

## MODALIDAD ACADÉMICA

<b>Asignatura</b>	<b>Ingeniería de Software de Fuentes Abiertas/Libre</b>	
<b>Carrera</b>	<b>INGENIERÍA EN SISTEMAS DE INFORMACIÓN</b>	
<b>Ciclo Lectivo</b>	2017	
<b>Vigencia del programa</b>	Desde el ciclo lectivo 2017	
<b>Plan</b>	2008	
<b>Nivel</b>	<input type="checkbox"/> 1er. Nivel <input type="checkbox"/> 2do. Nivel <input type="checkbox"/> 3er. Nivel <input type="checkbox"/> 4to. Nivel <input checked="" type="checkbox"/> 5to. Nivel	
<b>Coordinador de la Cátedra</b>	Dr. Ricardo Medel	
<b>Área de Conocimiento</b>	<input checked="" type="checkbox"/> Programación <input type="checkbox"/> Computación <input type="checkbox"/> Sistemas de Información <input checked="" type="checkbox"/> Gestión Ingenieril <input type="checkbox"/> Modelos <input type="checkbox"/> Complementaria	
<b>Carga horaria semanal</b>	6 horas semanales	
<b>Anual/ cuatrimestral</b>	cuatrimestral	
<b>Contenidos Mínimos, según Diseño Curricular-Ordenanza 1150 (sólo para asignaturas curriculares)</b>		
<b>Correlativas para cursar</b> (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	<ul style="list-style-type: none"> <li>Al menos 1 materia del conjunto PAVII-DLC</li> <li></li> </ul>	<ul style="list-style-type: none"> <li>Diseño de Sistemas</li> <li></li> </ul>
<b>Correlativas para rendir</b> (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	<ul style="list-style-type: none"> <li>Al menos 1 materia del conjunto PAVII-DLC</li> <li>Ingeniería de Software de Fuentes Abiertas/Libre (Electiva)</li> <li></li> </ul>	<ul style="list-style-type: none"> <li>Diseño de Sistemas</li> <li></li> </ul>
<b>Objetivos de la Asignatura</b>	<u>Generales</u> <ul style="list-style-type: none"> <li>Entender los conceptos fundamentales, historia y actualidad del software de fuentes abiertas/libre.</li> <li>Comprender los factores que afectan la gestión de proyectos de software de fuentes abiertas/libre y cómo enfrentarlos.</li> <li>Adquirir los conocimientos necesarios para participar de un proyecto de software de fuentes abiertas/libre.</li> </ul> <u>Específicos</u> <ul style="list-style-type: none"> <li>Tener los conocimientos necesarios que permitan evaluar la conveniencia de utilizar software de fuentes abiertas/libre en la práctica profesional.</li> <li>Saber utilizar herramientas conceptuales y de software que permiten participar</li> </ul>	

	<p>en proyectos de desarrollo de software de fuentes abiertas/libre.</p> <ul style="list-style-type: none"> <li>• Entender los conceptos de calidad e integración continua aplicados al desarrollo de software de fuentes abiertas/libre y utilizar herramientas apropiadas.</li> <li>• Obtener la capacidad de analizar cómo el software afecta las relaciones sociales y económicas en la sociedad.</li> <li>• Poder definir un modelo de negocios basado en software de fuentes abiertas/libre y el modelo cooperativo de creación de valor.</li> </ul>
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Programa Analítico**

**UNIDAD N° 1: Conceptos básicos e historia del software de fuentes abiertas/libre**

**OBJETIVOS ESPECÍFICOS:**

Desarrollar en el estudiante una comprensión de los conceptos fundamentales de software de fuentes abiertas/libre, así como sus diferentes definiciones, su historia y el estado actual del software de fuentes abiertas/libre.

Brindar al estudiante herramientas que le permitan comprender los aspectos legales a considerar cuando se hace uso del software de fuentes abiertas/libre en el contexto de su profesión.

**CONTENIDOS:**

Historia del software. Surgimiento del software privativo. Proyecto GNU: el nacimiento del software libre. Conceptos y definiciones de software libre. La Iniciativa Open Source (OSI): la conceptualización del software de fuentes abiertas. Historia del software de fuentes abiertas/libre en Latinoamérica. Licencias de software: privativas, permisivas, recíprocas totales, recíprocas parciales. Legislación sobre software de fuentes abiertas/libre.

**BIBLIOGRAFÍA:**

- González Barahona, Jesús, Seoane Pascual, Joaquín, Robles, Gregorio. “Introducción al software libre”, Fundació per a la Universitat Oberta de Catalunya, 2003. ISBN: 84-9788-028-5
- Williams, Sam. “Free as in Freedom: Richard Stallman's Crusade for Free Software”, O’Reilly, 2002.
- Raymond, Eric S., “La catedral y el bazar”, O’Reilly, 1997.
- Gomez Gomez, Víctor Manuel. “Aspectos Históricos del Software Libre en América Latina”, 2005.
- St. Laurent, Andrew M. “Understanding Open Source and Free Software Licensing”, O’Reilly, 2004, ISBN: 0-596-00581-4.
- Wheeler, David A. “Make Your Open Source Software GPL-Compatible. Or Else.”, 2014, <http://www.dwheeler.com/essays/gpl-compatible.html>

**EVALUACIÓN:**

**Este temario se incluye en los ejercicios prácticos a entregar y en el examen parcial integrador.**

**UNIDAD N° 2: Aspectos sociales del software de fuentes abiertas/libre**

**OBJETIVOS ESPECÍFICOS:**

Desarrollar en el estudiante la capacidad de analizar cómo el software afecta las relaciones sociales y económicas en la sociedad.

Brindar al estudiante herramientas que le permitan tomar decisiones sobre el uso adecuado del software de fuentes abiertas/libre en el contexto de su profesión.

**CONTENIDOS:**

El software de fuentes abiertas/libre como movimiento social. Software de fuentes abiertas/libre en la educación. Software de fuentes abiertas/libre en el estado. Modelos de negocios tradicionales y basados en software de fuentes abiertas/libre. Planes de negocios y *startups* basadas en software de fuentes abiertas/libre. Cooperativismo en el

mercado del software. Casos de estudio de negocios exitosos y fracasos basados en software de fuentes abiertas/libre.

**BIBLIOGRAFÍA:**

- D'Elia Branco, Marcelo. "El Software Libre y sus perspectivas para el desarrollo en América Latina y el Caribe", <http://www.bellanet.org>.
- Rosa, Fernando da; Heinz, Federico. "Guía práctica sobre software libre: su selección y aplicación local en América Latina y el Caribe", UNESCO, 2007. ISBN: 92-9089-103-3
- Stallman, Richard M. "Free Software, Free Society: Selected Essays of Richard M. Stallman", Free Software Foundation, 200
- Zúñiga, Lena. "Voces libres de los campos digitales: Una investigación social sobre el Software Libre en América Latina y el Caribe", 2006. ISBN: 9977-12-888-X
- Salas, Margarita. "Género y software libre en América Latina: Un estudio de caso", <http://www.sulabatsu.com/voces/Documentos/genero.pdf>
- Wheeler, David A. "Open Source Software (OSS or FLOSS) and the U.S. Department of Defense (DoD)", 2009. <http://www.dwheeler.com/essays/dod-oss.pdf>

**EVALUACIÓN:**

**Este temario se incluye en los ejercicios prácticos a entregar y en el informe sobre la participación del estudiante en un proyecto de software de fuentes abiertas/libre real.**

**UNIDAD N°3: Análisis y gestión de proyectos de software de fuentes abiertas/libre**

**OBJETIVOS ESPECÍFICOS:**

Desarrollar en el estudiante una comprensión de los factores que afectan la gestión de proyectos de software de fuentes abiertas/libre y cómo enfrentarlos.

Brindar al estudiante herramientas conceptuales y de software que le permitan participar o manejar una comunidad enfocada al desarrollo de software de fuentes abiertas/libre.

**CONTENIDOS:**

Tipos de comunicación en proyectos de software de fuentes abiertas/libre, ventajas y limitaciones (email, IRC, foros, blogs). Sistemas de control de versión y su impacto sobre el modelo de contribuciones (SVN, Git, Mercurial, Bazaar). Infraestructura para alojamiento de proyectos en la web (Bitbucket, Github, Google Code). Sistemas de seguimiento de Bugs (Bugzilla, Redmine). Distintas formas de participación en comunidades de software de fuentes abiertas/libre. Sistemas de reputación en proyectos de software de fuentes abiertas/libre y meritocracia.

**BIBLIOGRAFÍA:**

- Fogel, Karl. "Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre", O'Reilly, 2007.
- Raymond, Eric S. "La catedral y el bazar", O'Reilly, 1997.
- Hernández Gómez, Lorena. "Herramientas software para el trabajo científico colaborativo", Trabajo de Fin de Máster, Universidad de Salamanca, 2011.
- Collins-Sussman, Ben, Fitzpatrick, Brian W., Pilato, C. Michael. "Control de versiones con Subversion", O'Reilly, 2004.
- Loeliger, Jon, McCullough, Matthew. "Version Control with Git, 2nd Edition", O'Reilly, 2012.
- O'Sullivan, Bryan. "Mercurial: The Definitive Guide", O'Reilly, 2009.
- Gyerik, Janos. "Bazaar Version Control", Packt Publishing, 2013.
- Lesyuk, Andriy. "Mastering Redmine", Packt Publishing, 2013.

**EVALUACIÓN:**

**Este temario se incluye en los ejercicios prácticos a entregar y en el examen parcial integrador. Asimismo, su comprensión será evaluada por la participación del estudiante en un proyecto de software de fuentes abiertas/libre real.**

#### **UNIDAD N° 4: Diseño e implementación de software de fuentes abiertas/libre**

##### **OBJETIVOS ESPECÍFICOS:**

Desarrollar en el estudiante los conceptos básicos requeridos para analizar y valorar los aportes de codificación en una comunidad de software de fuentes abiertas/libre.

Brindar al estudiante los conocimientos necesarios para aportar código en un proyecto de software libre.

##### **CONTENIDOS:**

El código auto-documentado. Guías de estilo. Filosofía Unix de Diseño. *Refactoring*. Patrones de diseño.

Participación como desarrollador en un proyecto de software de fuentes abiertas/libre.

##### **BIBLIOGRAFÍA:**

- Fogel, Karl. "Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre", O'Reilly, 2007.
- Martin, Robert C. "Clean Code", 2008. ISBN-10: 0132350882
- Fowler, Martin, Beck, Kent, Brant, John, Opdyke, William, Roberts, Don. "Refactoring: Improving the Design of Existing Code", 1999. ISBN-10: 0201485672
- Gamma, Erich, Helm, Richard, Johnson, Ralph, Vlissides, John. "Design patterns : elements of reusable object-oriented software", 1994. ISBN-10: 0201633612
- Sokol, Francisco Zigmund, Aniche, Mauricio Finavaro, Gerosa, Marco Aurélio. "Does the Act of Refactoring Really Make Code Simpler? A Preliminary Study", Workshop Brasileiro de Métodos Ágeis, 2013.

##### **EVALUACIÓN:**

**Este temario se incluye en los ejercicios prácticos a entregar y en el examen parcial integrador. Asimismo, su comprensión será evaluada por la participación del estudiante en un proyecto de software de fuentes abiertas/libre real.**

#### **UNIDAD N° 5: Sistemas de software de fuentes abiertas/libre**

##### **OBJETIVOS ESPECÍFICOS:**

Brindar al estudiante conocimiento sobre los principales sistemas de software de fuentes abiertas/libres utilizados en la industria del software en la actualidad.

Permitir al estudiante obtener conocimiento práctico sobre el software de base y sistemas complejos que permiten el desarrollo de un completo ecosistema de software de fuentes abiertas/libre.

##### **CONTENIDOS:**

Lenguajes de programación libres (GCC, Clang/LLVM, Python). Sistemas operativos libres/de fuentes abiertas (GNU/Linux, FreeBSD, ReactOS). Bases de datos de fuentes abiertas/libres (MySQL, PostgreSQL, MongoDB).

##### **BIBLIOGRAFÍA:**

- Gough, Brian J. "An Introduction to GCC", 2004. ISBN: 0-9541617-9-3
- Lattner, Chris. "The Design of LLVM", *Dr. Dobbs's Journal*, Mayo de 2012.
- Lutz, Mark. "*Learning Python* (5th ed.)", O'Reilly Media, 2013. ISBN 978-0-596-15806-4
- Kroah-Hartman, Greg. "Linux Kernel in a Nutshell", O'Reilly Media, 2006. ISBN : 0-596-10079-5
- Dyer, Russell J. T. "MySQL in a Nutshell", O'Reilly Media, 2008.
- Redmond, Eric, Wilson, Jim R. "Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement", 2012.

##### **EVALUACIÓN:**

**Este temario se incluye en los ejercicios prácticos a entregar y en el examen parcial integrador.**

**UNIDAD N° 6: Mantenimiento y calidad en proyectos de software de fuentes abiertas/libre**

**OBJETIVOS ESPECÍFICOS:**

Desarrollar en el estudiante los conceptos de calidad e integración continua aplicados al desarrollo de software de fuentes abiertas/libre.

Brindar al estudiante conocimiento práctico sobre las herramientas de automatización de las tareas de calidad en proyectos de software de fuentes abiertas/libre.

**CONTENIDOS:**

Recepción y revisión de *patches*. Tests automatizados. *Triaging* de bugs. Herramientas de *build* e integración continua (Jenkins, Hudson, TravisCI).

**BIBLIOGRAFÍA:**

- Burns, Ed, Prakash, Winston. “Hudson Continuous Integration in Practice”, 2013. IS “BN-10: 0071804285
- Berg, Alan. “Jenkins Continuous Integration Cookbook”, Packt Publishing, 2012.
- Smart, John Ferguson. “Jenkins: The Definitive Guide”, 2011. ISBN-10: 1449305350
- Nelson-Smith, Stephen. “Test-driven Infrastructure with Chef, 2nd Edition”, O’Reilly, 2013.
- Black, Rex. “Managing the Testing Process, 3rd Edition”, Wiley, 2011.

**EVALUACIÓN:**

**Este temario se incluye en los ejercicios prácticos a entregar y en el examen parcial integrador. Asimismo, su comprensión será evaluada por la participación del estudiante en un proyecto de software de fuentes abiertas/libre real.**

<p><b>Metodología de enseñanza y aprendizaje</b></p>	<p>Clases teórico-prácticas Estas clases permitirán articular aspectos teóricos de la asignatura, desarrollados por el docente o por personalidades reconocidas de la comunidad de software libre local, invitados oportunamente, en forma expositiva con actividades prácticas que permitan ejercitar la temática abordada en casos reales planteados en clase y discusiones de las que deberán participar los alumnos. Con este tipo de clases se buscará interrelacionar los fundamentos teóricos con las experiencias prácticas en el desarrollo y aplicación de software de fuentes abiertas/libre y abrir la discusión a nuevas visiones de la problemática.</p> <p>Proyecto de Aplicación El alumno participará activamente en un proyecto de software de fuentes abiertas/libre ya existente, realizado por una comunidad externa a la asignatura y a la universidad. Este tipo de actividad está orientada a ejercitar en un ambiente real los temas teóricos y prácticos abordados en la asignatura. Debido a la complejidad de insertarse en tan poco tiempo en una comunidad de software de fuentes abiertas/libre, la participación de; alumno puede ser bien como usuario, como colaborador en la documentación, como tester o como desarrollador.</p>
<p><b>Sistema de evaluación</b></p>	<p>La evaluación de los alumnos se compone de los siguientes tres componentes mayores, sumados a su asistencia y participación activa en las clases.</p> <ol style="list-style-type: none"> <li>1. Ejercicios prácticos o ensayos de realización obligatoria, con entrega pautada.</li> <li>2. Examen parcial integrador y/o su instancia recuperatoria.</li> <li>3. Presentación (oral y escrita) sobre su participación en el proyecto de software libre mencionado anteriormente (Proyecto de Aplicación).</li> </ol> <p>Como se indica en el punto 2, habrá un examen de recuperación en el caso de que el alumno no alcanzara el mínimo esperado en el examen parcial integrador.</p>

<b>Regularidad: condiciones</b>	<p>Para regularizar la asignatura es necesario cumplir con todos los ejercicios prácticos de realización obligatoria, rendir el examen parcial o su recuperatorio, realizar la presentación de participación en el proyecto de software libre y tener al menos una nota de 4 en el promedio de los componentes mencionados anteriormente.</p> <p>Escala de notas de regularidad(*)</p> <table border="1" data-bbox="675 464 1248 842"> <thead> <tr> <th>NOTAS</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr><td>1</td><td></td><td>No Aprobado</td></tr> <tr><td>2</td><td></td><td>No Aprobado</td></tr> <tr><td>3</td><td></td><td>No Aprobado</td></tr> <tr><td>4</td><td>55% a 57%</td><td>Aprobado</td></tr> <tr><td>5</td><td>58% a 59%</td><td>Aprobado</td></tr> <tr><td>6</td><td>60% a 68%</td><td>Aprobado</td></tr> <tr><td>7</td><td>69% a 77%</td><td>Aprobado</td></tr> <tr><td>8</td><td>78% a 86%</td><td>Aprobado</td></tr> <tr><td>9</td><td>87% a 95%</td><td>Aprobado</td></tr> <tr><td>10</td><td>96% a 100%</td><td>Aprobado</td></tr> </tbody> </table> <p>(*) Escala acordada en reunión de Docentes Coordinadores</p> <p>El estudiante en condición de regular puede rendir en el plazo de un ciclo lectivo sin control de correlativas aprobadas</p>	NOTAS	PORCENTAJE	CALIFICACIÓN	1		No Aprobado	2		No Aprobado	3		No Aprobado	4	55% a 57%	Aprobado	5	58% a 59%	Aprobado	6	60% a 68%	Aprobado	7	69% a 77%	Aprobado	8	78% a 86%	Aprobado	9	87% a 95%	Aprobado	10	96% a 100%	Aprobado
NOTAS	PORCENTAJE	CALIFICACIÓN																																
1		No Aprobado																																
2		No Aprobado																																
3		No Aprobado																																
4	55% a 57%	Aprobado																																
5	58% a 59%	Aprobado																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Aprobado																																
8	78% a 86%	Aprobado																																
9	87% a 95%	Aprobado																																
10	96% a 100%	Aprobado																																
<b>Promoción: condiciones</b>	No corresponde.																																	
<b>Aprobación Directa: condiciones.</b>	<p>Para la aprobación directa de la asignatura es necesario tener una nota de al menos 7 en el promedio de los componentes de evaluación mencionados anteriormente.</p> <p>El estudiante puede registrar su nota en examen en el plazo de un ciclo lectivo, sin control de correlativas aprobadas, y después de ello se le exigirán correlativas aprobadas</p>																																	
<b>Modalidad de examen final</b>	<p>El alumno deberá presentar en forma oral y escrita un reporte de participación de al menos 2 meses en un proyecto de software libre, y superada esa instancia con una nota de al menos 6, pasará a una instancia oral de evaluación de todas las temáticas de la asignatura. La nota final será el promedio entre la nota de la presentación y la nota del oral.</p> <p>Escala de Notas para Examen Final (*)</p> <table border="1" data-bbox="724 1560 1292 1934"> <thead> <tr> <th>NOTA</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr><td>1</td><td></td><td>Insuficiente</td></tr> <tr><td>2</td><td></td><td>Insuficiente</td></tr> <tr><td>3</td><td></td><td>Insuficiente</td></tr> <tr><td>4</td><td></td><td>Insuficiente</td></tr> <tr><td>5</td><td></td><td>Insuficiente</td></tr> <tr><td>6</td><td>60% a 68%</td><td>Aprobado</td></tr> <tr><td>7</td><td>69% a 77%</td><td>Bueno</td></tr> <tr><td>8</td><td>78% a 86%</td><td>Muy Bueno</td></tr> <tr><td>9</td><td>87% a 95%</td><td>Distintivo</td></tr> <tr><td>10</td><td>96% a 100%</td><td>Sobresaliente</td></tr> </tbody> </table>	NOTA	PORCENTAJE	CALIFICACIÓN	1		Insuficiente	2		Insuficiente	3		Insuficiente	4		Insuficiente	5		Insuficiente	6	60% a 68%	Aprobado	7	69% a 77%	Bueno	8	78% a 86%	Muy Bueno	9	87% a 95%	Distintivo	10	96% a 100%	Sobresaliente
NOTA	PORCENTAJE	CALIFICACIÓN																																
1		Insuficiente																																
2		Insuficiente																																
3		Insuficiente																																
4		Insuficiente																																
5		Insuficiente																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Bueno																																
8	78% a 86%	Muy Bueno																																
9	87% a 95%	Distintivo																																
10	96% a 100%	Sobresaliente																																

	(*) Escala acordada en reunión de Docentes Coordinadores						
<b>Actividades en laboratorio</b>	Las actividades prácticas deberán ser realizadas asistidas por computadoras con diverso software libre, según el tema a tratar.						
<b>Horas/año totales de la asignatura</b> (hs. cátedra)	<b>90</b>						
<b>Cantidad de horas prácticas totales</b> (hs. cátedra)	<b>60</b>						
<b>Cantidad de horas teóricas totales</b> (hs. cátedra)	<b>30</b>						
<b>Tipo de formación práctica</b> (sólo si es asignatura curricular -no electiva-) <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	<input type="checkbox"/> Formación experimental <input type="checkbox"/> Resolución de problemas de ingeniería <input type="checkbox"/> Actividades de proyecto y diseño <input type="checkbox"/> Prácticas supervisadas en los sectores productivos y /o de servicios						
<b>Cantidad de horas cátedras afectadas a la formación práctica indicada en el punto anterior</b> (sólo si es asignatura curricular -no electiva-)	[en el caso de contar con 2 tipos de formación prácticas, indicar cantidad de horas por cada una]						
<b>Descripción de los prácticos</b>	<p>Las actividades prácticas constarán de una guía de ejercicios que el alumno debe realizar solo o en grupo, según se establezca previamente. Algunos de estos ejercicios serán de entrega obligatoria y aportarán a la nota del componente N° 1 del sistema de evaluación.</p> <p>Las actividades prácticas realizadas en el marco de un proyecto existente de software libre a elección del alumno serán pautadas previamente con el profesor, quien a su vez evaluará su ejecución en base a un informe del alumno (componente N° 3 del sistema de evaluación).</p>						
<b>Criterios generales</b> (los cuales serán tenidos en cuenta en las correcciones)	<p>Se evalúa la calidad de presentación de las resoluciones y la capacidad analítica del alumno o grupo para arribar a las conclusiones.</p> <p>Algunos de los conceptos evaluados como parte de las actividades descriptas anteriormente son:</p> <ul style="list-style-type: none"> <li>Precisión Conceptual</li> <li>Capacidad de análisis y síntesis</li> <li>Criterios para transferir los conocimientos adquiridos a situaciones prácticas</li> <li>Cumplimiento de los requerimientos de la actividad</li> <li>El lenguaje técnico utilizado sea el adecuado a la actividad</li> <li>La creatividad de la presentación de la actividad</li> </ul>						
<b>Cronograma de actividades de la asignatura</b> (contemplando las fechas del calendario 2017 y para cada unidad)	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">Día</th> <th style="width: 15%;">Unidad</th> <th style="width: 70%;">Contenido</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	Día	Unidad	Contenido			
Día	Unidad	Contenido					

	1/04	1	Historia del software. Surgimiento del software privativo. Proyecto GNU: el nacimiento del software libre. Conceptos y definiciones de software libre. La Iniciativa Open Source (OSI): conceptualización del software de fuentes abiertas. Historia del software de fuentes abiertas/libre en Latinoamérica. Licencias de software: privativas, permisivas, recíprocas totales, recíprocas parciales. Legislación sobre software de fuentes abiertas/libre.
	8/04	2	El software de fuentes abiertas/libre como movimiento social. Software de fuentes abiertas/libre en la educación. Software de fuentes abiertas/libre en el estado. Modelos de negocios tradicionales y basados en software de fuentes abiertas/libre. Planes de negocios y <i>startups</i> basadas en software de fuentes abiertas/libre. Cooperativismo en el mercado del software.
	22/04	3	Distintas formas de participación en comunidades de software de fuentes abiertas/libre. Sistemas de reputación en proyectos de software de fuentes abiertas/libre y meritocracia. Sistemas de control de versión y su impacto sobre el modelo de contribuciones (SVN, Git, Mercurial, Bazaar). Selección de la comunidad en la cual participar.
	29/04	3	Tipos de comunicación en proyectos de software de fuentes abiertas/libre, ventajas y limitaciones (email, IRC, foros, blogs). Sistemas de seguimiento de Bugs (Bugzilla, Redmine).
	6/05	4	<i>El código auto-documentado. Guías de estilo. Filosofía Unix de Diseño. Refactoring. Patrones de diseño.</i>
	13/05	-	Examen parcial
	20/05	5	Lenguajes de programación libres (GCC, Clang/LLVM, Python). Sistemas operativos libres/de fuentes abiertas (GNU/Linux, FreeBSD, ReactOS).
	27/05	6	Recepción y revisión de <i>patches</i> . Tests automatizados. <i>Triaging</i> de bugs.. Herramientas de <i>build</i> e integración continua
	03/06	-	Recuperatorio del examen parcial
	24/06	-	Presentación de resultados de participación en proyectos.
	1/07	-	Presentación de resultados de participación en proyectos.
<b>Propuesta para la atención de consultas y mail de contacto.</b>	El estudiante podrá contactar al coordinador de la cátedra de la siguiente manera: En la oficina 4 del Edificio Maders lunes y viernes de 18:00 a 20:00, o bien a través de la casilla de email: <a href="mailto:ricardo.h.medel@gmail.com">ricardo.h.medel@gmail.com</a>		
<b>Plan de integración con otras asignaturas</b>	Esta asignatura requiere de los conocimientos desarrollados en las siguientes asignaturas: <ul style="list-style-type: none"> <li>• Diseño de Sistemas</li> </ul>		

<p><b>Bibliografía Obligatoria</b></p>	<ul style="list-style-type: none"> <li>• González Barahona, Jesús, Seoane Pascual, Joaquín, Robles, Gregorio. “Introducción al software libre”, Fundació per a la Universitat Oberta de Catalunya, 2003. ISBN: 84-9788-028-5</li> <li>• Raymond, Eric S., “La catedral y el bazar”, O’Reilly, 1997.</li> <li>• Fogel, Karl. “Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre”, O’Reilly, 2007.</li> </ul>
<p><b>Bibliografía Complementaria</b></p>	<ul style="list-style-type: none"> <li>• Gomez Gomez, Víctor Manuel. “Aspectos Históricos del Software Libre en América Latina”, 2005.</li> <li>• Martin, Robert C. “Clean Code”, 2008. ISBN-10: 0132350882</li> <li>• Gamma, Erich, Helm, Richard, Johnson, Ralph, Vlissides, John. “Design patterns : elements of reusable object-oriented software”, 1994. ISBN-10: 0201633612</li> <li>• Black, Rex. “Managing the Testing Process, 3rd Edition”, Wiley, 2011.</li> <li>• D’Elia Branco, Marcelo. “El Software Libre y sus perspectivas para el desarrollo en América Latina y el Caribe”, <a href="http://www.bellanet.org">http://www.bellanet.org</a>.</li> <li>• Rosa, Fernando da; Heinz, Federico. “Guía práctica sobre software libre: su selección y aplicación local en América Latina y el Caribe”, UNESCO, 2007. ISBN: 92-9089-103-3</li> <li>• Williams, Sam. “Free as in Freedom: Richard Stallman's Crusade for Free Software”, O’Reilly, 2002.</li> <li>• St. Laurent, Andrew M. “Understanding Open Source and Free Software Licensing”, O’Reilly, 2004, ISBN: 0-596-00581-4.</li> <li>• Wheeler, David A. “Make Your Open Source Software GPL-Compatible. Or Else.”, 2014, <a href="http://www.dwheeler.com/essays/gpl-compatible.html">http://www.dwheeler.com/essays/gpl-compatible.html</a></li> <li>• Hernández Gómez, Lorena. “Herramientas software para el trabajo científico colaborativo”, Trabajo de Fin de Máster, Universidad de Salamanca, 2011.</li> <li>• Collins-Sussman, Ben, Fitzpatrick, Brian W., Pilato, C. Michael. “Control de versiones con Subversion”, O’Reilly, 2004.</li> <li>• Loeliger, Jon, McCullough, Matthew. “Version Control with Git, 2nd Edition”, O’Reilly, 2012.</li> <li>• O’Sullivan, Bryan. “Mercurial: The Definitive Guide”, O’Reilly, 2009.</li> <li>• Gyerik, Janos. “Bazaar Version Control”, Packt Publishing, 2013.</li> <li>• Lesyuk, Andriy. “Mastering Redmine”, Packt Publishing, 2013.</li> <li>• Fowler, Martin, Beck, Kent, Brant, John, Opdyke, William, Roberts, Don. “Refactoring: Improving the Design of Existing Code”, 1999. ISBN-10: 0201485672</li> <li>• Sokol, Francisco Zigmund, Aniche, Mauricio Finavaro, Gerosa, Marco Aurélio. “Does the Act of Refactoring Really Make Code Simpler? A Preliminary Study”, Workshop Brasileiro de Métodos Ágeis, 2013.</li> <li>• Stallman, Richard M. “Free Software, Free Society: Selected Essays of Richard M. Stallman”, Free Software Foundation, 200</li> <li>• Zúñiga, Lena. “Voces libres de los campos digitales: Una investigación social sobre el Software Libre en América Latina y el Caribe”, 2006. ISBN: 9977-12-888-X</li> <li>• Salas, Margarita. “Género y software libre en América Latina: Un estudio de caso”, <a href="http://www.sulabatsu.com/voces/Documentos/genero.pdf">http://www.sulabatsu.com/voces/Documentos/genero.pdf</a></li> <li>• Wheeler, David A. “Open Source Software (OSS or FLOSS) and the U.S. Department of Defense (DoD)”, 2009. <a href="http://www.dwheeler.com/essays/dod-oss.pdf">http://www.dwheeler.com/essays/dod-oss.pdf</a></li> </ul>

<b>Distribución de docentes</b>	<i>El docente de la asignatura es el Dr. Ricardo Medel. El horario del curso es Sábado de 8 a 10:30 hs.</i>
---------------------------------	-----------------------------------------------------------------------------------------------------------------

Firma: .....

Aclaración: .....