

## PROPUESTA DE MINOR

La siguiente ficha de proposición de *minor* será evaluada por el Consejo de Escuela para su aprobación. Un *minor* es un paquete autocontenido de cursos electivos coherentemente integrados que, de ser aprobados por el alumno, le permitirán agregar una especialización secundaria a su especialidad principal<sup>1</sup>. Los *minors* serán certificados en un documento diferente a los certificados de licenciatura y especialidad.

<b>Nombre del Minor</b>			
Computación			
<b>Nombre en Inglés</b>			
Computing			
<b>Resultados de Aprendizaje del Minor</b>			
El alumno que siga este Minor aprenderá los conceptos básicos de la computación que le permitirán desarrollar software eficiente de mediana complejidad, entendiendo los fundamentos de la Ciencia de la Computación y utilizando buenas prácticas en el área de su especialidad.			
Según las áreas de los electivos seleccionados, se priorizan ciertos resultados:			
<ul style="list-style-type: none"> <li>– Área de Algoritmos: Entender los fundamentos matemáticos de la Ciencia de la Computación.</li> <li>– Área Software de Sistemas: Programar sistemas cercanos a la máquina, muy eficientes.</li> <li>– Área Ingeniería de Software: Programar usando las mejores prácticas de trabajo en grupo.</li> <li>– Área de Aplicaciones a la Ingeniería: Resolver problemas de Ciencias de la Ingeniería usando computación.</li> </ul>			
<b>Plan de Estudio del Minor</b>			
<b>Cursos Obligatorios del Minor:</b>			
Código	Nombre de Curso	Unidades Docentes	SCT
CC3001	Algoritmos y Estructuras de Datos	10	6

<sup>1</sup> Los *Minors* son parte de las asignaturas electivas de licenciaturas consideradas en el plan de estudios de la Facultad vigente desde el 2007. En total suman 40 U.D., para completarlas, cada alumno puede elegir de los cursos ofrecidos por cualquier Departamento de la Facultad. Para Licenciaturas en Ciencias cada alumno puede elegir electivos de un conjunto de cursos definidos por la especialidad.

Cursos Electivos del Minor: Área de Algoritmos			
Código	Nombre de Curso	Unidades Docentes	SCT
CC3101	Matemáticas Discretas para la Computación	10	6
CC3102	Teoría de la Computación	10	6
CC4102	Diseño y Análisis de Algoritmos	10	6
	Algoritmos Avanzados	10	6
	Estructuras de Datos Avanzados	10	6
Cursos Electivos del Minor: Área de Software de Sistemas			
Código	Nombre de Curso	Unidades Docentes	SCT
CC3301	Programación de Software de Sistemas	10	6
CC4302	Sistemas Operativos	10	6
CC4301	Arquitectura de Computadores	10	6
CC4303	Redes	10	6
Cursos Electivos del Minor: Área de Ingeniería de Software			
Código	Nombre de Curso	Unidades Docentes	SCT
CC3002	Metodologías de Diseño y Programación	10	6
CC4401	Ingeniería de Software	10	6
CC4101	Lenguajes de Programación	10	6
CC5401	Ingeniería de Software II	10	6
CC3201	Bases de Datos	10	6
Cursos Electivos del Minor: Área de Aplicaciones a la Ingeniería			
Código	Nombre de Curso	Unidades Docentes	SCT
CC3003	Computación II	10	6
CC3501	Computación Gráfica, Visualización y Modelación para ingenieros	10	6
CC5502	Geometría Computacional	10	6
<b>Total Unidades Docentes y SCT cursos obligatorios y electivos</b>		<b>40</b>	<b>24</b>

Notas:

- El primer curso del *minor* debe tener como requisitos sólo cursos de Plan Común.
- El diseño del *minor* puede considerar cursos nuevos o ya existentes.

Propuesta elaborado por: José Miguel Piquer

### Breve descripción de los cursos que componen el Minor:

Código	Nombre			
CC3001	Algoritmos y Estructuras de Datos			
Nombre en Inglés				
Algorithms and data structures				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
MA1101, CC1001, MA2001			Obligatorio	
Resultados de Aprendizaje				
<p>Conocer, aplicar y analizar las estructuras de datos y los algoritmos más importantes. Ser capaz de diseñar e implementar aplicaciones utilizando estas estructuras de datos y algoritmos.</p>				

Código	Nombre			
CC3002	Metodologías de Diseño y Programación			
Nombre en Inglés				
Design and programming methodologies				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3001 (S)			Obligatorio	
Resultados de Aprendizaje				
<p>Los alumnos aprenderán los fundamentos necesarios para desarrollar software orientado a objetos fácil de entender, extender y mantener en el tiempo. En particular, aprenderán a diseñar y programar buenos objetos, a usar la herencia sólo cuando ésta provee ventajas reales, a integrar objetos para resolver un problema complejo, a diseñar y resolver problemas usando patrones de diseño, a evaluar diseños usando métricas y a enfrentar desarrollo de software de mediana complejidad usando metodologías estándares. Al final de este curso el alumno debe ser capaz de entender los conceptos más importantes de las etapas de análisis, diseño y programación orientada a objetos y de aplicarlos correctamente en el desarrollo de software de mediana y gran complejidad.</p>				

Código	Nombre			
CC3003	Computación II			
Nombre en Inglés				
Computing II				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC1001			Electivo para otra especialidad	
<b>Resultados de Aprendizaje</b>				
<p>El estudiante será capaz de modelar problemas y soluciones utilizando herramientas computacionales de alto nivel. Los problemas serán referidos a organizaciones y sistemas que puedan modelarse con servidores y clientes, colas, almacenamientos y procesos. Las soluciones suponen modelamiento de los datos del problema y la utilización de herramientas computacionales para su procesamiento.</p>				

Código	Nombre			
CC3101	Matemáticas Discretas para la Computación			
Nombre en Inglés				
Discrete Mathematics for Computer Science				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC1001, MA1101			Obligatorio	
<b>Resultados de Aprendizaje</b>				
<p>El propósito del curso es dotar al estudiante de herramientas matemáticas y formales para enfrentar, analizar y resolver problemas que involucren elementos discretos.</p> <p>Al finalizar el curso el alumno será capaz de:</p> <ul style="list-style-type: none"> <li>- Razonar matemáticamente acerca de tipos de datos y estructuras básicas (números, conjuntos, grafos) usados en algoritmos y sistemas; hacer demostraciones rigurosas usando distintos métodos de demostración.</li> <li>- Modelar y analizar procesos computacionales usando métodos analíticos y combinatoriales (recurrencias, técnicas de conteo, etc.). En particular, derivar expresiones asintóticas cerradas a partir de series y recurrencias que representan el crecimiento de funciones y utilizar modelos de teoría de grafos para resolver problemas de conectividad, recorridos y resolución de restricciones.</li> <li>- Demostrar propiedades elementales de teoría de números y explicar sus aplicaciones a la ciencia de la computación, como por ejemplo en criptografía y algoritmos de hashing.</li> </ul>				

Código	Nombre			
CC3102	Teoría de la Computación			
Nombre en Inglés				
Theory of Computation				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3001, CC3101			Obligatorio	
Resultados de Aprendizaje				
<p>Al finalizar el curso el alumno será capaz de:</p> <ul style="list-style-type: none"> <li>- Comprender lo que es un lenguaje regular y dominar sus representaciones en forma de autómatas finitos y expresiones regulares. Conocer las formas más elementales de demostrar que un lenguaje no es regular. Comprender las aplicaciones prácticas de los autómatas a búsqueda en texto.</li> <li>- Comprender lo que es un lenguaje libre del contexto y dominar sus representaciones en forma de autómatas de pila y gramáticas libres del contexto. Conocer las formas más elementales de demostrar que un lenguaje no es libre del contexto. Comprender las aplicaciones prácticas de esta teoría de parsing.</li> <li>- Comprender que existen problemas que no se pueden resolver por computador, la diferencia entre lenguajes decidibles y aceptables, y conocer algunos ejemplos emblemáticos, en particular el problema de la detención.</li> <li>- Comprender el formalismo de Máquinas de Turing y utilizarlo como modelo de computación. Comprender la Tesis de Church y tener noción de algunos formalismos alternativos.</li> <li>- Comprender el concepto de complejidad computacional y el significado de las clases de problemas P y NP. Comprender el concepto de NP-completitud y conocer varios de los problemas NP-completos más famosos. Comprender el mecanismo para establecer que un problema es NP-completo.</li> </ul>				

Código	Nombre			
CC3201	Bases de Datos			
Nombre en Inglés				
Databases				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3001			Obligatorio	
<b>Resultados de Aprendizaje</b>				
<p>El propósito del curso es que el estudiante domine las técnicas básicas de los sistemas para administrar grandes volúmenes de información:</p> <ul style="list-style-type: none"> <li>- Comprender el modelo relacional y poder aplicarlo para modelar grandes volúmenes de datos.</li> <li>- Poder implementar un modelo en forma computacionalmente eficiente.</li> <li>- Comprender los factores involucrados en las funcionalidades y eficiencia de los sistemas de bases de datos.</li> <li>- Haber enfrentado un problema de complejidad mediana, desde el modelamiento hasta la implementación eficiente.</li> </ul>				

Código	Nombre			
CC3301	Programación de Software de Sistemas			
Nombre en Inglés				
System software programming				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3001			Obligatorio	
<b>Resultados de Aprendizaje</b>				
<p>Este curso busca que los alumnos aprendan a programar en lenguaje C, utilizando las funciones básicas que provee el Sistema Operativo Unix tanto para la administración de sistemas, para hacer programas eficientes y para programación concurrente. Se entregan los conceptos básicos de arquitectura de computadores como direcciones de memoria, notación hexadecimal, representación de enteros, bits, etc. y se enfatizan las funciones de manejo de memoria, Entrada/Salida, sistema de archivos, sockets y threads.</p> <p>Al finalizar el curso el alumno será capaz de escribir y entender programas eficientes en lenguaje de programación C, manejar los conceptos básicos de la programación de sistemas y poder desarrollar sistemas para el Sistema Operativo Unix.</p>				

Código	Nombre			
CC3501	Computación Gráfica, Visualización y Modelación para Ingenieros			
Nombre en Inglés				
Computer graphics, visualization and modeling for engineers				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC1001, MA2601			CFB	
<b>Resultados de Aprendizaje</b>				
<ul style="list-style-type: none"> <li>- Desarrollar las habilidades de modelación, solución y visualización computacional de problemas aplicados que involucran geometrías / escenas tridimensionales complejas y datos asociados.</li> <li>- Enfrentar a los alumnos a problemas interdisciplinarios que requieran del uso de computación gráfica, visualización y modelación computacional de fenómenos físicos y sus resultados, diseño asistido por computador, así como a problemas del área entretenimientos.</li> <li>- Enfrentar a los alumnos al desarrollo de proyectos centrados en una aplicación usando OpenGL, MATLAB y/o algún sistema CAD.</li> <li>- Capacitar a los alumnos en el dominio de algunas de estas técnicas y en el uso de bibliotecas gráficas y herramientas computacionales que las utilicen.</li> </ul>				

Código	Nombre			
CC4101	Lenguajes de Programación			
Nombre en Inglés				
Programming Languages				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3102 / Autor			Obligatorio	
<b>Resultados de Aprendizaje</b>				
<ul style="list-style-type: none"> <li>- El alumno estará capacitado en los lenguajes de programación, tanto a nivel de su semántica y uso, como a nivel de su implementación. Para lo primero, el curso se centra en el estudio de intérpretes de lenguajes de programación, incluyendo en particular los paradigmas funcional y lógico, ya que éstos son en general desconocidos por los alumnos. Asimismo, se va llegando a mecanismos de lenguajes progresivamente más sofisticados a través de su uso y de su implementación (en el lenguaje Scheme). También se entregan elementos básicos relativos a técnicas de compilación e implementación eficiente.</li> <li>- Al terminar el curso, se espera que el alumno tenga un entendimiento claro de la semántica de distintos variantes de lenguajes, siendo capaz de elegir un lenguaje de programación para un problema dado, aplicando los conceptos vistos en clases. Además el alumno tendrá un sólido manejo del lenguaje Scheme.</li> </ul>				

Código		Nombre		
CC4102		Diseño y Análisis de Algoritmos		
Nombre en Inglés				
Design and Analysis of Algorithms				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
FI2003, CC3102, (MA3403/MA3401)/ Autor			Obligatorio	
<b>Resultados de Aprendizaje</b>				
<p>Al finalizar el curso el alumno será capaz de:</p> <ul style="list-style-type: none"> <li>- Comprender el concepto de complejidad de un problema como cota inferior, y conocer técnicas elementales para demostrar cotas inferiores.</li> <li>- Comprender las técnicas de diseño de algoritmos y estructuras de datos para memoria secundaria.</li> <li>- Comprender el concepto de algoritmo avaro (greedy). Ser capaz de diseñar algoritmos de este tipo y de demostrar que obtienen el óptimo.</li> <li>- Comprender el concepto de análisis amortizado de algoritmos. Ser capaz de diseñar algoritmos y estructuras de datos considerando costo amortizado, y ser capaz de analizar este costo.</li> <li>- Ser capaz de diseñar algoritmos eficientes usando la finitud del dominio. Poder distinguir estas situaciones.</li> <li>- Comprender el concepto de algoritmos en línea y competitividad. Ser capaz de diseñar algoritmos en línea y analizar su competitividad.</li> <li>- Comprender el concepto de algoritmos aleatorizados y probabilísticos, y cuándo son relevantes. Ser capaz de diseñar y analizar algoritmos de este tipo.</li> <li>- Comprender el concepto de algoritmos aproximados, y cuándo son relevantes. Ser capaz de diseñar y analizar algoritmos de este tipo.</li> <li>- Comprender el concepto de paralelización en algoritmos. Ser capaz de diseñar y analizar algoritmos paralelos.</li> <li>- Conocer un conjunto significativo de algoritmos y estructuras de datos de mediana complejidad para solución de problemas básicos.</li> </ul>				



Código	Nombre			
CC4301	Arquitectura de Computadores			
Nombre en Inglés				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3301			Obligatorio	
Resultados de Aprendizaje				
Al término del curso el alumno conocerá los elementos de Hardware de un computador.				

Código	Nombre			
CC4302	Sistemas Operativos			
Nombre en Inglés				
Operating systems				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3301			Obligatorio	
Resultados de Aprendizaje				
<ul style="list-style-type: none"> <li>- Este curso busca que los alumnos entiendan cómo funciona y se implementa un sistema operativo, el software que administra el hardware del computador, utilizando como ejemplo el sistema operativo Linux. Se entregan los conceptos básicos de procesos, administración de CPU, de memoria, de E/S y sistemas de archivos.</li> <li>- Al final del curso el alumno será capaz de escribir y entender programas que residen en el kernel de un sistema operativo (drivers), manejar los conceptos básicos de un sistema operativo y la sincronización de procesos.</li> </ul>				

Código	Nombre			
CC4303	Redes			
Nombre en Inglés				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC4302, (CC4401/Autor)			Obligatorio	
Resultados de Aprendizaje				
<ul style="list-style-type: none"> <li>- Este curso tiene como propósito darle al alumno una formación general sobre las redes de datos y los protocolos de comunicación actualmente existentes. En particular, se hará mucho énfasis en los sistemas de transporte independientes del medio físico, como TCP/IP y el funcionamiento de Internet, incluyendo el diseño, la implementación y la programación de aplicaciones basadas en Internet.</li> <li>- Al finalizar el curso el alumno será capaz de: diseñar y programar aplicaciones distribuidas en TCP/IP, comprendiendo el detalle del funcionamiento del sistema, lo que le permite también hacer ingeniería de redes, diseñando sistemas de tamaño mediano.</li> </ul>				

Código	Nombre			
CC4401	Ingeniería de Software			
Nombre en Inglés				
Software Engineering				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC3002, CC3201			Obligatorio	
Resultados de Aprendizaje				
<ul style="list-style-type: none"> <li>- El alumno estará capacitado para enfrentar el desarrollo de proyectos de desarrollo de software medianos a grandes, de una manera metódica y sistemática, aplicando técnicas apropiadas en cada etapa del desarrollo de software.</li> <li>- Al final del curso será capaz de: - escoger un modelo de ciclo de vida apropiado a las características del proyecto que quiere realizar; - escoger una notación apropiada para la especificación de los requisitos del software; - escoger el patrón de diseño arquitectónico más apropiado de acuerdo con los requisitos funcionales y de calidad se hayan establecido y finalmente, aplicar las técnicas de pruebas del software necesarias de acuerdo con los requisitos del sistema.</li> </ul>				

Código	Nombre			
CC5401	Ingeniería de Software II			
Nombre en Inglés				
Software Engineering II				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3	1,5	5,5
Requisitos			Carácter del Curso	
CC4401			Obligatorio	
<b>Resultados de Aprendizaje</b>				
<p>Al finalizar el curso el alumno deberá estar familiarizado con el desarrollo de sistemas de software reales.</p> <p>El alumno deberá ser capaz de:</p> <ul style="list-style-type: none"> <li>- Formular, planificar y gestionar un proyecto de software.</li> <li>- Trabajar en equipo, en forma coordinada, asumiendo un rol específico.</li> <li>- Identificar y administrar los riesgos del proyecto.</li> <li>- Desarrollar un sistema de información pequeño.</li> <li>- Asegurar la calidad del software ofrecido.</li> </ul>				

Código	Nombre			
CC5502	Geometría Computacional			
Nombre en Inglés				
Computational Geometry				
SCT	Unidades Docentes	Horas de Cátedra	Horas Docencia Auxiliar	Horas de Trabajo Personal
6	10	3		7
Requisitos			Carácter del Curso	
CC4102 / Autor			Electivo	
<b>Resultados de Aprendizaje</b>				
<ul style="list-style-type: none"> <li>- Al final del curso el alumno conocerá las técnicas más usadas para el modelamiento de objetos reales y podrá aplicarlos para resolver nuevos problemas.</li> <li>- Este curso tiene por objetivo estudiar y analizar algoritmos geométricos bi y tridimensionales (2-D y 3-D) para modelar objetos reales, tanto desde el punto de vista teórico como práctico. Estos algoritmos permiten modelar objetos para visualizarlos y/o simularlos numéricamente. Pueden ser usados en simulaciones numéricas de dispositivos y procesos, en representaciones de mapas en sistemas de información geográficos y para el reconocimiento de imágenes.</li> </ul>				