de 100 créditos SCT en las siguientes actividades:
- a) 60 créditos SCT en asignaturas formativas de nivel de postgrado (Programa de Estudios). La lista de asignaturas y sus respectivos programas se adjuntan en los Anexos N°4 y 6 de este reglamento, respectivamente.
 - b) 40 créditos SCT como Actividad de Graduación, que se compone de las asignaturas Seminario de Tesis I y II, en cuyo contexto el estudiante desarrolla su Tesis de Magíster, tal como se especifica en el Título VI de este reglamento.
- Art. 28** Cada asignatura de nivel postgrado será evaluada con una nota de 0 a 100, siendo 70 la nota mínima para su aprobación.
- Art. 29** El estudiante deberá aprobar el total de 100 créditos SCT que define el Plan de Estudios del MII, donde se cuentan también las asignaturas convalidadas u homologadas al estudiante por el CP.
- Art. 30** Una vez completados los 60 créditos SCT correspondientes a los cursos de nivel postgrado, el estudiante deberá informar al Director del MII el nombre del académico parte del CDTP que será su Director de Tesis. El Director del MII informará por escrito de esta decisión a la DP. En este momento, el Director de Tesis pasará a ser el tutor del estudiante.

TITULO VI: ACTIVIDAD DE GRADUACIÓN Y EXAMEN DE GRADO.

- Art. 31** La Tesis de Magíster constituye la última etapa de formación para cumplir con el perfil declarado por el MII, siendo ésta un trabajo personal de investigación que debe contribuir con conocimiento novedoso al desarrollo de la especialidad. El desarrollo de una Tesis de Magíster considera primero la aprobación de un Tema de Tesis, luego el desarrollo de un Trabajo de Tesis que debe concluir con la presentación de un escrito con los resultados de la investigación y, una vez aprobado éste, el Examen de Grado.

6.1 De la Inscripción y Aprobación del Tema de Tesis

- Art. 32** El Seminario de Tesis I corresponde a una asignatura donde el estudiante trabaja bajo la supervisión del Director de Tesis en definir el estado del arte del tema de tesis que el estudiante ha elegido y planificar el trabajo de investigación que deberá desarrollar.
- Art. 33** El Seminario de Tesis I concluye con la formulación de un proyecto de Tesis y la aprobación por parte del CP del Tema de Tesis presentado.



Art. 34 El Seminario de Tesis I podrá ser inscrito cuando un estudiante haya aprobado todas las asignaturas del Programa de Estudios del MII. El profesor de la asignatura inscrita será el Director de Tesis.

6.2 De la Presentación y Evaluación del Trabajo de Tesis

Art. 35 El Seminario de Tesis II corresponde al trabajo de investigación que realiza el estudiante en el desarrollo de su proyecto de tesis.

Art. 36 El Director de Tesis evaluará el Seminario de Tesis II una vez que el estudiante entregue un borrador completo con los resultados de investigación obtenidos en el Trabajo de Tesis.

Art. 37 El escrito del Trabajo de Tesis debe ser redactado en idioma español o inglés, según los formatos que define al respecto la Universidad.

Art. 38 La exigencia de productividad mínima para hacer entrega del escrito de la tesis es una publicación enviada a una conferencia internacional o revista indexada.

Art. 39 El CP una vez recibido un Trabajo de Tesis patrocinado por un Director de Tesis miembro del CDTP, procede a conformar un **Comité de Tesis**, que se compone al menos de los siguientes miembros:

- a) El Director de Tesis del estudiante.
- b) Un profesor co-referente interno a la UTFSM, designado por el CP.
- c) Un profesor o investigador co-referente externo e independiente a la UTFSM, experto en el área, designado por el Comité de Coordinación y Desarrollo de Investigación y Postgrado a proposición del CP.
- d) El Director del MII, o quién designe de entre los miembros del CP. Éste preside el Comité de Tesis, no pudiendo recaer esta responsabilidad en el Director de Tesis.

Art. 40 El Examen de Grado es público y consiste en una presentación y defensa oral de la Tesis de Magíster por parte del estudiante.

Art. 41 El Examen de Grado se dará por aprobado si la calificación es mayor o igual a 85, en escala de 0 a 100.

Art. 42 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 de este Examen, determinará conceder o no una última oportunidad para que el candidato al Grado lo rinda nuevamente en un determinado plazo no superior a 6 meses.

TITULO VII: DEL GRADO ACADÉMICO.

Art. 42 Una vez cumplidas por parte del estudiante todas las exigencias de Graduación, la Universidad otorga el grado académico de “Magíster en Ciencias de la Ingeniería Informática”.



TITULO VIII: DE LA RESPONSABILIDAD DEL PRESENTE REGLAMENTO.

Art. 43 La responsabilidad de la aplicación de las disposiciones contenidas en el presente reglamento al interior del Programa MII, será del Director del MII.

ARTÍCULOS TRANSITORIOS.

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

Art. T2 Aquellos alumnos de magister, aún no graduados, que hubiesen ingresado al MII en una modalidad distinta a la establecida en este reglamento, se acogerán a lo dispuesto en el Reglamento General N° 45-A, aprobado por el Consejo Académico el 1 de abril de 2003.



ANEXO Nº 1: ADMINISTRACIÓN DEL MII.

DIRECCIÓN

- MENDOZA ROCHA, MARCELO (DIRECTOR)

COMITÉ DE PROGRAMA

- MENDOZA ROCHA, MARCELO (PRESIDENTE)
- MONGE ANWANDTER, RAÚL
- MOREIRA WENZEL, ANDRÉS
- ÑANCULEF ALEGRÍA, RICARDO
- RIFF ROJAS, MARÍA CRISTINA

CLAUSTRO DE PROFESORES

- | | |
|------------------------------|--|
| ● ALLENDE OLIVARES, HÉCTOR | Máquinas de Aprendizaje |
| ● ARROYUELO BILLIARDI, DIEGO | Análisis y Diseño de Algoritmos |
| ● ASTUDILLO ROJAS, HERNÁN | Arquitectura de Software |
| ● BONNAIRE, XAVIER | Sistemas de Computación Distribuida |
| ● CASTRO VALDEBENITO, CARLOS | Optimización Combinatorial y Metaheurísticas |
| ● DOMBROVSKAIA, LIOUBOV | Interfaces Hombre-Máquina |
| ● LOBOS YÁÑEZ, CLAUDIO | Computación Gráfica |
| ● MENDOZA ROCHA, MARCELO | Recuperación de Información |
| ● MONGE ANWANDTER, RAÚL | Sistemas Distribuidos y Seguridad |
| ● MOREIRA WENZEL, ANDRÉS | Informática Teórica |
| ● ÑANCULEF ALEGRÍA, RICARDO | Máquinas de Aprendizaje |
| ● RIFF ROJAS, MARÍA CRISTINA | Computación Evolutiva y Metaheurísticas |
| ● SALINAS CARRASCO, LUIS | Computación Científica |
| ● SOLAR FUENTES, MAURICIO | Computación Paralela y Distribuida |

COLABORADORES

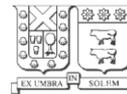
- | | |
|-----------------------------|---|
| ● ARAYA LÓPEZ, MAURICIO | Astro-informática |
| ● BUIL ARANDA, CARLOS | Datos y Semántica |
| ● HOFFMANN, HUBERT | Computación Gráfica |
| ● LÓPEZ MONCADA, CLAUDIA | Sistemas de Información |
| ● MONTERO URETA, ELIZABETH | Computación Evolutiva y Metaheurísticas |
| ● MORAGA ROCO, CLAUDIO | Lógica Borrosa |
| ● TORRES LÓPEZ, CLAUDIO | Computación Científica |
| ● VALLE VIDAL, CARLOS | Máquinas de Aprendizaje |
| ● VISCONTI ZAMORA, MARCELLO | Ingeniería de Software |
| ● VON BRAND SKOPNIK, HORST | Sistemas de Computación |



ANEXO Nº 2:

CDTP DEL MII (CLAUSTRO DE PROFESORES).

Nº	Área de especialización del Programa	Nombre	Grado / institución otorgante / año	Línea de especialización
01	Computación Científica e Inteligencia Computacional	ALLENDE OLIVARES, HÉCTOR	Dr. Rer. Nat. Computational Statistik / U. Dortmund. Dortmund, Alemania / 1988	Máquinas de Aprendizaje
02	Informática Teórica	ARROYUELO BILLIARDI, DIEGO	Doctor en Ciencias de la Computación / Universidad de Chile / 2009	Análisis y Diseño de Algoritmos
03	Arquitectura e Ingeniería de Software	ASTUDILLO ROJAS, HERNÁN	Ph.D. In Computer Science / Georgia Institute of Technology, USA / 1996	Arquitectura de Software
04	Sistemas de Computación y Bases de Datos	BONNAIRE, XAVIER	Docteur en Systèmes Informatiques / Université Pierre et Marie Curie, Paris VI, Paris, Francia / 1998	Sistemas de Computación Distribuida
05	Computación Científica e Inteligencia Computacional	CASTRO VALDEBENITO, CARLOS	Docteur en Informatique / Université Henri Poincaré, Nancy I. Nancy, Francia / 1998	Optimización Combinatorial y Metaheurísticas
06	Arquitectura e Ingeniería de Software	DOMBROVSKAIA, LIOUBOV	Doctor en Ciencias de la Ingeniería / Pontificia Universidad Católica de Chile. Santiago, Chile / 1998	Interfaces Hombre-Máquina
07	Sistemas de Computación y Bases de Datos	LOBOS YÁÑEZ, CLAUDIO	Docteur en Modèles, Méthodes et Algorithmes en Biologie, Santé et Environnement / Université Grenoble 1 - Joseph Fourier, Francia / 2009	Computación Gráfica
08	Sistemas de Computación y Bases de Datos	MENDOZA ROCHA, MARCELO	Doctor en Ciencias de la Computación / Universidad de Chile / 2007	Recuperación de Información
09	Sistemas de computación y Bases de Datos	MONGE ANDWANDTER, RAÚL	Dr. Ing / Universität Erlangen-Nürnberg, Alemania / 1992	Sistemas distribuidos y seguridad
10	Informática Teórica	MOREIRA WENZEL, ANDRÉS	Doctorado en Cs de la Ing, con mención en Modelamiento Matemático / Universidad de Chile / 2003	Informática Teórica



11	Computación Científica e Inteligencia Computacional	ÑANCULEF ALEGRÍA, JUAN RICARDO	Doctorado en Ingeniería Informática / UTFSM/ 2010	Máquinas de Aprendizaje
12	Computación Científica e Inteligencia Computacional	RIFF ROJAS, MARÍA CRISTINA	Docteur en Informatique et Mathématiques / Ecole National des Ponts et Chaussées, Paris, Francia / 1997	Computación Evolutiva y Metaheurísticas
13	Computación Científica e Inteligencia Computacional	SALINAS CARRASCO, LUIS	Dr. rer. Nat. con mención en Matemáticas / Universidad del Saarlandes. Saarbrücken, Alemania / 1976	Computación Científica
14	Sistemas de Computación y Bases de Datos	SOLAR FUENTES, MAURICIO	Doutorado em Engenharia de Sistemas e Computação / Universidad Federal de Rio de Janeiro, Brasil / 1992	Computación Paralela y Distribuida

COLABORADORES DEL MII

01	Sistemas de Computación y Bases de Datos	ARAYA LÓPEZ, MAURICIO	Ph. D. in Computer Science/Université de Lorraine, Francia/2013	Astro-informática
02	Arquitectura e Ingeniería de Software	BUIL ARANDA, CARLOS	Ph. D. in Computer Science/Universidad Politécnica de Madrid, España/2012	Datos y Semántica
03	Sistemas de Computación y Bases de Datos	HOFFMANN, HUBERT	Dr. Ing. Informatiker/Tech. U. Berlín, Alemania/1987	Computación Gráfica
04	Arquitectura e Ingeniería de Software	LÓPEZ MONCADA, CLAUDIA	Ph. D. in Computer Science/Universidad Politécnica de Madrid, España/2012	Sistemas de Información
05	Computación científica e Inteligencia Computacional	MONTERO URETA, ELIZABETH	Doctor en Informática, Université de Nice/ Sophia-Antipolis, Francia/2012	Computación Evolutiva y Metaheurísticas
06	Computación Científica e Inteligencia Computacional	MORAGA ROCO, CLAUDIO	Doctor en Ingeniería, Universidad Técnica Federico Santa María / 1972	Lógica Borrosa
07	Computación científica e Inteligencia Computacional	TORRES LÓPEZ, CLAUDIO	Ph.D. In Applied Mathematics/University of Delaware, USA/ 2012	Computación Científica
08	Computación científica e Inteligencia Computacional	VALLE VIDAL, CARLOS	Doctorado en Ingeniería Informática, Universidad Técnica Federico Santa María/2012	Máquinas de Aprendizaje



09	Arquitectura e Ingeniería de Software	VISCONTI ZAMORA, MARCELLO	Ph.D. In Computer Science / Oregon State University. USA / 1993	Ingeniería de Software
10	Sistemas de Computación y Bases de Datos	VON BRAND SKOPNIK, HORST	Ph.D. In Computer Science / Louisiana State University, Baton, Rouge, LA, USA / 1987	Sistemas de Computación

VISITANTES DEL MII

01	Computación científica e inteligencia computacional	CANESSA TERRAZAS, ENRIQUE	PH. D. In Computer Science, University of Michigan, USA (2002)	Agentes inteligentes, sistemas adaptativos
02	Computación científica e inteligencia computacional	FRERY ORGAMBIDE, ALEJANDRO	Doctor en Computación Aplicada, Instituto Nacional de Pesquisas Espaciais, INPE, Brasil, (1993)	Procesamiento digital de imágenes
03	Sistemas de Computación y Bases de Datos	FERNANDEZ BUGLIONI, EDUARDO	Ph.D. in Computer Science, University of California, Los Ángeles, California, USA (1972)	Sistemas distribuidos y seguridad



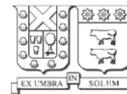
ANEXO Nº 3:

PLAN DE ESTUDIOS DEL MII.

AÑO 1				AÑO 2			
1ER. SEMESTRE		2do. SEMESTRE		1ER. SEMESTRE		2do. SEMESTRE	
INF-400	7 SCT.	INF-400	7 SCT.	INF-591	20 SCT	INF-592	20 SCT
INF-400	7 SCT.	INF-400	7 SCT.				
INF-500	8 SCT.	INF-500	8 SCT.				
INF-500	8 SCT.	INF-500	8 SCT.				
60 SCT ASIGNATURAS				40 SCT TESIS			

Observaciones:

- Los seminarios INF 591 e INF 592 corresponden a los seminarios de Tesis I y II indicados en el Título VI de este reglamento, respectivamente.



ANEXO N° 4: ASIGNATURAS DEL MII (PERÍODO 2010 - 2016)

SIGLA	ASIGNATURA	PROFESOR
Asignaturas Nivel 400		
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-424	Pruebas de Software	M. Visconti
INF-427	Interfaces Hombre-Máquina	L. Dombrovskaja
INF-435	Programación de Sistemas	H. von Brand
INF-440	Sistemas Distribuidos	R. Monge
INF-451	Computación Gráfica	C. Lobos/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-476	Algoritmos Evolucionistas	M.C. Riff
INF-477	Redes Neuronales Artificiales	H. Allende
INF-478	Análisis Inteligente de Datos	H. Allende/R. Ñanculef
INF-479	Reconocimiento de Formas en Minería de Datos	H. Allende/M. Mendoza
INF-480	Redes Complejas	A. Moreira
INF-481	Tecnologías de Búsqueda Avanzada en la Web	M. Mendoza
INF-484	Web Semántica	C. Buil
INF-485	Elementos de Análisis para Informática y Computación	L. Salinas
INF-486	Inferencia Estadística y Aprendizaje	H. Allende
INF-487	Computabilidad y Complejidad Computacional	A. Moreira
INF-488	Tópicos Actuales en Ingeniería de Software	H. Astudillo
INF-490	TEI: Computación Peer to Peer	X. Bonnaire
INF-491	TEI: Arquitecturas Empresariales	H. Astudillo
INF-493	TEI: Seminario de Sistemas Distribuidos	R. Monge
INF-494	TEI: Astro-informática	M. Solar
INF-495	TEI: Ingeniería de Software Experimental	M. Visconti
INF-496	TEI: Mejoramiento de Procesos de Software	M. Visconti
Seminarios de Tesis Magíster:		
INF-591	Seminario de Tesis I	Director de Tesis
INF-592	Seminario de Tesis II	Director de Tesis

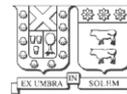
Nota:

TEI: Temas Especiales en Informática.





SIGLA	ASIGNATURA	PROFESOR
Asignaturas Nivel 500		
INF-510	Métodos Numéricos Computacionales	C. Torres
INF-520	Compresión de Texto	D. Arroyuelo
INF-524	Evaluación de Arquitectura de Software	H. Astudillo
INF-532	Hot Topics in Operating Systems	X. Bonnaire
INF-534	Programación Paralela Aplicada Avanzada	X. Bonnaire
INF-560	Modelos Computacionales en Series de Tiempo	H. Allende
INF-564	Diseño Avanzado de Algoritmos	D. Arroyuelo/M. Mendoza
INF-565	Metodología de la Investigación	H. Astudillo/H. Allende
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-577	Redes Neuronales Avanzadas	H. Allende
INF-578	Máquinas de Aprendizaje Computacional	H. Allende/R. Ñanculef
INF-582	Cloud Computing y Big Data	M. Solar
INF-584	Sistemas Complejos Discretos	A. Moreira



ANEXO N° 5

INFRAESTRUCTURA Y RECURSOS DISPONIBLES PARA EL MII

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 3°. 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 Postgrado).
- 3) 2 Auditorios debidamente equipados, con internet (fija más wifi) y sistema de videoconferencia.
- 4) 10 Espacios comunes (cocina, cafetería, circulación, terraza, entre otros).
- 5) 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).
- 6) 11 Baños (alumnos, profesores y funcionarios).

Además el Departamento de Informática dispone de instalaciones con una superficie de 294 m², en el Edificio A del campus San Joaquín de la UTFSM, en el piso 2, y y 267 m² en el Edificio B del mismo campus, piso 1. Estas superficies comprenden la siguientes áreas:

- 1) 12 oficinas (profesores, administrativos, secretarías)
- 2) 3 Laboratorios (1 de uso exclusivo para postgrado)
- 3) 1 Auditorio debidamente equipado, con internet y sistema de videoconferencia.
- 4) 3 Espacios comunes (sala profesores *part-time* con cafetería, sala de reuniones con sistema de videoconferencia, sala de estar).
- 5) 2 salas que prestan servicios (sala de servidores, centro de alumnos)
- 6) 8 baños (profesores y alumnos).

El Departamento de Informática cuenta con el siguiente equipamiento y laboratorios:

1. Laboratorio de Computación “LABCOMP” (Edificio F):

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 (Edificio F):

Laboratorio de soporte a docencia, tiene un número de 20 PCs, 1 pizarra, 1 proyector, 1 telón, cuenta con sistema de videoconferencias y sistema de audio.

3. Laboratorio Hardware (Edificio F):

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” (Edificio F):



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) (Edificio F):

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” (Edificio F):

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” (Edificio F):

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 “Postgrado” (Edificio F):

Laboratorio de uso específico de postgrado, cuenta con un total de 17 PCs, modelo MAC de 21, 5 pulgadas, 2 estaciones de trabajo con capacidad para 4 estudiantes cada una. Existen 8 cubículos individuales, para profesores visitantes y alumnos 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 alumnos, 1 pizarra, 1 proyector, 1 telón, área de cafetería y de descanso.

9. Laboratorio Aula Tecnológica (Edificio B):

Laboratorio de soporte a docencia, tienen 20 PCs, 1 pizarra, 1 proyector, 1 telón, 1 TV, sistema de videoconferencia.

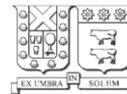
10. Laboratorio de Postgrado (Edificio B):

Laboratorio de uso específico para postgrado, cuenta con 3 estaciones de trabajo con capacidad para 4 estudiantes cada una, cada una con computador. Además hay 4 puestos adicionales también con computador. Se dispone de sistema de TV, kinect para apoyo a actividades de HCI, y área de descanso.



ANEXO N° 6

PROGRAMA DE ASIGNATURAS DEL MII



Asignatura	Teoría y aplicaciones de “Wavelets”		Sigla	INF-406
Créditos SCT	7	Conocimientos previos: MII-405 Procesamiento de Señales e Imágenes, o equivalente		

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.
- Tareas (unas 5) de nivel compatible con estudios de postgrado; una parte



importante de las tareas debe ser de naturaleza experimental (computador).

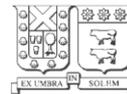
Bibliografía

- [1] P.L. Dragotti and M. Vetterli, Wavelet Footprints: Theory, Algorithms and Applications, IEEE Transactions on Signal Processing, Vol. 51, Nr. 5, pp. 1306-1323, 2003.
- [2] M.N. Do and M. Vetterli, Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance, IEEE Transactions on Image Processing, Vol. 11, Nr. 2, pp. 146-158, 2002.
- [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] Ch. Blatter: "Wavelets, eine Einfuehrung", Vieweg, Braunschweig-Wiesbaden, 1998.
- [8] A.K. Louis, P. Maass, A. Rieder, "Wavelets, Theorie und Anwendungen", B.G. Teubner, Stuttgart, 1994.
- [9] Y. Meyer, "Ondelettes et operateurs", Hermann, Paris, 1990.

Revistas

Proceedings del IEEE relacionados con procesamiento de señales, Teoría de la Información, etc.

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



Asignatura	Calidad y Productividad de Software	Sigla	INF-413
Créditos SCT	7	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

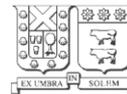
Trabajos prácticos (20%), Trabajo independiente de investigación (20%), Certamen: 20%, Examen final: 40%



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	



Asignatura	Arquitectura de Software	Sigla	INF-414
Créditos SCT	7	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.

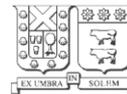
Objetivos

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



Asignatura	Pruebas de Software	Sigla	INF-424
Créditos SCT	7	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.

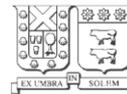
Objetivos

- 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



Asignatura	Interfaces Hombre-Máquina	Sigla	INF-427
Créditos SCT	7	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.

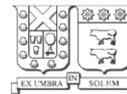
Objetivos

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'Reilly, 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.
Aprobado:	Depto. de Informática	
Fecha:	1/8/2000	
		Actualización: Diciembre 2012



Asignatura	Programación de Sistemas	Sigla	INF-435
Créditos SCT	7	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	



Asignatura	Sistemas Distribuidos		Sigla	INF-440
Créditos SCT	7	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

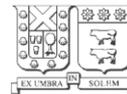
2 Certámenes: 67%

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	



Asignatura	Computación Gráfica	Sigla	INF-451
Créditos SCT	7	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.

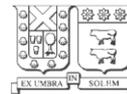
Contenidos

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

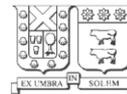


Presentaciones 20%
Trabajo Escrito 20%
Tareas 50%
Nota del Profesor 10%

Bibliografía

- [10] Edward Angel, Dave Shreiner: *“Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL™”*, Addison-Wesley, 6° Ed. 2012 (Texto Guía).
- [11] Robert Whitrow: *“OpenGL Graphics through Applications”*, Springer 2008.
- [12] Peter Shirley: *“Fundamentals of Computer Graphics”*, A.K. Peters 2° Ed. 2005.
- [13] Edward Angel: *“OpenGL: A Primer”*, Addison-Wesley 3° Ed. 2007.
- [14] Revista: IEEE Computer Graphics and Applications (ISSN 0272-1716).
- [15] Revista: ACM SIGGRAPH Computer Graphics (ISSN 0097-8930).
- [16] 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	



Asignatura	Optimización Combinatoria	Sigla	INF-472
Créditos SCT	7	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

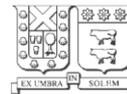
1. Modelos de optimización combinatoria: conceptos básicos, variables enteras, variables binarias, formulación de modelos, modelos clásicos.
2. Resolución de modelos de optimización combinatoria: enumeración exhaustiva, técnicas de ramificación y acotamiento, técnicas de enumeración implícita, planos 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

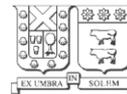
La asignatura se evalúa en base a certámenes (60%), análisis de artículos (20%) y tareas (20%).



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



Asignatura	Tópicos Avanzados en Inteligencia Artificial	Sigla	INF-474
Créditos SCT	7	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



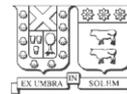
Cada tarea tiene una ponderación de un 10%, revisión de los papers un 10% y el proyecto un 40%.

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





Departamento de Informática

Asignatura	Modelamiento Estocástico y Simulación	Sigla	INF-475
Créditos SCT	7	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

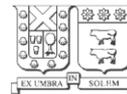
Tareas (30%), Examen (30%) y Proyecto (40%).

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
Departamento de Informática

Asignatura	Algoritmos Evolucionistas	Sigla	INF-476
Créditos SCT	7	Conocimientos previos: Programación en C y Pascal para la implementación de las diversas estrategias.	

Descripción

Este curso es un curso avanzado con énfasis en la propia exploración e investigación. La evaluación se realizará con trabajos, revisión de un paper y un proyecto. El proyecto será desarrollado individualmente. El proyecto puede sintetizar múltiples estrategias de diseño o ser una profundización de una en particular usando problemas y aplicaciones elegidas por el estudiante. Esta técnica es lo suficientemente flexible como para incluir aplicaciones clásicas de la investigación de operaciones, diseño en ingeniería, sistemas de manufactura y finanzas.

Objetivos

- .Presentar al estudiante los algoritmos evolucionistas como una técnica alternativa en la resolución de problemas de optimización y de satisfacción de restricciones.
- .Mostrar al alumno el uso de los algoritmos evolucionistas en la resolución de problemas del mundo real.

Contenidos

.*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).

.*Algoritmos Evolucionistas (AE)*: historia, y componentes de un AE (criterios para definir funciones de evaluación, criterios para definir operadores, noción de auto-adaptabilidad y tuning de parámetros).

.*Algoritmos Evolucionistas Híbridos para la Optimización Combinatoria*: aspectos considerar en la función de evaluación, aspectos a considerar en los operadores,



algoritmos propuestos en la literatura actual, criterios para comparar AE que resuelven problemas de optimización combinatoria.

.Co-evolución, problemas con múltiples objetivos, auto-adaptación, medioambiente cambiante, juegos.

.Aplicaciones de los algoritmos evolucionistas, su relación con vida artificial y sistemas multi-agentes.

.Conclusiones, investigación actual.

Metodología

Clases expositivas, presentación de avances del proyecto

Evaluación

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

Bibliografía

.Michalewicz, "Genetic Algorithm + Data Structures", Springer Ed., 1996.

.Back, "How to nature work", Oxford ED., 1996.

.ED. Bäck & Michalewics & Fogel, "Handbook of evolutionary Computation", Oxford University Press, 1997.

."Evolutionary Algorithms in Theory and Practice", Oxford University Press, 1995.

.Adicional: IEEE Transactions pon Evolutionary Computation, Journal of Heuristics, Complex Systems.

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



Asignatura	Redes Neuronales Artificiales	Sigla	INF-477
Créditos SCT	7	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. Web. *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	



Asignatura	Análisis Inteligente de Datos	Sigla	INF-478
Créditos SCT	7	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. Springer. 2003
- [2] Christopher 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:
Aprobado:	Depto. de Informática	
Fecha:	19/05/2010	

Actualización: Diciembre 2012





Universidad Técnica Federico Santa María
Departamento de Informática

Asignatura : Reconocimiento de Formas en Minería de Datos	Sigla : INF-479
Créditos SCT : 7	Prerrequisitos : Base de Datos Y Estadística Computacional.

Descripción:

Fundamentos teóricos de las técnicas usadas para el reconocimiento de entidades y patrones. Estos fundamentos entregan los conocimientos necesarios para reconocer la aplicación de estas técnicas así como la base para resolver diversos tipos de problemas de reconocimiento.

Objetivos: *Proporcionar al estudiante los fundamentos del Reconocimiento y de sus principales aplicaciones.*

Al término del curso el alumno será capaz de:

- .Conocer los principios básicos de reconocimiento de patrones
- .Conocer las diferentes técnicas para reconocimiento de formas .
- .Conocer las principales aplicaciones del reconocimiento de patrones, en Data Mining y Procesamiento de Imágenes.
- .Conocer los principales avances para los problemas de clasificación y reconocimiento de patrones.
- .Conocer los principales avances de las ANN, para los problemas de clasificación y reconocimiento de patrones.

Contenido

.Introducción al reconocimiento de Patrones : Conceptos básicos Fundamentos, Vectores Aleatorios e Inferencia ; análisis discriminante.



- .Aproximación paramétrica al reconocimiento de Patrones:
Clasificadores lineales; cuadráticos , Bayesianos otros.
- .Aproximación no paramétrica al reconocimiento de Patrones: función de densidad de Parzen , Vecino más cercano, K- means; LVQ,
- .Métodos de Clasificación no supervisada (Clustering): Clustering parametrico, Clustering no- parametrico,
- .Redes Neuronales y Clasificación: FANN; Redes Bayesianas, Redes Fuzzy, Métodos de Selección y Extracción de Características

Metodología

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

Evaluación

Examen (40%), tareas y presentación de trabajos (60%).

Bibliografía

- .R. Duda, P. Hart and D. Stork, “*Pattern Classification*”, Ed. J. Wiley ,2001.
- .C. Bishop *Pattern recognition and Machine Learning*”, Springer, 2006.
- .J. Han, M. Kamber, “*Data Mining Concepts and Techniques*”, Ed. Academic Press, 2001.
- .Revistas: Pattern Recognition, Pattern Recognition Letters IEEE Trans. on Neuronal Networks, , Statistics and Computing.

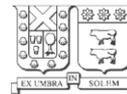
Revistas : IEEE Trans. on Neuronal Networks, Neuronal computing, Pattern Recognition, and Statistics and Computing.



--

Elaborado :	H. Allende Jul-2002	
Aprobado:	Comité IP del DI Oct/2002	
Actualizado	Jul-2011	





Departamento de Informática

Asignatura	Redes Complejas	Sigla	INF-480
Créditos SCT	7	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*. World 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
Departamento de Informática

Asignatura	Tecnologías de búsqueda en la Web		Sigla	INF-481
Créditos SCT	7	Conocimientos previos: estructuras de datos y algoritmos, bases de datos, probabilidad y estadística.		
Hrs. Cat. Sem. : 3	Hrs. Ayud. Sem. : 0		Hrs. Lab. Sem. : 0	

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:

1. Comprender los fundamentos de recuperación de información en texto.
2. Aplicar modelos de recuperación de información en sistemas de búsqueda en la Web.
3. Comprender los principales desafíos y limitaciones de los métodos existentes para realizar búsquedas en la Web social.
4. Comprender y conocer tendencias en búsqueda en la Web.
5. Analizar nuevas tecnologías y métodos de búsqueda, entre ellas búsqueda en la Web social, búsqueda vertical y búsqueda por facetas.

Competencias transversales

1. Comunicar información oral y escrita de manera eficaz.
2. Actuar con autonomía, flexibilidad, iniciativa, y pensamiento crítico al enfrentar problemas nuevos.

Competencias específicas

1. Comprender y evaluar nuevas tecnologías de búsqueda en la Web.
2. Implementar nuevos algoritmos de búsqueda para la Web 1.0 y 2.0.

Contenidos

- [1]Fundamentos de recuperación 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.
- [2]Modelos de recuperación de información en texto: Modelo Booleano, modelo Tf-Idf, modelo probabilístico, modelos con retroalimentación de usuarios.
- [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.
- [5]Temas avanzados: Búsqueda vertical, búsqueda por facetas, minería de opiniones, categorización de texto.



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: Tarea de procesamiento de texto en grupo y proyecto a desarrollar en el semestre que permitirá trabajar en torno al desarrollo de un sistema de búsqueda en la Web.
4. Actividades individuales: Presentación de un tema de investigación en clases, desarrollo de un trabajo escrito sobre un tema avanzado del estado del arte.

Evaluación

5. 3 certámenes
6. 1 Proyecto
7. 1 Tarea
8. 1 Presentación en clases
9. 1 Informe escrito sobre tema avanzado

Calificación

- Promedio certámenes: 40%
- Nota proyecto: 20%
- Nota tarea: 10%
- Nota presentación: 10%
- Nota informe escrito: 20%

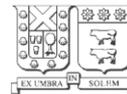
Bibliografía

1. Büttcher, S., Clarke, Ch. & Cormack, G. "Information Retrieval: Implementing and Evaluating Search Engines", MIT Press, 2010.
2. Manning, Ch., Raghavan, P., Schütze, H. "Introduction to Information Retrieval", Cambridge University Press, 2008.

Revistas

Journal of Information Retrieval (JIR), Journal of the American Society for Information Systems and Technology (JASIST), ACM Transactions on the Web (TWEB), ACM Transactions on Information Systems (TOIS), ACM Transactions on Intelligent Systems and Technology (TIST), Journal of Web Engineering (JWE).

Elaborado:	Marcelo Mendoza	Observaciones: Actualizado Diciembre 2012
Aprobado:	Depto. Informática 07-07-2011	
Fecha:	24-06-2011	



Asignatura	Web Semántica		Sigla	INF-484
Créditos SCT	7	Conocimientos previos: Bases de Datos		
Hrs. Cat. Sem. :	4	Hrs. Ayud. Sem. :	0	Hrs. Lab. Sem. : 0

Descripción

Al final de esta asignatura, el estudiante será capaz de utilizar tecnologías de la Web Semántica para lograr ese objetivo, logrando que sus propias aplicaciones puedan interpretar automáticamente el contenido de una página web, hacer que sus aplicaciones se entiendan entre ellas o que esas mismas aplicaciones razonen automáticamente acerca de los datos que ellas generan. Además, el estudiante será capaz de desarrollar páginas web cuyos contenidos sean entendidos por los grandes jugadores de Internet: Google, Bing, Yahoo y Yandex.

Objetivos

- Desarrollar, implantar y mantener sistemas de software confiables, eficientes y factibles
- Diseña modelos de datos de la Web Semántica, utilizando un framework reconocido universalmente.
- Diseña modelos de dominio para la Web Semántica, utilizando lenguajes estándares en el área.
- Desarrolla una Web Semántica, aplicando las tecnologías vistas en clases.

Contenidos

El curso está dividido en los siguientes módulos, en los cuales se estudian las distintas tecnologías que componen la llamada Web Semántica.

1. La Web actual

En este módulo se mostrarán brevemente los problemas de la Web actual y se introducirán las tecnologías de la Web Semántica. Todas estas tecnologías son estándares del World Wide Web Consortium (W3C¹).

2. El modelo de datos de la Web Semántica

En este módulo se verá el modelo de datos de la Web Semántica. Este modelo es el Resource Description Framework (RDF) [GKM], el cual se basa en describir los datos como relaciones de tres elementos (sujeto, predicado y objeto). Todos los datos de la Web Semántica utilizan este modelo de datos.

3. Modelado de dominios de datos

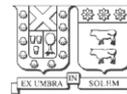
Para modelar los datos de un dominio concreto la Web Semántica dispone de dos lenguajes: RDF Schema y Web Ontology Language (OWL) [DS04, Gro12]. Mediante estos dos lenguajes es posible definir restricciones sobre los datos con diverso nivel de detalle. Se aprenderá desde cómo definir clases de datos a definir restricciones universales o existenciales sobre esos mismos datos, además de entender la complejidad asociada a esas restricciones.

4. El lenguaje de consulta de la Web Semántica

Una vez comprendidos cómo se estructuran los datos es el momento de acceder a ellos. Para acceder a esos datos se utilizará el lenguaje de consulta SPARQL [PS04, HS10]. Se verán los elementos principales del lenguaje y entenderá la complejidad del mismo.

5. Integración de datos

Integración de datos [Len02] es combinar datos de varias fuentes de tal forma que parezca que ellos provienen de una misma fuente. Las tecnologías de la Web Semántica están diseñadas para que el



proceso de integración de datos provenientes de diversas fuentes sea lo más sencillo posible. Se verá el conjunto de tecnologías Relational Database 2 RDF (RDB2RDF²) para acceder a datos almacenados en bases de datos relacionales y posteriormente convertir esos datos al modelo RDF.

6. Aplicaciones de la Web Semántica

En este módulo los alumnos verán dos aplicaciones que utilizan las tecnologías vistas durante el curso. Estas aplicaciones son Bio2RDF [CCAD13], la cual integra datos provenientes de más de 40 fuentes en el ámbito de la biomedicina y el proyecto de la Biblioteca del Congreso Nacional de Chile [CSSLG11] el cual modela las normas y leyes chilenas utilizando RDF, RDFSchema y OWL y provee acceso a esas normas a través de SPARQL.

Metodología

En este curso se trabaja con el formato de curso conocido como Blended Learning [GK04]. El curso estará dividido en dos partes fundamentales: en la primera parte se utilizarán clases no presenciales a través de videos en la plataforma Coursera (dictados por el actual profesor), complementadas con discusiones presenciales (en clase) acerca de los conceptos y problemas mostrados en estos videos. Es decir, los alumnos deberán ver los videos de la plataforma Coursera en casa y después en clase presencial se discutirán los conceptos aprendidos. El profesor se pondrá de acuerdo con los alumnos al inicio del curso.

En la segunda parte del curso, los alumnos propondrán soluciones a los problemas vistos anteriormente o bien posibles aplicaciones de las tecnologías anteriores en el dominio que el alumno proponga. Dichas propuestas se implementarán y formarán parte del proyecto final del curso. Este proyecto será individual.

Evaluación

Instrumentos de evaluación.	%
Proyectos Web del curso	15
Participación en Clase	10
Certamen contenidos (y lecturas adicionales)	25
Proyecto (desarrollo)	25
Proyecto (presentación)	25

Bibliografía

Básica

Grigoris Antoniou and Frank van Harmelen. A semantic web primer. MIT Press, 2004.

Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. Foundations of semantic web technologies. CRC Press, 2009.

Complementaria

[AGP09] Marcelo Arenas, Claudio Gutierrez, and Jorge Pérez. Reasoning web. Semantic technologies for information systems. Foundations of RDF Databases, pages 158–204. Springer-Verlag, Berlin, Heidelberg, 2009.

[AvH04] Grigoris Antoniou and Frank van Harmelen. A semantic web primer. MIT Press, 2004.

[CCAD13] Alison Callahan, Jose Cruz-Toledo, Peter Ansell, and Michel Dumontier. Bio2rdf release 2: Improved coverage, interoperability and provenance of life science linked data. In The Semantic Web: Semantics and Big Data, 10th International Conference, ESWC 2013, Montpellier, France, May 26-30, 2013. Proceedings, pages 200–212, 2013.

[CSSLG11] Francisco Cifuentes-Silva, Christian Sifaqui, and Jose Emilio Labra-Gayo. Towards an architecture and adoption process for linked data technologies in open government contexts: A case study for the library of congress of Chile. In Proceedings of the 7th International Conference on Semantic Systems, I-Semantics '11, pages 79–86, New York, NY, USA, 2011. ACM.

[DS04] Mike Dean and Guus Schreiber. OWL Web Ontology Language Reference, 2004.

[GHMP11] Claudio Gutierrez, Carlos A. Hurtado, Alberto O. Mendelzon, and Jorge Pérez. Foundations of semantic web databases. J. Comput. Syst. Sci., 77(3):520–541, May 2011.

[GK04] D Randy Garrison and Heather Kanuka. Blended learning: Uncovering its transformative potential in higher education. The internet and higher education, 7(2):95–105, 2004.

[GKM] J. J. Carroll G. Klyne and B. McBride. Resource description framework (RDF): Concepts and abstract



syntax.

[Gro12] W3C OWL Working Group. OWL 2 Web Ontology Language Document Overview (Second Edition), 2012.

[HKR09] Pascal Hitzler, Markus Krotzsch, and Sebastian Rudolph. Foundations of semantic web technologies. CRC Press, 2009.

[HS10] S. Harris and A. Seaborne. Sparql 1.1 query language, 2010.

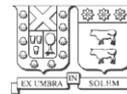
[Len02] M. Lenzerini. Data integration: A theoretical perspective. In PODS, pages 233–246, 2002.

[MAHP11] Alejandro Mallea, Marcelo Arenas, Aidan Hogan, and Axel Polleres. On blank nodes. In Proceedings of the 10th International Conference on The Semantic Web - Volume Part I, ISWC'11, pages 421–437, Berlin, Heidelberg, 2011. Springer-Verlag.

[PAG09] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. TODS, 34(3), 2009.

[PS04] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF, January 2004.

Elaborado:	Carlos Buil	Observaciones:
Aprobado:	Depto. Informática	
Fecha:	Marzo 2016	



Universidad Técnica Federico Santa María
Departamento de Informática

Asignatura	Elementos de Análisis para Informática y Computación.		Sigla	INF-485	
Créditos SCT	4	Conocimientos previos: Autorización del Profesor con base en una entrevista personal con el estudiante.			
Hrs. Cat. Sem.:	2	Hrs. Ayud. Sem.:	0	Hrs. Lab. Sem.:	0

Descripción

Asignatura centrada en los conceptos, resultados y métodos del Análisis necesarios para abordar los temas de la investigación avanzada propios de la Informática y la Ciencia de la Computación contemporáneas.

Objetivos

Capacitar al estudiante en los temas fundamentales del Análisis que aparecen en la investigación avanzada y las publicaciones recientes en la Informática y la Ciencia de la Computación contemporáneas.

Contenidos

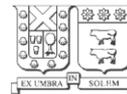
- Elementos del análisis matemático.
- Elementos del análisis funcional. Elementos de la teoría de operadores.
- Elementos de la teoría de los “kernels” y de los “kernels” reproductores. Elementos de la teoría de Nachman Aronszajn y Stefan Bergman.
- Elementos de la teoría de probabilidades e integración. Teoría de la información y entropía.

Metodología

Lecciones del Profesor, seguidas a lo largo del semestre, por una serie de exposiciones de los estudiantes. Trabajo tutorial del Profesor con cada estudiante para preparar las exposiciones de éstos. Discusión crítica de los temas tratados, con miras a proyectarlos como temas de investigación personal. Estudio y trabajo personal (textos y artículos).

Evaluación

Mediante tareas periódicas y exposiciones de artículos relevantes en la temática de la asignatura, por parte de los alumnos, y de las subsecuentes discusiones técnicas. Una parte importante de las tareas será de naturaleza experimental (computador). La calificación final de cada estudiante será el promedio aritmético de todas sus

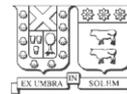


evaluaciones parciales.

Bibliografía

1. Walter Rudin. *Real and Complex Analysis*. McGraw-Hill Education; 3d edition (May 1, 1986).
2. Walter Rudin. *Functional Analysis*. McGraw-Hill Science/Engineering/Math; 2 edition (January 1, 1991).
3. John B. Conway. *A Course in Functional Analysis*. Springer Science & Business Media, Sep 7, 1990 - Mathematics - 399 pages.
4. Nachman Aronszajn, (1950). *Theory of Reproducing Kernels*. Transactions of the American Mathematical Society 68 (3): 337–404. doi:10.1090/S0002-9947-1950-0051437-7.JSTOR 1990404. MR 51437.
5. Alain Berlinet and Christine Thomas. *Reproducing kernel Hilbert spaces in Probability and Statistics*. Kluwer Academic Publishers, 2004.
6. Cucker, Felipe; Smale, Steve (2002). *On the Mathematical Foundations of Learning*. Bulletin of the American Mathematical Society 39 (1): 1–49. doi:10.1090/S0273-0979-01-00923-5.MR 1864085.
7. J. Weickert, H. Hagen (Eds.). *Visualization and Processing of Tensor Fields*. Springer, Berlin, 2006.
8. Thomas M. Cover, Joy A. Thomas. *Elements of Information Theory*. Wiley, 1991.
9. Bernhard Schölkopf, Alexander J. Smola. *Learning with Kernels*. The MIT Press. Cambridge, Massachusetts, 2002.

Elaborado:	L. Salinas	Observaciones:
Aprobado:	Departamento de Informática	
Fecha:	09/08/2015	



Universidad Técnica Federico Santa María Departamento de Informática

Asignatura	Inferencia Estadística y Aprendizaje	Sigla	INF-486
Créditos SCT	4	Conocimientos previos:	
Hrs. Cat. Sem. : 2	Hrs. Ayud. Sem. : 0	Hrs. Lab. Sem.: 0	

Descripción

Esta asignatura forma parte del núcleo de cursos fundamentales del plan de estudios de los postgrado en Ingeniería Informática y cubre fundamentos teóricos de las máquinas de Inferencia. Incluye tópicos básicos de probabilidades, inferencia estadística y aprendizaje, modelos estadísticos y métodos. Ella es una excelente preparación para el trabajo avanzado en Ciencias de la Computación, Inteligencia Artificial, Aprendizaje Automático, métodos cuantitativos en seguridad informática, Ingeniería de software entre otros.

Objetivos

Capacitar al estudiante en los temas fundamentales del Análisis y la Estadística que aparecen en la investigación avanzada y las publicaciones recientes en la Informática y la Ciencia de la Computación contemporáneas.

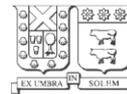
Contenidos

Parte I: Breve revisión de teoría de probabilidades

- 1.1 Probabilidad, medidas de probabilidad, independencia, probabilidad condicional, teoremas.
- 1.2 Variable aleatoria, función de distribución, Distribuciones Multivariadas, Esperanza, Varianza y Covarianza, Transformaciones, Inecuaciones de Markov, Chebychev y Hoeffding's.
- 1.3 Convergencia de variables aleatorias, tipos de convergencia, ley débil de los grandes números, teorema central del límite, método delta.

Parte II: Inferencia Estadística y Aprendizaje

- 2.1 Modelos paramétricos y no paramétricos
- 2.2 Estimación Puntual, estimación por regiones y de distribuciones de probabilidad
- 2.3 El método Bootstrap, intervalos de confianza Bootstrap.
- 2.4 Inferencia paramétrica: suficiencia, reducción y Verosimilitud teoremas
- 2.5 Contraste de hipótesis y el valor p: Lema de Neyman Pearson, comparación de Algoritmos.
- 2.6 Contraste de permutaciones, comparaciones Múltiples, Método de Bonferroni.
- 2.7 Inferencia Bayesiana: El método de Bayes, Funciones de parámetros, simulación, Estimación multi-paramétrica.
- 2.8 Teoría estadística de decisiones: Estimación del riesgo, funciones de pérdida, máxima verosimilitud, minimax y Bayes.



Parte III: Aplicaciones

Nota: Dado la extensión se sugiere seleccionar un tema para su formulación desde la perspectiva estadística.

- 3.1 Clasificación
- 3.2 Modelos lineales y log-lineales
- 3.3 Inferencia causal
- 3.4 Modelos de predicción

Metodología

Lecciones de los Profesores, seguidas a lo largo del semestre, por una serie de exposiciones de los estudiantes. Trabajo tutorial de los Profesores con cada estudiante para preparar las exposiciones de éstos. Estudio y trabajo personal (textos y artículos).

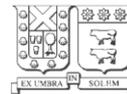
Evaluación

Promedio de Tareas 40%
Presentaciones 30%
Examen 30%

Bibliografía

- [1] Wasserman, L. (2004). All of Statistics: A concise course in statistical inference. Ed. Springer Verlag.
- [2] Casella, G. and Berger, R. L. (2002). Statistical Inference, 2nd. Ed Cengage Learning.
- [3] Bickel, P. J. and Doksum, K. A. (2015). Mathematical Statistics. 2nd. Ed CRC Press.
- [4] Rice, J. A. (2010). Mathematical Statistics and Data Analysis, 3^{era} Ed. Thomson

Elaborado:	H. Allende	Observaciones:
Aprobado:	Departamento de Informática	
Fecha:	Marzo 2016	



Universidad Técnica Federico Santa María Departamento de Informática

Asignatura	Computabilidad y Complejidad Computacional	Sigla	INF-487
Créditos SCT	4	Conocimientos previos: -	
Hrs. Cat. Sem. :	2	Hrs. Ayud. Sem. :	0
		Hrs. Lab. Sem.:	0

Descripción

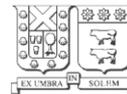
Este curso cubre los fundamentos de las teorías de computabilidad y complejidad computacional, bases de la informática teórica. En el caso de la computabilidad lo que se estudia son los límites de lo que un computador es, en principio, capaz de computar. Para esto se estudian las nociones de lenguajes formales, gramáticas y máquinas abstractas, con énfasis en el modelo de la máquina de Turing, su universalidad de cómputo y la existencia de problemas indecidibles. En el caso de la complejidad computacional lo que se estudian son los límites de lo que, para efectos prácticos, podemos calcular en un computador. Para esto se definen las clases más importantes de problemas (con énfasis en P y NP), la jerarquía de clases en que se insertan, sus problemas prototípicos, y las reducciones que permiten clasificar a otros problemas y con ello tener cotas para su dificultad.

Objetivos

Al aprobar la asignatura el alumno conocerá las nociones fundamentales de la computabilidad y la complejidad computacional, entendiendo la frontera entre lo indecidible y lo decidable, y dentro de esto último, la frontera entre lo tratable y lo intratable. Sabrá identificar cuando un problema tiene solución algorítmica, y demostrarlo. También sabrá identificar y demostrar cuando un problema tiene solución eficiente, y demostrarlo. Conocerá los problemas prototípicos de las principales clases de complejidad, y sabrá reducir problemas para clasificar su dificultad en función del tiempo y espacio que su resolución requiere.

Contenidos

- Máquinas de Turing. Lenguajes recursivos y recursivos enumerables; funciones computables.
- Máquina de Turing universal. Tesis de Church-Turing.
- Problemas indecidibles, teorema de Rice y funciones no computables.
- Complejidad algorítmica.
- Clases de complejidad computacional. Reducciones eficientes y completitud.
- P y NP. C-Value y C-Sat; teorema de Cook-Levin. Otros problemas NP-completos. CoNP, EXP, NEXP.
- Diagonalización y teoremas de jerarquía. Oráculos.



- Complejidad espacial. PSPACE y NL.
- Complementos: definición de otras clases importantes (BPP, NC, BQP); dificultad de aproximación y teorema PCP.

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. Adicionalmente habrá interrogaciones y evaluaciones escritas.

Evaluación

Tareas 50%
Interrogaciones y examen 50%

Bibliografía

- Arora & Barak, Computational Complexity: A Modern Approach. CUP, 2009.
- Sipser, Introduction to the Theory of Computation. Course Technology, 2005.
- Moore & Mertens, The Nature of Computation. OUP, 2011.
- Papadimitriou, Computational Complexity. Addison Wesley, 1994.

Elaborado:	A. Moreira	Observaciones:
Aprobado:	Departamento de Informática	
Fecha:	Marzo 2016	





Universidad Técnica Federico Santa María Departamento de Informática

Asignatura	Tópicos Actuales en Ingeniería de Software		Sigla	INF-488
Créditos SCT	4	Conocimientos previos: -		
Hrs. Cat. Sem. :	2	Hrs. Ayud. Sem. :	0	Hrs. Lab. Sem. : 0

Descripción

La Ingeniería de Software es la disciplina que se ocupa de la construcción sistemática, eficaz y eficiente de sistemas de software. Casi toda la Ciencia de Computación existe sobre la programación, pero el desarrollo de sistemas que satisfagan expectativas conflictivas y ambiguas de terceros es mucho más complejo, y la creación y mantención de equipos y organizaciones que hagan esto sistemáticamente es el foco de esta disciplina. Los participantes serán expuestos a las corrientes históricamente principales de investigación de la comunidad, y un panorama de la investigación actual que incluye sub-disciplinas, temas transversales, y enfoques inter- disciplinarios.

Objetivos

Comprender tendencias de investigación en Ingeniería de Software.

Analizar y redactar reviews sistemáticos del estado del arte de Ingeniería de Software. PAGE67

Contenidos

1. La comunidad de investigación en Ingeniería de Software

Motivaciones y temas iniciales; miembros destacados; hitos en la literatura; foros, conferencias y revistas "core".

2. Sub-disciplinas y temas recurrentes

La comunidad de Ing. Sw se divide en sub-comunidades, interesadas en problemas específicos y con criterios de validez para problemas y soluciones propuestos. El curso explorará:

3. Ingeniería de Requisitos: funcional/escenarios/goal-oriented
4. Diseño y Arquitectura de Software: ADLs, Architectural Knowledge
5. Sub-comunidades específicas a atributos de calidad: Seguridad, Escalabilidad, otras.
6. Desarrollo de Software: procesos/prácticas, agilidad, DevOps
7. Quality Assurance (QA): testing
8. Software Process Improvement (SPI): modelos de madurez, gestión de conocimiento
9. Ecosistemas de Software



3. Temas transversales a sub-disciplinas

Existen varios temas que son abordados en varias sub-disciplinas:

1. Variabilidad: incluyendo Model-Driven Engineering
2. Rationale y trazabilidad: incluyendo Design Rationales

4. Enfoques inter-disciplinarios

1. Search-based Software Engineering (SBSE): aplicaciones de técnicas de Inteligencia Computacional
2. Empirical Software Engineering (ESE): aplicaciones para generación de conocimiento confiable
3. Ontologías, Social Network Analysis, y otras técnicas que tratan a la Ing.Sw misma como objeto de conocimiento.

Metodología

El curso consistirá en:

1. Clases lectivas normales, con abundante material complementario previo. controles periódicos de lectura, e informes de lo tratado en cada clase.
2. Desarrollo de ensayos temáticos por los participantes.
3. Charlas de investigadores visitantes, reconocidos en la disciplina.

Evaluación

- Controles (de lectura y contenido) 40%
- Ensayos temáticos 30%
- Informes de clases/charlas 10%
- Propuesta (no accionable) de posible investigación 20%

Bibliografía

Básica.

- "Towards general theories of software engineering". Special issue of Science of Computer Programming. Pontus Johnson, Mathias Ekstedt, Michael Goedicke, Ivar Jacobson. Science of Computer Programming, 101 (2015) pp 1–5. DOI: 10.1016/j.scico.2014.11.005
- SWEBOK v3.0 - Guide to the Software Engineering Body of Knowledge. Pierre Bourque and Richard Fairley (Eds.). IEEE Computer Society (2014). ISBN: 978-0-7695-5166-1.
- Perspectives on the Future of Software Engineering: Essays in Honor of Dieter Rombach. Jürgen Münch and Klaus Schmid (Eds.). Springer-Verlag Berlin Heidelberg (2013). DOI: 10.1007/978-3-642-37395-4

Recomendada.

- Actas de ICSE (International Conference in Software Engineering) • Actas de ASE (I.C. on Automated Software Engineering), RE (I.C. on Requirements Engineering), ECSA (Euro.C. on Software Architecture), EuroSPI (Euro.C. on Software Process Improvement), y otras conferencias temáticas de similar categoría
- Revistas top: IEEE Transactions in Software Engineering (TSE), ACM Transactions on



Software Engineering and Methodology (TOSEM), IEEE Software, ACM Sigsoft Notes on Software Engineering (SEN)

- Revistas específicas: Journal of Requirements Engineering, Journal of Systems and Software, Science of Computer Programming, otras

- Revistas generales: IEEE Computer, Communications de la ACM, IT Pro

Elaborado:	H. Astudillo	Observaciones:
Aprobado:	Departamento de Informática	
Fecha:	Marzo 2016	



Asignatura	Tema Especial en Informática: Computación Peer to Peer	Sigla	INF-490
Créditos SCT	7	Conocimientos previos: Sistemas Distribuidos, Redes, Sistemas Operativos	

Descripción

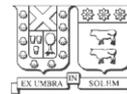
Este curso es un curso avanzado con énfasis en la propia exploración e investigación. La evaluación se realizará con dos certámenes y un proyecto. El proyecto está orientado a diseñar e implementar un servicio a gran escala usando técnicas Peer to Peer.

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.

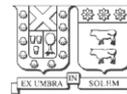
Bibliografía

- [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
Departamento de Informática

Asignatura	Tema Especial en Informática: Arquitecturas Empresariales	Sigla	INF-490
Créditos SCT	7	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

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

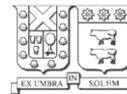
Al aprobar la asignatura el alumno será capaz de:

- 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: Tránsito 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

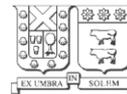
Evaluación

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 at al, Harward Business School Presss, 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:
Aprobado:	Depto. de Informática	Actualización: Diciembre 2012
Fecha:	15711/2010	



Asignatura	Tema Especial en Informática: Seminario de Sistemas Distribuidos	Sigla	INF-490
Créditos SCT	7	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%

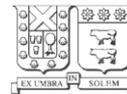
Bibliografía



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	





Departamento de Informática

Asignatura	Tema Especial en Informática: Astro- Informática	Sigla	INF-490
Créditos SCT	7	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

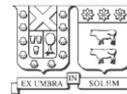
- Explicar movimientos terrestres y celestes.
- Realizar observaciones en telescopios virtuales (ópticos y radio).
- Conocer estándares de Observatorios Virtuales (OV).
- Explicar distintos tipos de observaciones astronómicas (óptica, radio, alta energía) y su relación con la informática.
- Crear implementaciones 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

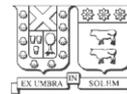
C1 = Nota Certamen 1; C2 = Nota Certamen 2; TA = Nota Talleres; E = Nota Exposición; P = Nota Participación.

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 Sekoranja, 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	



Asignatura	Tema Especial en Informática: Ingeniería de Software Experimental	Sigla	INF-492
Créditos SCT	7	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 Weslén. 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

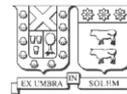
Conferencias

- [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	



Asignatura	Tema Especial en Informática: Mejoramiento de Procesos de Software	Sigla	INF-493
Créditos SCT	7	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, CMMI+ 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 Source, 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
Departamento de Informática

ASIGNATURA: Tema Especial en Informática: Métodos Numéricos Computacionales	SIGLA: INF-510	
Prerrequisitos: Lenguajes de Programación, Cálculo, Algebra Lineal Numérica, Computación Científica	Créditos SCT: 8	
Horas Semanales Cátedra: 4	Horas Semanales Ayudantía:	Horas Semanales Lab : 2

OBJETIVOS: Discutir el uso acoplado de herramientas clásicas y modernas en la computación científica. Dejando al estudiante capacitado para:

- Evaluar de diferentes algoritmos avanzados para la resolución de sistemas de ecuaciones lineales, comprender ventajas y desventajas de cada uno de los algoritmos propuestos y entender técnicas especiales recientes para mejorar su desempeño.
- Discutir, analizar y proponer aproximaciones a problemas descritos por ecuaciones diferenciales.
- Proponer mejoras, discutir y evaluar de métodos rápidos en Computación Científica.

CONTENIDOS:

1. Visión general del área
2. Algebra Lineal Computacional
 - a. Matrices: Completas, Ralas y Libres.
 - b. Gradiente Conjugado
 - c. GMRes: Reiniciación, Precondicionadores y Convergencia
 - d. Métodos de Precondicionadores
3. Introducción a Métodos de Perturbación
 - a. Perturbación Regular
 - b. Perturbación Singular
 - c. Análisis de Capa de Frontera
4. Valores Propios, Funciones Propias y Ecuaciones Integrales.
 - a. Expansiones Ortogonales
 - b. Ecuaciones Integrales
 - c. Funciones de Green
5. Diferencias Finitas y Métodos Espectrales
 - a. Introducción
 - b. Forma Matricial de Diferencias Finitas
 - c. Mallas Periódicas: DFT y la FFT
 - d. Suavidad y Precisión Espectral
 - e. Matrices de Diferenciación de Chebyshev
 - f. Problemas de Valor de Frontera
 - g. Problemas de Valor Inicial y Método de las Líneas
6. Tópicos Especiales (Si el tiempo lo permite)
 - a. Introducción a Métodos rápidos



- b. Transformada Rápida de Fourier
- c. Método Rápido de Multipolos en 2D

METODOLOGÍA DE TRABAJO:

Clases expositivas con participación del profesor y de los estudiantes. Discusión en grupos.

SISTEMA DE EVALUACIÓN:

30% Tareas, 40% Exámenes y 30% Monografía (Trabajo Final)

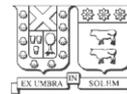
Nota: Consideración especial, dependiendo de la cantidad de estudiantes, el examen será take-home con exposición individual.

BIBLIOGRAFÍA:

1. Numerical Analysis, Second Edition, Timothy Sauer, Pearson, 2011, ISBN: 0321783670.
2. Meshfree Approximations Methods with MatLab, Gregory E. Fasshauer, World Scientific Publishing Company, 2007, ISBN: 9812706348.
3. Applied Mathematics, J. David Logan, Third Edition, Wiley, 2006, ISBN: 0471746622.
4. Iterative Methods for Sparse Linear Systems, Yousef Saad, Second Edition, SIAM, 2003, ISBN: 9780898715347.
5. Spectral Methods in MatLab, Lloyd N. Trefethen, SIAM, 2000, ISBN: 0898714656.
6. Applied Numerical Linear Algebra, James W. Demmel, SIAM, 1997, ISBN: 0898713897.
7. Numerical Linear Algebra, Lloyd N. Trefethen and David Bau, III, SIAM, 1997, ISBN: 0898713617
8. Domain Decompositions Methods, Barry Smith, Petter Bjorstad and William Gropp, Cambridge University Press, 1996, ISBN: 0521602866.

ELABORADO	Claudio Torres López	OBSERVACIONES:
APROBADO		
FECHA		

ACTUALIZADO	27 de junio de 2013	OBSERVACIONES:
APROBADO		
FECHA	12 agosto 2013	



Asignatura	Compresión de Texto		Sigla	INF-520
Créditos SCT	8	Conocimientos previos: Estructuras de datos		
Hrs. Cat. Sem. :	3	Hrs. Ayud. Sem. :	0	Hrs. Lab. Sem. : 0

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

1. Entregar los fundamentos de teoría de la información relacionados a compresión de secuencias de texto sin pérdida.
2. Estudiar las familias principales de compresores de texto.
3. 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

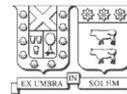
- [6] T. Cover and J. Thomas. Elements of Information Theory. Wiley, 2nd ed, 2006.
- [7] T. Bell, J. Cleary, and I. Witten. Text Compression. Prentice-Hall, 1990.
- [8] I. Witten, A. Moffat, and T. Bell. Managing Gigabytes. Morgan Kaufmann, 2nd ed, 1999.
- [9] D. Salomon. Data Compression: The Complete 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 por:	Diego Arroyuelo	Observaciones:
Fecha Emisión:	11/01/2012	
Fecha Aprobación por Comité I&P:	16-01-2012	



Asignatura	Evaluación de Arquitectura de Software	Sigla	INF-524
Créditos SCT	8	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:

- 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 modelos de calidad para arquitecturas de software
- distinguir los problemas de arquitectura de software de los que no lo son, y explicar articuladamente las nociones básicas de la disciplina
- 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

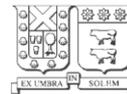
- Lectura sistemática de artículos de la literatura científica, reciente y/o fundamental.
- Talleres de técnicas específicas (e.g. árboles de utilidad).

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).

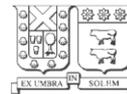


- [5] "Standard quality model to design software architecture". F.Losavio. Journal of Object Technology, 1(4), 2002, pp.165-178.
- [6] "Tools and Methods for Evaluating the Architecture". Software Engineering Institute, Carnegie Mellon University.
<http://www.sei.cmu.edu/architecture/tools/evaluate/>.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

Asignatura	Hot Topics in Operating Systems	Sigla	INF-532
Créditos SCT	8	Conocimientos previos: Operating Systems, Distributed Systems	
Hrs. Cat. Sem. : 2	Hrs. Ayud. Sem. : 0	Hrs. Lab. Sem. : 0	

Descripción

Los sistemas operativos representan una base indispensable para el uso de cualquier computador. Permiten usar de forma equitativa los recursos del computador a través de un conjunto de servicios para las aplicaciones y los usuarios. Los sistemas operativos han registrado una evolución muy importante durante los últimos años, para proveer servicios en adecuación con el avance de la tecnología de los componentes que se usan en los computadores.

Objetivos

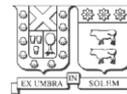
1. Comprender los conceptos fundamentales de los nuevos sistemas operativos.
2. Analizar y comprender el estado actual de la investigación en el tema de los sistemas operativos.
3. Seleccionar y profundizar uno de los conceptos nuevos de los sistemas operativos del futuro.

Contenidos

- Estudio de los nuevos conceptos de los sistemas operativos del futuro.
- Concepto de Virtualización de S.O. Y aspectos de seguridad.
- Concepto de Virtual Virtual Machine (VVM)
- Gestión de hardware de última generación (Procesadores Multi-core, Hot-Plug CPUs, ...).
- Seguridad de los sistemas operativos.

Metodología

- Clases de tipo seminario con discusión y análisis (1h30 semanal).
- La primera parte del curso consiste en la lectura de artículos de referencia sobre los últimos conceptos en los sistemas operativos modernos.
- Para el resto del curso, cada participante deberá seleccionar un tema para profundizarlo y revisar los artículos y libros relevante sobre este tema.
- Cada alumno deberá entregar un breve informe describiendo su avance en el tema (uno cada 2 semanas).
- La evaluación final se hará con un artículo sobre el tema de investigación elegido por el alumno, y con una presentación en el formato que se usa para las conferencias



internacionales de investigación

Evaluación

La evaluación se hará según los siguientes porcentajes:

- Informes : 25%
- Artículo Final : 45%
- Presentación oral : 20%
- Participación : 10%

Bibliografía

1. Ian Piumarta and Bertil Folliot and Lionel Seinturier, Highly configurable operating systems : the VVM approach, In Proceedings of the 3rd ECOOP Workshop on Object-Orientation and Operating Systems (OOOSWS), Cannes, France, June, 2000.
2. Yuan Xu, A Study of Scalability and Performance of Solaris Zones, <http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-8549>.
3. Haifeng Xue and Sihang Qing and Huanguo Zhang, XEN virtual machine technology and its security analysis, Wuhan University Journals Press, 2007.
4. Jeff Dike, User Mode Linux, Prentice Hall, 2006.
5. Yaozu Dong, Xen and Intel Virtualization Technology for IA-64, Proceedings of Gelato ICE: Itanium Conference and Expo: Spotlighting Linux on Itanium-based, Platforms, October 1-4, 2006, Biopolis, Singapore, 2006.
6. Jon Crowcroft and Keir Fraser and Steven Hand and Ian Pratt and Andrew Warfield, The Inevitability of Xen, USENIX Association newsletter, 30(4), August 2005.
7. Intel Virtualization, <http://download.intel.com/technology/magazine/archive/tim0507.pdf>.

Elaborado:	Xavier Bonnaire	Observaciones:
Aprobado:	Depto. Informática	
Fecha:	31-07/2007	



Asignatura	Programación Paralela Aplicada Avanzada	Sigla	INF-534
Créditos	8	Conocimientos previos: Sistemas Distribuidos, Redes, Sistemas Operativos	
SCT			

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



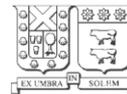
proyecto incluye:

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 SCT	8	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.

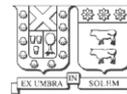
Evaluación

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 Departamento de Informática

Asignatura	Diseño Avanzado de Algoritmos		Sigla	INF-564
Créditos SCT	8	Conocimientos previos: Estructuras de Datos		
Hrs. Cat. Sem. :	4	Hrs. Ayud. Sem. :	0	Hrs. Lab. Sem. : 0

Descripción

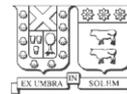
. El curso está orientado a desarrollar los conocimientos y habilidades asociadas al diseño avanzado de algoritmos. Es una asignatura auto contenida en el área. En ella se tratan las técnicas básicas de análisis y diseño de algoritmos y además se profundiza en aspectos avanzados de diseño como algoritmos aproximados, aleatorizados, algoritmos paralelos y algoritmos en línea.

Objetivos

- Analizar y diseñar algoritmos que permitan resolver problemas computacionales.
- Evaluar la pertinencia de algoritmos en la resolución de problemas computacionales.
- Potenciar la capacidad analítica en la resolución de problemas en la informática.

Contenidos

- Técnicas de Análisis de Algoritmos
 - Notación asintótica, análisis caso promedio, análisis amortizado, funciones generatrices.
- Técnicas Básicas de Diseño de Algoritmos
 - Divide y vencerás: MergeSort, conteo de inversiones, producto de enteros, *convex hull*, convolución y FFT.
 - Decrece y vencerás: Búsqueda con poca información (*galloping search*), búsqueda interpolada, búsqueda con recursos acotados. QuickSelect.
 - Programación dinámica: mínimos cuadrados segmentado, subsets sum, mochila y extensión a RNA, alineación de secuencias, producto matricial.
- Algoritmos en línea
 - Análisis de competitividad.
 - Problemas: predicción de demanda (productos, páginas en OS), predicción de valores.
- Algoritmos Aproximados
 - P/NP: reducciones de tiempo polinomial, SAT, coloreo de grafos, vertex cover.
 - Diseño con cotas de aproximación: balance de carga, set cover, center selection, disjoint paths.
 - Búsqueda local: metrópolis, *simulated annealing*, dinámica y equilibrio de Nash.
- Algoritmos Aleatorizados
 - Tipología: Montecarlo, Las Vegas, ejemplos.



- Cotas de Chernoff, balance de carga, ruteo, caching.
- Paralelismo: modelo PRAM, ejemplos.

Metodología

La asignatura se desarrollará con clases expositivas de 90 minutos. Se exponen los conceptos y se desarrollan problemas en clases.

Evaluación

3. 2 certámenes 40% cada uno
4. 2 tareas 20% en total

- Nota mínima en promedio de tareas: 55

Bibliografía

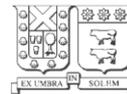
Básica

10. Levitin, A. (2011). Introduction to the design and analysis of algorithms, 3rd Ed., Pearson.
11. Kleinberg, J., Tardos, É. (2013). Algorithms design, 1st Ed., Pearson.

Recomendada

1. Cormen, Th., Leiserson, Ch., Rivest, R., Stein, C. (2009). Introduction to algorithms, 3rd Ed., MIT Press.

Elaborado:	Diego Arroyuelo y Marcelo Mendoza	Observaciones:
Aprobado:	Departamento de Informática	
Fecha:	Marzo de 2016	



Universidad Técnica Federico Santa María Departamento de Informática

Asignatura	Metodología de la Investigación		Sigla	INF-565
Créditos SCT	8	Conocimientos previos: -		
Hrs. Cat. Sem. :	4	Hrs. Ayud. Sem. :	0	Hrs. Lab. Sem. : 0

Descripción

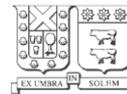
El curso está orientado a desarrollar los conocimientos y habilidades asociadas a la investigación científica. En él se tratan los temas clásicos de metodología de la investigación, y temas actuales que los investigadores deben necesariamente enfrentar: tales como habilidades de búsqueda y selección de información, habilidades comunicativas (escritas y orales), métodos de diseño de experimentos y validación elaboración de informes científicos y desarrollo de proyectos.

Objetivos

Concebir, modelar, diseñar, evaluar e implementar alternativas de soluciones tecnológicas informáticas, originales a partir del análisis de problemas específicos abiertos

Contenidos

- i. Epistemología de la ciencia.
 1. Introducción: tipos de ciencias (natural, artificial, exactas, etc.), Sociología de la ciencia: disciplinas y comunidades, praxis científica, rol de la publicación.
 - ii. Diversas corrientes en filosofía de la ciencia.
 1. Empirismo.
 2. Positivismo
 3. Popper y falsacionismo.
 4. Kuhn y paradigmas.
 5. Lakatos y programas de investigación.
 6. Feyerabend y la argumentación contra el método.
 7. Perspectiva bayesiana de la ciencia.
 - iii. Tipos de Inferencia: inferencia deductiva , inductiva y abductiva.
 - iv. La investigación científica.
 - v. La investigación: definición, características, formas y tipos.
 - vi. Método científico tradicional: elementos, etapas, características.
 - vii. La ética en la investigación científica.
 - viii. Investigación empírica en ciencias de la computación.
 1. El proceso de la investigación empírica.
 2. Tipos de investigación empírica: métricas, surveys, estudios de casos, cuasi-experimentos, experimentos y simulación.



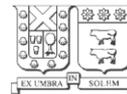
- ix. La Investigación científica como empresa cooperativa.
 - 1. Gestión de proyectos de investigación.
 - x. Formulación de proyectos: formulación, preguntas de la investigación y justificación de la investigación.
 - 1. Definición del problema, hipótesis de trabajo, objetivos, metodología y plan de trabajo, divulgación de los resultados.
 - 2. Elaboración del marco teórico: revisión de la literatura, tipos de fuentes bibliográficas y recursos para la búsqueda.
 - 3. Experimentación: unidad experimental, factor, tratamientos, bloqueos, variables dependientes, variables independientes, variables de bloqueo, sujetos, hipótesis, validación y replicación.
 - xi. Experimentos computacionales.
 - xii. Principios del diseño experimental.
 - xiii. Experimentos en informática: diseño, ejecución, interpretación, reporte.
 - xiv. Análisis de datos.
 - 1. Análisis estadístico de datos.
 - 2. Análisis descriptivo.
 - 3. Hipótesis estadística.
 - 4. Análisis paramétrico y no paramétrico.
 - 5. Métodos de reconocimiento de patrones.
 - 6. Conjuntos de validación
 - 7. Validación Cruzada
 - xv. Relación entre hipótesis y evidencia: el problema de la validación de la hipótesis.
 - xvi. Investigación reproducible e irreproducible
 - xvii. Elaboración de informes científicos.
 - xviii. Resultados intermedios del proceso científico: reporte técnico, artículo (workshops, conferencia, revista), tesis.
 - xix. Desarrollo de la argumentación.
 - xx. 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 proyecto de investigación.

Evaluación

Tareas (Análisis de Artículos de Investigación) 30%
Escritura de un artículo científico 30%
Presentación Proyecto de Investigación 40%



Bibliografía

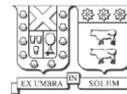
Básica

- T. Bartz-Beielstein “Experimental Research in Evolutionary Computation” Ed. Springer 2010.
- S. Bem, H. L. de Jong: *Theoretical Issues in Psychology: An Introduction (2nd ed.)*. Sage Publications (2006).
- Kai-Tai Fang, Runze Li and A. Sudjianto, T. “Design and Modeling for Computer Experiment” Ed. Chapman Hall/CRC Springer 2006.
- Thomas S. Kuhn: *The Structure of Scientific Revolutions* Ed. [University of Chicago Press](http://www.press.uchicago.edu) 1962. ISBN 9780226458113.
- Karl R. Popper: *The logic of scientific Discovery* Ed. Publisher: Routledge, 2002.
- Linda C N Fan & Angel O K Law (2012) An Assessment of Moral Judgment of Construction Professionals in Hong Kong, HKIE Transactions, 19:3, 17-24: <http://dx.doi.org/10.1080/1023697X.2012.10668993>

Recomendada

- 1.-R. A. Singleton, B: C. Straits, and M. M. Straits. “Approachs to Social Researchs. 1988. Second Ed. Oxford University Press.
- 2.-Freeman and F. Loua, *As As Time Goes By: From the Industrial Revolutions to the Information Revolution.*, Oxford University Press, 2001.
- 3.-A. F. Chalmers, “What is this thing called Science” University of Queensland Press, revised edition (1999).
- 4.-K. Hinkelmann, O. Kempthorne: *Design and Analysis of Experiments, Volume I: Introduction to Experimental Design*. John Wiley & Sons (1994).
- 5.-Jennifer Widom, “Tips for Writing Technical Papers”.
<http://infolab.stanford.edu/widom/paper-writing.html>
- Researcher's Bible, (Including contributions by Graeme Ritchie and Peter Ross.) 9 November 2004.
- 6.-Thomas S. Kuhn: *The Structure of Scientific Revolutions* Ed. [University of Chicago Press](http://www.press.uchicago.edu) 1962. ISBN 9780226458113.
- 7.-Karl R. Popper: *The logic of scientific Discovery* Ed. Publisher: Routledge, 2002.

Elaborado:	Héctor Allende y Hernán Astudillo	Observaciones:
Aprobado:	Departamento de Informática	
Fecha:	Marzo de 2016	



Asignatura	Simulación mediante Mallas Geométricas	Sigla	INF-568
Créditos SCT	8	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.

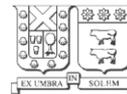
- Conocer las diferencias entre los distintos métodos de aproximación numérica.
- Conocer para que sirve una malla geométrica.
- Programar las técnicas más conocidas

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	



Asignatura	Métodos Cuantitativos en el Procesamiento Computacional de Imágenes	Sigla	INF-569
Créditos SCT	8	Conocimientos previos:	ILI-286 Computación Científica 2 + autorización del Profesor

Descripción

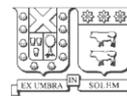
Asignatura centrada en los métodos matemáticos y computacionales aplicados al procesamiento de señales y, especialmente, imágenes. La asignatura introduce los elementos computacionales básicos del procesamiento de señales e imágenes, tales como los modelos vectoriales, matriciales y de campos tensoriales, 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

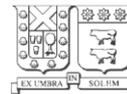
Desarrollo de artículos científicos

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	



Asignatura	Programación con Restricciones	Sigla	INF-571
Créditos SCT	8	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

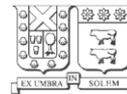


La asignatura se evalúa en base a certámenes (60%), análisis de artículos (20%) y tareas (20%).

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	



Asignatura	Computación Evolutiva	Sigla	INF-572
Créditos SCT	8	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



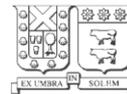
Evaluación

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	



Asignatura	Lógica Borrosa		Sigla	INF-575
Créditos SCT	8	Conocimientos previos: Lenguajes de programación, Cálculo Numérico, Estadística computacional, Álgebra		

Descripción

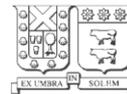
En este curso se dan los fundamentos formales de la Lógica Borrosa, a partir del contexto de la lógica matemática clásica, particularmente el cálculo de predicados. Los (sub)conjuntos borrosos emergen vinculados al lenguaje natural con una intencionalidad semántica: por una parte representan el uso del 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

Introducción; tipos de predicados
 Funciones de E en $[0,1]$
 t-normas, t-conormas y negaciones
 Conjuntos borrosos
 Consistencia lógica e inconsistencia en F (E)
 Modificadores lingüísticos
 Medidas de posibilidad
 Funciones de Agregación
 Principios de Extensión
 Cortes alfa y números borrosos
 Implicaciones borrosas
 Regla Composicional de Inferencia
 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 Departamento de Informática

Asignatura	Redes Neuronales Avanzadas	Sigla	INF-577
Créditos SCT	8	Conocimientos previos: Lenguajes de programación, cálculo numérico, estadística computacional, y redes neuronales artificiales	

Descripción

Este curso provee de un tratamiento de las distintas arquitecturas de redes neuronales artificiales (ANN) y sus aplicaciones a los problemas de aproximación universal, de clasificación y reconocimiento de patrones, tanto estáticos como dinámicos. Se analizan diferentes aplicaciones de las ANN, a la resolución de problemas de modelado matemático en ingeniería, reconocimiento de voz e imágenes, así como sus posibilidades y limitaciones. También se estudian las redes neuroborrosas y redes probabilísticas.

Objetivos

Al término del curso el alumno deberá capaz de:

- Diseñar las diferentes arquitecturas de las ANN.
- Conocer y diseñar ANN con de funciones de activación no monótonas.
- Diseñar y comprender los principales algoritmos para generar ajustar los pesos de las ANN.
- Seleccionar el modelo más adecuado para la resolución de un problema dado.
- Implantar la solución en un computador, incorporando los pre-procesamientos adecuados, la selección de conjuntos de datos y medir el desempeño del modelo.

Contenidos

- .Redes neuronales. Teoremas de aproximación universal.
- .Cotas de la tasa de convergencia del proceso de aprendizaje.
- .Redes neuronales utilizando funciones de activación no monótonas.
- .Diseño constructivo de redes neuronales.
- .Diseño evolutivo de redes neuronales. (algoritmos genéticos).
- .Sistemas neuro-borrosos para la extracción de modelos basados en reglas si-entonces con lógica borrosa, partiendo de datos numéricos de comportamiento.
- .Redes neuronales probabilísticas. Redes Neuronales Bayesianas, Cadenas de Markov Ocultas y Métodos de Monte Carlo.
- .Medidas de desempeño de una red neuronal.



Metodología:

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 será basado en una presentación oral y un reporte escrito de su tema de investigación.

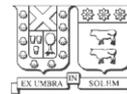
Evaluación:

Tareas e interrogaciones	30%
Proyecto	30%
Examen	40%

Bibliografía

- .Radford M. Neal, Bayesian Learning for Neural Networks, Ed. Springer Verlag, 1996.
- .Bishop C, Neural Networks for Pattern Recognition, Ed. Clarendon Press Oxford,1995.
- .Fa-Long luo and R. Unbehauen, “ Applied neural Networks for Signal Processing” , Ed. Cambridge University Press, 1998
- .B. D. Ripley, “Pattern recognition and Neural Network”, Ed. Cambridge university Press, 1996.
- .T.L. Fine, “Feedforward Neural Network Methodology”, Ed. Springer Verlag, 1999.
- .Revistas: Journal of Machine Learning, IEEE Trans. on Neuronal Networks, Neuronal Computing, Pattern Recognition.

Elaborado:	Héctor Allende	Observaciones:
Aprobado:	Depto. de Informática	
Fecha:	1/8/2004	



Asignatura	Máquinas del Aprendizaje Computacional	Sigla	INF- 578
Créditos SCT	8	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.

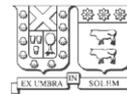
Objetivos:

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:

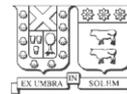
[10] T. Poggio, L. Rossaco, C. Fronger, y G. Canas, «Statistical Learning Theory and Applications, Spring 2012». [Online]. Available: <http://www.mit.edu/~9.520/>. [Accessed: 27-dic-2012].

[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	



Asignatura	Cloud Computing y Big Data	Sigla	INF-582
Créditos SCT	8	Conocimientos previos: Programación en C, Sistemas Operativos, Redes.	

Descripción

La asignatura introduce los conceptos de la computación en la “nube” como una plataforma alternativa para *High Performance Computing* (HPC), desde los elementos que componen un data center hasta los requerimientos de redes de interconexión, capacidades de procesamiento (*computing*) y capacidad de almacenamiento (*storage*). Se presentan las ventajas y desventajas del modelo en la “nube” para resolver aplicaciones técnicas y científicas que utilizan grandes volúmenes de datos (*Big Data*), que no son fáciles de mover. Se pone un especial énfasis en las soluciones costo/efectivas analizando, discutiendo y estableciendo comparaciones entre distintas soluciones de “nubes”: pública y privadas, comerciales y académicas, así como su análisis tipo *Total Cost of Ownership* (TCO).

La asignatura contempla también un acercamiento práctico al tema a través de la construcción de aplicaciones en la nube como servicios, *MapReduce* (Hadoop), PaaS (Azure), mover grandes volúmenes de datos (RDBMS tradicional) o mover los algoritmos hacia los datos (NoSQL).

Este curso está dedicado a la programación paralela en la “nube”. El objetivo del curso es ser capaz de modelar, diseñar, e implementar una solución paralela, cuyos resultados obtenidos deben ser difundidos a través de la escritura de un artículo técnico que debe ser presentado en el curso.

.Objetivos

- .Presentar las arquitecturas modernas para la programación paralela y distribuida.
- .Conocer problemas paralelizables y métodos para resolverlos.
- .Paralelizar aplicaciones con *Message Passing Interface* (MPI).
- .Saber programar una aplicación de *Big Data* para resolver problemas intensivos en cálculo.
- .Presentar resultados en un artículo técnico.
- .Conocer las ventajas y limitaciones de *cloud computing*.
- .Relacionarse con problemas de gran complejidad y computación intensiva.

.Contenidos

- .*Introducción*: Perspectiva histórica, nomenclatura, conceptos básicos, fuentes de grandes volúmenes de datos, ejemplos.
- .*Conceptos Teóricos*: Ley de Amdahl, Ley de Gustafson, *speedup*, Clase de Problemas de Nick (NC), Memoria distribuida y/o compartida, Algoritmos y complejidad con *Big Data*.



.Modelos Teóricos: Modelos secuenciales y paralelos, PRAM, BSP en la nube, MPI, MapReduce. Ejemplos comerciales: *Hadoop* como arquitectura de *cluster*.
.Diseño de algoritmo paralelo: *Mappers and Reducers*, Sistema de Archivo distribuido (HDFS), agregación local, *pairs and stripes*, etc.
.Algoritmos y aplicaciones con Big Data: Algoritmos sobre grafos (grandes), datos de sensores, geo-referenciación, datos de redes sociales, etc.

.Metodología

La metodología usada para el curso serán clases expositivas, investigación bibliográfica, trabajo práctico de programación en aplicación a elegir, escribir artículo, presentación técnica de los resultados.

.Evaluación

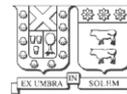
La nota final del curso de calculará de la siguiente forma:

- .Un certamen a mediados del semestre: 30%
- .Redacción de un artículo: 40%
- .Presentación del artículo: 30%.

.Bibliografía

- .Catlett Ch., Gentsch W., Grandinetti L., Joubert G. and J. Vazquez-Poletti (Eds.), *Cloud Computing and Big Data*. IOS Press BV, Netherlands, 2013.
- .De Bosschere K., D'Hollander E., Joubert G., Padua D. and F. Peters (Eds.), *Applications, Tools, and Techniques on the Road to Exascale*. IOS Press BV, Netherlands, 2012.
- .Aguilar, L. *Big Data: Análisis de Grandes Volúmenes de Datos en Organizaciones*. Alfaomega. 2014.
- .Grama A., Gupta A., Karypis G., Kumar V., *Introduction to Parallel Computing*, Second Edition, Addison Wesley, 2003.
- .White, T., *Hadoop, the definitive guide*, O'Reilly, 3rd Ed., 2012.

Elaborado:	Mauricio Solar	Observaciones:
Aprobado:	Comité IP del DI	
Fecha:	23-02-2015	



Asignatura	Sistemas Complejos Discretos	Sigla	INF-584
Créditos SCT	8	Conocimientos previos: Teoría de grafos, lenguajes formales, autómatas	

Descripción

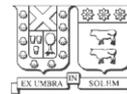
Este curso presenta una panorámica de los sistemas complejos discretos, desde los sistemas más homogéneos (autómatas celulares), donde es posible obtener más resultados teóricos exactos, hasta las redes heterogéneas y masivas, con las que debemos interactuar sin jamás llegar a conocerlas por completo; 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.

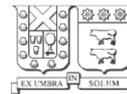
2. Conceptos generales de sistemas complejos discretos. Nociones de sistemas dinámicos, atractores, sensibilidad, no-linearidad, dinámicas caóticas. Fenómenos emergentes, nociones de complejidad.
3. Redes de autómatas: dinámicas sobre grafos de tamaño moderado, con énfasis en redes booleanas. Redes neuronales 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-Rényi). 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
Departamento de Informática

Asignatura	Seminario de Tesis I	Sigla	INF-591
Créditos SCT	20	Prerrequisitos	

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:	Prof. Guía de Tesis	Observaciones: Revisado Abril 2012.
Aprobado:	Depto. de Informática	
Fecha:	01/2002	



Universidad Técnica Federico Santa María
Departamento de Informática

Asignatura	Seminario de Tesis II	Sigla	INF-592
Créditos SCT	20	Prerrequisitos	Seminario de Tesis I

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 doctoral.

Bibliografía

Variable

Elaborado:	Profesor Guía de Tesis	Observaciones: Revisado Abril 2012.
Aprobado:	Depto. de Informática	
Fecha:		