

SAN LUIS, 16 de DICIEMBRE 1998

VISTO:

El Expte. "D"-3-594/98, mediante el cual el Departamento de Informática eleva el nuevo plan de estudios de la carrera: Licenciatura en Ciencias de la Computación ; y

CONSIDERANDO:

Que el Departamento de Informática ha advertido que el desarrollo de la carrera: Licenciatura en Ciencias de la Computación, si bien mantiene un nivel académico adecuado, posee indicadores negativos necesarios superar con el fin de brindar una oferta educativa de calidad, que satisfaga los requerimientos profesionales y científicos actuales.

Que el Departamento de Informática considera necesario efectuar modificaciones, debido a la actualización de conocimientos, incorporación de nuevas áreas temáticas, consolidación de los conceptos básicos de la disciplina con apertura y flexibilización de las orientaciones a través del dictado de asignaturas optativas.

Que el presente plan responde a la propuesta que el Departamento de Informática ha realizado ante el Fondo para el Mejoramiento de la Calidad Educativa (FOMECE).

Que conforme a las nuevas necesidades el plan incorpora asignaturas optativas que favorecerán la formación de acuerdo a las preferencias vocacionales de los alumnos.

Que el nuevo plan de estudios entrará en vigencia a partir del ciclo lectivo 1999.

Por ello y en virtud de lo acordado en su sesión del día 16 de Diciembre

**EL CONSEJO DIRECTIVO DE LA FACULTAD DE CIENCIAS
FISICO-MATEMATICAS Y NATURALES**

RESUELVE:

ARTICULO 1º.- Aprobar el Plan de Estudios de la carrera "LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN" de la Facultad de Ciencias Físico-Matemáticas y Naturales de la Universidad Nacional de San Luis, cuyo ordenamiento curricular obra en el ANEXO I de la presente disposición.

ARTICULO 2º.- Disponer que el presente plan de estudios entrará en vigencia en el ciclo lectivo 1999.

CDE. ORDENANZA N° 011-98

MBV
NNH

ARTICULO 3º.- Fijar los siguientes objetivos para la carrera “Licenciatura en Ciencias de la Computación” .

Objetivos Generales:

- Manejar con idoneidad los conceptos, principios y teorías que constituyen el ámbito de la competencia.
- Aplicar las metodologías adecuadas para las soluciones de los problemas de la especialidad y las formas de validación más coherentes.
- Desarrollar una permanente actitud crítica en la exploración del uso de la herramienta matemática y la flexibilidad en el uso de los sistemas de inferencia específicas de su campo.
- Transferir los conocimientos adquiridos a situaciones prácticas diversas.
- Comprender y usar la literatura científica actual y las técnicas experimentales del momento en el desempeño de sus profesiones.
- Investigar por sí, o bajo la supervisión de otros, las problemáticas actuales con referencia a las necesidades de la región, del país y del mundo.

Objetivos Específicos

Al término de sus estudios el egresado deberá estar capacitado para:

- Conocer y asesorar acerca del funcionamiento de equipos computacionales.
- Formular programas de computación, de diferentes niveles de complejidad.
- Elaborar y conducir proyectos de desarrollo de software de diversa complejidad.
- Analizar los resultados que brindan metodologías y técnicas alternativas de aplicación en un mismo campo o área, para aconsejar sobre la utilización de una de ellas.
- Seleccionar metodologías y técnicas que se ajusten al ámbito del problema estudiado.

CDE. ORDENANZA N° 011-98

MBV
NNH

- Evaluar resultados que surgan de la aplicación de las metodologías y técnicas.
- Diseñar algoritmos que resuelvan problemas de naturaleza diversa.
- Seleccionar enfoques, estructuras de programas y de datos más convenientes para cada aplicación.
- Concretar emprendimientos en entornos centralizados, distribuidos y/o paralelos.

ARTICULO 4°.-Establecer el siguiente perfil del Licenciado en Ciencias de la Computación:

El Licenciado en Ciencias de la Computación es un egresado con sólida formación en Programación, Estructuras de Datos y Algoritmos, Organización de Archivos, Bases de Datos, Arquitecturas, Sistemas Operativos, Redes, Sistemas Distribuidos, Paralelismo, Ingeniería de Software, Lenguajes, Construcción de Compiladores y Herramientas Inteligentes, lo que le brinda un marco conceptual que le permitirá:

- Desarrollarse sólidamente en el campo profesional, adaptándose con facilidad a los rápidos cambios tecnológicos propios de la disciplina.
- Iniciarse, si su vocación así lo requiriese, a la investigación científica y desarrollos propios de su área de interés.
- Comprender, utilizar y construir los esquemas y herramientas formales que se requieren para diseñar, desarrollar, implementar y validar los sistemas de software de alta calidad y complejidad que actualmente se requieren.

ARTICULO 5°.- Establecer las siguientes incumbencias:

Licenciatura en Ciencias de la Computación

- El título de **Licenciado en Ciencias de la Computación** habilitará para actuar profesionalmente tanto en industrias como en organismos nacionales y privados de todo el país.
- En la actividad profesional tanto independiente como en relación de dependencia podrá:

CDE. ORDENANZA N° 011-98

MBV
NNH

- Planificar, dirigir y co-auditar Proyectos de Desarrollo de Software de cualquier escala y tipo.
- Diseñar, desarrollar, validar y mantener programas de base y de aplicación (software).
- Efectuar estudios técnicos-computacionales de proyectos que involucren uso de computadoras.
- Promover las aplicaciones de la Informática a nuevas áreas. Desempeñarse en docencia e investigación Universitarias.

Analista-Programador Universitario:

- El título intermedio de **Analista-Programador Universitario** habilitará para actuar profesionalmente tanto en industrias como en organismos nacionales y privados de todo el país.
- En la actividad profesional tanto independiente como en relación de dependencia podrá:
 - Integrar equipos de Proyectos de Software, de cualquier escala, en las etapas de Análisis, Desarrollo e Implantación de Software de Aplicación.
 - Utilizar idóneamente diferentes herramientas de desarrollo de software.
 - Diseñar algoritmos que resuelvan problemas de naturaleza diversa y , posteriormente, codificarlos utilizando un lenguaje de programación adecuado.
 - Realizar el mantenimiento de programas de aplicación.
 - Asesorar acerca del funcionamiento del parque computacional de uso común en el mercado.

ARTICULO 6°.- El alumno que haya aprobado la totalidad del presente Plan de Estudios, se hará acreedor al título de **Licenciado en Ciencias de la Computación**, la orientación que hubiese seleccionado a través de las materias optativas cursadas, quedará reflejada en el correspondiente Certificado de Egreso consignando la denominación de las optativas.

ARTICULO 7°.- El título de Licenciado en Ciencias de la Computación habilitará para inscribirse en las Maestrías que brinde la Institución y/o en el Doctorado en Ciencias de la Computación.

CDE. ORDENANZA N° 011-98

MBV
NNH

ARTICULO 8°.- El alumno que haya aprobado hasta el 3° año completo del Plan de Estudios de la carrera de Licenciatura en Ciencias de la Computación, será acreedor al título intermedio de: **Analista-Programador Universitario**.

ARTICULO 9°.- Establecer los contenidos mínimos para las asignaturas correspondientes al **Ciclo Obligatorio** de la carrera Licenciatura en Ciencias de la Computación, obrantes en el ANEXO III de la presente disposición.

ARTICULO 10°.- Establecer los contenidos mínimos para el **Ciclo Optativo**, obrantes en el ANEXO IV.

Esta propuesta es para el inicio del Plan de Estudios, posteriormente podrán sufrir modificaciones acordes con las disponibilidades y necesidades del campo profesional, tecnológico y científico.

Las modificaciones serán propuestas por el Departamento de Informática, deberán aprobarse por el Consejo Directivo de la Facultad y homologarse por Consejo Superior.

ARTICULO 11°.- Establecer el régimen de correlatividades del Ciclo Obligatorio y del Ciclo Optativo, obrantes en los ANEXOS I y II respectivamente, de la presente disposición.

La **distribución de la carga horaria** es la siguiente:

Materias Básicas Específicas	Materias Básicas No Específicas	Materias Optativas	Trabajo Final de Licenciatura	TOTAL
2245 hs. (64,4%)	690 hs. (19,8%)	400 hs. (11,5%)	150 hs. (4,3%)	3485 hs.
Teórico - Práctico de Laboratorio	Teórico - Práctico de Aula	Teórico - Práctico de Laboratorio	Desarrollo teórico y/o experimental	

El régimen de correlatividades reconoce dos razones fundamentales:

1. La necesidad de asegurar los contenidos previos, como prerequisites para el cursado de algunas materias, tanto obligatorias como optativas. Este tipo de correlatividades son las denominadas, en el plan, correlativas para rendir y se materializan a través de la exigencia de aprobación de las asignaturas, que en cada caso se indican.
2. La necesidad de dotar de un ritmo de estudios que le permita al alumno finalizar la carrera en el tiempo previsto. Este tipo de correlatividades se denomina correlatividades para cursar. Las mismas se materializan a través de la exigencia de la aprobación de los trabajos prácticos y/o laboratorios que en cada caso se indican.

CDE. ORDENANZA N° 011-98

MBV
NNH

ARTICULO 12°.- Antes de comenzar a cursar materias de 4° año el alumno debe acreditar la aprobación de INGLÉS, dictado por el Centro de Lenguas Extranjeras (CELEX) de la UNSL ó solicitar equivalencias presentando las certificaciones correspondientes ó sometiéndose a la prueba de capacidad que requiera el CELEX.

ARTICULO 13°.- Establecer las equivalencias automáticas entre asignaturas de la carrera: Licenciatura en Ciencias de la Computación (Plan Ord. 3/79-R y 3/98-CD) y las del presente plan de estudios, obrantes en el ANEXO V.

Pautas para el otorgamiento de equivalencias:

1. Equivalencia automática para los alumnos de primer y segundo año, acorde con las equivalencias dadas.
2. Cambio de plan a elección de los alumnos para aquéllos que así lo desearan adaptándose a cursar las materias equivalentes del plan viejo que tengan equivalencia en el nuevo, en el plazo de tres años en el cual finalizará la vigencia del Plan de Estudios anterior, en cuanto al dictado de sus materias.
3. Los alumnos que opten por no cambiar de Plan de Estudios deberán concluir de cursar las materias de tercer, cuarto y quinto año en los tres años siguientes al de iniciación del nuevo Plan de Estudios, incluido 1999.
4. Los casos excepcionales no contemplados en la presente disposición, serán analizados por una Comisión Ad-Hoc, que se guiará por el principio de favorecer al alumno, en las situaciones que sean académicamente razonables o bien complementando con coloquios o exámenes.

ARTICULO 14°.- Establecer como Reglamento para la realización y aprobación del Trabajo Final de la Licenciatura en Ciencias de la Computación, el obrante en el ANEXO VI de la presente disposición.

ARTICULO 15°.- Elevar la presente Ordenanza al Consejo Superior de la Universidad Nacional de San Luis para su ratificación (Art. 85 inciso g del Estatuto Universitario).

ARTICULO 16°.- Comuníquese, insértese en el Libro de Ordenanzas y archívese.

CDE. ORDENANZA N° 011-98

MBV
NNH

ANEXO I
Ordenamiento Curricular y Correlatividades del Ciclo Obligatorio

Materia	Correlativa para cursar REGULAR	Correlativa para rendir APROBRADA	Crédito Horario Semanal	Crédito Horario Total
PRIMER AÑO				
PRIMER CUATRIMESTRE				
1 . Elementos de Algebra			8hs.	120 hs.
2 . Introducción a la Computación			6hs	90 hs.
3 . Introducción a la Matemática			6hs	90 hs.
SEGUNDO CUATRIMESTRE				
4 . Análisis Matemático I	3	3	8 hs	120 hs.
5 . Matemática Discreta	1	1	8 hs.	120 hs.
6 . Programación I	2	2	9 hs.	135 hs.
SEGUNDO AÑO				
PRIMER CUATRIMESTRE				
7 . Análisis Matemático II	4	4	9 hs	135 hs.
8 . Elementos de Probabilidad y Estadística	4	4	7 hs.	105 hs.
9 . Programación II	6	6	9 hs	135 hs.
SEGUNDO CUATRIMESTRE				
10. Ingeniería de Software I	9		8 hs.	120 hs.
11. Estructuras de Datos y Algoritmos	5-7-8 - 9	5-8	9 hs	135 hs.
12. Arquitectura del Procesador I	9	9	8 hs	120 hs.
TERCER AÑO				
PRIMER CUATRIMESTRE				
13. Ingeniería de Software II	10	10	8 hs	120 hs.
14. Organización de Archivos y Bases de Datos I	11	11	9hs.	135 hs.
15. Análisis Comparativo de Lenguajes	11-12	11-12	8 hs.	120 hs.
SEGUNDO CUATRIMESTRE				
16. Arquitectura del Procesador II	12	12	6 hs.	90hs
17. Simulación	7-8	7-8	6 hs.	90hs.
18. Sistemas Operativos y Redes	12-14	12	9 hs.	135hs.
19. Planeamiento Informático	13	13	4 hs.	70hs.
CUARTO AÑO				

CDE. ORDENANZA N° 011-98

MBV
NNH

PRIMER CUATRIMESTRE				
20. Lógica para Computación	5	5	6 hs.	90 hs.
21. Sistemas Distribuidos y Paralelismo	18	18	7 hs.	105 hs.
22. Autómatas y Lenguajes	15	15	7 hs.	105 hs.
23. Inteligencia Artificial	11-17	11-17	6 hs.	90 hs.
SEGUNDO CUATRIMESTRE				
24. Diseño y Construcción de Compiladores	12-15-18-22	12-15-22	10 hs.	150 hs.
25. Computabilidad y Complejidad	22		6 hs.	90 hs.
26. Base de Datos II	14-20	14	8 hs.	120 hs.
QUINTO AÑO				
Optativas	ver ANEXO II			400 hs.
Trabajo Final	ver ANEXO VI			150 hs.
total hs.				3485hs.

ANEXO II

Régimen de Correlatividades del Ciclo Optativo

Materia	C/C	C.P/R	C/H	Cr H
1. Aprendizaje de Máquina	24 (*)	19(*)-20(*)	7 hs.	105hs.
1. Inteligencia Artificial Distribuida	24(*)-25(*)	24(*)-25(*)	7 hs.	105hs.
1. Computación Evolutiva I	2		6 hs.	90hs.
1. Computación Evolutiva II	3	2	6 hs.	90hs.
1. Redes Neuronales	1-2 y 20(*)	1-2 y 20(*)	6 hs.	90hs.
1. Sistemas Expertos	20(*)-23(*)	20(*)-23(*)	7 hs.	105hs.
1. Ingeniería del Conocimiento	23 (*)	23 (*)	6 hs.	90hs.
1. Sistemas Inteligentes	4-5-6	4-5	7 hs.	105hs.
1. Semántica Formal de los Lenguajes de Programación.	20(*)-24(*)	20(*)-24(*)	7 hs.	105hs.
1. Teoría de la Información	8(*)-16 (*)	8(*)-16 (*)	6 hs.	90hs.
1. Teoría de Grafos y sus Aplicaciones.	11 (*)	11 (*)	7 hs.	105hs.
1. Computación Paralela: Fundamentos	20(*)-21(*)-25(*)	20(*)-21(*)-25(*)	8 hs.	120hs.

CDE. ORDENANZA N° 011-98

MBV
NNH

1. Estudio Comparativo de Metodologías	13 (*)	13 (*)	6 hs.	90hs.
1. Arquitecturas de Sistemas de Software	13 (*)	13(*)	6 hs.	90hs.
1. Métodos Formales y Análisis de Herramientas para la Producción de Software	13-14 y 20(*)	20 (*)	8 hs.	120hs.
1. Diseño de Software de Tiempo Real	21 (*)	21 (*)	8 hs.	120hs.

OBSERVACIÓN: El (*) se refiere a la numeración de las asignaturas del ciclo obligatorio, ver ANEXO I.

ANEXO III

CONTENIDOS MÍNIMOS DEL CICLO OBLIGATORIO

1. **ELEMENTOS DE ALGEBRA:** Conjuntos. Relaciones Binarias. Relaciones de Orden y Equivalencia. Funciones. Representación de números en distintas bases y operaciones. Principio de Inducción. Números complejos. Polinomios y Ecuaciones Algebraicas. Cálculo combinatorio. Sumas y Productos indexados. Sistemas de Ecuaciones lineales, matrices y determinantes.

2. **INTRODUCCION A LA COMPUTACION:** Introducción al manejo de computadoras. Problemas, modelos y abstracciones: Representación de problemas; Búsqueda de soluciones a problemas; Inferencia, analogía, similitud entre problemas, generalización, particularización. Algoritmo: concepto; algoritmos computacionales; lenguaje de diseño de algoritmos; datos y acciones; datos de entrada y datos de salida; acciones primitivas; estructuras de control. Resolución de problemas: Métodos de resolución de problemas, descripción utilizando un lenguaje de diseño de algoritmos; refinamiento paso a paso. Programación: breve descripción de una computadora; introducción a la programación estructurada utilizando lenguajes didácticos. *Laboratorios sugeridos:* los necesarios para la ejecución y depuración de los programas escritos en los lenguajes didácticos que se utilicen.

3. **INTRODUCCION A LA MATEMATICA:** Revisión de los conceptos de números: naturales, enteros, racionales, expresiones decimales. Potencias y raíces. Números complejos. Revisión de conceptos algebraicos: ecuaciones, sistemas e inecuaciones. Valor Absoluto. Ecuaciones de segundo grado. Expresiones Algebraicas. Revisión del concepto de función: las funciones y sus gráficas, expresión algebraica de una función, funciones representadas mediante rectas, la parábola. Revisión de conceptos trigonométricos. Noción intuitiva sobre máximos y mínimos relativos y absolutos. Funciones periódicas. Función exponencial y logarítmicas. Cónicas. Lenguaje matemático.

4. **ANALISIS MATEMATICO I:** Funciones reales de una variable. Sucesiones y series. Límite. Continuidad. Derivación. Integración. Aproximación de funciones.

5. **MATEMATICA DISCRETA:** Introducción a la Teoría de Grafos: grafos no dirigidos, caminos, ciclos y conectividad; caminos Eulerianos, circuitos Hamiltonianos; isomorfismos. Árboles: mínima spanning tree. Nociones de Álgebra Universal. Reticulados Distributivos. Algebras de Boole. Estructuras Algebraicas:

Monoides, grupos, anillos, cuerpos. Espacios Vectoriales: Transformaciones lineales, matriz asociada, transformaciones lineales simétricas, autovalores y autovectores.

6. **PROGRAMACION I:** Paradigma de programación Imperativa: Estructuras de control a nivel de sentencias. Variables simples y estructuradas, tipos de datos predefinidos y definidos por el usuario y sus usos en resolución de problemas, registros, archivos. Estructuras de control a nivel de unidades, parámetros formales y actuales. Su uso en programas en un lenguaje que responda al paradigma.

Laboratorios sugeridos: Implementación y corrida de programas cortos usando un lenguaje imperativo.

7. **ANALISIS MATEMATICO II:** Funciones reales de varias variables. Límite. Continuidad. Diferenciación en 2 ó más variables. Funciones implícitas. Extremos Vinculados. Integrales múltiples. Ecuaciones Diferenciales.

8. **ELEMENTOS DE PROBABILIDAD Y ESTADISTICA:** Conceptos probabilidad elemental: Experimentos, sucesos, suma y producto de sucesos, sucesos condicionales, sucesos excluyentes, espacio muestral, sucesos independientes. Estadística descriptiva: variables aleatorias discretas y continuas, distribuciones discretas y continuas. Teorema del límite: enunciados y aplicaciones, estimación de parámetros, inferencia estadística, prueba de hipótesis, regresión lineal, análisis de varianza. Introducción a los procesos de Markov.

9. **PROGRAMACION II:** Estructuras dinámicas y árboles. Recursividad. Encapsulamiento de datos. Tipos de datos abstractos. Introducción a la verificación de programas. Paradigma de la programación funcional: uso de funciones de orden superior, evaluación perezosa. Resolución de problemas usando un lenguaje que responda al paradigma. Paradigma de programación Orientado a Objetos: Elementos básicos y avanzados. Resolución de problemas usando un lenguaje que responda al paradigma.

Laboratorios sugeridos: Programas que implementen estructuras dinámicas y tipos de datos abstractos. Implementación de recursividad. Implementación y corrida de programas cortos utilizando lenguajes que respondan a los diferentes paradigmas de programación.

10. **INGENIERIA DE SOFTWARE I:** Naturaleza y cualidades del software. Principios de la Ingeniería del Software. Diseño de software. Especificación de software. Verificación de software. El proceso de producción de software. El gerenciamiento

de la producción de software. Herramientas y ambientes de la Ingeniería de Software.

Laboratorios sugeridos: Implementación de un prototipo para una especificación dada. Uso de herramientas formales. Dado un conjunto de especificaciones informales producir un conjunto de especificaciones formales para el mismo problema. Dada una presentación informal de un problema realice un análisis de requerimientos para el proyecto de implementación y produzca un documento de análisis de requerimientos.

11. ESTRUCTURAS DE DATOS Y ALGORITMOS I: Grafos dirigidos. Digrafos, Problemas de Caminos en los digrafos. Aplicaciones: Ruteo en Redes de Comunicación. Estructuras de Información básicas: definición, uso, implementaciones contiguas y linkeadas. Adecuación a los requerimientos del problema. Análisis de complejidad: análisis asintótico de límites superior y promedio, balance entre tiempos vs. espacio en los algoritmos. Clases de complejidad: P, NP P-space, problemas tratables e intratables, soluciones aproximadas para problemas intratables, análisis de límite inferior NP-completos. Ordenamiento: $O(n^2)$ y $O(n \log n)$ algoritmos de ordenamiento, complejidad espacio - tiempo, mejor - peor casos. Búsqueda serial, binaria, complejidad espacio-tiempo, mejor - peor casos. Hashing, resolución de colisiones.

Laboratorios sugeridos: Uso de los distintos tipos de estructuras en aplicaciones típicas de computer science. Estudio comparativo de las diferentes estrategias de implementación. Diseño de nuevas estructuras y su implementación. Corroboración de la complejidad teórica de algoritmos de ordenamiento y búsqueda por métodos experimentales, identificando diferencias entre los comportamientos del mejor, promedio y peor casos.

12. ARQUITECTURA DEL PROCESADOR I: Sistemas digitales: sistemas combinatorios y secuenciales, autómatas; circuitos básicos. Representación de la información: alfanumérico, punto fijo y flotante, representación de signo, complemento a 1 y a 2, etc.. Estructura de una computadora: máquina Von Neumann, máquina algorítmica (Gluskov), UCP, ALU, memorias, periféricos, procesadores de entrada-salida, buses. Assemblers, registros accesibles al programador, ciclos de búsqueda, ejecución de instrucción, buses internos, mecanismos de acceso a memorias, memorias entrelazadas, formato y conjunto de instrucciones, direccionamiento, subrutinas, interrupciones y excepciones.

Laboratorios sugeridos: Programación en assembler, detección de errores o debugging, en su defecto, usando simuladores.

13. INGENIERIA DE SOFTWARE II: Paradigmas de la Ingeniería de Software. Introducción a la administración de proyectos de Software. Análisis de requerimientos. Distintas corrientes metodológicas incluidas en la Ingeniería del Software. Introducción a las métricas de Software. Introducción al aseguramiento de Calidad de Software.

Laboratorios sugeridos: Construir un diseño usando una de las metodologías estudiadas para un conjunto dado de especificaciones. Implementar el diseño mencionado anteriormente.

14. ORGANIZACION DE ARCHIVOS Y BASES DE DATOS I: Organizaciones básicas. Organizaciones híbridas. Evaluación del Sistema de Archivos. Técnicas. Visión global, modelos y aplicaciones de bases de Datos. Historia y motivación. Componentes de un sistema de base de datos. Modelización conceptual: entidad-relación, orientado a objetos. Funciones soportadas por un sistema típico de bases de datos. El modelo relacional: terminología, transformación de un esquema conceptual a uno relacional, representación de relaciones, integridad de la entidad y referencial. Lenguajes de consulta. Teoría de diseño de bases relacionales. Álgebra relacional. Un caso de estudio.

Laboratorios sugeridos: Interacción con un sistema administrador de base de datos para que los estudiantes creen una pequeña base de datos y evalúen cómo el sistema soporta las funciones introducidas en las teorías. Usando un lenguaje procedural implementar la operación de join del álgebra relacional, intentar al menos dos implementaciones a los efectos de demostrar la eficiencia relativa de las diferentes técnicas.

15. ANALISIS COMPARATIVO DE LENGUAJES: Historia y espectro de los lenguajes de programación. Evolución de los paradigmas de programación. Máquinas virtuales. Estudio de las características esenciales de los lenguajes de programación: tipos de datos y su representación, control de secuencia, binding, etc.. Sistemas de traducción de lenguajes. Introducción a la especificación formal de semántica de los lenguajes de programación: breve descripción de la semántica axiomática, de notacional y operacional. Paradigmas de programación: Funcional y aspectos teóricos subyacentes: teoría de funciones recursivas; Imperativo y aspectos teóricos subyacentes: Sistema de prueba Hoare/Kijkstra (pre/post conditions); Lógico y aspectos teóricos subyacentes: cálculo de predicado de primer orden; Paralelo y aspectos teóricos del álgebra de procesos; algebraico y aspectos teóricos de la teoría de categorías.

16. ARQUITECTURA DEL PROCESADOR II: Arquitecturas Alternativas: procesadores RISC, procesadores de alta performance: pipeline, procesamiento

CDE. ORDENANZA N° 011-98

MBV
NNH

paralelo, procesamiento vectorial. Arquitecturas no Von Neumann: data flow, reduction machines. Arquitecturas multiprocesadores.

17. SIMULACION: Introducción. Definición. Propiedades. Clasificación. Formulación del problema. Los modelos matemáticos. Validación. Análisis de Datos. Técnicas de generación de números aleatorios. Teoría de Colas. Simulación de Sistemas. Lenguajes de Simulación. Diseño de los experimentos de simulación y análisis de los resultados.

Laboratorios sugeridos: Implementación de distintos problemas de simulación usando un lenguaje de simulación.

18. SISTEMAS OPERATIVOS y REDES: Historia, evolución y filosofía. Métodos de estructuración: modelos de capas y cliente-servidor de objetos. Tareas y procesos: definición, bloques de control, listas ready, dispatching, context switch. Coordinación y sincronización de procesos. Deadlocks: causas, condiciones, prevención; modelos y mecanismos. Scheduling and dispatching. Administrador de memoria. Administrador de dispositivos. Sistema de Archivos. Seguridad. Protección. Un caso de estudio. Redes de computadoras: Transmisión de datos, frecuencia, espectro, bandwidth; señal como función del tiempo, señal como función de la frecuencia. Representación de señales periódicas por medio de series de Fourier, transmisión analógica y digital. Arquitectura de Redes Locales, topologías más comunes. Internetworking. Protocolos de Internet.

Laboratorios sugeridos: Diseño e implementación de un context switcher simple y múltiples tareas usando un reloj para causar context switch hecho en un lenguaje de alto nivel. Implementación de algún mecanismo de sincronización de accesos y probar la falta de conflictos. Correr varias mezclas de jobs bajo varios algoritmos de scheduling y analizar los resultados (a través de simulaciones).

19. PLANEAMIENTO INFORMÁTICO: Aspectos conceptuales y habilidades adquiridas durante el desarrollo: elaboración del planeamiento de proyectos de software, análisis de riesgo, establecimiento de cronogramas, determinación de costos y habilidad para efectuar un correcto control presupuestario. Aplicación de los conceptos relacionados con el Modelo de Madurez. Uso adecuado de los principios y criterios de "métricas de software". Integración de soluciones provenientes de diversas fuentes con desarrollos propios. Aspectos legales relacionados con los proyectos y productos de software. Distinción de la finalidad y limitaciones de los estándares vigentes para la producción de software, uso de los estándares para documentar proyectos de software.

20. **LOGICA PARA COMPUTACION:** Razonamiento válido; formas lógicas, lenguaje objeto y metalenguaje. Introducción a la lógica proposicional. Introducción a la lógica de predicados. Elementos de prueba (incluyendo inducción y contradicción). Teorías formales: Su caracterización, axiomas, reglas de inferencia, deducción, teoremas e interpretaciones. Profundización de conceptos de la lógica de predicados. El cálculo λ . Correctitud: construcción correcta a través del método de aserciones. Inducción como método de prueba de programas. Conceptos fundamentales de lógica difusa.

21. **SISTEMAS DISTRIBUIDOS Y PARALELISMO:** Sistemas Distribuidos: Metas, conceptos de hard, conceptos de software, usos en el diseño de sistemas distribuidos, comunicación: modelo cliente-servidor, invocaciones remotas de procedimientos, comunicación de grupo. Sincronización en sistemas distribuidos: Algoritmos, control de concurrencia, transacciones, manejo de transacciones. Administración de procesos y procesadores: threads, modelos de sistemas, asignación del procesador. Sistemas de archivos distribuidos. Un caso de estudio. Computaciones paralelas: Algunas técnicas básicas: computación por pasaje de mensajes. Conceptos de Balance de carga.

Laboratorios sugeridos: Desarrollo de aplicaciones cliente-servidor, uso de mecanismos de comunicación entre procesos. Implementación de threads.

22. **AUTOMATAS Y LENGUAJES:** Introducción a los lenguajes formales. Autómatas finitos y lenguajes regulares, un ejemplo de aplicación: analizador léxico gráfico, construcción automática de analizadores lexicográficos. Autómatas a pila y lenguajes libres de contexto, un ejemplo de aplicación: analizador sintáctico. Autómatas acotados linealmente y lenguajes sensibles al contexto. Computabilidad: Máquina de Turing y tesis de Turing-Church, lenguajes estructurados por frases. Redes de Petri: definición, marcado y transición. Redes de Petri como reconocedoras de lenguajes, secuencia de disparos, lenguaje de secuencia de disparos, etiquetado. Poder computacional de las Redes de Petri en relación con el de las máquinas de Turing.

Laboratorio sugerido: Construcción de un scanner y de un analizador sintáctico para un lenguaje simple.

23. **INTELIGENCIA ARTIFICIAL:** Historia, alcance y limitaciones. Representación del conocimiento, búsqueda en espacios de estado, búsquedas heurísticas, sistemas expertos, modelos cognitivos y de aprendizaje. Procesamiento no simbólico: introducción a las redes neuronales y computación evolutiva.

Laboratorio Sugerido: De ser posible interacción con un sistema experto, observando su comportamiento, capacidades y limitaciones o, alternativamente, construir un

CDE. ORDENANZA N° 011-98

MBV
NNH

pequeño sistema experto usando un lenguaje habitual en IA. Implementación de una red neuronal sencilla. Implementación de un algoritmo evolutivo sencillo.

24. DISEÑO Y CONSTRUCCION DE COMPILADORES: Conceptos básicos: traducción, compiladores, ensambladores e intérpretes, lenguaje fuente, lenguaje de implementación y lenguaje objeto. Fases de un compilador. Pasadas. Tabla de símbolos. Análisis sintáctico ascendente y descendente. Recuperación de Errores. Traducción dirigida por la sintaxis: definiciones y esquemas de traducción. Atributos. Definiciones y esquemas de traducción S-atribuidos y L-atribuidos. Evaluación de atributos. Analizador Semántico: sistemas de tipos, especificación para un controlador de tipos. Conversiones. Sobrecarga. Polimorfismo. Soportes de tiempo de ejecución. Generación de código intermedio. Algunas nociones básicas de optimización.

Laboratorio sugerido: Construcción de un compilador para un minilenguaje.

25. COMPUTABILIDAD Y COMPLEJIDAD: Computabilidad: Introducción a cardinales transfinitos. Formalización del concepto de problema. Lenguaje formal de programación S. Funciones computables. Funciones y predicados recursivos primitivos. Minimización acotada. Funciones parcialmente computables. Forma normal de Kleene. Programa Universal. Problema de Halting. Conjuntos recursivos enumerables. Máquinas de Turing. Algunos problemas no computables. Clasificación de problemas no computables usando oráculos. Complejidad: Modelos de computación. Computaciones acotadas en tiempo y espacio. Tiempo de ejecución y espacio de trabajo de las máquinas de Turing. Clases de complejidad. Relaciones entre clases de complejidad. Reducibilidad de problemas. Reducibilidad acotada en el tiempo. Reducibilidad acotada en el espacio. Complejidad de circuitos booleanos. Complejidad no uniforme. Complejidad uniforme. Modelo de computación probabilístico. Jerarquía de tiempo polinomial. Máquinas de Turing alternantes.

26. BASES DE DATOS II: Modelos algebraico, red, jerárquico y relacional. Estructuras de Datos. Resolución de consultas. Estudio de productos comerciales administradores de tales bases. Revisión de teoría de diseño de bases de datos relacionales. Lógica formal. Algebra Relacional. Cálculo relacional. Optimización de queries. Estudio de SQL. Estudio de productos comerciales administradores de bases de datos relacionales. Teoría de modelos finitos y teoría de bases de datos. Computabilidad de queries. Expresividad. Clases de queries. Diferentes formalismos para representación de clases de queries. Bases de datos distribuidas. Descomposición de queries localización de datos. Bases de datos de valores complejos. Algebra de valores complejos. Cálculo de valores complejos.

CDE. ORDENANZA N° 011-98

MBV
NNH

TRABAJO FINAL: El trabajo final consistirá de un trabajo donde el alumno integre los conocimientos adquiridos en la carrera tanto en el ciclo obligatorio como el opcional. El mismo consistirá en una propuesta que permita evaluar la aplicación del método científico y/o de trabajo profesional bajo la supervisión de un Director. Básicamente tendrá por objeto mostrar la capacitación del alumno para desempeñarse en su actividad teórica o aplicada, según la orientación por éste libremente elegida.

ANEXO IV CONTENIDOS MÍNIMOS DEL CICLO OPTATIVO

1. **APRENDIZAJE DE MÁQUINA:** Aspectos generales de aprendizaje de máquina. Aprendizaje a través árboles de decisiones. Aprendizaje a partir de ejemplos. Aprendizaje de Resolución de Problemas y soluciones de planeamiento. Aprendizaje a partir de la observación y descubrimiento. Aprendizaje por refuerzo.
2. **INTELIGENCIA ARTIFICIAL DISTRIBUIDA:** Definición. Taxonomía. Resolución distribuida de problemas. Sistemas Multi-agentes. Inteligencia Artificial Paralela. Formulación, descripción, descomposición y asignación de tareas. Interacción, lenguaje y comunicación. Coherencia y coordinación. Modelización de otros agentes y la actividad organizada. Disparidades inter-agentes. Aprendizaje y sistemas multi-agentes. Herramientas para Inteligencia Artificial Distribuida. Aplicaciones.
3. **COMPUTACION EVOLUTIVA I:** Introducción. Motivación de la evolución como modelo de simulación. Ventajas y desventajas de la Computación Evolutiva sobre otros enfoques. Fundamentos de los procesos evolutivos. Fundamentos de la Genética. Algoritmos Genéticos. Estrategias Evolutivas. Programación Evolutiva. Representación del espacio de soluciones. Evaluación del fitness. Mecanismos de Selección y operadores genéticos: alternativas. Convergencia de algoritmos evolutivos. Adaptabilidad y autoadaptabilidad. Otras metaheurísticas. Hibridización de metaheurísticas. Aplicaciones: Problemas clásicos de administración y optimización de recursos.
4. **COMPUTACION EVOLUTIVA II:** Selección: métodos alternativos. Representación: dependencia del problema. Multimodalidad. Decepción. Niching, sharing y crowding. Manejo de restricciones: penalidades, reparación, decodificadores. Aplicaciones. Pareto optimality. Algoritmos Genéticos multisexuados. TSP, otros crossovers. Múltiples. Coevolución. Paralelismo. Problemas de ordenamiento, secuenciamiento y otros schedulings de diferentes tipos. Estrategias Evolutivas aplicadas a distintos problemas.
5. **REDES NEURONALES:** Modelos estadísticos: La red de Hopfield, función de energía, la máquina de Boltzman y sus aplicaciones. Modelos Auto-organizados: Introducción y consideraciones básicas, concepto de mapas computacionales; modelo self-organizing featuring-mapping (SOFM), principio de funcionamiento y

CDE. ORDENANZA N° 011-98

MBV
NNH

algoritmo de aprendizaje, learning vector quantization, aplicaciones. Redes con función de base radial: teoría de regularización, aplicaciones. Redes modulares: modelo de mezcla gaussiana asociativo, estructura jerárquica de expertos adaptativos, aplicaciones. Procesamiento temporal: back propagation a través del tiempo. Redes neuro-difusas: fundamentos, modelos y aplicaciones.

6. **SISTEMAS EXPERTOS:** Estructura de los sistemas expertos. Tipos de conocimiento. Marcos. Reglas. Redes Semánticas. Tipos. Representación. Motor de inferencia. Sistemas de deducción hacia adelante y hacia atrás. Modelos de tratamiento de la incertidumbre. Modelos probabilísticos. Factores de certeza. Teoría de la evidencia.. Herramientas de construcción de sistemas expertos: generales y específicas.
7. **INGENIERIA DEL CONOCIMIENTO:** Introducción a la ingeniería del conocimiento. Naturaleza de la experiencia. Clasificación. Sistemas basados en conocimiento. La ingeniería del software y los sistemas basados en conocimiento. Arquitectura. Representación del conocimiento. Control del uso de conocimiento. Metodología de construcción de sistemas basados en conocimiento. Adquisición del conocimiento. Aprendizaje automático. Desarrollo del prototipo. Etapas de desarrollo. Conceptualización. Formalización. Verificación. Evaluación.
8. **SISTEMAS INTELIGENTES:** Técnicas de inteligencia artificial no simbólica aplicables a sistemas de computación y de producción. Sistemas de computación y producción inteligentes. Problemas de scheduling e hibridización de meta heurísticas aplicadas a problemas de optimización de recursos.
9. **SEMANTICA FORMAL DE LOS LENGUAJES DE PROGRAMACION:** Introducción a la semántica operacional: evaluación de expresiones, ejecución de comandos, pruebas simples. Introducción a la semántica denotacional: motivación, equivalencia de las semánticas, ordenes parciales completas y funciones continuas. El teorema de Knaster-Tarski. Semántica axiomática: lenguaje de aserciones, semántica de las aserciones, reglas de prueba para corrección parcial. Uso de las reglas de Hoare.
10. **TEORIA DE LA INFORMACION:** Entropía de esquemas finitos. Teoremas de Shanon. Codificación. Problemas. Códigos auto-correctores. Códigos lineales. Códigos cíclicos. Códigos recurrentes. Códigos de longitud variable.

11. **TEORIA DE GRAFOS Y SUS APLICACIONES:** Recapitulación de definiciones generales. Descendencia. Generalidades sobre redes infinitas. Operaciones sobre las redes. Los números fundamentales de la teoría de redes. Núcleo de una red. Juegos sobre una red. El problema del camino más corto. Sistema de Transporte. Matriz asociada a una red. Matrices de incidencia. El problema de Euler. Conectividad de una red.
12. **COMPUTACION PARALELA: FUNDAMENTOS:** Paralelismo y programación: Una perspectiva. Una notación de programación. Una lógica de programación: aserciones, un modelo de ejecución de programas; Conceptos fundamentales, teoremas sobre conceptos fundamentales. Propiedades adicionales. Arquitecturas y transformaciones: esquemas de programas, el problema de la atomicidad. Un caso de estudio. Estructuración de programas: Composición de programas por unión, superposición. Formato para el diseño de especificaciones.
13. **ESTUDIO COMPARATIVO DE METODOLOGÍAS:** Estudio de propuestas metodológicas y al análisis de sus fortalezas y debilidades al soportar desarrollos de sistemas. Aspectos conceptuales y habilidades adquiridas durante el desarrollo, tales como: utilización de por lo menos tres metodologías de desarrollo de sistemas (se pondrá especial énfasis en el paradigma de Orientación a Objetos), selección de metodologías que se ajusten al ámbito del problema estudiado, utilización del soporte CASE correspondiente a las metodologías estudiadas.
14. **ARQUITECTURAS DE SISTEMAS DE SOFTWARE:** Descripción de arquitecturas con notaciones adecuadas, modelos de arquitectura de uso exitoso de mayor difusión, selección de la arquitectura más adecuada para un determinado sistema, asociación de determinados tipos de arquitecturas con aplicaciones específicas, integración en una determinada arquitectura de las especificaciones funcionales y no funcionales de un sistema de software, concreción de emprendimientos en entornos distribuidos, sistemas concurrentes y sistemas reactivos.
15. **MÉTODOS FORMALES Y ANÁLISIS DE HERRAMIENTAS PARA LA PRODUCCIÓN DE SOFTWARE:** Sustentación de la actividad profesional en los métodos formales de la ingeniería de software, métodos de análisis de herramientas para la producción de software, evaluación del potencial de distintas herramientas para la producción de software, definición de la viabilidad de la incorporación de nuevas herramientas de producción de software, utilización de lógicas de descripción de estados, de grafos, redes de Petri y otros temas relacionados.

16. **DISEÑO DE SOFTWARE DE TIEMPO REAL:** Definición de especificaciones consistentes respecto de consideraciones de concurrencia y sincronización desde las más tempranas etapas del análisis. Detección de la necesidad de utilizar especificaciones de software de tiempo real, definición de arquitecturas de hardware y software de tiempo real, selección de herramientas de implementación adecuadas para software de tiempo real, utilización de técnicas de verificación de especificaciones de tiempo real, definición y verificación del rendimiento esperado en software de tiempo real.

ANEXO V

EQUIVALENCIAS Y ASIMILACIÓN CON PLAN ORD. 3/79-R y 3/98-CD

PLAN ANTERIOR	PLAN NUEVO
1. Cálculo I	Intr. a la Mat. y Análisis Matemático I
1. Álgebra y Geometría	Elementos de Álgebra
1. Probabilidad Elemental	Matemática Discreta
1. Diagramación y Programación	Intr. a la Computación y Programación I
1. Elementos de Probabilidad y Estadística	Elementos de Probabilidad y Estadística
1. Matemáticas Especiales	Análisis Matemático II
1. Arquitectura del Procesador	Arquitectura del Procesador I
1. Álgebra y Programación Lineal	Matemática Discreta
1. Programación Avanzada	Programación II
1. Estructuras de la Información	Estructuras de Datos y Algoritmos
1. Organización de Archivos	Org. de Archivos y Bases de Datos I
1. Análisis Comparativo de Lenguajes	Análisis Comparativo de Lenguajes
1. Sistemas Administrativos	Ingeniería de Software I
1. Optativa: Met. de Desarrollo de Software	Ingeniería de Software II
1. Optativa: Herramientas Inteligentes	Inteligencia Artificial
1. Optativa: Ingeniería del Conocimiento	Inteligencia Artificial y Planeamiento Informático
1. Autómatas y Lenguajes	Autómatas y Lenguajes
1. Simulación	Simulación y Lógica para Computación
1. Diseño y Construcción de Compiladores	Compiladores
1. Generación Automática de Sistemas	Inteligencia Artificial y Computabilidad y Complejidad
21. Teoría de la Información	
22. Sistemas operativos	Sistemas Operativos y Redes
23. Recuperación de la Información.	Bases de Datos II
24. Análisis Comparativo de Sistemas	Arquitectura del Procesador II y Sistemas Distribuidos y Paralelismo
Trabajo Final	Se admiten planes previamente aprobados, si los hubiera

CDE. ORDENANZA N° 011-98

MBV
NNH

ANEXO VI REGLAMENTACION DEL TRABAJO FINAL

I. DESCRIPCIÓN:

El Trabajo Final de la Licenciatura en Ciencias de la Computación consistirá de un trabajo donde el alumno integre los conocimientos adquiridos en la carrera tanto en el ciclo obligatorio como el opcional. El mismo consistirá en una propuesta que permita evaluar la aplicación del método científico y/o de trabajo profesional bajo la supervisión de un Director. Básicamente tendrá por objeto mostrar la capacitación del alumno para desempeñarse en su actividad teórica o aplicada, según la orientación por éste libremente elegida.

I. OBJETIVOS:

1. Integrar y afianzar conocimientos mostrando una correcta aplicación del método científico y/o una adecuada metodología para el desempeño profesional.
2. Introducir al futuro egresado en el campo de su posible orientación: académico, profesional o de la investigación.

III. CONDICIONES PARA LA DIRECCION:

1. Podrán ser Directores del Trabajo Final, Profesores y/o Investigadores, de la Universidad Nacional de San Luis y/o de otras Universidades Nacionales, que reúnan la condición de ser profesionales con probada idoneidad en el área de Ciencias de Computación.
2. En caso de ser necesario podrá contar a lo más con un Co-Director. Podrán cumplir las funciones de co-director: Profesores y/o Investigadores de otras disciplinas y/o Profesionales de Empresas Estatales o Privadas.

Tanto los Directores como los Codirectores podrán cumplir tal función con más de un alumno.

CDE. ORDENANZA N° 011-98

MBV
NNH

IV. CONDICIONES PARA LA REALIZACION DEL TRABAJO FINAL:

1. El Trabajo Final se considera individual, pero en los casos en que esté justificado por la amplitud temática, podrá hacerse a lo sumo con dos integrantes.
2. El alumno podrá iniciar su Trabajo Final al comenzar a cursar el quinto año de su carrera. Se considerará como fecha de iniciación del Trabajo Final, la aprobación por la Facultad, del Director, tema del Trabajo, y programa de actividades.
3. El Crédito horario de las materias optativas de quinto año es de un total de 400 horas más un máximo de 150 horas requeridas por el Trabajo Final, por lo cual si habiendo transcurrido un lapso de 18 meses el alumno no hubiese rendido el trabajo, deberá solicitar a la Facultad, con fundamento la ratificación o rectificación del tema. En este caso, el nuevo plazo para rendir será de 3 meses, vencido el cual el tema habrá perdido validez y el alumno deberá efectuar una nueva presentación.
4. Excepcionalmente el alumno que en cuarto año tenga definida la orientación de su Trabajo Final y haya aprobado las materias que a juicio del Director se requieran, podrá solicitar la correspondiente inscripción.
5. Para iniciar el Trabajo Final, el alumno deberá inscribirse, mediante la presentación de una nota al Departamento Alumnos solicitando la inscripción, se incluirá:
 - a) Tema y Plan de Trabajo Final.
 - b) Director y Codirector (en caso que fuere necesario).

V. DE LA APROBACION DEL PLAN DE TRABAJO FINAL:

El Plan de Trabajo Final será evaluado por una Comisión Asesora de Trabajo Final nombrada a tal efecto por el Departamento de Informática, la cual asesorará al Consejo de Departamento acerca de la aprobación o no del mismo, quien finalmente lo elevará al Sr. Decano de la Facultad para su protocolización definitiva.

El plan será evaluado considerando los siguientes aspectos:

1. Tema a desarrollar.
2. Cronograma de tareas.
3. Detalle de los elementos que se requerirán para su desarrollo (equipamiento, insumos, bibliografía, etc..)
4. Tiempo estimado de finalización (no más de 150 horas)

VI. EVALUACION DEL TRABAJO FINAL:

CDE. ORDENANZA N° 011-98

MBV
NNH

Una vez finalizado el Trabajo, el mismo será presentado por el alumno, en original y dos copias, ante un Tribunal Examinador designado por el Sr. Decano de la Facultad, a propuesta del Departamento de Informática. El Tribunal estará integrado por el Director como Presidente, dos vocales (Profesores del Departamento) y dos suplentes. El suplente nato del Director es el Codirector (en caso que correspondiere).

En un término no mayor de 30 días corridos el Tribunal Examinador deberá expedirse aprobando, rechazando total o parcialmente el Trabajo. En caso de rechazo total el tema perderá validez y el alumno podrá solicitar un nuevo tema; en caso de rechazo parcial el alumno deberá ajustar el trabajo a las observaciones del Tribunal, de producirse más de dos rechazos parciales el alumno podrá solicitar un nuevo tema.

La defensa del Trabajo Final será calificado por parte del Tribunal Examinador, con una nota de cuatro (4) a diez (10).

