

MODALIDAD ACADÉMICA

Asignatura	Ingeniería de Software de Fuentes Abiertas/Libre	
Carrera	INGENIERÍA EN SISTEMAS DE INFORMACIÓN	
Ciclo Lectivo	2019	
Vigencia del programa	<i>Desde el ciclo lectivo 2019</i>	
Plan	2008	
Nivel	<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	
Coordinador de la Cátedra	<i>Dr. Ricardo Medel</i>	
Área de Conocimiento	<input type="checkbox"/> Programación <input type="checkbox"/> Computación <input type="checkbox"/> Sistemas de Información <input type="checkbox"/> Gestión Ingenieril <input type="checkbox"/> Modelos <input type="checkbox"/> Complementaria <input checked="" type="checkbox"/> Asignatura Electiva	
Carga horaria semanal	<i>6 horas semanales</i>	
Anual/ cuatrimestral	<i>Cuatrimstral (1° cuatrimestre)</i>	
Contenidos Mínimos, según Diseño Curricular-Ordenanza 1150 (sólo para asignaturas curriculares, no electivas)	N/A	
Correlativas para cursar (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	<ul style="list-style-type: none"> Al menos 1 materia del conjunto PAVII-DLC 	<ul style="list-style-type: none"> Diseño de Sistemas
Correlativas para rendir (según Diseño Curricular-Ordenanza 1150)	Regulares	Aprobadas
	N/A	N/A
Objetivos generales de la Asignatura	<ul style="list-style-type: none"> Entender los conceptos fundamentales, historia y actualidad del software de fuentes abiertas/libre. Comprender los factores que afectan la gestión de proyectos de software de fuentes abiertas/libre y cómo enfrentarlos. Adquirir los conocimientos necesarios para participar de un proyecto de software de fuentes abiertas/libre. 	
<u>Programa Analítico</u>		
Unidad Nro. 1: CONCEPTOS E HISTORIA DEL SOFTWARE DE FUENTES ABIERTAS/LIBRE		
Resultados de Aprendizaje:		
<ul style="list-style-type: none"> Definir los conceptos fundamentales del software libre y de fuentes abiertas a fin de diferenciarlo del software privativo y explicar las diferencias en las modalidades de producción, difusión y apropiación de ambos tipos de software. 		

- Describir el devenir histórico tanto del movimiento de software libre como de la iniciativa del software de fuentes abiertas a fin de situarse en el momento histórico actual y prever las tendencias a futuro de ambas iniciativas.

Contenidos: Historia del software. Surgimiento del software privativo. Proyecto GNU: el nacimiento del movimiento 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.

Bibliografía Obligatoria:

- 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
- Gomez Gomez, Víctor Manuel. “Aspectos Históricos del Software Libre en América Latina”, 2005.

Bibliografía Complementaria:

- 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.
- St. Laurent, Andrew M. “Understanding Open Source and Free Software Licensing”, O’Reilly, 2004, ISBN: 0-596-00581-4.
- OpenExpo Europe, “Tendencias Open Source y Software Libre 2019”, 2019.

Evaluación: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, y en la aplicación de los conocimientos a casos prácticos durante el examen parcial.

Unidad Nro. 2: LICENCIAS DE SOFTWARE Y PROPIEDAD INTELECTUAL

Resultados de Aprendizaje:

- Distinguir los aspectos legales que deben considerarse al producir y/o utilizar software de fuentes abiertas/libre en el contexto de la profesión de la Ingeniería en Sistemas de Información, permitiéndole tomar decisiones acorde a la ley y las buenas prácticas.

Contenidos: Propiedad intelectual y derechos de autor. Herramientas legales para protección de la propiedad intelectual, patentes, licencias, registro como obra. Estado de la propiedad intelectual del software en Argentina y el mundo. Licencias de software privativas y libres, estrictas y permisivas. Compatibilidad de licencias.

Bibliografía Obligatoria:

- 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
- St. Laurent, Andrew M. “Understanding Open Source and Free Software Licensing”, O’Reilly, 2004, ISBN: 0-596-00581-4.

Bibliografía Complementaria:

- Raymond, Eric S., “La catedral y el bazar”, O’Reilly, 1997.
- Wheeler, David A. “Make Your Open Source Software GPL-Compatible. Or Else.”, 2014, <http://www.dwheeler.com/essays/gpl-compatible.html>

Evaluación: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, en la correcta descripción de las licencias y herramientas legales en uso en la comunidad donde realizará la práctica, lo cual debe ser explicitado en los informes parciales que serán solicitados, y en la aplicación de los conocimientos a casos prácticos durante el examen parcial.

Unidad Nro. 3: GESTIÓN Y CALIDAD DE PROYECTOS DE SOFTWARE LIBRE

Resultados de Aprendizaje:

- Definir los factores técnicos, de comunicación y humanos que afectan la gestión de proyectos de software de fuentes abiertas/libre en el marco de su desarrollo por medio de comunidades globales a fin de poder participar en o liderar una comunidad de software libre.
- Describir los diferentes modelos de organización de comunidades donde se desarrolla software libre a fin de poder realizar una participación efectiva en una comunidad de software libre.
- Descubrir cuáles son las herramientas informáticas utilizadas en estas comunidades de modo de ser un miembro productivo en la comunidad seleccionada para la práctica.

Contenidos: Modelos de desarrollo de software libre: comunitarios, unitarios. Modelos de gestión y comunicaciones en comunidades de software libre. Herramientas de desarrollo distribuido: software de control de versiones, infraestructura web para alojamiento de proyectos, sistemas de control y seguimiento de bugs/issues, herramientas de testing automatizado, sistemas de integración continua. Formas de participación en comunidades de software libre.

Bibliografía Obligatoria:

- Fogel, Karl. “Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre”, O’Reilly, 2007.
- Loeliger, Jon, McCullough, Matthew. “Version Control with Git, 2nd Edition”, O’Reilly, 2012.
- Black, Rex. “Managing the Testing Process, 3rd Edition”, Wiley, 2011.

Bibliografía Complementaria:

- 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.
- O’Sullivan, Bryan. “Mercurial: The Definitive Guide”, O’Reilly, 2009.
- 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.

Evaluación: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, en la correcta descripción del modelo de desarrollo y herramientas de gestión utilizadas en la comunidad donde realiza la práctica, lo cual debe ser explicitado en los informes parciales que serán solicitados, y en la aplicación de los conocimientos a casos prácticos durante el examen parcial.

Unidad Nro. 4: MODELOS DE NEGOCIOS CON SOFTWARE LIBRE

Resultados de Aprendizaje:

- Describir cuáles son los posibles modelos de negocio aplicables al software libre y cuáles las características del software libre y del mercado que condicionan qué modelos utilizar, a fin de poder seleccionar el o los modelos de negocios que aseguren la viabilidad comercial de los proyectos de software libre que el ingeniero pueda desarrollar en su vida profesional.

Contenidos: Modelos de negocios tradicionales y basados en software libre. Planes de negocios y startups basadas en software libre. Cooperativismo en la industria del desarrollo de software y servicios informáticos. Casos de estudio de negocios exitosos y fracasos basados en software libre.

Bibliografía Obligatoria:

- Fogel, Karl. “Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre”, O’Reilly, 2007.
- 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

Bibliografía Complementaria:

- D’Elia Branco, Marcelo. “El Software Libre y sus perspectivas para el desarrollo en América Latina y el Caribe”, <http://www.bellanet.org>.

Evaluación: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, en la correcta descripción del modelo de negocios (si existiera) del proyecto de software libre donde realiza la práctica, el cual debe ser explicitado en los informes parciales que serán solicitados, y en la aplicación de los conocimientos a casos prácticos durante el examen parcial.

Unidad Nro. 5: SISTEMAS DE OPERATIVOS LIBRES

Resultados de Aprendizaje:

- Describir los principales sistemas operativos libres utilizados en la industria del software, sus características técnicas y su historia a fin de poder seleccionar los que sean más adecuados a cada proyecto a desarrollar en la vida profesional.

Contenidos: Historia y actualidad de los principales sistemas operativos libres (UNIX, GNU/Linux, familia BSD, ReactOS). Conceptos de los lenguajes de programación más utilizados en el desarrollo de sistemas operativos libres y sus compiladores (GCC, Clang/LLVM). Bases de datos libres (MySQL, PostgreSQL, MongoDB).

Bibliografía Obligatoria:

- Kroah-Hartman, Greg. “Linux Kernel in a Nutshell”, O’Reilly Media, 2006. ISBN : 0-596-10079-5
- Redmond, Eric, Wilson, Jim R. “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, 2012.

Bibliografía Complementaria:

- Gough, Brian J. “An Introduction to GCC”, 2004. ISBN: 0-9541617-9-3
- Lattner, Chris. “The Design of LLVM”, Dr. Dobb’s Journal, Mayo de 2012.
- Dyer, Russell J. T. “MySQL in a Nutshell”, O’Reilly Media, 2008.

Evaluación: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, y en la aplicación de los conocimientos a casos prácticos durante el examen parcial

Unidad Nro. 6: DISEÑO E IMPLEMENTACIÓN DE SOFTWARE LIBRE

Resultados de Aprendizaje:

- Participar como un miembro productivo en una comunidad de software libre con el fin de analizar, valorar y realizar aportes de código y tests de calidad en dicha comunidad.
- Describir las guías de estilo y de diseño utilizadas por las comunidades de software libre para asegurar la calidad de sus proyectos de modo de participar de acuerdo a las reglas de comunidades de software libre o crear la propia comunidad con reglas para un adecuado funcionamiento productivo.

Contenidos: El código auto-documentado. Guías de estilo de diferentes comunidades de software libre. Filosofía Unix de Diseño. Refactoring. Patrones de diseño. Comportamiento esperado de los desarrolladores en un proyecto de software libre.

Bibliografía Obligatoria:

- Fogel, Karl. “Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre”, O’Reilly, 2007.

Bibliografía Complementaria:

- 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: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, en la correcta descripción de las guías de estilo y diseño utilizadas en la comunidad donde realiza la práctica y en la calidad de la participación del estudiante en dicha comunidad, lo cual debe ser explicitado, respectivamente, en los informes parciales y en el final que le serán solicitados, y en la aplicación de los conocimientos a casos prácticos durante el examen parcial .

Unidad Nro. 7: SOFTWARE LIBRE Y SU FILOSOFÍA APLICADOS EN OTROS ÁMBITOS

Resultados de Aprendizaje:

- Describir los aportes de la filosofía del software libre a otros ámbitos (educación, Estado y gobierno, movimientos sociales, cooperativismo) y su retroalimentación, a fin de descubrir nuevos ámbitos de participación, fomentando la producción, uso y apropiación del software libre como medio para mejorar las relaciones sociales y el desempeño económico de la sociedad.

Contenidos: El software libre como movimiento social. La filosofía del software libre en otros ámbitos: educación, Estado, movimientos sociales, cooperativismo. El software libre y su relación con el voto electrónico.

Bibliografía Obligatoria:

- Stallman, Richard M. “Free Software, Free Society: Selected Essays of Richard M. Stallman”, Free Software Foundation, 2001.
- 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

Bibliografía Complementaria:

- 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
- Salas, Margarita. “Género y software libre en América Latina: Un estudio de caso”, <http://www.sulabatsu.com/voces/Documentos/genero.pdf>
- Wheeler, David A. “Make Your Open Source Software GPL-Compatible. Or Else.”, 2014, <http://www.dwheeler.com/essays/gpl-compatible.html>

Evaluación: Según la metodología definida para esta asignatura, los resultados de aprendizaje se evaluarán en base a la participación en clase del estudiante, para lo cual el equipo docente registrará la cantidad y calidad de dichas participaciones, y aplicando los conocimientos a casos prácticos durante el examen parcial.

<p>Metodología de enseñanza y aprendizaje (Planificar estrategias centradas en el aprendizaje activo del estudiante)</p>	<p>Clases participativas Antes de cada clase el cuerpo docente indicará qué conceptos se cubrirán en dicha clase y se seleccionará un grupo de estudiantes para dirigir la discusión. Durante las clases el grupo seleccionado utilizará medios audiovisuales para guiar la discusión sobre los temas establecidos. Estos estudiantes y el cuerpo docente promoverán la participación de todos los estudiantes en la discusión. Con este tipo de clases se busca 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>Práctica en una comunidad de software libre Durante el cuatrimestre, el estudiante participará activamente en una comunidad de software libre ya existente (en la cual no haya participado previamente). Este tipo de actividad está orientada a ejercitar en un ambiente real los temas discutidos durante las clases de la asignatura. Debido a la complejidad de insertarse en tan poco tiempo en una comunidad de software de fuentes abiertas/libre, la participación del estudiante puede ser bien como colaborador en la documentación, como tester o como desarrollador.</p>
<p>Sistema de evaluación (Nombrar y describir cada una de las diferentes instancias de evaluación, pensando en la Evaluación como proceso continuo de recolección de evidencias)</p>	<p>La evaluación de los estudiantes se compone de los siguientes cuatro componentes:</p> <ol style="list-style-type: none"> 1. Participación en clase. El cuerpo docente hará un seguimiento de la participación de cada estudiante en las discusiones durante las clases, tratando de obtener evidencia del aprendizaje en base a la cantidad y calidad de la participación de cada estudiante. Para esto, cada estudiante se deberá identificar antes de intervenir y los docentes tomarán nota de la pertinencia de la participación y el conocimiento demostrado al participar, además de la cantidad de participaciones en clase. 2. Dos reportes parciales y un reporte final sobre la experiencia del estudiante en la comunidad de software libre seleccionada. Los reportes mostrarán la aplicación de las herramientas de análisis a la comunidad bajo estudio y la calidad de la participación del estudiante en dicha comunidad. 3. Un examen parcial integrador y/o su instancia recuperatoria evaluando la comprensión de los conceptos a través de su aplicación en casos prácticos. 4. Presentación oral sobre la participación del estudiante en la comunidad de software libre seleccionada.
<p>Criterios de evaluación (los cuales serán tenidos en cuenta en las correcciones)</p>	<ol style="list-style-type: none"> 1. La participación de los estudiantes en clase se evaluará en base a la cantidad y calidad de las intervenciones de cada estudiante en las clases. Con calidad nos referimos a que no sean conceptos mencionados previamente y que no sean obviedades. 2. Las respuestas en el examen parcial (y/o su recuperación) serán evaluadas respecto a la correcta aplicación de los conceptos en los casos presentados en las preguntas. 3. Los tres reportes (2 parciales y uno final) sobre la participación del estudiante serán evaluados respecto al apego a los aspectos formales (longitud, ortografía, estructura) y a la calidad del contenido en cuanto a

	<p>la profundidad y corrección del análisis de la comunidad de software donde se realizó la intervención.</p> <p>4. La presentación oral y el informe final permitirán evaluar la calidad de la participación del estudiante en la comunidad de software libre: siendo el aporte de código (nuevas funciones o corrección de errores) la participación más valiosa, mientras que el reporte de errores y la colaboración en la documentación siendo menos valiosos.</p>
<p>Regularidad: condiciones (Describir las condiciones necesarias para regularizar. Se sugiere incluir la aclaración que el estudiante en condición de regular puede rendir en el plazo de un ciclo lectivo sin control de correlativas aprobadas)</p>	La asignatura no permite la Regularidad.
<p>Promoción: condiciones (Aclarar si hubiera promoción de alguna parte de la asignatura, las condiciones y si tiene duración, con el mayor detalle posible)</p>	La asignatura no permite la Promoción.
<p>Aprobación Directa: condiciones. (la calificación será la nota registrada como Nota Final en Autogestión) (Se sugiere incluir la aclaración que el estudiante, en esta condición, 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>	<p>La calificación del estudiante en la materia será el promedio de las notas de los cuatro componentes mencionados en la sección Sistema de Evaluación.</p> <p>El estudiante estará en condiciones de Aprobación Directa si su calificación es igual o mayor a 7 (siete).</p> <p>El estudiante en esta condición 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>
<p>Modalidad de examen final (Describir las características metodológicas del examen final para los distintos estados del estudiante)</p>	N/A
<p>Actividades en laboratorio</p>	Las actividades prácticas, de participación en la comunidad de software libre elegida, se realizarán fuera de los horarios de la asignatura.
<p>Cantidad de horas prácticas totales (en el aula)</p>	N/A
<p>Cantidad de horas teóricas totales (en el aula)</p>	60
<p>Cantidad de horas estimadas totales de trabajo (extra áulicas).</p>	30
<p>Horas/año totales de la</p>	90

asignatura (en el aula).	
Tipo de formación práctica (sólo si es asignatura curricular -no electiva-)	<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
Cantidad de horas cátedras afectadas a la formación práctica indicada en el punto anterior (sólo si es asignatura curricular -no electiva-)	N/A
Descripción de los prácticos	<p>Las actividades serán realizadas en el marco de una comunidad de software libre preexistente, a elección del estudiante. El tipo de actividad específica a realizar por el estudiante en dicha comunidad serán pautadas previamente con el profesor. Una actividad que todos los estudiantes realizarán es el análisis, con las herramientas conceptuales brindadas en la asignatura, sobre la gestión de la comunidad, estructura y herramientas utilizadas, modelo de negocios y estándares de desarrollo. La participación en sí del estudiante en la comunidad variará dependiendo del estudiante y la comunidad, pero pueden ser:</p> <ul style="list-style-type: none"> • Codificación de nuevas funcionalidades • Codificación para la corrección de errores • Corrección de errores de configuración del software desarrollado o de las herramientas utilizadas para su desarrollo • Detección de errores en el software desarrollado • Creación de y/o mejoras en la documentación del código, manuales de operación y manuales de usuario • Traducción de documentación, manuales o interfaces de usuario • Mejoras en los protocolos de gestión de la comunidad • Desarrollo o mejora de material educativo basado en el software desarrollado <p>Esta lista se da a modo de sugerencia pero de ninguna manera restringe las actividades posibles de realizarse en la comunidad a las indicadas aquí.</p>

<p>Cronograma de actividades de la asignatura (contemplando las fechas del calendario 2019 y para cada unidad)</p>	<table border="1"> <thead> <tr> <th>Fecha</th> <th>Clase</th> <th>Tema</th> </tr> </thead> <tbody> <tr> <td>23/3/2019</td> <td>1° clase</td> <td>Inducción a la asignatura. Herramientas básicas.</td> </tr> <tr> <td>30/03/2019</td> <td>2° clase</td> <td>Conceptos e historia del software libre.</td> </tr> <tr> <td>06/04/2019</td> <td>3° clase</td> <td>Propiedad intelectual. Licencias de software.</td> </tr> <tr> <td>13/04/19</td> <td>4° clase</td> <td>Gestión de proyectos de SL.</td> </tr> <tr> <td>20/04/19</td> <td>Feriado Semana Santa</td> <td>-</td> </tr> <tr> <td>27/04/19</td> <td>5° clase</td> <td>Modelos de negocios con SL.</td> </tr> <tr> <td>04/05/19</td> <td>6° clase</td> <td>Sistemas operativos de SL.</td> </tr> <tr> <td>11/05/19</td> <td>7° clase</td> <td>Reglas de diseño y programación.</td> </tr> <tr> <td>18/05/19</td> <td>8° clase</td> <td>SL en otros ámbitos.</td> </tr> <tr> <td>25/05/19</td> <td>Feriado</td> <td>-</td> </tr> <tr> <td>01/06/19</td> <td>9° clase</td> <td>Examen parcial.</td> </tr> <tr> <td>08/06/19</td> <td>10° clase</td> <td>Repaso y consultas.</td> </tr> <tr> <td>15/06/19</td> <td>11° clase</td> <td>Recuperatorio examen parcial.</td> </tr> <tr> <td>22/06/19</td> <td>12° clase</td> <td>Presentaciones orales (1° sesión).</td> </tr> <tr> <td>29/06/19</td> <td>13° clase</td> <td>Presentaciones orales (2° sesión).</td> </tr> </tbody> </table>	Fecha	Clase	Tema	23/3/2019	1° clase	Inducción a la asignatura. Herramientas básicas.	30/03/2019	2° clase	Conceptos e historia del software libre.	06/04/2019	3° clase	Propiedad intelectual. Licencias de software.	13/04/19	4° clase	Gestión de proyectos de SL.	20/04/19	Feriado Semana Santa	-	27/04/19	5° clase	Modelos de negocios con SL.	04/05/19	6° clase	Sistemas operativos de SL.	11/05/19	7° clase	Reglas de diseño y programación.	18/05/19	8° clase	SL en otros ámbitos.	25/05/19	Feriado	-	01/06/19	9° clase	Examen parcial.	08/06/19	10° clase	Repaso y consultas.	15/06/19	11° clase	Recuperatorio examen parcial.	22/06/19	12° clase	Presentaciones orales (1° sesión).	29/06/19	13° clase	Presentaciones orales (2° sesión).
Fecha	Clase	Tema																																															
23/3/2019	1° clase	Inducción a la asignatura. Herramientas básicas.																																															
30/03/2019	2° clase	Conceptos e historia del software libre.																																															
06/04/2019	3° clase	Propiedad intelectual. Licencias de software.																																															
13/04/19	4° clase	Gestión de proyectos de SL.																																															
20/04/19	Feriado Semana Santa	-																																															
27/04/19	5° clase	Modelos de negocios con SL.																																															
04/05/19	6° clase	Sistemas operativos de SL.																																															
11/05/19	7° clase	Reglas de diseño y programación.																																															
18/05/19	8° clase	SL en otros ámbitos.																																															
25/05/19	Feriado	-																																															
01/06/19	9° clase	Examen parcial.																																															
08/06/19	10° clase	Repaso y consultas.																																															
15/06/19	11° clase	Recuperatorio examen parcial.																																															
22/06/19	12° clase	Presentaciones orales (1° sesión).																																															
29/06/19	13° clase	Presentaciones orales (2° sesión).																																															
<p>Propuesta para la atención de consultas y mail de contacto.</p>	<p>El estudiante podrá contactar al coordinador de la cátedra de las siguientes maneras:</p> <ul style="list-style-type: none"> • En la oficina 4 del Edificio Maders lunes de 18:00 a 20:00 • A través de las casillas de email: <ul style="list-style-type: none"> ◦ rmedel@frc.utn.edu.ar ◦ ricardo.h.medel@gmail.com 																																																
<p>Plan de integración con otras asignaturas</p>	<p>N/A</p>																																																
<p>Bibliografía Obligatoria</p>	<ul style="list-style-type: none"> • Black, Rex. “Managing the Testing Process, 3rd Edition”, Wiley, 2011. • Fogel, Karl. “Producir software de código abierto: Cómo llevar a buen puerto un proyecto de código libre”, O’Reilly, 2007. • Kroah-Hartman, Greg. “Linux Kernel in a Nutshell”, O’Reilly Media, 2006. ISBN : 0-596-10079-5 • Gomez Gomez, Víctor Manuel. “Aspectos Históricos del Software Libre en América Latina”, 2005. • 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 • Loeliger, Jon, McCulloug, Matthew. “Version Control with Git, 2nd Edition”, O’Reilly, 2012. • Redmond, Eric, Wilson, Jim R. “Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement”, 2012. • St. Laurent, Andrew M. “Understanding Open Source and Free Software Licensing”, O’Reilly, 2004, ISBN: 0-596-00581-4. • tallman, Richard M. “Free Software, Free Society: Selected Essays of Richard M. Stallman”, Free Software Foundation, 2001. • Zúñiga, Lena. “Voces libres de los campos digitales: Una investigación social sobre el Software Libre en América Latina y el Caribe”, 2006. 																																																

Bibliografía Complementaria	ISBN: 9977-12-888-X																	
	<ul style="list-style-type: none"> • Berg, Alan. “Jenkins Continuous Integration Cookbook”, Packt Publishing, 2012. • Collins-Sussman, Ben, Fitzpatrick, Brian W., Pilato, C. Michael. “Control de versiones con Subversion”, O’Reilly, 2004. • D’Elia Branco, Marcelo. “El Software Libre y sus perspectivas para el desarrollo en América Latina y el Caribe”, http://www.bellanet.org. • Dyer, Russell J. T. “MySQL in a Nutshell”, O’Reilly Media, 2008. • 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 • Gough, Brian J. “An Introduction to GCC”, 2004. ISBN: 0-9541617-9-3 • Hernández Gómez, Lorena. “Herramientas software para el trabajo científico colaborativo”, Trabajo de Fin de Máster, Universidad de Salamanca, 2011. • Lattner, Chris. “The Design of LLVM”, Dr. Dobb's Journal, Mayo de 2012. • Martin, Robert C. “Clean Code”, 2008. ISBN-10: 0132350882 • Nelson-Smith, Stephen. “Test-driven Infrastructure with Chef, 2nd Edition”, O’Reilly, 2013. • OpenExpo Europe, “Tendencias Open Source y Software Libre 2019”, 2019. • O’Sullivan, Bryan. “Mercurial: The Definitive Guide”, O’Reilly, 2009. • Raymond, Eric S., “La catedral y el bazar”, O’Reilly, 1997. • 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 • Salas, Margarita. “Género y software libre en América Latina: Un estudio de caso”, http://www.sulabatsu.com/voces/Documentos/genero.pdf • Smart, John Ferguson. “Jenkins: The Definitive Guide”, 2011. ISBN-10: 1449305350 • 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. • Wheeler, David A. “Make Your Open Source Software GPL-Compatible. Or Else.”, 2014, http://www.dwheeler.com/essays/gpl-compatible.html • Williams, Sam. “Free as in Freedom: Richard Stallman's Crusade for Free Software”, O’Reilly, 2002. 																	
Distribución de docentes	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;"><i>Curso</i></th> <th style="text-align: center;"><i>Turno</i></th> <th style="text-align: center;"><i>Día y Horas</i></th> <th style="text-align: center;"><i>Profesor</i></th> <th style="text-align: center;"><i>JTP</i></th> <th style="text-align: center;"><i>Ayudante</i></th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><i>Único</i></td> <td style="text-align: center;"><i>Mañana</i></td> <td style="text-align: center;"><i>Sábado 8-10:30 hs.</i></td> <td style="text-align: center;"><i>Medel</i></td> <td style="text-align: center;">-</td> <td style="text-align: center;">-</td> </tr> </tbody> </table>						<i>Curso</i>	<i>Turno</i>	<i>Día y Horas</i>	<i>Profesor</i>	<i>JTP</i>	<i>Ayudante</i>	<i>Único</i>	<i>Mañana</i>	<i>Sábado 8-10:30 hs.</i>	<i>Medel</i>	-	-
<i>Curso</i>	<i>Turno</i>	<i>Día y Horas</i>	<i>Profesor</i>	<i>JTP</i>	<i>Ayudante</i>													
<i>Único</i>	<i>Mañana</i>	<i>Sábado 8-10:30 hs.</i>	<i>Medel</i>	-	-													

--	--

Firma:

Aclaración: