

Universidad Nacional

Córdoba

Argentina

Expte. 03-08-03401.-

CÓRDOBA, 19 MAR 2008

VISTO las presentes actuaciones en las que la Facultad de Matemática, Astronomía y Física, eleva para su aprobación el Convenio Específico a celebrarse entre esta Universidad, la empresa Intel Software de Argentina SA (Intel) y el Gobierno de la Provincia de Córdoba con el objeto de establecer los términos y condiciones respecto a las donaciones que efectuarán las citadas instituciones a esta Casa, a los efectos de financiar cursos de formación para graduados de las carreras de Computación, Sistemas y Afines; atento lo informado por la Subsecretaría de Postgrado dependiente de la Secretaría de Asuntos Académicos a fojas 46 y por la Secretaría de Planificación y Gestión Institucional a fs. 48; lo dictaminado por la Dirección de Asuntos Jurídicos bajo el nro. 38863, y teniendo en cuenta lo dispuesto por las Resoluciones N° 344/99 y 458/03 del H. Consejo Superior.

LA Rectora de la Universidad Nacional de Córdoba

RESUELVE:

ARTÍCULO 1.- Aprobar el Convenio Específico de que se trata, a celebrarse entre esta Universidad, la empresa Intel Software de Argentina SA y el Gobierno de la Provincia de Córdoba, obrante a fs. 1/8 y su Anexo I que corre agregado a fs. 9/43, que en fotocopia forman parte integrante de la presente resolución, y autorizar al Sr. Decano de la Facultad de Matemática, Astronomía y Física, Dr. Daniel E. Barraco Díaz, a suscribirlo en representación de esta Casa.

ARTÍCULO 2.- Previo a la firma del instrumento pertinente, la Facultad de origen deberá modificar el encabezamiento en el sentido que corresponderá ser celebrado a nombre de la UNC, agregar "2º piso" respecto al domicilio legal, llenar los espacios en blanco en los articulados, contar con la conformidad del Dr. Gabriel I. López y agregar los Convenios Marcos oportunamente suscriptos entre las partes, como así también los representantes legales de esas instituciones deberán acreditar adecuadamente la representación que invocan, todo de conformidad a lo dictaminado por la Dirección de Asuntos Jurídicos a fs. 50/vta.

ARTÍCULO 3.- Comuníquese y dése cuenta al H. Consejo Superior.

Ae

[Firma manuscrita]

RESOLUCIÓN Nro. -Mgter. JHON BORETTO SECRETARIO GENERAL UNIVERSIDAD NACIONAL DE CÓRDOBA

489

[Firma manuscrita]

Dra. SILVIA CAROLINA SCOTTO Rectora UNIVERSIDAD NACIONAL DE CÓRDOBA

CONVENIO ESPECIFICO

Entre la Facultad de Matemática, Astronomía y Física (la "FaMAF") de la Universidad Nacional de Córdoba, (la "Universidad"), con domicilio en la Av. Haya de la Torre s/n, Pabellón Argentina, Ciudad Universitaria, Ciudad de Córdoba, representada en este acto por el Señor Decano, Dr. Daniel Barraco, según autorización conferida por la Señora Rectora mediante R.R. N° [__]; Intel Software de Argentina S.A. ("Intel"), con domicilio en la calle 25 de Mayo 267, Piso 3º, Ciudad de Córdoba, representada en este acto por su Director Titular, Sr. Luis Blando; y el Gobierno de la Provincia de Córdoba, República Argentina (la "Provincia" y conjuntamente con la Universidad e Intel, las "Partes" e individualmente cada una de ellas, la "Parte"), con domicilio en Chacabuco 1300, Ciudad de Córdoba, representada en este acto por el Señor Ministro de Industria, Comercio y Trabajo, Dn. Roberto Avalor, se conviene en celebrar el presente convenio específico (el "Convenio Específico") para la implementación de un programa educativo para formación de graduados en sistemas y servicios distribuidos.

CONSIDERANDO

- A. Que, con fecha 24 de abril de 2007, la Universidad e Intel suscribieron un Convenio Marco manifestando su voluntad recíproca de desarrollar diversos emprendimientos educativos conjuntos entre los que se encuentra la formación de graduados.
- B. Que, a su vez, Intel ha celebrado con la Provincia, con fecha 08 de noviembre de 2007, un convenio (el "Convenio") en virtud del cual ambas partes acordaron la realización de sendos aportes de dinero destinados a financiar la investigación y educación técnica en Instituciones Universitarias, actividades que incluyen, entre otras, cursos de formación para graduados.
- C. Que, en consecuencia, las Partes han convenido celebrar el presente Convenio Específico a fin de establecer los términos y condiciones para la financiación e implementación de cursos de formación para graduados.

POR ELLO, EN VIRTUD DE LO PRECEDENTEMENTE EXPUESTO, las Partes acuerdan lo siguiente:

1. **OBJETO.**

El presente Convenio Específico tiene por objeto establecer los términos y condiciones en los que (i) Intel y la Provincia efectuarán sendas donaciones a la Universidad para financiar cursos de formación para graduados y (ii) la Universidad a través de la FaMAF deberá implementar cursos de formación para graduados.

2. **CUESTIONES RELATIVAS A LA ORGANIZACIÓN DE LOS CURSOS.**

- 2.1. Los cursos de formación para graduados tendrán una duración mínima total de un año (360 horas de clases), y se desarrollarán de acuerdo con el calendario académico que apruebe el Honorable Consejo Directivo de la FaMAF, debiendo comenzar a más tardar el 1º de Abril de 2008.
- 2.2. La coordinación académica de los cursos de formación para graduados estará a cargo de la Universidad, con el alcance establecido en el apartado 2.8 del presente Convenio Específico, quien encomendará tales actividades al Dr. Gabriel Infante López, profesor de reconocido prestigio profesional y académico (Se debe contar con su consentimiento en el expediente).
- 2.3. La Universidad a través de FaMAF implementará los cursos de formación para graduados debiéndose ajustar en lo sustancial a lo previsto en el Plan de Estudios (el "Plan de Estudios") que se acompaña como **Anexo I** al presente Convenio Específico.
- 2.4. Los cursos de formación para graduados estarán orientados a graduados de las Carreras de Computación, Sistemas y afines y no tendrán costo alguno para los cursantes, salvo los costos asociados a los materiales de estudio, los que estarán a exclusivo cargo de los cursantes.
- 2.5. La designación de los profesores para los diferentes cursos de formación para graduados deberá ser previamente aprobada por la Secretaría de Postgrado de la FaMAF y por el Honorable Consejo Directivo de la FaMAF, quienes evaluarán sus antecedentes y el cumplimiento de los requisitos académicos.
- 2.6. Los postulantes deberán inscribirse en la Secretaría de Postgrado de la FaMAF. El proceso de admisión y selección de los postulantes estará exclusivamente a cargo de la Universidad de conformidad con la normativa vigente.
- 2.7. La FaMAF tendrá a su cargo el diseño del sistema educativo de enseñanza de los cursos de formación para graduados, la elección de los cursos a dictarse, y sus contenidos, la Metodología de la Evaluación (presencial final), la organización temporal de los cursos, el régimen de correlativas y el horario en el que los mismos se dictarán. Lo que estará sujeto a la revisión y aprobación de la Junta Directiva (tal como se define en la Cláusula 4).
- 2.8 Con una anticipación mínima de 30 días respecto de la fecha prevista para el inicio de cada semestre, la Secretaría de Postgrado de la FaMAF deberá remitir a la Junta Directiva el cronograma de actividades y toda demás documentación referida a la implementación de los cursos para graduados correspondiente al semestre siguiente.
- 2.9. La Junta Directiva evaluará el material suministrado por la Secretaría de Postgrado de

la FaMAF, pudiendo hacer sugerencias y solicitar cambios. El material definitivo será acordado de común acuerdo entre la Junta Directiva y la Secretaría de Postgrado de la FaMAF.

2.10. Las Partes acuerdan que la Universidad tendrá a su exclusivo cargo la evaluación de los cursantes, de conformidad con los exámenes previstos.

2.11. La FaMAF estará a cargo de la entrega de los correspondientes certificados de aprobación de los cursos de aquellos cursantes que aprueben los exámenes y el trabajo final escrito.

3. FINANCIACIÓN DE LOS CURSOS.

3.1. Los cursos de formación para graduados serán financiados por (i) Intel hasta la suma total de U\$S 116.767 (dólares estadounidenses ciento dieciséis mil setecientos sesenta y siete) y (ii) por la Provincia hasta la suma total de U\$S 116.767 (dólares estadounidenses ciento dieciséis mil setecientos sesenta y siete); durante la vigencia de este Convenio (en adelante, las "Donaciones") en los términos de la presente Cláusula. Este monto será afectado a los siguientes ítems : a) honorarios docentes y de administración U\$S 142.000 (dólares estadounidenses ciento cuarenta y dos mil); b) equipamiento y bibliografía U\$S 32.700 (dólares estadounidenses treinta y dos mil setecientos); c) mantenimiento de equipos, insumos y gastos de implementación U\$S 58.834 (dólares estadounidenses cincuenta y ocho mil ochocientos treinta y cuatro).

3.1.1. Las Partes convienen que las Donaciones, serán pagadas en un 100% por Intel a la Universidad, en los plazos indicados en la Cláusula 3.5, debiendo la Provincia reintegrar a Intel el 50% de dichas sumas, correspondiente a la proporción de los aportes por ella comprometida, (los "Reintegros") dentro de los sesenta (60) días siguientes a ser fehacientemente notificada por Intel de la solicitud de reintegro con la debida acreditación de su efectivización mediante la correspondiente documentación respaldatoria. Intel y la Provincia reconocen que el trámite de pago de los Reintegros obedecerá a las disposiciones del Convenio.

3.2. Al respecto, las Partes acuerdan que las Donaciones son efectuadas con cargo a la Universidad, en los términos del artículo 1826 del Código Civil de la República Argentina, de acreditar fehacientemente ante Intel y la Provincia, según se establece en la Cláusula 4.4, a entera satisfacción de estas últimas, que todo el dinero de las Donaciones serán destinadas íntegramente a financiar los cursos de formación para graduados

3.3. En este mismo acto, la Universidad acepta expresamente las Donaciones en los

términos establecidos precedentemente y se obliga a aplicar tales fondos a la implementación de los cursos de formación para graduados.

3.4. Intel, hará efectivas las Donaciones en seis cuotas que serán abonadas mediante una única transferencia bancaria en los plazos establecidos en la Cláusula 3.5. En consecuencia, el total de las cuotas a ser pagadas a la Universidad será igual a la suma de U\$S 233.534 (dólares estadounidenses doscientos treinta y tres mil quinientos treinta y cuatro).

3.5. El monto de la primera de las cuotas será de U\$S 54.699 (dólares estadounidenses cincuenta y cuatro mil seiscientos noventa y nueve) y será efectivizado dentro de los treinta (30) días corridos de la firma del presente Convenio Específico. Las cinco cuotas restantes, cada una por un monto de U\$S 35.767 (dólares estadounidenses treinta y cinco mil setecientos sesenta y siete) serán efectivizadas dentro de los treinta (30) días previos al inicio de cada uno de los semestres lectivos, comenzando a partir del segundo semestre lectivo, y a condición de lo previsto en la Cláusula 4 del presente Convenio Específico.

3.6. A los fines de hacer efectivas las Donaciones, Intel girará los fondos correspondientes a las cuotas previstas en el apartado 3.4 de la presente Cláusula a la cuenta corriente N° [_____] abierta en el Banco [_____] a nombre de [_____] , debiendo la Universidad entregar los correspondientes recibos a Intel dentro de las veinticuatro (24) horas siguientes a la acreditación de los fondos en la referida cuenta bancaria.

3.7. Cualquier costo derivado de la implementación de los cursos de formación para graduados que supere los montos de las Donaciones será asumido exclusivamente por la FaMAF, no existiendo obligación alguna de Intel ni de la Provincia de asumir costo alguno que resulte superior a las sumas comprometidas para las Donaciones.

4. LA JUNTA DIRECTIVA.

4.1. La FaMAF deberá rendir cuentas acerca de la efectiva aplicación de las sumas correspondientes a las Donaciones a la implementación de los cursos de formación para graduados ante una junta directiva (la "Junta Directiva") integrada por los representantes de Intel y la Provincia, en los términos descritos en el Convenio. 4.2. A tales fines, la FaMAF deberá presentar, dentro de los [_____] días siguientes a la finalización de cada semestre, ante la Junta Directiva la documentación que acredite la aplicación de las Donaciones a la financiación de los cursos de formación para graduados.

4.3. La Junta Directiva tendrá amplias facultades para solicitar a la FaMAF la exhibición de la documentación pertinente y auditar los gastos incurridos por la Universidad en la implementación de los cursos de formación para graduados en cualquier momento durante

la vigencia del presente Convenio Específico.

4.4. En el caso que la Junta Directiva detectase irregularidades en la aplicación de las sumas correspondientes a las Donaciones, deberá intimar fehacientemente a la FaMAF para que en el término de cinco (5) días presente su descargo y/o documentación faltante. Si vencido dicho plazo la FaMAF no contestara la intimación o habiéndolo hecho su respuesta no fuera satisfactoria, a exclusivo criterio de la Junta Directiva, la mora se producirá en forma automática y de pleno derecho, e Intel y la Provincia podrán: (i) abstenerse de efectivizar las cuotas pendientes de las Donaciones, si aun no hubieran sido desembolsadas en su totalidad; y (ii) rescindir el presente Convenio Específico y revocar total o parcialmente las Donaciones.

5. OBLIGACIONES DE LAS PARTES.

5.1. Intel y la Provincia tendrán a su cargo: (i) la financiación de los cursos de formación para graduados en los términos previstos en la Cláusula 3 del presente Convenio Específico. En tal sentido, Intel y la Provincia asumen las obligaciones de pago en forma indistinta y en proporción a las Donaciones a ser efectuadas por cada una de ellas para la financiación de los cursos de formación para graduados y (ii) la fiscalización, a través de la Junta Directiva, según Cláusula 4, de la efectiva aplicación de las Donaciones a los fines convenidos en este Convenio Específico, bien como aprobación del cronograma de actividades y toda demás documentación referida a la implementación de los cursos para graduados.

5.2. La FaMAF tendrá a su cargo: (a) la organización e implementación de los cursos de formación para graduados de conformidad con los términos del presente Convenio Específico; (b) la rendición periódica de cuentas ante la Junta Directiva según lo previsto en las Cláusulas 4.1 y 4.2 precedentes; (c) el diseño de la metodología educativa, los exámenes, el cronograma de actividades y demás documentación vinculadas a la implementación de los cursos de formación para graduados en los términos de las Cláusulas 2.3 y 2.7 de este Convenio Específico; (d) la obtención de la aprobación de la Junta Directiva respecto del Plan de Estudios de los cursos de formación para graduados y demás material relacionado con la implementación de dichos cursos; (e) la conformación del plantel de profesores que dictará los cursos de formación para graduados conforme lo establecido en la Cláusula 2.5; (f) la admisión de los postulantes para cursar los cursos según Cláusula 2.6; (g) la evaluación de los alumnos conforme Cláusula 2.11; y (h) la entrega a los cursantes de los correspondientes certificados.

5.2. Las Partes manifiestan expresamente que (i) la FaMAF será la única responsable por la

implementación de los cursos de formación para graduados y (ii) ni Intel ni la Provincia asumirán responsabilidad alguna relacionada con la implementación de los cursos. En consecuencia, la FaMAF deberá mantener indemne a Intel y a la Provincia con relación a cualquier reclamo que les fuera cursado por terceros con relación a la implementación de los cursos de formación para graduados.

6. RELACIÓN ENTRE LAS PARTES.

6.1. Las Partes acuerdan que la FaMAF actúa en forma independiente y que tendrá a su exclusivo cargo todas las obligaciones fiscales, impositivas, previsionales y laborales que le resulten de aplicación, debiendo mantener indemnes a Intel y a la Provincia por todo reclamo que pueda surgir con relación a dichas obligaciones.

6.2. Todas las personas designadas por la FaMAF para participar, directa o indirectamente, en la implementación de los cursos de formación para graduados estarán bajo su exclusiva responsabilidad y no existirá obligación alguna respecto de ellas a cargo de Intel y/o de la Provincia.

6.3. La FaMAF deberá mantener indemne a Intel y a la Provincia respecto de cualquier reclamo de sus dependientes, empleados, subcontratistas, autoridades nacionales, provinciales o municipales o cualquier otro tercero, en relación al cumplimiento de sus obligaciones bajo el presente Convenio Específico.

6.4. En ningún caso las estipulaciones previstas en el presente Convenio Específico podrán ser interpretadas como la creación de sociedad alguna entre las Partes ni dará lugar a la creación de relaciones laborales y/o de agencia entre las Partes.

6.5. Ninguna de las Partes podrá actuar en representación de las otras ni obligar a las otras Partes frente a terceros.

7. PLAZO.

El presente Convenio Específico tendrá una duración hasta 31 de diciembre de 2010 y podrá ser prorrogado por sucesivos períodos adicionales de un (1) año si las Partes así lo acordaran y lo instrumentaran por escrito.

8. RESCISIÓN.

8.1. Sin perjuicio de las disposiciones que rigen la relación entre Intel y la Provincia en virtud del Contrato Marco y del Convenio, en el caso que Intel y/o la Provincia incumplieran con sus obligaciones bajo el presente Convenio Específico, la FaMAF deberá

intimar por escrito en forma fehaciente a ambas Partes para que subsanen dicho incumplimiento en un plazo de quince (15) días. Vencido dicho plazo sin que el incumplimiento hubiera sido subsanado, la Universidad podrá rescindir el presente Convenio Específico.

8.2 Intel y la Provincia podrán rescindir el presente Convenio Específico en el supuesto previsto en la Cláusula 4.4 precedente y revocar total o parcialmente las Donaciones.

8.3 Se deja convenido, que las partes podrán rescindir el presente convenio, para el supuesto de que no se contaran con las donaciones suficientes y en tiempo y la forma convenida, en forma previa al inicio de cada año lectivo.

9. **NOTIFICACIONES.**

Todas las notificaciones que deban efectuarse en virtud de este Convenio Específico se cursarán por escrito y serán válidas en la medida en que sean entregadas personalmente en, o remitidas a, los domicilios que se indican a continuación, por carta documento o telegrama, en todos los casos con recibo firmado como acuse de entrega. Las notificaciones también podrán cursarse por fax, pero sólo serán válidas en la medida en que el destinatario confirme su recepción.

Las direcciones postales y números de fax de las Partes son las siguientes:

PARA LA PROVINCIA:

Chacabuco 1300
(5000) Córdoba
Atención: [Sr. Gobernador] [please confirm]
Fax (0351) 468-8447

PARA INTEL:

25 de Mayo 267, Piso 3º, Ciudad de Córdoba
Atención: Señor Luis Blando, Director Titular
c.c.: Dirección Legal de Intel
Fax (011) 4515-4299

PARA LA FAMA F:

Av. Haya de la Torre s/n, Pabellón Argentina,
Ciudad Universitaria, Ciudad de Córdoba
Atención: [_____]
Fax [_____]

10. **DISPOSICIONES VARIAS.**

10.1. Las Partes se obligan a mantener la confidencialidad de toda la información recibida de conformidad o en relación con el presente Convenio Específico, incluyendo la información referente a los valores exactos de las Donaciones, sujeto a la legislación correspondiente.

10.2. Ninguna disposición del presente Convenio Específico podrá ser interpretada una licencia expresa o implícita de cualquier derecho de propiedad intelectual o industrial de propiedad de las Partes.

10.3. Ninguna disposición del presente Convenio Específico podrá ser interpretada de modo tal que se prohíba y/o impida a Intel y/o a la Provincia celebrar con terceros otros convenios y/o contratos de objeto igual o similar al del presente Convenio Específico.

10.4. La FaMAF no podrá ceder total o parcialmente los derechos y obligaciones que le correspondan en virtud del presente Convenio, sin el previo consentimiento de Intel y la Provincia expresado por escrito.

10.5. Las Partes acuerdan que el presente Convenio Específico se regirá por la legislación aplicable de la República Argentina y la Provincia de Córdoba.

10.6. Negociación previa. Las Partes deberán intentar resolver todo reclamo o controversia relacionado con el presente Convenio Específico, su existencia, validez, calificación, interpretación, alcance, cumplimiento, resolución o rescisión (la "Controversia") a través de consultas y negociaciones realizadas de buena fe y en un ámbito de cooperación mutua, tomando en consideración los objetivos del presente Convenio Específico. Tales esfuerzos estarán a cargo de las personas físicas autorizadas para resolver el conflicto en representación de cada una de las Partes, según lo dispuesto a continuación:

En representación de la Provincia, actuará el titular del Poder Ejecutivo Provincial.

En representación de Intel, actuarán Luis Blando y/o quienes lo reemplacen en el futuro.

En representación de la Universidad, actuará [_____].

10.7. Arbitraje. Si las Partes no logran resolver una Controversia de conformidad con lo establecido en el apartado 10.6 precedente en un plazo máximo de 30 (treinta) días corridos, la Controversia se resolverá definitivamente por el Tribunal de Arbitraje, constituido por un integrante designado por cada uno de las partes, y dos elegidos de común acuerdo por las mismas, que funcionará en la ciudad de Córdoba

10.8. Las Partes manifiestan expresamente que el presente Convenio Específico sustituye todas las propuestas previas, verbales y/o escritas, y todos los convenios y/o comunicaciones previos intercambiados por las Partes respecto de los cursos de formación para graduados.

En prueba de conformidad, se suscriben tres (3) ejemplares de un mismo tenor y a un sólo efecto, en la Ciudad de Córdoba, a los [__] días del mes de [____] de 2008.



ANEXO I

Plan de Estudios de la Especialización en Sistemas y Servicios Distribuidos

Facultad de Matemática, Astronomía y Física
Universidad Nacional de Córdoba



Índice

1. Sistemas Distribuidos	5
1.1. Objetivos	5
1.2. Contenidos	5
2. Construyendo Aplicaciones sobre Infraestructuras Distribuidas	7
2.1. Objetivos	7
2.2. Contenidos	7
3. Seguridad	10
3.1. Objetivos	10
3.2. Contenidos	10
4. Ingeniería de Documentos	14
4.1. Objetivos	14
4.2. Contenidos	14
5. Técnicas formales para mejorar la calidad del software	16
5.1. Objetivos	16
5.2. Contenidos de Aprendizaje	16
5.3. Formas Metodológicas de Enseñanza y Aprendizaje	18
5.4. Trabajos Prácticos	19
6. Programación Concurrente en Java	21
6.1. Objetivos	21
6.2. Programa	22
7. Evaluación Final integradora y Trabajo de Tesis	24
7.1. Objetivo	24
7.2. Generalidades	24
7.3. Sobre el pre-proyecto	24
7.4. Evaluación del pre-proyecto	25
7.5. Sobre la evaluación del trabajo especial	25
8. Performance de Bajo Nivel	26
8.1. Objetivos	26
8.2. Programa	26



9. Ingeniería de Performance en Sistemas Distribuidos	28
9.1. Objetivos	28
9.2. Contenidos	28
10. Proceso de Desarrollo de Software	30
10.1. Objetivos	30
10.2. Programa	30
11. Patrones Seguros y Diseño Seguro de Sistemas	32
11.1. Objetivos	32
11.2. Programa	32
12. Seminario de Software Libre	34
12.1. Descripción	34
12.2. Temas	34
12.3. Metodología de trabajo	34
12.4. Evaluación	34



Esquema General y Distribución en Cuatrimestres

Primer Cuatrimestre			
Nombre Materia	Hs.	Docentes	Régimen
Sistemas Distribuidos	60	Matías Cuenca Acuña y Mariano Cilia	obligatoria
Ingeniería de Documentos	60	Gabriel Infante-Lopez y Laura Alonso Alemany	obligatoria
Seminario de Software Libre	20	Ricardo Medel	optativa
Performance de Bajo Nivel	20	Arzhan Kinzhalov	optativa
Programación Concurrente en Java	40	Javier Blanco y Nicolás Wolovick	obligatoria
Patrones Seguros y Diseño Seguro de Sistemas	20	Eduardo Fernández	optativa
Segundo Cuatrimestre			
Nombre Materia	Hs.	Docentes	Régimen
Seguridad	60	Daniel Fridlender y Daniel Penazzi	obligatoria
Construyendo Aplicaciones sobre Infraestructuras Distribuidas	60	Mariano Cilia y Matías Cuenca Acuña	obligatoria
Ingeniería de Performance en Sistemas Distribuidos	20	Samuel Kounev	optativa
Técnicas formales para mejorar la calidad del software	10	Pedro D'Argenio y Nazareno Aguirre	obligatoria
Proceso de Desarrollo de Software	20	Patricio Maller	optativa
Trabajo Final Integrador	20	Gabriel Infante-Lopez	obligatoria



1. Sistemas Distribuidos

Docentes: Matías Cuenca Acuña (responsable) y Mariano Celis.

Duración: 60hs

Régimen: Obligatoria.

1.1. Objetivos

Presentar la problemática de los sistemas distribuidos abarcando temas como diseño, administración, tolerancia a fallas, etc. Introducir el marco teórico en que se basan de dichos sistemas, incluyendo los algoritmos y modelos más usados.

1.2. Contenidos

- Los problemas más comunes: Alta disponibilidad, tolerancia a fallas, distribución de carga, administración.
- Manejo de procesos distribuidos: Migración, exclusión mutua, dead-lock, Distribución y balanceo de carga: Scheduling distribuido, migración de conexiones TCP.
- Mantenimiento del estado distribuido: Memoria distribuida, manejo con transacciones del estado, Comunicación virtual synchronous, orden total y parcial.
- Almacenamiento masivo de datos: RAID, NASD, Replicación de datos, sistemas de archivos distribuidos, bases de datos distribuidas.
- Alta disponibilidad y tolerancia a fallas. Administración de sistemas distribuidos: Técnicas, virtualización.
- Modelos de programación: Multi-threaded, asincrónico, orientado a eventos.
- Diseño de sistemas distribuidos de gran escala. Patrones de uso comunes en sistemas distribuidos: Grid Computing, Peer to Peer, Application Servers.



Bibliografía

Randy Chow, Theodore Johnson. *Distributed Operating Systems & Algorithms*. Addison-Wesley. 1997

William Stallings. *Operating Systems: Internals and Design Principles*, Fifth Edition. Prentice Hall. 2004



2. Construyendo Aplicaciones sobre Infraestructuras Distribuidas

Docentes: Mariano Cilia (responsable) y Matías Cuenca Acuña.

Duración: 60hs

Régimen: Obligatoria.

2.1. Objetivos

Presentar la diversidad, complejidad y los conceptos detrás de la construcción de aplicaciones sobre infraestructuras distribuidas. Esto incluye una revisión de la evolución de las tecnologías y las distintas plataformas; el concepto detrás de la partición de la lógica del negocio; el middleware para las comunicaciones; la inversión del control en la aplicación de servicios básicos (persistencia, transacciones, seguridad, control de acceso, etc.); la arquitectura orientada a servicios (Service Oriented Architecture, SOA); la integración de aplicaciones y negocios.

2.2. Contenidos

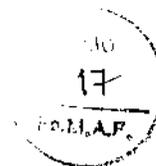
1. Evolución de las aplicaciones. Aplicaciones Cliente/Servidor. El modelo de Cartner. El equilibrio fat client vs. fat server. Middleware. Escalabilidad. Evolución de las Tecnologías Cliente/Servidor.
2. Entendiendo el middleware para comunicaciones. El modelo RPC. Modelos de interacción. Casos de estudio. Comunicación Asíncrona. Middleware Oriented Middleware (MOM). Queuing. Publish/Subscribe. Estrategia de ruteo de notificaciones. Modelos de direccionamiento.
3. Plataformas de objetos distribuidos. CORBA. Transaction Processing Monitors (TPM). Object Transaction Monitors (OTM).
4. Desarrollo basado en componentes. La plataforma J2EE. Application Servers. Server-side Components. Lógica de la aplicación. Enterprise Java Beans (EJBs). Session Beans. Entity Beans. Message-Driven Beans. EJB Patterns. Aspectos de Performance. SPECjAppServer.
5. El modelo de servicios SOA. Analisis y diseño orientado a servicios. WebServices: SOAP, WSDL, UDDI. Composición de servicios. BPEL4WS. Los estándares. Herramientas. Otros modelos.



6. Conceptos de Integración. Integración de datos heterogéneos. Tecnologías para integración. Estilos de integración. Patrones de integración. Enterprise Application Integration (EAI). Business to Business (B2B). Enterprise Service Bus (ESB).
7. Aplicaciones de Internet. Evolución. Web 2.0. Rich Internet Applications (RIA). AJAX. OpenLaszlo. Mixing Services. Frameworks y bibliotecas.
8. Caso de Estudio. Hacia la identificación automática de objetos. Códigos de barra y códigos de barra bidimensionales. Identificadores de radiofrecuencia (RFID tags). Electronic Product Code (EPC). El sucesor del código de barras. AutoID. EPCglobal Network. Ubiquitous and pervasive systems. Web presence model: CoolTown.
9. Revisión de conceptos y tecnologías en contexto. Ciclo de adopción de las tecnologías.

Bibliografía

- P.A. Bernstein, E. Newcomer: *Principles of Transaction Processing for the Systems Professional*, Morgan Kaufmann, 1997.
- J. Gray, A. Reuter: *Transaction Processing: Concepts and Techniques*, Morgan Kaufmann, 1993.
- Q. Maimoudi (editor), *Middleware for Communications*, Wiley, 2004.
- E. Roman, *Mastering EJB and the Java 2 Platform*, Enterprise Edition, 3rd Ed, Wiley, 2004.
- R. Mounson-Haefel, *Enterprise JavaBeans*, 4th edition, O'Reilly, June 2004.
- Orfali, Harkey and Edwards, *Client/Server Survival Guide*, 3rd Edition, J. Wiley, 1999.
- F. Marinescu, *EJB Design Patterns: Advanced Patterns, Processes, and Idioms*, Wiley, Feb 2002.
- Thomas Erl, *Service-Oriented Architecture (SOA): Concepts, Technology and Design*, Prentice Hall, 2005.
- Professional Java Web Services*, Wrox Press, 2002.
- S. Dustdar, H. Gall, M. Hauswirth, *Software-Architekturen für Verteilte Systeme*, (in German) Springer 2003.



G. Alonso, F. Casati, H. Kimo, V. Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer 2004.

Lindhium. *Enterprise Application Integration*. Addison Wesley, 2000.

G. Hohpe, B. Woolf. *Enterprise Integration Patterns: Designing, Building and Deploying Messaging Solutions*. Addison Wesley, 2004.

Ch. Bussler. *Business to Business Integration: Concepts and Architecture*. Springer 2003.

D. Chappell. *Enterprise Service Bus: Theory in Practice*, O'Reilly, June 2004.



3. Seguridad

Docentes: Daniel Fridlender (responsable) y Daniel Penazzi

Duración: 60hs

Régimen: Obligatoria.

3.1. Objetivos

Seguridad es la disciplina que aborda la construcción de sistemas que puedan permanecer confiables en presencia de malicia o error. Estudia las herramientas, los procesos y los métodos necesarios para el diseño, la implementación y la verificación de tales sistemas. Abarca el estudio de las vulnerabilidades de los sistemas, de las diferentes políticas de seguridad, de herramientas criptográficas, de principios de diseño y de métodos formales para establecer la confiabilidad de sistemas.

3.2. Contenidos

1. Introducción

- a) Comportamiento no intencional.
- b) Ataques.
- c) Vulnerabilidades.
- d) Políticas, mecanismos y aseguramiento.

2. Políticas

- a) Propiedades de seguridad (integridad, secreto, disponibilidad).
- b) Variaciones.
- c) Principales.
- d) Políticas de seguridad.
- e) Seguridad multinivel.
- f) No interferencia.
- g) Control de flujo de información.

3. Mecanismos

- a) El modelo de control de acceso.



- b)* Autenticación, autorización, auditoría.
- c)* Implementación: lista de control de acceso, capacidades.
- d)* Grupos y roles.
- e)* Permisos.
- f)* Protección y confinamiento.
- g)* Riesgos comunes.
- h)* Teorías y sistemas generales.

4. Aseguramiento

- a)* Especificación y prueba.
- b)* Mecanismo económico.
- c)* Base computacional confiable.
- d)* Diseño abierto.
- e)* Código abierto.

5. Seguridad en lenguajes de programación

- a)* Safety y security.
- b)* Otros aspectos de seguridad
- c)* Java: control de acceso en Java, verificación de bytecode.
- d)* Proof-carrying code.
- e)* Control del flujo de información.

6. Criptografía

- a)* Encriptación simétrica: métodos, tipos de ataques.
- b)* Stream ciphers, block ciphers.
- c)* Encriptación múltiple.
- d)* Encriptación probabilística.
- e)* Encriptación asimétrica.
- f)* RSA.
- g)* Encriptación híbrida.
- h)* Diffie-Hellman.
- i)* Funciones de hash.



- j)* Código de autenticación de mensaje.
- k)* Aleatoriedad.

7. Protocolos de seguridad

- a)* Generalidades.
- b)* Protocolos de autenticación.
- c)* Protocolo de Denning-Sacco.
- d)* Protocolo Needham-Schroeder de clave compartida.
- e)* Descendientes de Needham-Schroeder.
- f)* Protocolo SSL.

8. Métodos formales para protocolos de seguridad

- a)* El Pi-Calculus aplicado.
- b)* Descripción y ejemplos.
- c)* Análisis automático de protocolos.

9. Principios de diseño de protocolos

- a)* Ejemplos de ataques y moralejas.
- b)* Autenticidad.
- c)* Dos conceptos de autenticidad.
- d)* Autenticación de usuarios.
- e)* Protección del archivo de claves.
- f)* Comunicación segura con claves.
- g)* Infraestructura de clave pública (PKI)
- h)* Autoridades de certificación.
- i)* Ejecutando una autoridad de certificación.
- j)* Expiración y revocación.
- k)* Clave compartida vs. clave pública.
- l)* Jerarquía de autoridades de certificación.

10. Seguridad en redes

- a)* Ataques a redes y defensas.
- b)* Ataques de denegación de servicio.



- c) Detección de ataques.
- d) Firewalls.
- e) Combatiendo spam.
- f) Spam como problema de seguridad.

Bibliografía

Ross Anderson. *Security engineering: A guide to building dependable distributed systems*, 2001.

Matt Bishop. *Computer Security: Art and Science*, 2002.

Trust in Cyberspace Committee on Information Systems Trustworthiness, National Research Council, 1999.

Bruce Schneier. *Applied cryptography*, 1996.

Bruce Schneier. *Secrets & Lies: Digital Security in a Networked World*, 2000.

Kevin D. Mitnick & William L. Simon. *The Art of Deception: Controlling the Human Element of Security*, 2002.

William R. Cheswick, Steven M. Bellovin. *Firewalls and Internet Security*, 1991.

Robin Milner. *Communicating and Mobile Systems: the Pi-Calculus*, 1999.

Eric Rescorla. *SSL and TLS: Building and Designing Secure Systems*, 2000.

Lé Gong, Gary Ellison, Mats Dageforde. *Inside Java 2 Platform Security: Architecture, API Design, and Implementation*, 2003.

Web Security & Commerce, Simson Garfinkel & Eugene H. Spafford, 1997.



4. Ingeniería de Documentos

Docentes: Gabriel Infante-López (responsable) y Laura Alonso Alemany

Duración: 60hs

Régimen: Obligatoria.

4.1. Objetivos

La ingeniería de documentos sintetiza ideas complementarias de sistemas de información y análisis, publicación electrónica, análisis de procesos de negocios e informática de negocios. Su visión unificadora, con el documento como concepto central, nos ayuda a concebir y entender los nuevos modelos de negocios basado en redes. La esencia de la ingeniería de documentos es el análisis y el diseño de métodos que conllevan:

- especificaciones precisas o modelos para la información que el modelo de negocio requiere, y
- reglas por las cuales se coordinan procesos relacionados.

En este curso nos focalizamos principalmente en las tecnologías que hacen posible la ingeniería de documentos: XML, Modelos, Patrones, y Reusabilidad.

4.2. Contenidos

- Introducción y motivación
 - Contexto histórico de los lenguajes de marcado
 - Características básicas de XML
 - Panorámica de aplicaciones y herramientas
 - Unicode e Infosets
- Fundamentos de XML
 - Sintaxis y semántica básicas, DTD
 - XML Schema, modelado de elementos y atributos
 - Namespaces, XPath, Xlink, XPointer



- XSL, XSLT
- Más Schema: Composición, Documentación, Tipos reusables, Derivación simple.
- Tipos complejos y extensibilidad
- Programando XML.
 - SAX
 - DOM, JDOM
 - JAXP, JAXB, JAXM, Emerging Java APIs for XML
- Ingeniería Documental y Arquitectura de Información
 - Formato de Documentos, TEI, DocBook
 - XML en la web, XHTML, CSS
 - RDF, OWL, y la web semántica
 - Diseñando lenguajes a medida
- Aplicaciones empresariales
 - JSP/Servlet
 - Servicio de mensajería, JMS
 - Bases de datos, XInclude, XML Base, XML Id
 - SOAP, WSDL, y Web Services
- Herramientas y proyectos
 - Herramientas libres
 - Herramientas no libres
 - Empresas innovadoras
 - Líneas de investigación activas

Bibliografía

Robert J. Glushko and Tim McGrath. *Document Engineering Analyzing and Designing Documents for Business Informatics and Web Services*. The MIT Press, 2005.

Elliott Rusty Harold, W. Scott. *XML in a Nutshell*, 3rd Edition, 2004.



5. Técnicas formales para mejorar la calidad del software

Docentes: Pedro D'Argenio (responsable) y Nazareno Aguirre

Duración: 40hs

Régimen: Obligatoria

5.1. Objetivos

El objetivo esencial de la materia es lograr que los alumnos se familiaricen con las técnicas más importantes de validación y verificación de software disponibles en la actualidad, que comprendan los fundamentos teóricos que hacen posibles la aplicación de estas técnicas en la práctica, y sepan aplicarlas.

5.2. Contenidos de Aprendizaje

1. El Problema de construir programas correctos. ¿Qué es un programa correcto? Necesidad de las especificaciones de programas. Anotaciones en programas. Revisión de enfoques tradicionales a la validación y verificación de programas. Testing y sus limitaciones. Lógica de Hoare. Programas anotados. Programas correctos por construcción. Transformación de programas. Problemas en la automatización de las técnicas tradicionales de validación y verificación de programas.
2. Model checking. La lógica modal. Definiciones fundamentales. Operadores modales y sus distintas interpretaciones. Sintaxis y semántica de la lógica modal proposicional. Estructuras de Kripke. Relación entre estructuras de Kripke y la semántica de programas concurrentes. Lógicas temporales. La lógica temporal como una lógica modal. Descripción de propiedades relacionadas con el tiempo mediante la utilización de operadores modales. Lógicas de puntos de tiempo y lógicas de intervalos. Nuevos operadores modales de las lógicas temporales. Lógicas temporales proposicionales y de puntos de tiempo. Lógica temporal lineal y lógica temporal ramificada. CTL*, CTL y LTL.
3. Sintaxis y semántica formales de la lógica temporal lineal proposicional. Autómatas de Büchi y su relación con modelos de la lógica temporal lineal proposicional. Autómatas de Büchi y su relación con



fórmulas de la lógica temporal lineal proposicional. Traducción de interpretaciones y fórmulas de la lógica temporal lineal proposicional a autómatas de Büchi. Reducción del model checking en lógica temporal lineal proposicional a un problema en Teoría de Autómatas. Automatización en la verificación de propiedades temporales mediante el model checking. Ejemplos de su uso en la práctica. La concurrencia y las dificultades en la comprensión de programas concurrentes. Utilización de la herramienta SPIN. El lenguaje ProMeLa. Especificación de programas concurrentes en ProMeLa. Caracterización de propiedades de programas concurrentes mediante el uso de lógica temporal lineal. Ejemplos. Limitaciones del model checking en relación a la lógica temporal lineal proposicional. Limitaciones del modelo de representación explícito. Introducción al model checking simbólico. Introducción a software model checking.

4. La lógica como lenguaje formal de especificaciones y su expresividad. Lógica proposicional. Sus ventajas y sus limitaciones en poder expresivo. Reducción del problema de la deducción a SAT solving proposicional. Automatización de la deducción. La lógica de primer orden. Características y poder expresivo. Restricciones para la aplicación de SAT solving para deducción en lógica de primer orden. Representación de diseños de sistemas mediante lógica de primer orden. Algunas extensiones importantes a la lógica de primer orden. La lógica relacional. Descripción de diseños y propiedades estructurales mediante la lógica relacional. El poder de SAT solving de primer orden como una técnica de validación de sistemas. Limitaciones. La herramienta Alloy. Definición de diseños mediante firmas y funciones de transformación de estados. Validación de diseños en Alloy. Utilización en la práctica. Ejemplos.
5. Limitaciones de las técnicas automáticas de validación y verificación de sistemas. Ejemplos de la no aplicabilidad de técnicas automáticas. La deducción sobre la exploración exhaustiva. Deducción semi-automática y asistida por computadora. Algunos cálculos de prueba útiles para la deducción automática. La lógica temporal lineal de primer orden. Comparación entre las lógicas temporales lineales proposicionales y de primer orden y sus respectivos poderes expresivos. Deducción en lógica temporal lineal de primer orden. Combinación de la deducción con mecanismos automáticos de inferencia. Programas concurrentes y su especificación en lógica temporal lineal de primer orden. La herra-



mienta STeP. Especificación axiomática de programas concurrentes. Verificación de programas concurrentes usando STeP. Ejemplos.

6. El ciclo de desarrollo de sistemas de software. Necesidad de herramientas formales en la construcción de sistemas. Los métodos formales a lo largo del proceso de desarrollo de software. Software correcto por construcción. El método B. Fundamentos teóricos. Teoría de Conjuntos de Zermelo-Fraenkel. Caracterización de tipos de datos y construcciones básicas del método B en Teoría de Conjuntos. Especificaciones modulares. La noción de máquina abstracta. Módulos consistentes. Obligaciones de prueba de máquinas abstractas. La noción de refinamiento. Construcción progresiva de implementaciones de máquinas abstractas. Limitaciones del método B en poder expresivo.
7. Aplicación de técnicas formales al testing. Testing basado en modelos. Máquinas de estados finitas como modelos abstractos de software. Modelos directos y modelos inversos. Técnicas para derivación de modelos a partir del software. Generación de casos de prueba a través de model checking y lógicas temporales. Generación de casos de prueba a través de constraint solving. Generación de casos de prueba a través de ejecuciones simbólicas. La herramienta Spec Explorer. Generación de casos de prueba usando Spec Explorer.

5.3. Formas Metodológicas de Enseñanza y Aprendizaje

Debido a que la validación y verificación de software son áreas de las Ciencias de la Computación fuertemente basadas en lógica y matemática discreta, es necesario estudiar con cierto detalle los resultados teóricos que hacen posible la aplicación de las técnicas de validación y verificación presentadas en esta asignatura. Sin embargo, si bien nos preocuparemos por intentar lograr que los alumnos comprendan los resultados teóricos subyacentes a las distintas técnicas presentadas, se pondrá especial énfasis en la aplicación de las principales técnicas de validación y verificación de software en problemas concretos. Las clases serán por lo tanto teórico-prácticas. Se fomentará el uso de herramientas de software para la asistencia en la verificación (las técnicas elegidas cuentan, todas ellas, con herramientas de soporte).

Los trabajos prácticos obligatorios, cuya aprobación es parte del requisito para la aprobación del curso, tienen por objetivo lograr que los alumnos puedan aplicar la teoría aprendida en la resolución, mediante la asistencia de programas de computadora, de problemas de validación y verificación de



software. Se buscará que los problemas a resolver en los trabajos prácticos hagan evidentes las bondades y limitaciones de las distintas técnicas de validación y verificación presentadas. Se intentará utilizar ejemplos y problemas interesantes, en los cuales las fallas sean difíciles de reconocer, intentando estimular al alumnado.

Se fomentará la lectura de material adicional y la auto-organización de los alumnos en sus actividades. Además, se dejará en manos de los alumnos la instalación y manejo de las herramientas de software utilizadas en la asignatura, como una manera de estimular la práctica en cuestiones técnicas no necesariamente ligadas a los tópicos que la asignatura abarca, y la experiencia en la utilización de herramientas nuevas.

5.4. Trabajos Prácticos

La materia contará con trabajos prácticos obligatorios, que serán evaluados por el docente de la materia. El plazo para la resolución de cada uno de los trabajos prácticos es de dos semanas.

Además, se proveerá una serie de ejercicios adicionales (cuya resolución es opcional) que acompañarán cada una de las clases teórico-prácticas.

El objetivo de los trabajos prácticos obligatorios (cuya resolución deberá ser individual) es poder evaluar la aplicación de las técnicas aprendidas en situaciones concretas de tamaño mediano, y la correcta comprensión de los fundamentos teóricos subyacentes a las técnicas estudiadas. Permiten también detectar y corregir problemas tanto en las actividades de cálculo y deducción en matemática y lógica como en la construcción de programas.

Bibliografía

- J-R Abrial. *The B-Book, Assigning Programs to Meanings*. Cambridge University Press, 1996.
- C. Baier y J-P. Katoen. *Principles of Model Checking*. Forthcoming.
- E. M. Clarke, Orna Grumberg y Doron Peled. *Model Checking*. MIT Press, 1999.
- D. Jackson. Automating First-Order Relational Logic, en *Proceedings de Conference on the Foundations of Software Engineering FSE 2000*, 2000.
- D. Jackson, I Shiyakhter y M. Sridharan. A Micromodularity Mechanism, en *Proceedings de FSE/ESEC'01*, 2001.



D. Jackson. *Software Abstractions: Logic, Language, and Analysis*. MIT Press, 2006.

J.-P. Katoen. *Concepts, Algorithms and Tools for Model Checking*. Arbeitsberichte der Informatik, Friedrich-Alexander-Universität Erlangen Nuerberg, vol. 32 no. 1. Genuer Druck GmbH, 1999.

M. Huth y M. Ryan. *Logic in Computer Science: Modelling and Reasoning about Systems*. 2nd Edition. Cambridge University Press, 2004.

P. Jorgensen. *Software Testing: A Craftsman's Approach*. Second Edition. CRC Press, 2002.

Z. Manna y A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems Specification*. Springer-Verlag, 1992

Z. Manna y A. Pnueli. *Temporal Verification of Reactive Systems: Safety*. Springer, 1995.



6. Programación Concurrente en Java

Docentes: Javier Blanco (responsable), Nicolás Wolovick

Duración: 40hs

Régimen: Obligatoria

6.1. Objetivos

La programación concurrente se basa en construir programas que se ejecutan en paralelo. Mediante comunicación y sincronización estos programas cooperan para realizar una tarea en conjunto. Ejemplos típicos son las interfaces gráficas, las bases de datos distribuidas y las simulaciones a gran escala.

Dentro de los posibles modelos de concurrencia (Bal *et. al.* 1989), el modelo de procesos secuenciales compartiendo memoria (*thread model*), es uno de los más utilizados, en parte porque las tecnologías de lenguajes de programación (Java), de sistemas operativos (WindowsXP, Linux) y de arquitecturas de procesador (Intel CoreDuo (Creger, M. 2005)), ponen a los *threads* al tope de sus prioridades.

Los *threads* o multiprogramas intentan preservar la sintaxis de los programas secuenciales, pero cambian el modelo computacional de manera drástica. Este modelo presentan un no-determinismo salvaje (Lee, E. 1999), que causa que toda la intuición y entendimiento, así como herramientas, técnicas y metodologías utilizadas para la programación secuencial deban ser abandonadas. La tarea del programador consiste entonces en poder este no-determinismo de manera adecuada (semáforos, exclusión mutua, monitores, etc.) para que las ejecuciones sean correctas (*safety*) y progresen (*liveness*).

A fin de atacar este problema, usualmente se utilizan métodos formales que decoran el programa con *aserciones* sobre su estado que deben ser válidas en cada punto (Consot, P. 1990). Existen herramientas formales que permiten decidir cuando el conjunto multiprogramas--aserciones resulta válido (Apt, K. *et. al.* 1991; Schneider, F. 1997; Feijen, W. 1999; Rodríguez, E. 2005) (además de ser muy útiles para construir el programa mismo).

Este curso se enfoca en estudiar los problemas del modelo de hilos haciendo énfasis en la herramienta de implementación Java. Los métodos asercionales serán la herramienta que nos permitirán razonar de manera rigurosa sobre los programas de este modelo.



Java incorpora desde su versión 5 casi todos los mecanismos usuales de control de concurrencia además de objetos y bloques de código en exclusión mutua *synchronized*; instrucciones atómicas de comparación y asignación, semáforos, barreras y *threadpools*, entre otros.

Java al ser un lenguaje orientado a objetos, permite dar soluciones a problemas genéricos de concurrencia a través de monitores, estructuras de datos *wait-free*, estructuras de datos *thread-safe*, además de poder implementar varios patrones de computación concurrente (Lea, D. 2000). Todas estas soluciones "enlatadas" son interesantes no solo desde el punto de vista externo, sino también en su estructura interna.

En este curso se trabajará sobre conceptos de programación concurrente que permitirán analizar estas soluciones y decidir los contextos adecuados de uso. Estos conceptos permitirán también desarrollar nuevas soluciones.

6.2. Programa

- Tipos de concurrencia: problemas de la concurrencia; limitaciones del razonamiento operacional. Programación concurrente orientada a objetos (Lea, D. 2000).
- Programas paralelos con *shared memory*: lógica elemental para demostrar y razonar; teoría de Owicki-Gries; modelo computacional; desarrollo sistemático de multiprogramas; algunos problemas y sus soluciones (sincronización de fase, *ABP*, región crítica, barrera, lectores y escritores, etc.); implementación de algoritmos en Java.
- Exclusión mutua: concepto de exclusión mutua; uso para simplificar los programas y las pruebas; objetos y métodos sincronizados en Java; confinamiento; manejo de locks; algoritmos para asegurar exclusión mutua.
- Primitivas de sincronización de alto nivel: semáforos y semáforos binarios divididos (Dijkstra, E. 1979); regiones críticas con condición, su mapeo a *SBS*; barreras; monitores; algunos problemas típicos de concurrencia y sus soluciones (Andrews, G. 2000) (algoritmos paralelos de datos, lectores y escritores con prioridad, filósofos comensales, barbero dormilón, etc.); dependencia de estados en Java; monitores en Java (Dongol, 2006); problemas con las excepciones.
- Progreso: deadlock, inanición, justicia del planificador. Técnicas para asegurar el progreso.



- Temas avanzados: transacciones; creación dinámica de threads; mensajes; computación paralela; especificación de programas concurrentes en JML (Rodríguez, E. 2005); verificación automática de algunas propiedades; modelo de memoria (Pugh, W. 1999).

Bibliografía

- Andrews, G. *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison Wesley, 2000.
- Apt, K. and Ernst-Rüdiger Olderog. *Verification of Sequential and Concurrent Programs*. Graduate texts in Computer Science. Springer-Verlag, New York, 1991.
- Bal II., Jennifer G. Steiner, and Andrew S. Tanenbaum. Programming languages for distributed computing systems. *ACM Computing Surveys*, 21(3), 1989.
- Cousot, P. Methods and logics for proving programs. In *Handbook of Theoretical Computer Science*, Volume B: Formal Models and Semantics (B), pages 841-994, 1990.
- Greger, M. Multicore cpus for the masses. *ACM Queue*, 3(7), 2005.
- Dijkstra, E. *A tutorial on the split binary semaphore*. EWD Series, (703), 1979.
- Dongol, B. Derivation of java monitors. In *Australian Software Engineering Conference*, pages 211-220. IEEE Computer Society, 2006.
- Feijen, W. and A.J.M. van Gasteren. *On a Method of MultiProgramming*. Monographs in Computer Science. Springer, 1999.
- Lea, D. *Concurrent programming in Java*. Java. Addison-Wesley, 2000.
- Lee, E.. The problem with threads. *Computer*, 39(5):33-42, 2006.
- Pugh, W.. Fixing the java memory model. In *Java Grande*, pages 8-998, 1999.
- Rodríguez, E., Matthew B. Dwyer, Cormac Flanagan, John Hatcliff, Gary T. Leavens, and Robby. Extending JML for modular specification and verification of multi-threaded programs. ECOOP. In Andrew P. Black, editor, volume 3586 of *Lecture Notes in Computer Science*. Springer, 2005.
- Schneider, F. *On Concurrent Programming*. Springer, 1997.



7. Evaluación Final integradora y Trabajo de Tesis

Responsable: Gabriel Infante Lopez

Docentes: Cuerpo de Tutores.

Duración: Equivalente a 20hs de curso.

Régimen: Obligatoria.

7.1. Objetivo

El objetivo del trabajo integrador es que el alumno sea capaz de demostrar que puede integrar los conocimientos aprendidos durante la especialidad en un trabajo de su autoría.

7.2. Generalidades

Cada alumno será asistido y dirigido por un tutor de su elección del cuerpo de tutores. Esta materia tendrá uno o más responsables a cargo.

El alumno interesado presentará un *pre-proyecto* de trabajo integrador. Los docentes encargados y el director evaluarán el trabajo propuesto y la conveniencia de designar un colaborador de dirección.

Al finalizar el trabajo especial, el alumno entregará un informe escrito denominado "Trabajo Integrador" que será evaluado por los miembros del *tribunal examinador*.

7.3. Sobre el pre-proyecto

El pre-proyecto es una descripción del trabajo especial a realizar. Este debe especificar:

1. Nombre de/llos alumnos que realizará/an el trabajo.
2. Nombre de/llos docente/s que lo asistirá/án.
3. El título (no necesariamente definitivo) del Trabajo Integrador.
4. Un resumen de lo que se propone hacer, explicando la motivación, los objetivos y la metodología a seguir.
5. La bibliografía a utilizar.



7.4. Evaluación del pre-proyecto

Junto con el director, los docentes encargados deberán:

- determinar si el trabajo propuesto en el pre-proyecto se enmarca dentro de los estándares de un trabajo integrador
- determinar si el trabajo necesita supervisión adicional aparte del director responsable del trabajo especial, y proponer en ese caso un colaborador adicional.
- Fiscalizar que el trabajo propuesto no exceda el equivalente a un curso de 20hs.

Una vez que se aprobó el pre-proyecto, el alumno presentará una nota a la CAP adjuntando el pre-proyecto con el aval de los docentes encargados indicando, si lo hubiere, el nombre del colaborador de dirección.

7.5. Sobre la evaluación del trabajo especial

El director y los docentes encargados propondrán las personas que acompañarán al director en el tribunal examinador.

El alumno deberá entregar el Trabajo Integrador para su evaluación por los miembros de un Tribunal Examinador, quienes recibirán una copia del mismo con el pre-proyecto originalmente presentado.

Los miembros del Tribunal Examinador podrán sugerir mejoras para la versión final del Trabajo Integrador, basándose en la copia recibida e interactuando con el alumno (por ejemplo, solicitando la demostración de un desarrollo asociado al trabajo, si lo hubiere).

Tras la entrega de la versión final del Trabajo Integrador, y con la aprobación de los miembros del Tribunal Examinador, el alumno enviará el Trabajo Integrador a imprimir y enviará una nota al Consejo Asesor Profesional correspondiente.

La defensa del trabajo tendrá lugar en la fecha que correspondiere a la mesa de examen inscripto. El Tribunal Examinador determinará la calificación definitiva del trabajo.



8. Performance de Bajo Nivel

Docentes: Arzhan Kinzhalin (Inv.^o Argentina)

Duración: 20hs

Régimen: Optativa

8.1. Objetivos

El propósito del curso es estudiar diferentes factores que afectan la performance de las aplicaciones. Estudiaremos estos factores desde el nivel de diseño hasta la microarquitectura del procesador. El curso además cubrirá el uso de herramientas y de distintas técnicas disponibles para los desarrolladores que ayudan a mejorar la performance de sus aplicaciones.

8.2. Programa

- Performance Cuantitativa: workloads, benchmarks, ley de Amdahl's, notaciones de performance.
- Arquitectura de Computadoras: Vista de alto nivel, bases de microarquitectura, jerarquía de memoria, subsistemas de Entrada/Salida críticos de performance.
- Técnicas de optimización de software, evitando penalidades de ejecución, uso eficiente de la memoria, técnicas básicas de optimización de compiladores.
- Herramientas de cambio: PMU, herramientas de monitoreo de performance de CPU (Intel VTune Analyzer), Herramientas a nivel de Sistema Operativo.
- Núcleo Múltiple y paralelismo. Tendencias futuras en computación.

Bibliografía

J. L. Hennessy, D. A. Patterson, *Computer Architecture: A Quantitative Approach* (3rd edition). Morgan Kaufmann Series in Computer Architecture and Design. Elsevier Science, 2003.



A. Fog. *How to Optimize for the Pentium Microprocessors*. <http://www.agner.com/assem/>. March 2000.

Intel Corp. "Intel Architecture Software Developer's Manual, Volume 1: Basic Architecture", 1997.

Intel Corp. "Intel Architecture Software Developer's Manual, Volume 2: Instruction Set Reference Manual", 1997.

Intel Corp. "Intel Architecture Software Developer's Manual, Volume 3: System Programming Guide", 1997.



9. Ingeniería de Performance en Sistemas Distribuidos

Docentes: Samuel Komney (Universidad de Cambridge)

Duración: 20hs

Régimen: Optativa

9.1. Objetivos

Aprender a comprender los aspectos de performance en sistemas distribuidos. Esto incluye el marco teórico, modelos, concepto de workload, métricas, herramientas y técnicas.

9.2. Contenidos

- Modelos de e-Business. Modelos de referencia. Comportamiento del usuario. Arquitecturas típicas de sistemas de comercio electrónico. Diagramas de interacción entre cliente y servidor (CSID).
- Caracterizando la carga en e-Business. Características de carga en la Web. Caracterizando el comportamiento del usuario.
- El costo de seguridad en e-Business. Seguridad en e-Business. Análisis cuantitativo del servicio de autenticidad. Análisis cuantitativo de los servicios de pago electrónico.
- Métricas de performance. Que es una métrica? Características de las buenas métricas. Métricas estándares para el sistema y el procesador. Aceleración y cambio relativo.
- Performance promedio y variabilidad. Índices para estimar la tendencia. Componiendo métricas de distintos benchmarks. Cuantificando variabilidad.
- Errores en toma de datos experimentales. Fuentes de errores. Precisión, resolución. Modelo matemático para errores. Intervalos de certidumbre para proporciones y promedios.
- Comparando alternativas. Comparando dos alternativas. Antes y después de las comparaciones. Mediciones que no se corresponden. Comparando más de dos alternativas.



- Herramientas y técnicas para mediciones. Eventos. Estrategias fundamentales de toma de mediciones. Profiling. Seguimiento de eventos. Toma de mediciones indirectas.

Bibliografía

D. Menasce y V. Almeida, *Scaling for e-business: Technologies, Models and Methods*. Prentice Hall, 2000.

D. Menasce y V. Almeida, *Capacity planning for Web Performance (Metrics, Models and Methods)*. Prentice Hall, 2001.

C. Loosley y E. Douglas, *High-Performance Client/Server*. Wiley & Sons, 1997.



10. Proceso de Desarrollo de Software

Docentes: Patricio Maller (Intel Argentina)

Duración: 20hs

Régimen: Optativa

10.1. Objetivos

Con el avance de la computación hacia una disciplina de ingeniería, los procesos utilizados para desarrollar software tomaron una fuerte relevancia tanto en el mundo académico como en la industria. Tomando como marco conceptual los procesos de elaboración de productos, y una buena parte de resultados empíricos, los procesos de producción de software fueron rápidamente reconocidos como determinantes de productos de calidad y entregados a tiempo. Aunque los procesos de producción de software tendieron a homogeneizarse bajo estándares de la IEEE y el SEI, en los últimos años han emergido nuevas corrientes que tienden a modernizar y "agilizar" los procesos, de forma tal de producir mejoras drásticas en la producción. En esta materia se estudiarán los conceptos básicos de procesos de software. Los modelos tradicionales serán visitados, tomando ejemplos de la industria para ilustrar los diferentes paradigmas.

Finalmente, se explorarán las metodologías de tipo "agile", sus áreas de aplicación y los resultados. Se pretende, mostrando todo el espectro en procesos de producción de software, que los asistentes sean capaces de encontrar el balance en procesos a través de un cabal conocimiento de los extremos.

10.2. Programa

1. Introducción y marco histórico
2. Procesos de software y modelos de calidad
3. CMMI, conceptos básicos y estructura
4. Arquitectura de procesos
5. Implementaciones clásicas de procesos de SW
6. Metodologías ágiles de desarrollo (énfasis en XP y Scrum)
7. Evaluaciones del proceso, métricas y auditorías



8. Adaptación del proceso a las necesidades de la organización (tailoring)

Bibliografía

- Ivan Jacobson, Grady Booch, James Rumbaugh. *The Unified Software Development Process*, 2003.
- Roger S Pressman. *Software Engineering: A Practitioner's Approach*, 2003.
- Alistair Cockburn. *Agile Software Development: The Cooperative Game*, 2006.
- Mary Beth Chrissis, Mike Kourad, and Sandy Shrum. *CMMI : Guidelines for Process Integration and Product Improvement*, 2003.
- Ken Schwaber. *Agile Project Management with Scrum*, 2004.
- Kent Beck and Cynthia Andres. *Extreme Programming Explained: Embrace Change*, 2004.



11. Patrones Seguros y Diseño Seguro de Sistemas

Docentes: Eduardo Fernández (Florida Atlantic University)

Duración: 20hs

Régimen: Optativa

11.1. Objetivos

Los análisis y diseños de patrones están bien establecidos para crear software orientado a objetos de alta calidad. Los patrones combinan la experiencia y las buenas prácticas para desarrollar modelos básicos que pueden ser usados en nuevos diseños. Los patrones de seguridad tienen el extenso conocimiento acumulado sobre seguridad con las estructuras brindada por los patrones para proveer las guías para el diseño y evaluación de sistemas seguros. Esta tecnología está siendo adoptada por compañías como IBM, Sun o Microsoft.

En este curso mostramos la anatomía de un patrón de seguridad, una variedad de estos patrones, y su uso en la construcción de sistemas seguros. Estos patrones incluyen autenticación, autorización, acceso controlado por roles, firewalls, web Services security (SAML, XACML, XML Firewall), y otros.

Aplicaremos estos patrones en el desarrollo de un sistema seguro basado en una arquitectura jerárquica cuyas capas definen el alcance de cada mecanismo de seguridad. Primero, los ataques posibles se consideran por análisis de casos de uso. Luego, los permisos de los usuarios se definen a partir de los casos de uso, utilizando Role-Based Access Control (RBAC) u otro modelo de seguridad.

Los ataques son usados para definir las políticas que pueden detenerlos. Los permisos se reflejan en el modelo de la clase conceptual. Después definiremos restricciones adicionales que se aplican a la distribución, interfaces, y componentes. Los patrones serán mostrados usando modelos UML, y algunos ejemplos se tomarán de del libro "Security Patterns", de reciente aparición.

11.2. Programa

1. Introducción. Motivación, conceptos básicos. El contexto para la seguridad. Ataques.



2. El diseño de sistemas seguros. Diseño orientado a objetos. UML, y patrones. La necesidad para una buena ingeniería del software. Principios de Seguridad. Patrones de Seguridad. Estándares.
3. Anatomía de un patrón seguro.
4. Modelos de seguridad y sus patrones – políticas, matrices de acceso, modelos de varios niveles. RBAC.
5. Definiendo autorizaciones desde casos de uso – aspectos no funcionales de casos de uso, RBAC y políticas de seguridad.
6. Firewall, IDS, y patrones de sistemas operativos.
7. Modelo de autorización conceptual.
8. Arquitectura de sistemas seguros – efecto de la distribución y de las interfaces de usuario.
9. Servidores de aplicación web y componentes – mapeo de RBAC a componentes, J2EE y .NET
10. Patrones para servicios web: SAML, XACML, Liberty Alliance, WS-Security. Comparando estándares utilizando patrones. Aplicaciones y firewalls XML.
11. Coordinación de varios niveles – mapeo de autorizaciones a través de niveles de arquitectura.
12. Conclusiones – el futuro.

Bibliografía

M. Schumacher, E. B. Fernández, F. Buschmann, and D. Hybertson, *Security Patterns*, Wiley, 2006.



12. Seminario de Software Libre

Docentes: Ricardo Medel (Intel Argentina)

Duración: 20hs

Régimen: Optativa

12.1. Descripción

Dado que buena parte de la infraestructura de Internet está basada en software libre, éste es un tema de relevancia al momento de estudiar sistemas distribuidos y servicios web. Este seminario tiene como objetivo introducir a los alumnos en el tema del software libre y desarrollar las habilidades requeridas para realizar investigación independiente.

12.2. Temas

Temas propuestos: historia del software libre, licencias de software y aspectos legales, GNU/Linux, casos de éxito (Mozilla, Debian, Eclipse, Apache), procesos de desarrollo basados en software libre, infraestructura para el desarrollo de software libre, negocios abiertos, adopción del software libre en el Tercer Mundo, aplicación de los procesos de creación de software libre a otros ámbitos de creación de conocimiento (Wikipedia, Gleducar), software libre en el Estado.

12.3. Metodología de trabajo

Se formarán grupos de 3 ó 4 alumnos. Cada grupo trabajará en un tema específico durante el cuatrimestre, dirigidos por un tutor. Al final del cuatrimestre cada grupo deberá presentar un documento con una elaboración original del tema elegido y realizará una presentación oral grupal. Durante el cuatrimestre, figuras relevantes de la comunidad del software libre darán un ciclo de charlas sobre los temas propuestos. Cada charla durará al menos 1 hora y habrá tiempo para preguntas y respuestas.

12.4. Evaluación

Los alumnos deberán asistir al 80% de las charlas y a todas las reuniones pactadas con los tutores. La nota final de cada estudiante estará basada en



Facultad de Matemática, Astronomía y Física
Universidad Nacional de Córdoba



su aporte al trabajo del equipo, el nivel de la bibliografía consultada, la calidad del documento producido y la presentación oral.