

EXP-UNC 44032/2011

RESOLUCIÓN CD N° 136/2012

VISTO

La Resolución CONEAU N° 1106/2011 por la cual no se da lugar a la solicitud de acreditación de la Licenciatura en Ciencias de la Computación; y

CONSIDERANDO

Que a través de la Sra. Rectora de esta Universidad, Dra. Carolina Scotto, se ha presentado oportunamente un recurso de reconsideración de acuerdo a lo establecido por la Ordenanza CONEAU N° 52/2008;

Que la CONEAU ha decidido dar lugar a dicha solicitud, otorgando un plazo de seis meses para superar las deficiencias señaladas por la Resolución CONEAU N° 1106/2011;

Que la CONEAU ha observado que no se asegura el dictado de algunos de los contenidos básicos curriculares enunciados en el Anexo I-1 de la Resolución Ministerial N° 786/2009, correspondiente a los estándares de acreditación de la Licenciatura en Ciencias de la Computación;

Que en consecuencia, la Secretaria Académica asesorada por docentes de la carrera, ha propuesto modificaciones en los programas de las asignaturas de manera de cubrir adecuadamente dichos contenidos;

Que es conveniente elevar a CONEAU los programas actualizados de todas las asignaturas de la carrera.

**EL CONSEJO DIRECTIVO
DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA**


RESUELVE:

ARTÍCULO 1°: Aprobar los programas de cada una de las asignaturas obligatorias de la carrera, que figuran en el Anexo que forma parte de la presente resolución.

ARTÍCULO 2°: Elévese a la Comisión Nacional de Evaluación y Acreditación Universitaria, comuníquese y archívese.

DADA EN LA SALA DE SESIONES DEL CONSEJO DIRECTIVO DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA, A VEINTICINCO DÍAS DEL MES DE JUNIO DE DOS MIL DOCE.

pk


Dra. NOEMI PATRICIA KISBY
SECRETARIA ACADEMICA
Fa.M.A.F. UNC


Dra. ESTHER GALINA
VICE DECANA
Fa.M.A.F.

ANEXO**PROGRAMA DE ASIGNATURA**

ASIGNATURA: Álgebra	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Primer año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Compenetrar al alumno con el lenguaje y teoremas básicos del álgebra lineal, necesarios para la redacción matemática de problemas de otras ciencias o de la matemática.

CONTENIDO

Unidad I: Cuerpos. Definición y Ejemplos. El cuerpo de los números complejos. Descomposición polar, Teorema de Moivre, raíces n -ésimas, raíces de la unidad.

Unidad II: Sistemas de ecuaciones lineales, sistemas de ecuaciones equivalentes, matriz asociada a un sistema de ecuaciones, operaciones elementales por filas, matrices reducidas por filas en escalera, matrices equivalentes por filas. Matrices, operaciones con matrices, propiedades de las operaciones con matrices, matrices invertibles.

Unidad III: Espacios vectoriales, subespacios, combinación lineal de vectores, conjuntos linealmente independientes y linealmente dependientes, bases y dimensión, Teorema de la dimensión de la suma de subespacios. Bases ordenadas, coordenadas lineales, matriz de cambio de base, aplicación de las operaciones por filas al cálculo de subespacio generado por un conjunto finito de vectores.

Unidad IV: Transformaciones lineales, imagen y núcleo, teorema de la dimensión, el

Anexo Res. CD N° 136/12



álgebra de las operadores lineales, matriz de una transformación lineal, rango fila igual a rango columna de una matriz, dimensión del espacio de las transformaciones lineales, cambio de bases, caracterización de las transformaciones lineales biyectivas, isomorfismos, matrices semejantes, funcionales lineales, el espacio dual, la transpuesta de una transformación lineal.

Unidad V: Definición y cálculo de determinantes, alternancia, desarrollo por una fila o columna, determinante de un producto. Matrices invertibles y determinantes.

Unidad VI: Autovalores y autovectores de un operador lineal, polinomio característico, Teorema de Cayley Hamilton, operadores diagonalizables, operadores autoadjuntos en \mathbb{R}^n .

Unidad VII: Productos Internos en espacio euclideo \mathbb{R}^n , propiedades elementales.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- HOFFMAN, K. y KUNZE, R. *Álgebra Lineal*. México: Prentice-Hall, 1973. Cap. 1,2,3,4,5.
- GENTILE, E. *Espacios Vectoriales*. Buenos Aires, 1968.
- MEYER, C. *Matrix analysis and applied linear algebra*. Philadelphia : Society for Industrial and Applied Mathematics. SIAM, c2000.
- LANG, S. *Álgebra Lineal*. Bogotá : Fondo educativo interamericano, 1976.

METODOLOGÍA DE TRABAJO

La asignatura se organiza en clases teóricas y prácticas, de cuatro horas reloj cada una.

Las clases teóricas son expositivas, y las clases prácticas se organizan en comisiones donde los alumnos resuelven de manera independiente o grupal ejercicios prácticos, bajo la supervisión y acompañamiento de los docentes.





UNC

Universidad
Nacional
de Córdoba



1613 - 2013
400
AÑOS



FAMAF

Facultad de Matemática,
Astronomía y Física

EVALUACIÓN

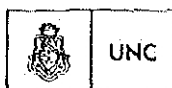
FORMAS DE EVALUACIÓN

Examen final teórico-practico escrito, para los alumnos de la Licenciatura en Matemática se tomara teórico oral.

CONDICIONES PARA OBTENER LA REGULARIDAD Y PROMOCIÓN

Aprobar con cuatro o más, cada parcial o sus recuperatorios. Se requiere setenta por ciento de asistencia a las clases prácticas.

No hay promoción.



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Algoritmos y estructuras de datos I	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Primer año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Se pretende que el alumno adquiera:

- capacidad para analizar algoritmos
- capacidad y hábito de identificar abstracciones al abordar un problema
- familiaridad con técnicas frecuentes de diseños de algoritmos

CONTENIDO

Unidad I: Expresiones cuantificadas: repaso de especificaciones con cuantificadores lógicos, revisión de la sustitución y la regla de Leibniz, reglas generales para las expresiones cuantificadas, cuantificadores aritméticos y lógicos.

Unidad II: Construcción y corrección de programas: especificaciones formales, ejemplos y resolución de problemas.

Unidad III: Técnicas elementales para la construcción de programas: definiciones recursivas, reemplazo de constantes por variables, modularización, uso de tuplas, generalización por abstracción.

Unidad IV: Tipos abstractos de datos: aritmética de precisión arbitraria, especificaciones implícitas, especificaciones implícitas con invariante de representación.

Unidad V: Recursión final: funciones recursivas finales, recursión lineal y final, recursión final para listas.

Unidad VI: Programación imperativa: estados y predicados, transformador de

Anexo Res. CD N° 136/12



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

predicados wp, definición de un lenguaje de programación imperativo (Skip, Abort, Asignación, Composición, Alternativa, Repetición).

Unidad VII: Cálculo de programas imperativos: derivación de ciclos. Técnicas para determinar invariantes (tomar términos de una conjunción, reemplazo de constantes por variables, fortalecimiento de invariantes).

Unidad VIII: Recursión final y programación imperativa; programas imperativos sobre listas.

Trabajos prácticos especiales

Proyecto 1: Tipos de datos en Haskell.

Proyecto 2: Recursión, alto orden, expresiones lambda y otros elementos de Haskell.

Proyecto 3: Lenguaje C primera parte.

Proyecto 4: Lenguaje C segunda parte.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Cálculo de programas. Javier Blanco, Silvina Smith, Damián Barsotti, Córdoba: Universidad Nacional de Córdoba, c2008.

BIBLIOGRAFÍA COMPLEMENTARIA

- Programming: the derivation of algorithms, Anne Kaldewaij, Prentice-Hall, 1990.

METODOLOGÍA DE TRABAJO

La asignatura se organiza en clases teóricas y prácticas, de cuatro horas reloj cada una.

Las clases teóricas son de carácter expositivo, con participación de los estudiantes en la resolución de problemas planteados.

Se asignan además trabajos prácticos especiales que son desarrollados por los alumnos de manera grupal.

En las clases prácticas los alumnos trabajan de manera individual o grupal, bajo la supervisión y acompañamiento de los docentes.



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

En las clases prácticas de taller los alumnos acceden al laboratorio de informática, desarrollando las prácticas en computadora con software libre (Haskell, GCC).

EVALUACIÓN

Se toman dos parciales, evaluándose respectivamente conceptos de programación funcional y programación imperativa.

Requisitos para obtener la regularidad

- Aprobación de los dos parciales o un parcial y un recuperatorio.

Requisitos para la aprobación:

- Evaluación integral de la materia, o promoción directa.

Promoción:

- Aprobación de ambos parciales con nota no inferior a 7 (siete).



UNC
Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Algoritmos y Estructuras de Datos II	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Se pretende que el alumno adquiera:

- capacidad para analizar algoritmos
- capacidad y hábito de identificar abstracciones al abordar un problema
- familiaridad con técnicas frecuentes de diseños de algoritmos

CONTENIDO

Unidad I: Análisis de algoritmos

Motivación

Problema de Ordenación. Ordenación por selección. El ciclo for. Conteo de operaciones de un programa. Definición en símbolos (ops). Conteo de comparaciones de la ordenación por selección. Incidencia del crecimiento del tamaño de los datos en la performance del algoritmo. Introducción del término “del orden de”. Ordenación por inserción. Conteo. Peor caso, mejor caso y caso medio.

La notación O

Significado de peor caso y caso medio. Operaciones elementales. Análisis aproximado. La notación O. Definición formal. Ejemplos. Insignificancia de las constantes aditivas y multiplicativas. Reflexividad y transitividad. Igualdad entre los O's de funciones. Equivalencia entre logaritmos de diferente base. Regla del límite. Jerarquía: logaritmos, polinomios, exponenciales, factoriales. El O de la suma y el producto. El O de un polinomio. Terminología: funciones y algoritmos logarítmicos, cuadráticos, cúbicos, polinomiales, exponenciales. Balance entre tiempo y espacio de los algoritmos.

Notación complementaria

Las notaciones Ω y Θ . Definición formal. Equivalencia entre expresiones con Ω , Θ y O. Distinción entre Ω , Θ y O por un lado y el análisis del mejor caso, caso medio y

peor caso por el otro. Notación $f(n) + O(g(n))$, $f(n) * O(g(n))$, $f(n)^{O(g(n))}$, etc.

Ejemplos

Búsqueda lineal. Análisis de mejor caso, peor caso y caso medio. Búsqueda lineal en un arreglo ordenado. Análisis de mejor caso, peor caso y caso medio. Búsqueda binaria. Análisis de mejor caso, peor caso y caso medio. Contraste entre el algoritmo lineal y el logarítmico cuando el tamaño de la entrada crece.

Motivación de la recurrencias

Transformación gradual de la ordenación por selección en la ordenación por intercalación. Versión funcional de la ordenación por intercalación. Versión imperativa. Análisis de la ordenación por intercalación. Resolución de la recurrencia.

Recurrencias

Recurrencias homogéneas. Formulación, resolución y demostración. Ejemplos. Recurrencias no homogéneas. Formulación y resolución. Ejemplos. Recurrencias divide y vencerás. Formulación y resolución. Ejemplos. Funciones eventualmente no decrecientes. Funciones λ -uniformes y uniformes. Regla de la uniformidad. Ejemplos. Demostración de la resolución de recurrencias divide y vencerás.

Unidad II: Estructura de datos

Introducción

Importancia de la elección de estructuras de datos adecuadas. Los tipos concretos como concepto relativo a un lenguaje de programación. Los tipos abstractos como concepto asociado a un problema que se quiere resolver. Tipos abstractos y sus diferentes representaciones.

Estructuras concretas

Estructuras concretas más comunes en los lenguajes de programación. Arreglos. Operaciones para manipularlos. Almacenamiento en memoria. Representación gráfica. Eficiencia de las operaciones. Diferentes tipos de índices. Tipos enumerados. Ciclo for generalizado. Listas como tipos concretos. Operaciones para manipularlos. Almacenamiento en memoria. Representación gráfica. Eficiencia de las operaciones. Registros. Operaciones para manipularlos. Almacenamiento en memoria. Representación gráfica. Problema de aliasing.

Tipos abstractos de datos (TAD's)

Tipos abstractos más usuales. TAD Lista. Operaciones. Ecuaciones. Representaciones usando arreglos. TAD Pila. Operaciones. Ecuaciones. Ejemplo: versión iterativa de la ordenación por intercalación usando una pila. Ejemplo: evaluación de expresiones en notación polaca inversa usando una pila. Representaciones usando arreglos y listas. TAD cola. Operaciones. Ecuaciones. Representaciones usando arreglos y listas. Listas enlazadas. Representación gráfica. Representaciones de listas, pilas y colas usando listas enlazadas, listas enlazadas con puntero al último y listas circulares. Aliasing y errores usuales al programar con punteros. Manejo de memoria en ejecución. TAD árbol binario. Representación gráfica. Operaciones. Ecuaciones. Terminología botánica y genealógica. Posiciones. Subárbol correspondiente a una posición. Posiciones de un



árbol. Elemento alojado en una posición de un árbol. Representación usando punteros. Árboles binario de búsqueda (ABB). Operaciones: versiones recursiva e iterativa. Eficiencia. TAD cola de prioridades. Operaciones. Ecuaciones. Heap. Implementación de cola de prioridades usando un heap. Eficiencia de las operaciones. Heap usando arreglos. Eficiencia. Ordenación con heap. Eficiencia. Ordenación con heap sin arreglo auxiliar.

Otras estructuras

Problema unión-find. Inicialización virtual.

Unidad III: Estrategias conocidas de resolución de problemas

Uso de heurísticas en algoritmos. Estrategias de diseño de algoritmos.

Algoritmos voraces

Propiedades generales de los algoritmos voraces (o greedy o glotones o golosos). Esquema general. Problema de la moneda simplificado. Problema de la mochila simplificado. Problema del camino de costo mínimo. Algoritmo de Dijkstra. Problema del árbol generador de costo mínimo. Algoritmos de Prim y de Kruskal.

Divide y vencerás

Propiedades generales de la técnica divide y vencerás. Esquema general. Búsqueda binaria. Ordenación por intercalación. Ordenación rápida (quicksort). Cálculo eficiente de la potencia n -ésima de un número. Multiplicación de grandes números.

Programación dinámica

Funciones recursivas potencialmente exponenciales. Confección de tablas. Fibonacci. Problema de la moneda. Problema de la mochila. Funciones con memoria. Revisión de los problemas de la moneda y de la mochila. Problema de los caminos de costo mínimo. Algoritmo de Floyd. Cómputo de números combinatorios. Reducción del espacio necesario para las tablas.

Backtracking

Motivación: algoritmo para salir de un laberinto. Recorrida de árboles binarios. Pre-orden, in-orden y pos-orden de izquierda a derecha y de derecha a izquierda. In-orden para listar ordenadamente un ABB. Recorrida de árboles finitarios. Precondicionamiento. Pre-orden y pos-orden para resolver el problema del ancestro. Recorrida de árboles dirigidos o no. DFS recursivo e iterativo con pila. BFS con cola. Grafos implícitos. Problema de la mochila con pesos reales. Problema de las ocho reinas. Podas graduales al grafo de búsqueda.

Trabajos prácticos especiales

Desarrollo de cuatro trabajos de laboratorio

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Brassard & Bratley. Fundamentos de Algoritmia.



- Brassard & Bratley. Fundamentals of Algorithmics.
- Cormen, Leiserson, Rivest & Stein. Introduction to Algorithms.
- Manber. Introduction to Algorithms: A Creative Approach.
- Fridlender. Notas de Algoritmos y Estructuras de Datos II.
- Gramaglia. Algoritmos y Estructuras de Datos II: Guía de Ejercicios Resueltos.
- Tiraboschi. Algoritmos y Estructuras de Datos II: Guía de Ejercicios.

BIBLIOGRAFÍA COMPLEMENTARIA

- Balcázar. Programación Metódica.
- Biggs. Matemática Discreta.
- Blanco & Smith. Cálculo de Programas
- Kaldewajj. Programming: the Derivation of Algorithms.

METODOLOGÍA DE TRABAJO

El desarrollo de los contenidos se basa fuertemente en la modalidad de clase magistral, empleando en este tipo de actividad un 33% del tiempo de aula. En las mismas se estimula la participación de los alumnos a través del análisis de casos de estudio y preguntas, vinculando la actividad con la práctica.

Para el desarrollo de actividades prácticas (33%) se trabaja con una guía de ejercicios adoptando una modalidad de resolución grupal o individual según la preferencia del alumno, bajo la supervisión y orientación de los docentes y atendiendo a los tiempos de cada alumno o grupo de alumnos.

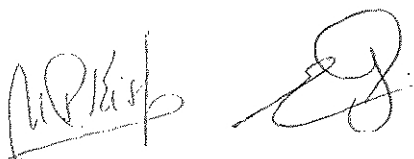
Las actividades de implementación en máquina se organizan con la modalidad de clase magistral seguida de trabajo grupal supervisado con entregas pautadas.

EVALUACIÓN

FORMAS DE EVALUACIÓN

Se toman tres evaluaciones parciales, uno para cada módulo principal de la materia. Los exámenes evalúan contenidos teóricos y prácticos.

Requisitos para obtener la regularidad





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

-
- Aprobación de dos de los tres exámenes parciales.

Requisitos para la aprobación:

- Evaluación escrita integral de la materia, de contenidos teóricos y prácticos. En caso de ser necesario, se toma un examen oral.

Promoción

- Aprobación de todos los parciales con nota mínima 6(seis) y promedio no inferior a 7(siete).



UNC
Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Análisis Matemático I	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Primer año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Este curso tiene como objetivo general introducir al estudiante en los conceptos básicos del cálculo matemático en una variable, abordando contenidos fundamentales de funciones, límite, continuidad, derivada e integrales.

Se pretende que a lo largo del curso el estudiante logre

1. apropiarse de conceptos básicos del cálculo diferencial en una variable,
2. desarrollar habilidades para interpretar y representar funciones, algebraica y gráficamente,
3. habilidad para aplicar procedimientos y reglas de cálculo para funciones en una variable.

CONTENIDO

Unidad I: Números y Funciones.

Números enteros, racionales y reales. Desigualdades. Valor absoluto. Funciones. Definición. Ejemplos. Gráficas de funciones. Funciones inyectivas, suryectivas y biyectivas. Rectas, parábolas, circunferencia, elipse. Funciones trigonométricas. Funciones exponenciales y logarítmicas. Propiedades y ejemplos.

Unidad II: Límite y continuidad.

Definición intuitiva de límite. Ejemplos. Límites laterales. Relación entre la existencia de límites laterales y la de límite. Límites infinitos. Límite cuando la variable tiende a infinito. Límites infinitos cuando la variable tiende a infinito. Límites notables. Definición de continuidad en un punto. Continuidad por derecha y por izquierda. Definición de continuidad en un intervalo. Propiedades. Teorema de Weierstrass. Teorema de los Valores Intermedios.

Unidad III: Derivada.

Definición de función derivable en un punto. Ejemplos. Reglas de derivación. Propiedades. Regla de la cadena. Derivadas de orden superior. Derivada de funciones trigonométricas. Derivada de funciones exponenciales. Derivada de la función inversa. Derivada de funciones trigonométricas inversas. Algo sobre el número e . Derivada de funciones logarítmicas.

Unidad IV: Valores máximos y mínimos. Gráficas.

Definición de punto de máximo (mínimo) y de valor máximo (mínimo) locales y absolutos. Ejemplos. Teorema de Fermat. Máximos y mínimos en intervalos cerrados. El Teorema de Rolle y el Teorema del valor medio. Teorema del valor medio de Cauchy. La regla de L'Hopital. Funciones crecientes y decrecientes. Propiedades. Concavidad y puntos de inflexión. Prueba de concavidad. Prueba de la segunda derivada. Gráficas.

Unidad V: Integrales.

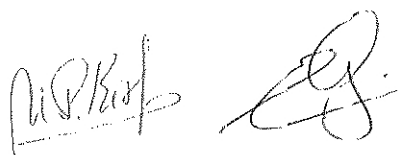
La integral indefinida de una función continua. Área. Suma de Riemann. Teorema fundamental del cálculo. Propiedades básicas de la integral indefinida. Técnicas de integración: Método de sustitución, integración por partes. Aplicaciones al cálculo de áreas y volúmenes.

BIBLIOGRAFÍA**BIBLIOGRAFÍA BÁSICA**

- Marta Urciuolo, Patricia Kisbye, *Notas de Análisis Matemático I* (en edición)
- Louis Leithold, *El cálculo*
- Serge Lang, *Cálculo*
- Carina Boyallian, Elida Ferreyra, Marta Urciuolo, Cynthia Will, *Un Segundo curso de cálculo.*

BIBLIOGRAFÍA COMPLEMENTARIA

Spivak, Michael David, *Cálculus: cálculo infinitesimal*. Barcelona: Reverté, 1980.



METODOLOGÍA DE TRABAJO

El curso estará organizado en tres comisiones de aproximadamente 50 alumnos cada una. Cada comisión tendrá un docente responsable, docentes de práctico y ayudantes alumnos.

Las clases se organizarán en modalidad teórico-práctica, en las cuales el docente a cargo de la comisión dictará las clases promoviendo la interacción docente-alumno. Estas clases tendrán intercaladas jornadas de clases prácticas, para que los alumnos puedan completar las guías de práctico.

Los docentes de práctico atenderán consultas de práctico al finalizar cada clase teórico práctica, como también las jornadas de clases prácticas.

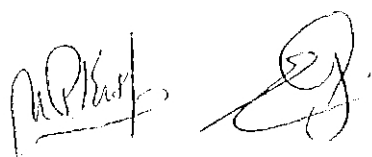
EVALUACIÓN

FORMAS DE EVALUACIÓN

- Se tomarán dos parciales, con posibilidad de recuperar uno de ellos.
- El examen final contará de una evaluación escrita sobre contenidos teórico-prácticos.

CONDICIONES PARA OBTENER LA REGULARIDAD

- Para la regularidad se deberá aprobar ambos parciales, o uno de ellos y el recuperatorio del otro.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Análisis Matemático II	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Primer año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Que el estudiante:

1. maneje con solvencia las nociones básicas del cálculo diferencial e integral en una variable.
2. adquiera nociones básicas del cálculo diferencial e integral en varias variables.

CONTENIDO

Unidad I: Integrales

Primera parte:

Repaso de los siguientes temas: Primitiva. Tabla de primitivas. Sumas superiores e inferiores de Riemann de funciones continuas en un intervalo cerrado. Definición de función integrable y de su integral definida. Linealidad de la integral. Teorema fundamental del cálculo y regla de Barrow. Métodos de integración: integración por sustitución y partes. Cálculo de área entre curvas.

Segunda parte:

Integración de funciones racionales por el método de fracciones simples. Integración de funciones racionales de funciones trigonométricas. Integrales impropias. Criterios de comparación para la convergencia de integrales impropias. Aplicaciones.

Unidad II: Sucesiones y Series Numéricas

Definición de sucesión. Límite de una sucesión. Límite de la suma, producto y cociente de sucesiones. Teorema del sándwich. Teoremas que relacionan límite de

funciones de variable real con límite de sucesiones. Sucesiones acotadas superior e inferiormente. Sucesiones crecientes y decrecientes. Convergencia de sucesiones monótonas y acotadas. Subsucesiones. Subsucesiones de una sucesión convergente.

Series numéricas. Definición de convergencia. Las series geométrica y armónica. Propiedades: suma y producto por una constante de series convergentes. Criterio de la divergencia. Criterios de convergencia: de comparación, de la integral, de comparación en el límite, para series alternantes. Definición de convergencia absoluta y su relación con la convergencia. Criterio del cociente y de la raíz.

Unidad III: Series de potencias. Series y polinomios de Taylor

Series de potencias. Radio e intervalo de convergencia. Criterio del cociente para series de potencias. Derivada y primitivas de una serie de potencias. Serie de Taylor de una función. Polinomio de Taylor. Fórmula de Lagrange del resto de Taylor. Aproximación de funciones por polinomios de Taylor.

Unidad IV: Espacio euclídeo de dimensión n . Curvas en R^n

El espacio euclídeo de dimensión n . Producto escalar en R^n . Propiedades. La desigualdad de Cauchy Schwartz. La norma euclídea. Propiedades. Desigualdad Triangular.

Geometría analítica: definición de una recta en R^n , ecuaciones vectorial, paramétrica y standart. Definición de Plano en R^3 , ecuaciones vectorial, normal y cartesiana. Concepto de planos paralelos y ángulo entre planos.

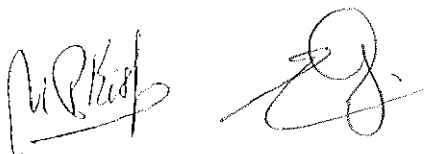
Funciones vectoriales y curvas en el espacio: límite, continuidad y derivada. Ecuación de la recta tangente a una curva en un punto.

Unidad V: Funciones de varias variables

Funciones de varias variables a valores reales. Definición de límite y continuidad. Propiedades. Derivadas parciales de una función. Plano tangente a una superficie gráfico de una función de dos variables, en un punto. Diferenciabilidad de una función de dos variables. Continuidad de una función diferenciable. Continuidad de las derivadas parciales implica diferenciabilidad. Regla de la cadena. Gradiente de una función a valores reales. Derivadas direccionales. El gradiente como dirección de máximo crecimiento. Derivadas parciales de ordenes superiores. Recta tangente a curvas de nivel en el plano. Plano tangente a superficies de nivel en el espacio. Igualdad de las derivadas cruzadas. La matriz Hessiana. Criterios para determinar máximos, mínimos (locales) y puntos de sillan, de funciones de dos variables a valores reales.

Unidad VI: Integrales múltiples

Integrales dobles en rectángulos. Integrales iteradas. Integrales dobles en regiones





UNC

Universidad
Nacional
de CórdobaFAMAF
Facultad de Matemática,
Astronomía y Física

generales. Cambio de variables. Coordenadas polares. Integrales triples en rectángulos. Cálculo de integrales múltiples por integración iterada.

BIBLIOGRAFÍA

- James Steward, Calculus.
- James Steward, Cálculo multivariable.
- Michael Spivak, Cálculo infinitesimal.
- C. Boyallian, E. Ferreyra, M. Urciuolo, C. Will; Un segundo curso de cálculo. Trabajos de matemática Serie C- FaMAF

METODOLOGÍA DE TRABAJO

La asignatura se dicta en dos turnos, uno por la mañana y otro en horario vespertino. Cada turno se organiza en dos módulos por semana de cuatro horas reloj cada uno. Cada módulo consta de una clase teórica de dos horas y una clase práctica de dos horas.

Las clases teóricas son expositivas, con resolución de ejemplos y dando cabida a preguntas de los estudiantes.

Las clases prácticas se organizan en comisiones donde los alumnos resuelven de manera independiente o grupal ejercicios prácticos, bajo la supervisión y acompañamiento de los docentes. También los docentes interactúan con los alumnos mediante exposiciones para la resolución de algunos problemas.

EVALUACIÓN

FORMAS DE EVALUACIÓN

- Tres evaluaciones parciales durante el cursado de la materia.
- Un examen final.

CONDICIONES PARA OBTENER LA REGULARIDAD

- Aprobación de dos de los tres parciales.
- Asistencia al menos al 70% de las clases prácticas.



UNC
Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Análisis Numérico	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año– Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

- Lograr que el estudiante adquiera las herramientas básicas para resolver problemas de matemática aplicada mediante el uso de técnicas computacionales para diseñar algoritmos numéricos.

CONTENIDO

- Análisis de errores
Error absoluto y relativo. Redondeo y truncamiento. Propagación de errores. Sistemas de punto flotante. Errores de representación. Estrategias para minimizar los errores.
- Solución de ecuaciones no lineales
Métodos de Bisección, Newton, Secante, métodos de punto fijo.
- Interpolación numérica
Interpolación polinomial. Formas de Lagrange y de Newton. Splines.
- Aproximación de funciones
Teoría de cuadrados mínimos.
- Integración numérica
Reglas simples y compuestas: rectángulo, punto medio, trapecio, Simpson. Reglas Gaussianas.
- Resolución de sistemas de ecuaciones lineales
Eliminación gaussiana y factorización LU. Métodos iterativos: Jacobi y Gauss-Seidel.
- Introducción a la Programación Lineal
Convexidad y desigualdades lineales. Programación lineal. Método Simplex.

BIBLIOGRAFÍA

- Análisis Numérico: las matemáticas del cálculo científico. D. Kincaid, W. Cheney. Addison – Wesley, 1994.
- Análisis Numérico. R. Burden, J. Faires, 7ma. Edición, 2001.
- Numerical Analysis: an introduction. L. Eldén, L. Wittmeyer-Koch. Academic Press, 1990.

METODOLOGÍA DE TRABAJO

Las clases teóricas son de carácter expositivo.

Las clases prácticas se organizan en comisiones donde los alumnos resuelven de manera independiente o grupal los ejercicios prácticos, bajo la supervisión y acompañamiento de los docentes.

La asignatura se organiza en clases teóricas y prácticas, de cuatro horas reloj cada una, con talleres de laboratorio de informática incluidos.

Las clases teóricas son expositivas, y las clases prácticas se organizan en comisiones donde los alumnos resuelven de manera independiente o grupal ejercicios prácticos, bajo la supervisión y acompañamiento de los docentes.

En las clases de laboratorio se resuelven en computadora una lista de problemas seleccionados.

EVALUACIÓN

Requisitos para obtener la regularidad

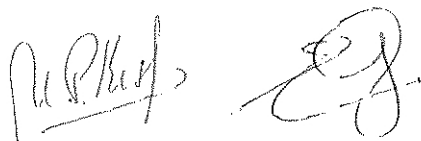
- Aprobación de los dos parciales o un parcial y su recuperatorio.
- 70% de asistencia en las clases de laboratorio.

Requisitos para la aprobación:

- Examen escrito y prueba de laboratorio.

Promoción:

- No se prevé régimen de promoción.



PROGRAMA DE ASIGNATURA

ASIGNATURA: Arquitectura de Computadoras		AÑO: 2012
CARÁCTER: Obligatoria		
CARRERA: Licenciatura en Ciencias de la Computación		
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.	
UBICACIÓN en la CARRERA: Tercer año - Segundo cuatrimestre		

FUNDAMENTACIÓN Y OBJETIVOS

Que el alumno sea capaz de interpretar el funcionamiento de los bloques "internos" asociados a Arquitectura de Computadoras No Convencionales (No "Von Neumann", Procesadores de Alta Prestación y Computadoras Reconfigurables).

CONTENIDO**Unidad 1: Computación SISD ("Single Instruction, Single Data")**

- 1.1.-Arquitecturas tipo SISD, subtipo RISC (Reduced Instruction Set Computer).
- 1.2.-Arquitecturas tipo SISD, subtipo CISC (Complex Instruction Set Computer).
- 1.3.-Ejemplos de Arquitecturas SISD tipo No-Von Neuman.
- 1.4.- Concepto de Segmentación Encausada, (Pipe-Line).
- 1.5.- Ejemplos de Arquitecturas SISD, RISC.
- 1.6.- Ejemplos de Arquitecturas SISD, CISC.
- 1.7.- Arquitectura y Set de Instrucciones de Procesadores con Pipe-line.
- 1.8.- Ejemplos de Procesadores con Pipe-Line.
- 1.9.- Organización, Jerarquía y Administración de Memorias en Sistemas con Pipe-Line.
- 1.10.- Arquitectura y Set de Instrucciones de Procesadores Vectoriales.
- 1.11.- Características de los lenguajes para procesamiento Vectorial.
- 1.12.- Características de los Compiladores para Procesadores Vectoriales.
- 1.13.- Ejemplos de Procesadores con Segmentación Encausada y Vectoriales.
- 1.14.- Caso de estudio práctico para integración de conceptos.

-Unidad 2: Computación SIMD ("Single Instruction, Multiple Data")

- 2.1.-Procesadores Matriciales o SIMD. Concepto.
- 2.2.-Arquitecturas de los Procesadores SIMD.
- 2.3.-Redes de Interconexión.
- 2.4.- Algoritmos para procesadores SIMD. Ejemplos.
- 2.5.- Procesadores SIMD Asociativos.
- 2.6.- Memorias Asociativas.
- 2.7.- Algoritmos para procesadores SIMD, Asociativos.
- 2.8.- GPGPU (General Purpose Graphic Processor Unit) y Computación heterogénea: Origen y Conceptos.
- 2.9.- Implementación: OpenCL (Open Computer Language)
- 2.10.- Arquitectura y Modelo de Plataforma.
- 2.11.- Modelo de Ejecución y de Memoria.
- 2.12.- Lenguaje e Interfaces.
- 2.13.- Operaciones con Matrices en OpenCL
- 2.14.- Caso de estudio Práctico para integración de conceptos.

-Unidad 3: Computación MIMD (Multiple Instruction, Multiple Data) y de Flujo de Datos

- 3.1.-Computadores MIMD ligeramente acoplados.
- 3.2.-Computadores MIMD estrechamente acoplados.
- 3.3.-Distintos tipos de Buses y Redes de Interconexión.
- 3.4.-Estructuración y Organización de la Memoria.
- 3.5.-La Problemática de los Sistemas Multicaché.
- 3.6.-Características de los S.O. para Sistemas MIMD.
- 3.7. Algoritmos en Procesadores SIMD.
- 3.8. Ejemplos de sistemas MIMD.
- 3.9. Nociones de computadores de Flujo de Datos.
- 3.10. Arquitecturas de computadoras de Flujo de Datos.
- 3.11.- Casos de estudio de ejemplo.

-Unidad 4: Nociones de Computación Reconfigurable (C. R.) y de Alta Performance (HPC)

- 4.1.-Conceptos generales, historia y estado del arte de la C. R.
- 4.2.-El uso de HDL en computación reconfigurable.
- 4.3.-Nociones de Codiseño Hardware-Software y su aplicación en C.R.
- 4.4.- Conceptos generales de HPC.
- 4.5.- Historia y estado del arte HPC.
- 4.6.- Nociones de HPC y distribuida. Nociones básicas de Clusters.
- 4.7.- Nociones básicas de Arquitecturas Grid.



BIBLIOGRAFÍA

- Arquitectura de Computadoras y Procesamiento Paralelo, Kai Hwang y Faye A. Briggs. Mc Graw-Hill (1988).
- David A. Patterson and John L Hennessy: "Computer Organization and Design – The Hardware/Software Interface". Fourth Edition. Elsevier – Morgan Kaufmann - 2009
- John L Hennessy and David A. Patterson: "Computer Architectura – A quantitative Approach". Fourth Edition. Elsevier – Morgan Kaufmann. (2007)
- Volnei. Pedroni. Circuit Design Using VHDL. MIT Press, Cambridge, Massachusetts, 2004.
- Douglas Perry. VHDL: Programming by Example. Mc. Graw Hill, NY, 2002.
- Enoch Hwang. Microprocessor Design: Principles and Practices with VHDL. Brooks/Cole. 2004.

METODOLOGÍA DE TRABAJO

CLASES TEÓRICAS: Cuatro horas semanales.

CLASES PRÁCTICAS: Cuatro horas semanales. Corresponden a prácticas de laboratorio en donde el alumno resolverá problemas utilizando el centro de cómputo.

EVALUACIÓN

FORMAS DE EVALUACIÓN


-Examen final teórico/práctico escrito. Para aprobar deberán obtener como mínimo un 40% los regulares y un 60% los libres.

CONDICIONES PARA OBTENER LA REGULARIDAD

Se toman tres (3) parciales teóricos/prácticos que deberán aprobar con un mínimo del 40%, pudiéndose recuperar solamente uno de los tres.

CONDICIONES PARA OBTENER LA PROMOCIÓN

Deberán aprobar los tres parciales ya mencionados con un mínimo del 60%, pudiéndose recuperar solamente uno de los tres. Además deberán tener un promedio mínimo del 70%, no teniéndose en cuenta en el mencionado promedio la calificación del parcial recuperado ya que, en su lugar se usará la nota obtenida en el recuperatorio. El promedio que se trata será la calificación del examen final.



PROGRAMA DE ASIGNATURA

ASIGNATURA: Bases de Datos		AÑO: 2012
CARÁCTER: Obligatoria		
CARRERA: Licenciatura en Ciencias de la Computación		
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.	
UBICACIÓN en la CARRERA: Tercer año – Segundo cuatrimestre		

FUNDAMENTACIÓN Y OBJETIVOS

Lograr que los alumnos sean capaces de:

- Construir modelos de datos de calidad y restricciones de integridad económicas.
- Especificar consultas, disparadores y restricciones de seguridad.
- Realizar planificaciones, analizar su calidad, definir las contemplando protocolos e interbloqueos.

CONTENIDO

Unidad I: Introducción a las Bases de Datos.

Ejemplos de aplicaciones de bases de datos. Sistemas de bases de datos frente a sistemas de archivos. Diseño y administración de Sistemas de Bases de Datos. Escalabilidad, eficiencia y efectividad.

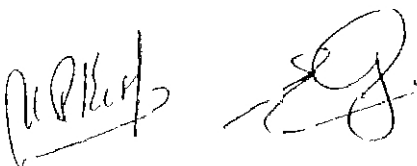
Abstracciones de datos. Ejemplares y Esquemas. Modelado y Calidad de datos. Lenguajes de . DBMS.

Distintos tipos de usuarios de una base de datos. Arquitectura de aplicaciones de bases de datos. Nociones de minería de datos.

Unidad II: Modelo de Entidad-Relación.

Entidades, atributos y conjuntos de entidades. Relaciones y conjuntos de relaciones. Clasificación de

Atributos. Aridad de los conjuntos de relaciones. Correspondencia de cardinalidades. Restricciones de participación. Claves, superclaves y claves primarias tanto de conjuntos de entidades como de conjuntos de relaciones. Diagrama de entidad-



relación. Conjuntos de entidades débiles. Especialización y generalización. Herencia de atributos. Restricciones de diseño sobre las generalizaciones. Agregación. Decisiones de diseño al construir un diagrama de entidad-relación. Reducción de un esquema de entidad-relación a tablas.

Unidad III: Dependencias Funcionales.

Estructura básica de las bases de datos relacionales. Esquema de la base de datos relacional. Claves. Dependencias funcionales: conceptos básicos, cierre de un conjunto de dependencias funcionales, cierre de un conjunto de atributos, recubrimiento canónico. Descomposición. Propiedades deseables de una descomposición: descomposición de reunión sin pérdida y conservación de las dependencias. Repetición de la información.

Unidad IV: Formas Normales.

Primera forma normal. Forma normal de Boyce-Codd (FNBC): definición, algoritmo de descomposición, conservación de las dependencias. Tercera forma normal (3FN): definición, algoritmo de descomposición. Comparación de Forma normal de Boyce-Codd con tercera forma normal.

Unidad V: Modelo Relacional.

Lenguajes de consulta. Álgebra relacional: operaciones fundamentales, otras operaciones, operaciones del álgebra relacional extendida, modificación de la base de datos, vistas. El cálculo relacional de tuplas: definición formal, seguridad de las expresiones.

Unidad VI: SQL

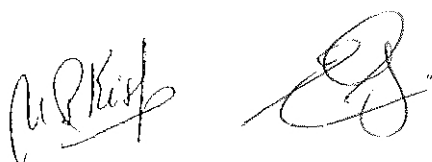
Cláusulas *select*, *from* y *where*. La operación de renombramiento. Variables tupla. Operaciones sobre Cadenas. El orden de la presentación de las tuplas. Operaciones sobre conjuntos. Funciones de agregación. Manejo de valores nulos. Subconsultas anidadas. Vistas. Modificación de la base de datos. Reunión de relaciones. Lenguaje de definición de datos: tipos de dominios en SQL, definición de esquemas en SQL.

Unidad VII: Integridad y Seguridad.

Restricciones de los dominios en SQL. Integridad referencial. Integridad referencial en SQL. Aserciones. Aserciones en SQL. Disparadores. Disparadores en SQL. Seguridad y autorización: medidas de seguridad en varios niveles, autorizaciones, concesión de privilegios, papeles, trazas de auditoría. Autorización en SQL: privilegios en SQL, papeles, el privilegio de conceder privilegios.

Unidad VIII: Transacciones.

Concepto de transacción. Propiedades ACID. Estados de una transacción. Planificaciones. Planificaciones Secuenciales. Importancia de Ejecuciones concurrentes de varias transacciones. Secuencialidad. Secuencialidad en cuanto a conflictos. Secuencialidad en cuanto a vistas. Recuperabilidad. Planificaciones recuperables. Retrocesos en cascada. Planificaciones sin cascada. Definición de



transacciones en SQL. Grafo de precedencia para una planificación. Comprobación de la secuencialidad en cuanto a conflictos.

Unidad IX: Control de Concurrencia.

Esquemas de control de concurrencia. Protocolos de control de concurrencia. Bloqueos. Funciones de compatibilidad. Interbloqueos. Protocolos de bloqueo. Planificaciones legales bajo un protocolo de bloqueo. Inanición de transacciones. Protocolo de bloqueo de dos fases y variantes estricta, rigurosa y con conversión de bloqueos del mismo. Protocolo de ordenación por marcas temporales. Regla de escritura de Thomas. Protocolo basado en validación. Granularidad múltiple: modos de bloqueo intencionales, función de compatibilidad, protocolo de bloqueo de granularidad múltiple.

Unidad X: XML.

Documentos XML, DTD, lenguajes de consulta XPATH y XQuery, XSLT.

Proyecto de Taller.

Desarrollo de base de datos de un Estudio Demográfico.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Silberschatz, Korth y Sudarshan. Fundamentos de Bases de Datos. Mc Graw Hill, Cuarta Edición (2002).
- García-Molina, Ullman, Widom. Database System Implementation. Prentice-Hall (2000).

METODOLOGÍA DE TRABAJO

Las clases teóricas son en modalidad de clase magistral, con desarrollo de los contenidos del programa.

En las clases prácticas los alumnos resuelven problemas tendientes a adquirir las competencias descriptas en los objetivos.

Se realiza un proyecto de taller grupal donde los alumnos desarrollan una base de datos para un problema del mundo real.

Se realiza además una actividad para la familiarización con el uso de algún sistema de bases de datos comercial.

EVALUACIÓN

Se toman tres parciales, y se realiza un proyecto de laboratorio grupal con evaluación.

Para la regularidad

- Aprobación de dos de los tres parciales.
- Aprobación del proyecto de laboratorio.

Para la aprobación

- Examen escrito
- Defensa del proyecto de laboratorio

Promoción

- Aprobación de los tres parciales con promedio mínimo 7 (siete).
- Aprobación del proyecto de laboratorio.





UNC

Universidad
Nacional
de CórdobaFAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Curso de Nivelación	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: mensual	CARGA HORARIA: 100 hs.
UBICACIÓN en la CARRERA: Primer año	

FUNDAMENTACIÓN Y OBJETIVOS

Se pretende crear un espacio que permita al ingresante:

- 1- mejorar su preparación y reforzar sus conocimientos matemáticos para los primeros cursos de las carreras de la Facultad,
- 2- revisar y subsanar posibles deficiencias en los temas de aritmética, álgebra, lógica elemental, conjuntos y funciones, abarcados por el curso,
- 3- desarrollar la capacidad de leer y comprender enunciados matemáticos,
- 4- desarrollar habilidad de traducir al lenguaje matemático enunciados y problemas expresados en lenguaje coloquial.
- 5- informarse sobre temas inherentes a su futura vida universitaria.

CONTENIDO

Cálculo Algebraico:

- Revisión de los conjuntos numéricos y su operaciones. Los números naturales, enteros, racionales, irracionales, reales y complejos.
- Transformación algebraica de situaciones problemáticas. Incógnitas y constantes.
- Polinomios. Suma, resta, multiplicación y división de polinomios. Algoritmo de División. Teorema del Resto.
- Ecuaciones lineales con una incógnita. Sistemas de ecuaciones lineales. Resolución de sistemas de dos ecuaciones con dos incógnitas.
- Resolución de ecuaciones de segundo grado. Raíces, propiedades de las raíces.

Anexo Res. CD N° 136/12

- Resolución de ecuaciones fraccionarias. Ecuaciones que pueden transformarse a ecuaciones lineales o cuadráticas. Simplificación de expresiones algebraicas.

Elementos de Lógica y Teoría de Conjuntos

- Teoría básica de conjuntos. Pertenencia, inclusión. Operaciones entre conjuntos: unión, intersección, diferencia, complemento. Producto cartesiano de conjuntos.
- Teoría elemental de lógica. Proposiciones, conectivos lógicos: negación, conjunción, disyunción, condicional o implicación, bicondicional o doble implicación. Combinación de proposiciones con conectivos lógicos.
- Cuantificadores. Negación de cuantificadores.

Funciones

- Funciones: concepto y representación gráfica. Distancia entre puntos. Desigualdad triangular.
- Función constante. Función lineal. Función cuadrática.
- Trigonometría. Funciones trigonométricas. Identidades trigonométricas.

Introducción a la Vida Universitaria

Ambientación del estudiante a la vida universitaria.

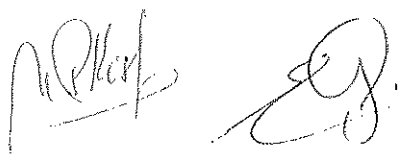
BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Kisbye, P. , Merlo, David. Cálculo algebraico.
- Kisbye, P. , Tiraboschi, Alejandro: Elementos de Lógica y Teoría de Conjuntos.
- Sánchez, C. Funciones

METODOLOGÍA DE TRABAJO

- Modalidad presencial:
 - Clases teóricas y prácticas en grupos, con interacción entre docentes y alumnos.
 - Guías de ejercicios de aplicación que se resuelven grupal o individualmente, con la asistencia y orientación de los docentes.
- Modalidad a distancia
 - Tutorías a través de aula virtual, con evaluaciones presenciales conjuntas con la modalidad presencial.





UNC

Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

EVALUACIÓN

FORMAS DE EVALUACIÓN

- Dos evaluaciones parciales.

CONDICIONES PARA OBTENER LA REGULARIDAD

- Aprobación de uno de los dos parciales
- 70% de asistencia a clases.

CONDICIONES PARA OBTENER LA PROMOCIÓN

- Obtener la regularidad con nota mínima 6(seis) en ambos parciales.



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Física	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Cuarto año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Introducir al alumno en nociones sobre aspectos fundamentales de la física clásica, incorporando el método científico de las ciencias naturales.
Capacitar al alumno para integrar equipos multidisciplinarios de investigación y desarrollo.

CONTENIDO

Unidad I: Repaso de algunos conceptos de álgebra y análisis.

Producto escalar, propiedades; producto vectorial, cálculo con determinantes; integrales de líneas, sistemas de coordenadas: cartesiano y esférico.

Unidad II: Cinemática.

Sistemas de referencia; coordenadas de una partícula puntual. Transformaciones entre sistemas de coordenadas; velocidad y aceleración. Velocidad angular; movimiento circular uniforme. Aceleración centrípeta.

Unidad III: Dinámica.

Leyes de Newton; discusión. El concepto de fuerza. Unidades de fuerza. Ejemplos de aplicación de la leyes de Newton: partícula libre; oscilador armónico. Propiedades del movimiento oscilatorio. Movimiento de un campo de fuerza uniforme. Fuerzas de rozamiento. Ejemplos. Trabajo de una fuerza. Energía potencial y energía cinética. Ley de conservación de la energía. Centro de masa. Conservación del momento lineal.

Unidad IV: Electrostática.

Fenómenos eléctricos elementales. Carga eléctrica. Ley de Coulomb. Campo

Anexo Res. CD N° 136/12

eléctrico. Ejemplos de cálculo: carga puntual y dipolo eléctrico. Ley de Gauss: concepto de flujo de un campo vectorial e integral de superficie. Aplicación al cálculo del campo de una carga puntual. Esfera uniformemente cargada. Conductores. El campo eléctrico en la cercanía de un conductor. Potencial eléctrico. Capacidad.

Unidad V: Corriente Eléctrica.

Cargas eléctricas en movimiento; Resistencia; Ley de Ohm. Resistencias en serie y en paralelo. Resistencia equivalente. Circuitos simples; Leyes de Kirchhoff; Ejemplos de aplicación. Disipación de potencia en una resistencia. Circuitos con capacitores y resistencias.

Unidad VI: Campo Magnético.

Fenómenos magnéticos simples. Definición del campo magnético. Ley de Ampere: campo magnético asociado con una corriente lineal. Campo de un solenoide. Fuerza de Lorentz.

Unidad VII: Inducción electromagnética

Ley de Faraday. Flujo de campo magnético y fuerza electromotriz. Inductancia y auto inductancia. Circuitos con Capacitor-Resistencia e Inductancia. Corriente alterna. Algunos circuitos simples. Fenómeno de Resonancia.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Physics for Computer Science Students, N. García y A. Damask, Springer 1991.

BIBLIOGRAFÍA COMPLEMENTARIA

- Física, Vol. 1 y 2, Serway, R, McGraw-Hill Interamericana, 2001.
- University Physics, Sears y Zemansky, Adison Wesley, 2001.

METODOLOGÍA DE TRABAJO

Se realizan clases teóricas y prácticas.

En las clases teóricas se desarrollan los conceptos básicos. Las clases prácticas están destinadas a la resolución de problemas de aplicación de los conceptos, en base a guías de problemas sugeridos.

Se realizan clases de experimentos demostrativos de las leyes fundamentales discutidas en las clases teóricas y prácticas.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

EVALUACIÓN

Para la regularidad: Se requiere

- el 80% de asistencia a las clases teóricas y prácticas
- aprobación de dos de las tres evaluaciones parciales.

Para la aprobación: Se requiere

- aprobar un examen escrito de resolución de problemas con un nivel equivalente al desarrollado en la materia.

Promoción:

- Aprobar los tres exámenes parciales con calificación mayor o igual a 7 (siete).



Universidad
Nacional
de Córdoba



FAMA F
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Ingeniería del Software I	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Tercer año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno sea capaz de:

- entender actividades de análisis, diseño y testing del software.
- Manejar elementos de planificación, especificación y documentación de proyectos usando el paradigma de OO en análisis y diseño de sistemas.

CONTENIDO

1 - Introducción:

- El dominio del problema. Teoría general de sistemas. Sistemas de Información. Privacidad, integridad y seguridad en sistemas de información.
- El desafío de la Ingeniería del Software.
- El enfoque de la Ingeniería del Software.

2 - Ciclo de vida del software:

- Procesos. Modelo de procesos. Componentes. Enfoque ETVX.
- Características deseadas del proceso del software: Predecible y repetible, Tolerante a cambios, Testeable y Mantenible.
- Proceso de desarrollo del software, etapas fundamentales.
- Reingeniería
- Modelos de procesos de desarrollo: Cascada, Prototipado, Iterativo.
- Otros procesos del software: Administración del proyecto, Proceso de inspección, Administración de configuración, Administración de cambios, Administración del proceso (CMM). Sistemas colaborativos.

Anexo Res. CD N° 136/12

- Ingeniería de software de sistemas de tiempo real
- Nociones de auditoría y peritaje.

3 - Ingeniería de requerimientos:

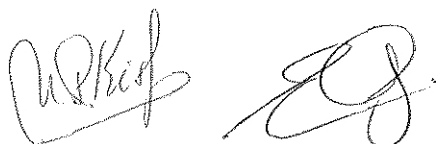
- Requerimientos del software, Necesidad de la especificación de requerimientos, Proceso de requerimientos.
- Análisis del problema: Enfoque informal, Modelo de flujo de datos (DFD), Modelo orientado a objetos (UML), Prototipado.
- Especificación de los requerimientos del software: Características, Componentes, Lenguajes de especificación, Estructura de un documento.
- Especificación funcional con Casos de Uso: Conceptos, Estructura, Abstracción.
- Validación.
- Métricas: Tamaño, Calidad.

4 - Arquitectura del software:

- Rol de la arquitectura del software.
- Vistas: Módulos, Componentes y conectores, Asignación de recursos.
- Vista de Componentes y Conectores (C&C). Estilos arquitectónicos para C&C: Tubos y Filtros, Datos compartidos, Cliente-servidor, Estratificados.
- Conceptos de Arquitecturas Orientadas a Servicios
- Documentación del diseño arquitectónico.
- Arquitectura en comparación con el diseño. Preservación de la integridad de una arquitectura.
- Evaluación de las arquitecturas (método de análisis ATAM).

5 - Planeamiento del proyecto de software:

- Planeamiento del proceso.
- Estimación del esfuerzo: Incertidumbres, Construcción de los modelos (estimaciones top-down y bottom-up), El modelo COCOMO.
- Planificación y recursos humanos: Planificación global y detallada, Estructura del equipo de trabajo.
- Plan del Control de Calidad: Introducción y eliminación de errores, Enfoques, Plan.
- Administración del Riesgo: Conceptos, Evaluación, Control.
- Planeamiento del seguimiento del proyecto: Mediciones, Seguimiento observacional, Registro del seguimiento.



6 - Diseño orientado a funciones:

- Niveles en el proceso de diseño.
- Principios del diseño: Particionado y jerarquía, Abstracción, Modularidad. Estrategias top-down y bottom-up.
- Acoplamiento y Cohesión.
- Notación y especificación del diseño.
- Metodología de diseño estructurado: Cuatro pasos elementales, Heurísticas de diseño, Análisis de transacción.
- Verificación.
- Métricas: de red, de estabilidad y de flujo de información.
- Patrones

7 - Diseño orientado a objetos:

- Conceptos de la orientación a objetos: Clases, Objetos, Relación entre objetos, Herencia, Polimorfismo.
- Conceptos de diseño: Acoplamiento, Cohesión, Principio abierto-cerrado.
- UML.
- Una metodología de diseño: Modelado dinámico, Modelado funcional, Definición de clases y operaciones, Optimización.
- Métricas.
- Diseño centrado en el usuario
- Patrones

8 - Diseño detallado:

- Lenguaje de diseño de procesos (PDL). Diseño lógico (del algoritmo). Modelo de estado de clases (Autómatas de estado finito). Refinamiento en abstracciones de datos e invariantes de representación.
- Verificación: "Recorrido" del diseño, Revisión crítica (bajo proceso de inspección), Verificación de consistencia y Uso de técnicas formales.
- Métricas: Complejidad Ciclomática, Vínculos de datos, Métricas de cohesión.

9 - Codificación:

- Principios y pautas para la programación: Errores comunes, Programación estructurada, Ocultamiento de la información, Prácticas de programación, Estándares de codificación.
- Proceso de Codificación: Incremental, Dirigido por test,



Programación de a pares, Control del código fuente y construcción (build).

- Refactorización: Conceptos básicos, "Malos olores", Refactorizaciones comunes.
- Verificación: Inspección del código, Test de unidad, Análisis Estático, Métodos formales.
- Métricas: Tamaño y Complejidad.

10 – Testing:

- Conceptos fundamentales: Defecto y desperfecto (fault & failure), Oráculos, Casos de test y criterios de selección, Psicología del test.
- Testing de caja negra: Particionado por clases de equivalencia, Análisis de valores límites, Grafo de causa-efecto, Testing de a pares, Casos especiales, Testing basado en estados (Máquinas de estado finitas).
- Testing de caja blanca: Criterios basados en flujo de control, Criterios basados en flujo de datos, Testing por mutación, Generación de casos de tests y herramientas de soporte.
- El proceso de testing: Niveles, Plan, Especificación de los casos de test, ejecución de los casos de test, Análisis, Registro de defectos y seguimiento.
- Análisis y Prevención de los defectos.
- Métricas. Estimación de la confiabilidad

11 – Aspectos profesionales y sociales

- Computación y sociedad.
- Propiedad intelectual, licencia de software y contratos informáticos.
- Aspectos legales.
- Responsabilidad y ética profesional.

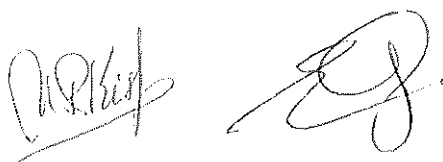
BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Pankaj Jalote. An Integrated Approach to Software Engineering, Third Edition. Springer. 2005. ISBN: 0-387-20881-X.

BIBLIOGRAFÍA COMPLEMENTARIA

- F. Brooks. The Mythical Man Month. Addison-Wesley. 1995.
- R. Glass. The Relationship Between Theory and Practice in Software



Engineering. Communications of the ACM, 39(11):11–13. Nov. 1996.

- R. Glass. Managing for Innovation. Communications of the ACM, 51(3):17–18. Mar. 2008.
- IEEE. Diversos estándares de la IEEE sobre la Ingeniería del Software.

http://standards.ieee.org/reading/ieee/std_public/description/se/.

- B. Meyer. Object-Oriented Software Construction (2nd Edition). Prentice Hall. 2000.
- D. Parnas. Software Engineering: An Unconsummated Marriage. Communications of the ACM, 40(9):128. Sep. 1997.
- Sun Microsystems. Java Code Convention. 1997
- R. Stallman et al. GNU coding standards. 2007.

METODOLOGÍA DE TRABAJO

La materia se organiza en clases teóricas, clases prácticas y actividades de laboratorio. En las clases teóricas se brindan los contenidos fundamentales de la asignatura.

En las clases prácticas se ejercita sobre los temas cubiertos en la teoría, con especial énfasis en actividades de análisis y diseño orientado a objetos, especificación de requisitos y testing.

Las clases de laboratorios se utilizan para llevar adelante un proyecto de desarrollo, de tamaño mediano, que es resuelto en grupos y en el cual los alumnos experimentan los problemas que surgen en el desarrollo de un sistema real, siguiendo todas las etapas que involucra el desarrollo de un proyecto real.

Las clases teóricas son complementadas con charlas de temáticas variadas vinculadas a la ingeniería de software, brindadas por docentes de la asignatura e invitados de la industria local.

EVALUACIÓN

Condiciones para la regularidad

- Aprobación del proyecto, pasando satisfactoriamente cuatro instancias de seguimiento/evaluación del mismo.

Condiciones para la aprobación

- Aprobación de un examen final escrito.
- Defensa del proyecto.

Promoción

- Aprobación del proyecto
- Aprobación de dos evaluaciones parciales con nota no inferior a 7 (siete).





Universidad
Nacional
de Córdoba



FAMA F
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Ingeniería del Software II	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Quinto año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno

- se familiarice con las técnicas de análisis de software y de garantía de calidad avanzadas.
- Comprenda los fundamentos detrás del análisis estático, el model-checking, el análisis basado en constraint solving y el testing.

CONTENIDO

I. El problema de la corrección del software: (1) Definición de sistemas críticos, (2) Limitaciones del testing y la simulación, (3) Discusiones sobre verificación, (4) Métodos formales para Ingeniería de Software.

II. Programación concurrente: (1) Definición de sistemas reactivos, (2) Interacción entre procesos, (3) Los problemas de la concurrencia, (4) Semántica de los programas concurrentes, (5) Interleaving y no determinismo, (6) Razonamiento sobre programas concurrentes, (7) La necesidad de abstraer para modelar, (8) El lenguaje de modelado FSP: sintaxis y semántica, (9) La herramienta LTSA.

III. Sincronización de procesos concurrentes: (1) Recursos compartidos: interferencia y exclusión mutua, (2) Detección de errores, (3) Monitores, sincronización condicional e invariantes del monitor, (4) Semáforos y su invariante, (5) Buffers acotados, (6) Bisimulación como equivalencia de procesos, (7) Comunicación mediante pasaje de mensajes, (8) Pasaje sincrónico de mensajes, (9) Recepción selectiva, (10) Pasaje asincrónico de mensajes, (11) Rendezvous. (12) Transacciones distribuidas.

Anexo Res. CD N° 136/12

IV. Propiedades de los sistemas concurrentes: (1) Categorías de propiedades: *alcanzabilidad*, *safety*, *liveness*, y *fairness*, (2) Necesidad de la categorización de propiedades, (3) Propiedades como conjuntos de trazas, (4) Lenguajes ω -regulares, (5) Formalización de las propiedades de *safety* y *liveness*, (6) Otras propiedades, (7) Análisis automatizado de propiedades usando FSP: *deadlock*, *safety* y *liveness*.

V. Lógicas temporales: (1) Limitaciones de los métodos previos y de las lógicas usuales, (2) Lógicas modales, (3) Introducción a las lógicas temporales, (4) La lógica temporal lineal LTL, (5) Sintaxis y semántica, (6) Operadores derivados y leyes, (7) Especificación de propiedades con LT: *Safety* y *Liveness*, (8) *Fairness*: incondicional, débil y fuerte, (9) Otros tipos propiedades en LTL.

VI. Model checking: (1) El modelo de un sistema, (2) Autómatas de Büchi: definición y uso para presentar programas y propiedades, (3) Model Checking de propiedades LTL con enfoque en la teoría de autómatas, (4) Herramientas de model checking, (5) El model checking de propiedades descritas en LTL Spin, (6) Promela: modelado y análisis, (7) El model checker de propiedades descritas en CTL (computational tree logic) SMV, (8) El model checker de propiedades de tiempo Uppaal, (9) Otros model checkers: *Kronos* y *Zeus*, *Prism* y *Liquor*, etc.

VII. Especificaciones de sistemas: (1) Características de los lenguajes de especificación, (2) Las lógicas como lenguajes de especificación, (3) Lógica proposicional: Sintaxis, semántica y poder expresivo, (4) SAT solving en la lógica proposicional: ventajas y desventajas, (5) Lógica de primer orden: Sintaxis, semántica y poder expresivo, (6) SAT solving en la lógica de primer orden, (7) El álgebra relacional. Sintaxis, Semántica y Axiomas

VIII. El lenguaje de especificación Alloy: (1) Sintaxis del lenguaje Alloy, (2) Características de Alloy, (3) Uso de Alloy para la resolución de problemas con restricciones (constraint solving), (4) Modelos de ejecuciones, (5) Uso de Alloy para verificar refinamientos, (6) Análisis de especificaciones en Alloy: Cotas, cuantificadores no acotados, axiomas de generación.

IX. Algoritmos para verificar satisfactibilidad en lógica proposicional: (1) Algoritmos simples: Tablas de verdad y argumentos semánticos, (2) Algoritmos avanzados, (3) Tablas de verdad revisadas, (4) Conversión a forma normal conjuntiva, (5) Regla de resolución clausal, (6) Propagación de restricciones booleanas, (7) El algoritmo de Davis, Putnam, Logemann & Loveland, (8) Cláusulas de Horn, (9) Linealidad de la resolución en la lógica de Horn, (10) La lógica de Horn como base de la programación lógica y los demostradores automáticos de teoremas.

X. Testing: (1) Definición del testing basado en modelos, (2) Testing con modelos formales, (3) El proceso de testing formal, (4) Conformidad corrección y exhaustividad, (5) La teoría de conformidad de testing basada en entradas y salidas (ioco: Input/Output Conformance Testing), (6) Extensión con tiempo y canales de la teoría ioco, (7) Definición de cubrimiento semántico.



BIBLIOGRAFÍA

- [1] J. Magee y J. Kramer. *Concurrency: State Models & Java Programs*, 2nd edition. Wiley 2006.
- [2] B. Alpern y F. Schneider. Defining Liveness. *Information Processing Letter* 21:181-185. 1985
- [3] B. Alpern y F. Schneider. Recognizing Safety and Liveness. *Distributed Computing* 2 (3): 117-126. 1987.
- [4] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
- [5] B. Berard, M. Bidoit, A. Finkel, F. Laroussinie, A. Petit, L. Petrucci, P. Schnoebelen. *Systems and Software Verification Model-Checking Techniques and Tools*. Springer, 2001.
- [6] M. Müller-Olm, D. Schmidt, B. Steffen. Model-Checking: A Tutorial Introduction. En A. Cortesi, G. Filé (Eds.), *Procs. Of SAS'99*, LNCS 1694, pp. 330-354. Springer 1999.
- [7] G. J. Holzmann. *The SPIN Model Checker: Primer and Reference Manual*. Addison-Wesley, 2003.
- [8] D. Jackson. *Software Abstractions: Logic, Language, and Analysis*. MIT Press, 2006.
- [9] A.R. Bradley y Z. Manna. *The Calculus of Computation: Decision Procedures with Applications to Verification*. Springer, 2007.
- [10] P. Jalote. *An Integrated Approach to Software Engineering*, Third Edition. Springer. 2005.
- [11] J. Tretmans. *A formal Approach to Conformance Testing*. PhD Thesis. Univeristeit Twente, The Netherlands, 1992.

METODOLOGÍA DE TRABAJO

La materia se organiza en clases teóricas, clases prácticas y actividades de laboratorio. En las clases teóricas se brindan los contenidos fundamentales de la asignatura siguiendo la selección de capítulos de la bibliografía..

En las clases prácticas se ejercita sobre los temas cubiertos en la teoría, con especial énfasis en actividades de modelado y análisis.

La asignatura incluye el desarrollo de dos trabajos prácticos con entrega, que son resueltos en grupo en un período de una (1) semana cada uno. En estos trabajos prácticos se experimenta con problemas de tamaño más significativo, en comparación con los ejercicios de las clases prácticas, y se integran contenidos de diferentes tópicos de la materia.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

EVALUACIÓN

Condiciones para la regularidad

- Aprobación del proyecto, pasando satisfactoriamente cuatro instancias de seguimiento/evaluación del mismo.

Condiciones para la aprobación

- Aprobación de un examen final escrito.
- Defensa del proyecto.

Promoción

- Aprobación del proyecto
- Aprobación de dos evaluaciones parciales con nota no inferior a 7 (siete).



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Introducción a la Lógica y la Computación	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año – Segundo cuatrimestre	

FUNDAMENTACIÓN y OBJETIVOS

Objetivos:

Esta es una materia en la que se abordan algunos contenidos que constituyen las bases teóricas de las ciencias de la computación. El objetivo es crear un marco de referencia desde el cual fundamentar las prácticas profesionales tanto en el campo de las aplicaciones y los desarrollos, como en el de la investigación teórica o aplicada.

Fundamentación:

Se han definido para esta materia tres grandes ejes de contenidos teóricos que contribuirán a lograr los objetivos propuestos. El primer eje trata de **estructuras ordenadas**, que constituyen la base para la definición de modelos matemáticos, tanto de los lenguajes de programación como de las lógicas que se utilizan para razonar sobre los programas. El segundo eje aborda la **lógica proposicional** a través de una presentación diferente a la ofrecida en materias anteriores, que no pone énfasis en el cálculo, sino en el concepto de demostración. Este abordaje establece las bases para conectar la lógica con otras áreas fundamentales de las Ciencias de la Computación, como el cálculo lambda (a través del isomorfismo de Curry-Howard), y la inteligencia artificial.

Por último, el tercer eje trata sobre **mecanismos de computación y formas de definición de lenguajes formales**, con aplicaciones directas en el desarrollo de los lenguajes de programación, por ejemplo mediante las técnicas de parsing.

CONTENIDO

Unidad I: Relaciones y orden

Noción de Relación. Relaciones de Equivalencia y Particiones. Relaciones de orden. Ordenes Parciales. Conjuntos Parcialmente Ordenados (CPO). Máximos, mínimos, elementos maximales y minimales, ínfimos y supremos. Diagramas de Hasse. Isomorfismo de CPO's y sus propiedades.

Unidad II: Reticulados y Álgebras de Boole.

Noción de reticulado como CPO. Noción de reticulado como estructura algebraica. Equivalencia de las definiciones. Isomorfismo de reticulados (como estructura algebraica). Equivalencia entre las nociones de isomorfismo. Cotas y Complementos. Reticulados distributivos y su caracterización. Álgebras de Boole y sus propiedades. Representación de las álgebras de Boole finitas como álgebras de conjuntos. Teorema de Birkhoff, de Representación de reticulados distributivos finitos. Filtros, Filtros Primos y Maximales.

Unidad III: Cálculo Proposicional: Sintaxis y Semántica.

Nociones básicas de la lógica proposicional: el lenguaje de las proposiciones, los conectivos. Propiedades sintácticas y sus métodos de demostración. La inducción estructural. Noción de verdad. Valuaciones y sus propiedades. Tablas de verdad. Noción de Completitud de un conjunto de conectivos.

Unidad IV: Cálculo Proposicional: Deducción Natural.

Noción de demostración: el sistema de deducción natural de Gentzen-Prawitz. Caso intuicionista y clásico: la reducción al absurdo. Teoremas de Corrección y Completitud del cálculo proposicional. Álgebra de Lindembaum.

Unidad V: Autómatas Finitos y Expresiones regulares.

Noción de sistema de estados finito. Cadenas, Alfabetos y Lenguajes. Noción de Autómata finito determinístico (DFA). Transformación de estados. Lenguaje aceptado por un DFA. Autómatas no determinísticos (NFA). Autómatas no determinísticos con movimientos "ε". Construcción de un DFA que acepte el lenguaje de un NFA con mov. "ε". Expesiones regulares. Lenguajes regulares. Teorema de Kleene. Equivalencias entre las diversas maneras de definir un lenguaje regular. Pumping Lemma. Aplicaciones.

Unidad VI: Introducción a las Gramáticas

Noción de Gramáticas Libre de Contextos (CFG). Derivación. Lenguajes Libres de Contexto. Gramáticas Regulares. Obtención de una gramatica regular desde una expresión regular.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Apunte de Cátedra: Lenguajes y Automatas, Alejandro Tiraboschi.
- Apunte de Cátedra: Lógica, Pedro Sanchez Terraf.
- Apunte de Cátedra: Reticulados y Álgebras de Boole, Alejandro Tiraboschi y Héctor Gramaglia.

BIBLIOGRAFÍA COMPLEMENTARIA

- B. Davey, H. Priestley, Introduction to Lattices and Order, Cambridge University Press.
- Jeffrey Ullman; John Hopcroft; Rajeev Motwani. "Introducción a la teoría de autómatas, lenguajes y computación". Prentice Hall, 2002.
- D. Van Dalen, Logic and Structure. Springer Verlag, Berlín

METODOLOGÍA DE TRABAJO

Las actividades que sustentan el proceso de enseñanza-aprendizaje se diseñan articulando principalmente dos metodologías consideradas de base en esta propuesta de enseñanza: por un lado la asimilación de cuerpos de conocimiento, fuertemente centrada en los métodos inductivos y de instrucción; y la resolución de problemas estructurados.

Las actividades en el aula se organizan según el esquema tradicional de división en clases teóricas y clases prácticas. Las clases teóricas se sustentan con exposiciones por parte del docente, alentando la intervención de los estudiantes a través de preguntas. Las clases prácticas se organizan en torno a una guía de ejercicios que los alumnos van desarrollando (generalmente trabajando en grupo), contando con la asistencia de una cantidad suficiente de docentes que les permitan cumplir con las actividades propuestas.

EVALUACIÓN

FORMAS DE EVALUACIÓN

Se tomarán 3 (tres) exámenes parciales teórico-prácticos. Las evaluaciones parciales serán sobre contenidos teórico-prácticos.

- Si la cátedra lo considera necesario se podrán incorporar otras instancias de evaluación formativa.



- La materia contempla un régimen de promoción.
- La aprobación de la materia se dará por promoción, o mediante la aprobación de un examen final en las fechas destinadas a exámenes en el calendario académico. El examen final contará de una evaluación escrita sobre contenidos teórico-prácticos, complementada con una instancia oral si el tribunal lo considera necesario.

CONDICIONES PARA OBTENER LA REGULARIDAD

Aprobación de 2 exámenes parciales, con calificación mayor o igual a 4.

CONDICIONES PARA OBTENER LA PROMOCIÓN

- Aprobación de los 3 exámenes parciales, con una calificación mayor o igual a 6 y promedio mínimo 7.
- Aprobar un coloquio final.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Introducción a los algoritmos	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Primer año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno adquiera:

- capacidad de análisis de problemas
- formalización de soluciones a problemas
- manipulación de expresiones formales
- pruebas de corrección de expresiones formales
- familiaridad con conceptos básicos de programación.

CONTENIDO

Unidad I: Introducción.

Objetivos de la materia. Metodología de trabajo. Historia de la computación. Software libre.

Unidad II: Cálculo proposicional.

Operadores booleanos: equivalencia, disyunción, conjunción, implicación, negación, discrepancia. Tablas de verdad. Leyes para manipularlos. Propiedades. Estructura de las pruebas formales. Demostraciones.

Unidad III: Aplicaciones del cálculo proposicional.

Representación del conocimiento en lógica proposicional. Caballeros y pícaros. Análisis de razonamientos.

Unidad IV: Introducción a la programación funcional.

Formalismo básico. Modelo computacional. Desarrollo de programas recursivos. Demostraciones por inducción. Introducción a la derivación de programas. Listas: Constructores y operadores y propiedades. Ejemplos y resolución de problemas.

Unidad V: Cálculo de predicados.

Noción de predicado. Cuantificador universal. Cuantificador existencial. Enfoque

Anexo Res. CD N° 136/12

semántico (interpretación) y enfoque sintáctico (leyes). Demostraciones.

Unidad VI: Especificaciones.

Representación del conocimiento en lógica de predicados. Concepto de especificación formal de un problema. Ejemplos y resolución de problemas.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- [1] J. Blanco and S. Smith and D. Barsotti, *Cálculo de Programas*, 2008.

BIBLIOGRAFÍA COMPLEMENTARIA

- [1] D. Gries and F. Schneider, *A Logical Approach to Discrete Math*, Springer Verlag, 1993.
- [2] R. Smullyan, *What is the name of this book?*, Penguin, 1990.

METODOLOGÍA DE TRABAJO

La materia se articula en dos sesiones semanales de 4 (cuatro) horas de teórico-práctico con el grupo dividido en comisiones.

El 50% de las sesiones prácticas se llevan a cabo en laboratorios de computación, donde se usan herramientas específicas para la implementación computacional y verificación de los ejercicios resueltos en papel.

EVALUACIÓN

Condiciones para la regularidad


- Aprobación de los dos parciales, o un parcial y su recuperatorio con nota igual o superior al 50%.

Condiciones para la aprobación

- Aprobación de un examen final escrito o promoción directa.

Condiciones para la promoción

- Aprobación de los dos parciales, o un parcial y su recuperatorio con nota igual o superior al 70%.



PROGRAMA DE ASIGNATURA

ASIGNATURA: Lenguajes Formales y Computabilidad	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Cuarto año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno maneje con madurez los siguientes conceptos:

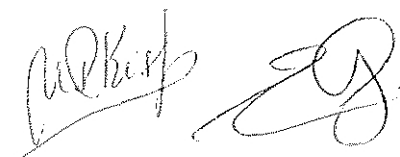
- lenguajes libres de contexto
- maquinas de estado finito (autómatas a pilla y maquinas de Turing)
- funciones recursivas, funciones computables y funciones Turing computables, y su equivalencia
- computabilidad efectiva y Tesis de Church
- conjuntos recursivamente enumerables y conjuntos recursivos
- el halting problem

Estos conceptos le permitirán acceder a ideas y habilidades fundamentales para el desempeño en las ciencias de la computación teórica.

CONTENIDO

Capítulo 1

Gramáticas libres de contexto. Lenguajes libres de contexto. Derivaciones leftmost. Automatas a pila. Equivalencia de lenguajes aceptados por vaciamiento de pila y por alcance de estado final. Equivalencia entre los lenguajes libres de contexto y los lenguajes aceptados por automatas a pila.



Capítulo 2

Funciones Σ -mixtas. Identificación entre Σ^* y ω para un orden total sobre Σ . Funciones Σ -recursivas y Σ -recursivas primitivas. Conjuntos Σ -recursivos y Σ -recursivos primitivos. Lema de división por casos. Iteración de funciones Σ -recursivas primitivas. Cuantificación acotada de predicados Σ -recursivos primitivos. Minimización acotada de predicados Σ -recursivos primitivos. Lema de independencia del alfabeto (sin demostración).

Capítulo 3

El lenguaje imperativo S asociado a un alfabeto finito Σ . Sintaxis y semántica. Macros. Funciones Σ -computables. Equivalencia entre funciones Σ -computables y Σ -recursivas. Forma normal de Kleene. El halting problem. Caracterización de los conjuntos Σ -recursivamente enumerables.

Capítulo 4

Maquinas de Turing. Lenguaje aceptado por una maquina de Turing.(por detención y por alcance de estado final). Equivalencia entre funciones Σ -Turing computables y Σ -recursivas y entre lenguajes Σ -recursivamente enumerables y lenguajes aceptados por maquinas de Turing.

BIBLIOGRAFÍA

- BELL and MACHOVER, A Course in Mathematical Logic, North-Holland, 1986.
- M. DAVIS and E. WEYUKER, Computability, Complexity and Languages, Academic Pres 1983.
- J. HOPCROFT and J. ULLMAN, Introduction to Automata Theory, Languages and Computacion, Addison-Wesley 1979.

METODOLOGÍA DE TRABAJO

Clases teóricas de aproximadamente dos horas, en las cuales se cubren los contenidos teóricos de la materia y clases prácticas de dos horas en las cuales se asiste al alumno en la resolución de los ejercicios con la finalidad de que se afirmen y esclarezcan los conceptos introducidos en la teoría.

EVALUACIÓN

FORMAS DE EVALUACIÓN

Se toman tres parciales para evaluar regularidad y/o promoción de la parte práctica de la materia. Los exámenes finales consisten de una parte práctica y una teórica, en general tomadas por separado. La parte práctica se toma por medio de un escrito





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

de cuatro horas aproximadamente y la parte teórica se toma ya sea por medio de un escrito de dos horas o por medio de un examen oral de duración aproximada de una hora.

CONDICIONES PARA OBTENER LA REGULARIDAD Y PROMOCIÓN

Dos o más parciales aprobados con nota mayor o igual a 4 aseguran la regularidad y la parte práctica se promociona con los tres parciales aprobados con nota mayor o igual a 4 y promedio de los tres mayor o igual a 7.



UNC
Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Lenguajes y Compiladores	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Quinto año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Los lenguajes de programación son un ejemplo fructífero para mostrar las diversas maneras de dar significado a un lenguaje, haciendo cada una de ellas un aporte sustantivo a la comprensión y utilidad del mismo. Decimos que el ejemplo es fructífero porque los diferentes ambientes de uso y estudio de los mismos proponen maneras muy distintas de acceder al significado de un programa, que van desde la descripción intuitiva de un sentido finito y dinámico (un manual), hasta una modalidad estática del significado, vigente en la lógica formal y la matemática (denotación). En este curso ocupa un lugar destacado el último enfoque. A partir del desarrollo de la Teoría de Dominios la semántica denotacional adquiere un relevancia especial, no sólo por tratarse de objetos matemáticos perfectamente definidos en el contexto de una teoría particular, sino además porque comienza a ser utilizada como "la definición" del lenguaje y luego, si se proponen otras semánticas (operacional, axiomática), se las demuestra correctas con respecto a dicha definición.

El objetivo general de la asignatura es lograr que los estudiantes se apropien de las herramientas más importantes que actualmente se utilizan para dar significado a las frases de un lenguaje de programación y de esta manera adquirir los conceptos fundamentales que permitan la construcción de compiladores. Dentro de los objetivos específicos, mencionamos como relevantes:

- el acceder al uso de herramientas matemáticas apropiadas
- disponer de recursos para evaluar las características principales de los distintos lenguajes actualmente en uso
- reconocer propiedades deseables en lenguajes de programación y las herramientas para garantizarlas
- proveer de recursos para que el estudiante pueda diseñar e implementar lenguajes de programación

Anexo Res. CD N° 136/12

CONTENIDO

Unidad I (Introdutoria) Herramientas básicas para dar semántica a lenguajes formales

Las ideas de semántica axiomática, operacional y denotacional. Gramática abstracta, sintaxis abstracta. Lenguaje y metalenguaje. Dirección por sintaxis. Semántica composicional. Lógica de predicados: sintaxis y semántica. Variables y ligadura. Sustitución. Propiedades.

Unidad II: Semántica de los lenguajes imperativos.

Semántica denotacional de las construcciones básicas de un lenguaje imperativo. El problema de dar significado a la iteración. Introducción a la Teoría de Dominios. Función continua y teorema del menor punto fijo. Aplicación al problema motivador. Semántica denotacional de un lenguaje imperativo simple. Propiedades. Fallas y manejo de excepciones. Output. Input. Semántica operacional para el lenguaje imperativo.

Unidad III: Semántica de los lenguajes aplicativos.

Lenguaje aplicativo. Sintaxis. Semántica operacional eager y normal: la noción de evaluación, formas canónicas y reglas de evaluación. Tratamiento de errores. Semántica denotacional directa del lenguaje aplicativo. Sintaxis y semántica de la recursión en las modalidades eager y normal. Propiedades.

Unidad IV: Semántica de la combinación de paradigmas.

Los problemas de la combinación de paradigmas. Las nociones de estado, ambiente, identificador y variable. Un lenguaje que combina los paradigmas. Funciones vs. procedimientos. Pasaje de parámetros. Semántica denotacional.

Unidad V: Otros tópicos de interés

Semántica de continuaciones. Tuplas, patrones y unión disjunta en los lenguajes aplicativos puros. Tipos.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Fridlender, D., Apuntes de Cátedra (basados en el libro de Reynolds).
- Reynolds, Theories of Programming Languages, Cambridge University Press, 1998.

BIBLIOGRAFÍA COMPLEMENTARIA

- Tennet, R., Semantic of Programming Languages, Prentice Hall.



METODOLOGÍA DE TRABAJO

El desarrollo de los contenidos se basa en la modalidad de exposición, empleando en este tipo de actividad un 50% del tiempo de aula. En las mismas se estimula la participación de los alumnos a través del análisis de casos de estudio y preguntas, vinculando la actividad con la práctica.

Para el desarrollo de actividades prácticas se trabaja con una guía de ejercicios adoptando una modalidad de resolución grupal o individual según la preferencia del alumno. Se trabaja bajo la orientación de los docentes respetando los tiempos de cada grupo o individuo.

En las clases teóricas y prácticas se presentan ejercicios ejemplificadores para su discusión en clase. La guía de práctica consta por un lado de una serie de ejercicios que tiene por objetivo afianzar los contenidos trabajados en el teórico, y por otro de un conjunto de problemas más complejos que fomentan la integración de todos los saberes, incentivando el pensamiento analítico y crítico.

EVALUACIÓN

FORMAS DE EVALUACIÓN

- Se tomarán 3 (tres) exámenes parciales teórico-prácticos. Las evaluaciones parciales serán sobre contenidos teórico-prácticos.
- Si la cátedra lo considera necesario se podrán incorporar otras instancias de *evaluación formativa*.
- La materia contempla un régimen de promoción.
- La aprobación de la materia se dará por promoción, o mediante la aprobación de un examen final en las fechas destinadas a exámenes en el calendario académico. El examen final contará de una evaluación escrita sobre contenidos teórico-prácticos, complementada con una instancia oral si el tribunal lo considera necesario.

CONDICIONES PARA OBTENER LA REGULARIDAD

- Aprobación de 2 exámenes parciales, con calificación mayor o igual a 4.

CONDICIONES PARA OBTENER LA PROMOCIÓN

- Aprobación de los 3 exámenes parciales, con una calificación mayor o igual a 7.
- Aprobación del taller.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Lógica	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Cuarto año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Lograr que el alumno maneje con madurez conceptos básicos de la lógica de primer orden. Estos conceptos le permitirán acceder a ideas y habilidades fundamentales para el desempeño en las ciencias de la computación teórica.

CONTENIDO

Capítulo 1

Conjuntos parcialmente ordenados. Diagramas de Hasse. Elementos maximales, máximos y supremos. Homomorfismos de posets. Reticulados. Equivalencia de la definición geométrica y la algebraica. Subreticulados. Homomorfismos de reticulados. Congruencias de reticulados. Relación entre congruencias y homomorfismos. Reticulados acotados. Subreticulados acotados. Homomorfismos y congruencias de reticulados acotados. Reticulados complementados. Subreticulados complementados. Homomorfismos y congruencias de reticulados complementados. El teorema del filtro primo. Lema de Rasiowa y Sikorski.

Capítulo 2

Tipos de primer orden. Términos. Unicidad de la lectura de términos. Fórmulas. Unicidad de la lectura. Variable libres y acotadas. Reemplazos.

Capítulo 3

Estructuras de tipo τ . Valor de un término para una asignación en una estructura. Valor de verdad de una fórmula para una asignación en una estructura (Tarski). Substitución. Sentencias universalmente válidas. Equivalencia de fórmulas.

Capítulo 4

Tipos algebraicos. Algebras. Subuniversos y subalgebras. Producto directo de dos

Anexo Res. CD N° 136/12

álgebras. Homomorfismos. Congruencias. Teorema del isomorfismo. El álgebra de términos. Identidades y el teorema de Completitud de la lógica ecuacional (Birkhoff).

Capítulo 5

Teorías de primer orden. Modelos. Concepto de prueba formal. Teorema de corrección. Consistencia. El álgebra de Lindembaum de una teoría. Teorema de completitud de Godel. Teorema de compacidad. Aplicaciones.

Capítulo 6

La aritmética de Peano. Algunos teoremas básicos. Inducción completa. El modelo estandard. Existencia de modelos no estandard. Análisis de recursividad del lenguaje de primer orden: los teoremas forman un conjunto recursivamente enumerable. Funciones representables. La función β de Godel. Toda función primitiva recursiva es representable. Teorema de incompletitud de Godel.

BIBLIOGRAFÍA

- BELL and MACHOVER, A course in mathematical logic, North-Holland, 1986.
- EBBINGHAUS, FLUM and THOMAS, Mathematical Logic, UTM, Second Edition, Springer-Verlag.

METODOLOGÍA DE TRABAJO

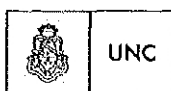
Clases teóricas de aproximadamente dos horas, en las cuales se cubren los contenidos teóricos de la materia y clases prácticas de dos horas en las cuales se asiste al alumno en la resolución de los ejercicios con la finalidad de que se afirmen y esclarezcan los conceptos introducidos en la teoría.

EVALUACIÓN

FORMAS DE EVALUACIÓN

Se toman tres parciales para evaluar regularidad y/o promoción de la parte práctica de la materia. Los exámenes finales consisten de una parte práctica y una teórica, en general tomadas por separado. La parte práctica se toma por medio de un escrito de cuatro horas aproximadamente y la parte teórica se toma ya sea por medio de un escrito de dos horas o por medio de un examen oral de duración aproximada de una hora.





UNC

Universidad
Nacional
de Córdoba



1613 - 2013
400
AÑOS



FAMAF

FAMAF
Facultad de Matemática,
Astronomía y Física

CONDICIONES PARA OBTENER LA REGULARIDAD Y PROMOCIÓN

Dos o más parciales aprobados con nota mayor o igual a 4 aseguran la regularidad y la parte práctica se promociona con los tres parciales aprobados con nota mayor o igual a 4 y promedio de los tres mayor o igual a 7.

PROGRAMA DE ASIGNATURA

ASIGNATURA: Matemática Discreta I	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Primer año – primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

- Aprendizaje y manejo de los siguientes temas:
- Números naturales, principio de inducción y recursividad.
- Técnicas de conteo y nociones de combinatoria básica.
- Números enteros, divisibilidad, números primos y congruencias.
- Rudimentos de la teoría de grafos.

CONTENIDO

Unidad I: Números naturales.

Números reales: definición axiomática. Conjuntos inductivos. Definición de los números naturales. El Principio de Inducción. Definiciones recursivas. Conjuntos bien ordenados. Principio de Buena Ordenación. Equivalencia entre los Principios de Inducción y Buena Ordenación. El Principio de Inducción Fuerte.

Unidad II: Conteo.

Técnicas de conteo. Principio de los casilleros. Principio de adición y multiplicación. Arreglos, permutaciones y combinaciones. El número combinatorio: definición, propiedades y aplicaciones. Fórmula del binomio. Funciones inyectivas, suryectivas y biyectivas.

Unidad III: Divisibilidad.

Números enteros: definición y propiedades básicas. Algoritmo de la división. Desarrollos en base b . Suma y productos en base b . Máximo común divisor y mínimo común múltiplo. Algoritmo de Euclides. Números primos. Teorema fundamental de la Aritmética. Aplicaciones.



Unidad IV: Congruencias.

Definición y propiedades de la relación de congruencia. Reglas de divisibilidad. Ecuaciones de congruencias. Criterios para la existencia de soluciones de ecuaciones lineales de congruencia. El pequeño teorema de Fermat. Encriptación RSA.

Unidad V:

Números complejos. Definición y propiedades básicas. Representación gráfica: el plano complejo. La norma y el módulo de un número complejo. Argumento de un número complejo. Representación trigonométrica o polar de un número complejo.

Unidad VI: Grafos.

Definición y distintos tipos de representación. Valencia y paridad de un vértice. Lema del apretón de manos. Isomorfismo de grafos. Caminos, caminatas, ciclos. Ciclo Hamiltoniano y caminata Euleriana. Circuito Euleriano. Existencia de caminatas y circuitos Eulerianos. Árboles: definición y propiedades.

BIBLIOGRAFÍA**BIBLIOGRAFÍA BÁSICA**

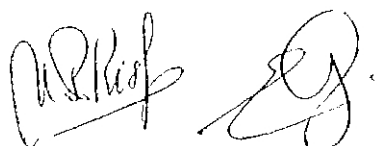
- P. Kisbye, R. Miatello, Notas de Álgebra I, Matemática Discreta I, Trabajos de Matemática, Serie C, FaMAF, 2004/32.

BIBLIOGRAFÍA COMPLEMENTARIA

- E. Gentile, Notas de Álgebra I, Eudeba, Universidad de Buenos Aires, 1976.

METODOLOGÍA DE TRABAJO

- Se busca intercalar la teoría con los ejercicios, motivando los resultados teóricos previamente con ejemplos ilustrativos. A continuación de ellos se presentan los teoremas, intentando dar aplicaciones de los mismos. A manera de ejemplo, para introducir la teoría de conteo, se hace primero hincapié en técnicas de conteo y luego en la definición formal por medio de funciones inyectivas, suryectivas y biyectivas.
- Se estimula la interacción con los estudiantes por medio de preguntas, con interrupciones para permitir su participación activa, logrando una muy buena respuesta. En particular, se incentiva a los alumnos a realizar programas que ejecuten diversos algoritmos, como el algoritmo de división, el algoritmo de Euclides para el m.c.d. y algoritmos para encontrar primos. La interacción de los alumnos de distintas comisiones se facilita a través del aula virtual y de redes sociales.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

- En la segunda parte, principalmente dedicada a ejercicios se plantean problemas, tomados de las guías de prácticos, que luego son resueltos en el pizarrón por algún alumno o bien por el profesor. Esta modalidad permite resolver al frente la gran mayoría de los problemas de las prácticas.

EVALUACIÓN

FORMAS DE EVALUACIÓN

El examen final constará de una evaluación escrita de cuatro horas de duración sobre contenidos teóricos (% 30) y prácticos (% 70).

La materia no tiene régimen de promoción.

CONDICIONES PARA OBTENER LA REGULARIDAD

1. ASISTENCIA: Asistencia al 70% de la totalidad de las clases prácticas.
2. EXÁMENES PARCIALES: Aprobación de 2 exámenes parciales, con posibilidades de un recuperatorio si no se aprueba alguno de los parciales.

PROGRAMA DE ASIGNATURA

ASIGNATURA: Matemática Discreta II	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Tercer año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Que los alumnos adquieran:

- experiencia en el desarrollo de algoritmos complejos y en el análisis de su complejidades.
- Comprensión de conceptos de flujos maximales, matchings, complejidad computacional, P-NP, códigos de corrección de errores, importancia de algoritmos polinomiales.

CONTENIDO

Unidad I: Coloreo

Repaso de la noción de grafo. Notaciones. Coloreo de Grafos. Número cromático. Algoritmo de fuerza bruta. Problema k -Color. Algoritmo 1-COLOR. Definición de bipartito. Repaso de BFS.

Propiedad: un grafo es bipartito si y solo si no tiene ciclo impares. Algoritmo polinomial para 2-COLOR. Algoritmo Greedy de Coloreo. Ejemplo de aplicación. Ejemplo de que no funciona. Propiedad: $\chi \leq \Delta + 1$. Ejemplos de que se alcanza la cota. Algoritmo de Brooks. Teorema de Brooks (sin prueba).

Unidad II: Flujos

Grafos Dirigidos. Ejemplos. Networks. Flujos. Propiedad: $out(s) - in(s) = in(t) - out(t)$. Valor de un flujo. Definición de corte y capacidad de un corte. Propiedad: $v(f) = f(S, S) - f(S, S)$. Corolario: el valor de cualquier flujo es menor o igual que la capacidad de cualquier corte. Corolario: Si el valor de un flujo es igual a la capacidad de un corte entonces el flujo es maximal y el corte es minimal. Teorema : Existen flujos maximales. Algoritmo "Naive" o "Greedy" para encontrar un flujo maximal.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

Ejemplo donde funciona. Ejemplo donde falla. Como modificar el algoritmo para que funcione. El Teorema del flujo maximal-Corte minimal (Max-Flow-Min-Cut Theorem). Algoritmo de Ford-Fulkerson. Ejemplos de aplicación del algoritmo de Ford-Fulkerson. Debilidades del algoritmo de Ford-Fulkerson: ejemplo donde la complejidad no depende del número vértices o lados. Ejemplo donde el algoritmo no termina. Refinamientos: Algoritmos fuertemente polinomiales: Algoritmo de Edmonds-Karp. Complejidad. Algoritmo de Dinic. Complejidad. Algoritmos de pre-flow/push: algoritmo "wave". Complejidad.

Unidad III: Matchings

Matchings en grafos bipartitos. Matchings perfectos y Matchings completos. Ejemplos. Algoritmos para encontrar matchings como aplicación de los algoritmos para encontrar flujos maximales. Modificaciones. Uso de matrices. Definición de $\Gamma(S)$. Condición de Hall. Teorema de Hall. Teorema del matrimonio. (Todo grafo bipartito regular tiene un matching perfecto). Problemas de Matchings Óptimos en grafos bipartitos con pesos. Resolución de "bottleneck problem": problema del asignamiento óptimo cuando se desea minimizar el máximo (o maximizar el mínimo) de los pesos.

Resolución del problema del asignamiento óptimo cuando se desea minimizar (o maximizar) la suma de los pesos: Algoritmo Húngaro. Codificación de complejidad $O(n^3)$ del algoritmo Húngaro.

Unidad IV: P y NP

La clase P. La clase NP. Ejemplos. El problema SAT. El problema k -COLOR. Ejemplo: 2-COLOR es P. Reducción polinomial. Reducción de 3-COLOR a SAT. Las clases de problemas NP-hard y NP-completo. Teorema de Cook (sin prueba): SAT es NP-completo. Teorema: 3-SAT es NP-completo. 2-SAT está en P. Horn-SAT está en P. Teorema: 3-COLOR es NP-completo.

Unidad V: Tópicos de Inteligencia Artificial

Algoritmos de búsqueda. Hill Climbing. Simulated Annealing.

Algoritmos genéticos: Codificación del problema. Fitness. Reproducción de Población. Terminación.

Elementos de la Reproducción: Selección, Crossover, Mutación, Reemplazo.

Algunas posibilidades de Selección para Reproducción: Roulette, SUS, Rank-based selection.

Algunas posibilidades de Crossover. Single point, double or multiple points.

Algunas posibilidades de Mutación. Algunas posibilidades de Reemplazo: Ambos progenitores, Progenitor más débil, Individuos más débiles, Randmo. Ventajas y desventajas.

Arquitectura alternativa a la tradicional: Steady State.

No Free Lunch Theorem. Ejemplos de aplicación.

Unidad VI: Códigos

Definiciones básicas. Distancia de Hamming. Detección de errores. Corrección de

errores. Ejemplos de códigos. Chequeo de paridad. Códigos de repetición. Cota de Hamming. Códigos lineales. Propiedad: C lineal entonces $\delta(C)$ es igual al mínimo peso no nulo. Matrices Generadoras. Códigos lineales como espacios filas de una matriz. Códigos lineales como núcleos de matrices. Matrices de chequeo. Equivalencias entre matrices generadoras y de chequeo. Propiedad: todo código lineal tiene un matriz de chequeo. Proposición: Si en la matriz de chequeo no hay columnas repetidas ni nulas entonces el código correspondiente corrige al menos un error. Algoritmo para corregir un error. Códigos de Hamming. Códigos perfectos. Propiedad: Hamming es perfecto. Propiedad: $\delta(C)$ es igual al menor número de columnas linealmente dependientes de una matriz de chequeo. Singleton Bound. Códigos MDS.

Unidad VII: Códigos Cíclicos

Rotación de una palabra. Códigos cíclicos. Códigos cíclicos mirados como polinomios. Propiedad: todo código lineal (sobre \mathbb{Z}_2) tiene un único polinomio no nulo de menor grado. Polinomio generador de un código cíclico. Propiedades del polinomio generador. Uso del polinomio generador para codificación: dos métodos. Matrices generadoras asociadas a los dos métodos. Obtención en forma directa a partir del polinomio generador de una matriz de chequeo con la identidad a izquierda. Polinomio chequeador. Corrección de errores: error trapping.

Unidad VIII: Códigos de Reed-Solomón

Cuerpos finitos. Definiciones y construcción. Propiedades básicas de los cuerpos finitos. Teorema del elemento primitivo. Códigos de Reed-Solomon. Teorema: los códigos de Reed-Solomon son MDS.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Matemática Discreta. N. Biggs, 1989.
- Applied Combinatorics. Roberts, 1989. Prentice-Hall.
- Data Structures and Network Algorithms. R.E. Tarjan. 1983, Society for Industrial and Applied Mathematics.
- Computers and Intractability: A Guide to the Theory of NP-completeness. Garey and Johnson, 1979, Bell Telephone Laboratories.
- .pdfs de algunos temas en página de Daniel Penazzi.



BIBLIOGRAFÍA COMPLEMENTARIA

- Applied Combinatorics. A. Tucker, 2nd Ed., 1984.
- Network Flows: Theory, Algorithms and Applications. Ahoja-Magnani-Orlin, 1993, Prentice Hall.
- Combinatorial Optimization: Algorithms and Complexity. Papadimitriou-Steiglitz, 1998, Dover Publications.

METODOLOGÍA DE TRABAJO

La asignatura se organiza en clases teóricas y prácticas, de cuatro horas reloj cada una.

Las clases teóricas son expositivas, y las clases prácticas se organizan en comisiones donde los alumnos resuelven de manera independiente o grupal ejercicios prácticos, bajo la supervisión y acompañamiento de los docentes.

Además los alumnos implementan un proyecto en grupos de no más de 5 estudiantes, en su propio tiempo, bajo la orientación y supervisión de los docentes.

EVALUACIÓN

FORMAS DE EVALUCIÓN

- Se tomarán dos parciales (solo prácticos) más un recuperatorio para los que hayan aprobado uno solo.
- Se deben entregar proyectos de programación, el número y el tipo especificados en clase.
- El examen final consta de una parte práctica y una parte teórica, ambas valen 50% de la nota y deben aprobarse por separado. Los alumnos libres deben hacer un ejercicios extra, que vale -1 punto si no se hace o se hace mal. Los alumnos que no aprueben el proyecto deben hacer otro ejercicio extra más, que vale -7 puntos si no se hace o se hace mal.

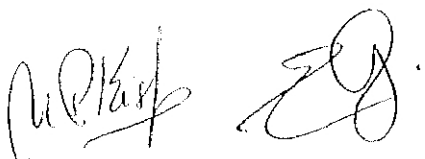
CONDICIONES PARA OBTENER LA REGULARIDAD

1. EXÁMENES PARCIALES

- Aprobar los dos parciales con al menos 4 (cuatro), o uno de ellos con al menos 4 (cuatro) y aprobar el recuperatorio.

2. PROYECTO DE LABORATORIO

- Aprobar un número mínimo de proyectos.





Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Modelos y Simulación	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Cuarto año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

El objetivo de la asignatura es proporcionar al estudiante una formación sólida en el manejo de los conceptos y técnicas utilizados en la simulación de sistemas, a través del procesamiento digital de modelos matemáticos probabilísticos.

CONTENIDO

Unidad I: Revisión de fundamentos de Probabilidad y Estadística.

Axiomas de probabilidad, probabilidad condicional e independencia. Variables aleatorias. Valor esperado y varianza. Desigualdad de Chebyshev y Ley de los grandes números.

Variables aleatorias discretas: Distribuciones binomial, Poisson, geométrica, binomial negativa, hipergeométrica.

Variables aleatorias continuas: Uniforme, normal, exponencial, gamma.

El proceso de Poisson homogéneo y no homogéneo.

Unidad II: Generación de números pseudoaleatorios

Generadores congruenciales y combinaciones. El método de Monte Carlo. Aplicaciones del método de Monte Carlo para el cálculo de integrales.

Unidad III: Generación de variables aleatorias discretas

Método de la transformada inversa. Generación de una permutación aleatoria. Método de composición. Método del alias. Método de aceptación y rechazo. Casos especiales para la generación de variables aleatorias binomiales, geométricas y de Poisson.



Unidad IV: Generación de variables aleatorias continuas.

Método de la transformada inversa. Método de aceptación y rechazo. Método polar para la generación de variables aleatorias normales. Generación de un proceso de Poisson homogéneo. Generación de un proceso de Poisson no homogéneo.

Unidad V: Simulación por medio de eventos discretos

Sistema de cola de espera con uno y dos servidores. Implementación en proyecto.

Unidad VI: Análisis estadístico de datos simulados

Técnicas de inferencia estadística. Histogramas. Diagramas de caja. Diagramas de scattering. Estimación de parámetros de una distribución. Estimadores de máxima verosimilitud.

La media muestral y la varianza muestral. Estimadores por intervalos de la media de una población y de una proporción. La técnica Bootstrap para la estimación del error cuadrático medio de un estimador.

Unidad VII: Técnicas de validación estadística

Tests de bondad de ajuste. El test ji-cuadrado para datos discretos. El test de Kolmorov-Smirnov para datos continuos. Bondad de ajuste con parámetros no especificados. El problema de dos muestras: test de rangos de Mann-Whitney o Wilcoxon. El problema de varias muestras: test de Kruskal-Wallis.

Validación de hipótesis de un proceso de Poisson homogéneo y no homogéneo.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Sheldon M. Ross, *Modelos y Simulación*, Prentice Hall, 2da. edición, (1999).
- Sheldon M. Ross, *Simulation*, Academic Press, 3rd. edition, (2002).

BIBLIOGRAFÍA COMPLEMENTARIA

- Averill M. Law, W. David Kelton, *Simulation Modelling and Analysis*, Mc. Graw Hill, 3ra. edición, 2000
- George Marsaglia and Arif Zaman, *Some portable very-long-period random number generators*, *Computers in Physics*, (8)1, 117 (1994).
- Numerical Recipes: <http://www.nr.com/oldverswitcher.html>



UNC

Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

METODOLOGÍA DE TRABAJO

Las clases teóricas se desarrollan con filminas y pizarrón, con modalidad de clase magistral.

Las clases prácticas se desarrollan mayormente en el laboratorio de computación, a fin de que los estudiantes puedan desarrollar los algoritmos en computadora.

La asignatura incluye el desarrollo de dos proyectos, en la mitad y hacia el final de la cursada respectivamente, bajo la orientación y supervisión de los docentes.

EVALUACIÓN

Condiciones para la regularidad

- Aprobación de dos de los tres parciales.
- Presentación de los dos proyectos especiales, en tiempo y forma.

Condiciones para la aprobación

- Aprobación de un examen final escrito.
- Defensa de los proyectos especiales.

Promoción:

- Aprobación de los tres parciales, con promedio mínimo 6(seis).
- Aprobación de los dos proyectos especiales.



Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Organización del computador	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Que el alumno sea capaz de reconocer las unidades constitutivas de un sistema de computación, comprender su funcionamiento interno y la interacción entre ellas.

CONTENIDO

Sección 1: Sistemas Digitales

Unidad 1: Circuitos Lógicos Combinacionales

1.1-Funciones lógicas. Postulados del álgebra de conmutación (Boole). Minimización mediante el uso de los mismos y con métodos numéricos (Quine - McCluskey).

1.2-Circuitos lógicos de bajo y medio nivel de integración.

Unidad 2: Circuitos Lógicos Secuenciales

2.1-Celda básica de memoria ("Flip-Flop D").

2.2-Circuitos lógicos secuenciales síncronos.

2.3-Autómatas de Moore y Mealy.

2.4-Introducción a los circuitos lógicos secuenciales programables.

2.5- "Latches" y "Shift Registers"



Universidad
Nacional
de Córdoba



FAMAFA
Facultad de Matemática,
Astronomía y Física

Unidad 3: Aritmética Binaria

- 3.1-Sistemas binarios de numeración.
- 3.2-Representación de números negativos.
- 3.3-Puntos fijo y flotante.
- 3.4- Máquinas algorítmicas para aritmética binaria. Errores en la representación de los datos a nivel máquina.

Sección 2: Estructuras del Computador Tipo "Von Neumann y Harvard"

Unidad 4: Procesadores

- 4.1-Líneas de direccionamiento, datos y control.
- 4.2-Registros internos.
- 4.3-Modos de direccionamientos.
- 4.4-Instrucciones (Incluye conceptos sobre lenguaje ensamblador ("assembly")).
- 4.5-Interrupciones.

Unidad 5: Memorias

- 5.1- Conceptos fundamentales sobre memorias "Read Only Memory" - ROM, "Programmable Read Only Memory" - PROM, "Erasable Programmable Only Memory" - EPROM y "Electrically Erasable Programmable Read Only Memory" - EEPROM (Introducción a los "Programmable Logic Devices" - PLD). Memoria "FLASH".
- 5.2-Conceptos fundamentales sobre memorias "Random Access Memory" - RAM estáticas (SRAM) y dinámicas (DRAM).
- 5.3-Estructuración o decodificado de bancos de memorias ("Memory Mapped").
- 5.4- Otros tipos de Memorias. Ancho de banda. Jerarquía de memorias. Componentes principales de la jerarquía. Organización funcional.
- 5.5-Sistemas de detección de errores en datos almacenados en memoria ("CHEKSUM").

Unidad 6: Puertos de Entrada/Salida

- 6.1-Puerto paralelo. Su estructuración y utilización.
- 6.2- Puerto serie. Su estructuración y utilización.

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL:

- Patterson, David y Hennesy, John. *Estructura y diseño de computadores. La interfaz hardware/software*. Editorial Reverté. 4ta. Edición. Año 2011.
- Stallings, William. *Organización y arquitectura de computadores*. Prentice Hall, 2007.
- Morris Mano, M.: "Ingeniería Computacional, diseño del hardware". Prentice Hall Hispanoamericana S.A., 1992.
- Tanenbaum, A. S.: "Organización de Computadoras, un enfoque estructurado". Prentice Hall Hispanoamericana S. A., 2000.

METODOLOGÍA DE TRABAJO

Clases teóricas: son con exposición dialogada con la intervención del alumno en la elaboración de los conceptos fundamentales. Se utilizan filminas y pizarrón.

Clases prácticas: se realizan resolución de ejercicios, utilización de programas de simulación, lenguajes de programación y prácticas de laboratorio.

EVALUACIÓN

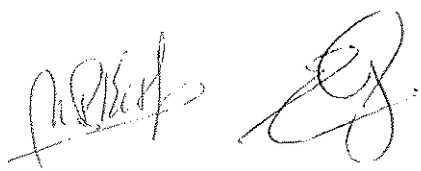
Condiciones para la regularidad:

Los alumnos deberán aprobar tres (3) parciales teóricos/prácticos con un mínimo del 40%, pudiéndose recuperar solamente uno de los tres.

Condiciones para la aprobación:

Aprobación bajo el régimen de promoción: Los alumnos deberán aprobar los tres parciales ya mencionados con un mínimo del 60%, pudiéndose recuperar solamente uno de los tres. Además deberán tener un promedio mínimo del 70%, no teniéndose en cuenta en el mencionado promedio la calificación del parcial recuperado ya que, en su lugar se usará la nota obtenida en el recuperatorio. El promedio que se trata será la calificación del examen final.

Aprobación de finales para quienes no hayan promocionado: Los alumnos deberán aprobar un examen final teórico/práctico escrito. Para aprobar deberán obtener como mínimo un 40% quienes hayan regularizado la materia, y un 60% quienes rindan la materia en condición de libre.



PROGRAMA DE ASIGNATURA

ASIGNATURA: Paradigmas de Programación	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Tercer año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Que el alumno pueda:

- Aprender conceptos fundamentales de lenguajes de programación.
- Reconocer aspectos comunes en lenguajes de programación desestimando las diferencias superficiales.
- Conocer nociones fundamentales sobre paradigmas de programación mayoritarios.

CONTENIDO

1- Introducción

- Introducción a los paradigmas de programación: Imperativo, Orientado a Objetos, Funcional, Lógico.
- Criterios de Diseño y de Implementación de Lenguajes de Programación.
- Relación entre los distintos formalismos de cómputo.
- Entidades y ligaduras.
- Sistema de tipos. Niveles de polimorfismo.
- Encapsulamiento y Abstracción.
- Tutorial de Oz
-

2- Semántica de Modelos Declarativos

- Sintaxis, Semántica, Stores, Entornos, Tipos Declarativos.
- Máquina Abstracta, Semántica del Lenguaje Formal de Núcleo.

3- Traducción de Lenguajes

- Manejo de Memoria, Excepciones, Unificación, Típeo Dinámico y Estático

4- Técnicas para Programación Declarativa

- Recursión como Iteración, Diseño Orientado a Tipos
- Acumuladores e Invariantes, Recursión en Listas.
- Programación de Alto Orden.
- ADTs, Componentes de Software

5- Concurrencia y Paralelismo

- Threads, Streams, Ejecución Perezosa.
- Programas de tiempo real, Haskell, Concurrencia con pasaje de mensajes.

6- Programación con Estados

- Estado Explícito.
- Polimorfismo, mecanismo de pasaje de parámetros.
- Programación Orientada a Objetos.
- Herencia Simple/Múltiple.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

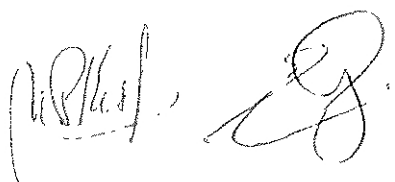
- Peter Van Roy and Seif Haridi. *Concepts, Techniques, and Models of Computer Programming*. MIT Press. 2004

METODOLOGÍA DE TRABAJO

Clases teóricas: clases magistrales, con la utilización de filminas que quedan disponibles para los alumnos. Se promueve el intercambio de ideas y participación de los alumnos.

Clases prácticas: se resuelven guías de ejercicios con y sin requerimientos de programación.

Clases de laboratorio: se resuelven ejercicios con programación.





UNC
Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

EVALUACIÓN

Condiciones de regularidad:

- Aprobación de ambos parciales, o un parcial y un recuperatorio.

Condiciones de aprobación:

- Aprobación de un examen final escrito.

Promoción:

- Aprobación de los dos parciales con nota mínima 8 (ocho)



Universidad
Nacional
de Córdoba



FAMA F
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Probabilidad y Estadística	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

El propósito de este curso es proporcionar una base sólida de la teoría de la Probabilidad necesaria para una mejor comprensión de la Estadística Inferencial a un nivel universitario inicial. Otro objetivo de la materia es destacar su importancia en la resolución de problemas de diversas disciplinas.

CONTENIDO

Unidad I: Estadística descriptiva.

Modelos matemáticos; modelos determinísticos y aleatorios. Población y muestra. Estadística descriptiva de conjuntos numéricos de datos. Métodos gráficos y tabulares para resumir y describir. Histogramas. Distribución de frecuencia de la muestra. Formas cualitativas de histogramas. Medidas de posición: media muestral, mediana muestral y cuartiles. Propiedades de equivariancia. Medidas de variabilidad: desviación estándar, distancia intercuartílica. Box-plot. El coeficiente de variación. Distribución de frecuencias representada por curva normal. Regla empírica.

Unidad II: Probabilidad

Modelos matemáticos; modelos determinísticos y aleatorios. Elementos de un modelo aleatorio o probabilístico: espacio muestral, Familia de eventos, función de probabilidad. Propiedades. Probabilidad de unión de eventos. Espacios finitos equiprobables. Probabilidad condicional. Propiedades. Fórmula multiplicativa, fórmula de la probabilidad total, teorema de Bayes. Independencia de eventos. Esquema de extracción sin reposición.

Unidad III: Variables aleatorias discretas.

Variable aleatoria: definición. Variable aleatoria discreta. Distribución de probabilidad

Anexo Res. CD N° 136/12

o función de probabilidad de masa. Función de distribución acumulada de una variable aleatoria. Propiedades. Esperanza, valor esperado o media de una variable aleatoria (v.a.) discreta. Valor esperado de funciones de variable aleatoria discreta. Varianza y desviación estándar. Propiedades de varianza. Ejemplos de v.a. discretas: distribución de probabilidad binomial, media y varianza. Extracción con reposición de una caja con bolas numeradas. Distribución de Poisson. Aproximación binomial a la distribución de Poisson. Media y varianza de la distribución de Poisson. Distribución hipergeométrica. Extracción sin reposición de una caja con bolas numeradas. Esperanza y varianza de la distribución hipergeométrica. Aproximación binomial a la hipergeométrica. Distribución binomial negativa. Esperanza y varianza.

Unidad IV: Variables aleatorias continuas.

Definición de variable aleatoria continua. Función densidad de probabilidad. Función de distribución acumulada. Percentil de una v.a. con densidad. Valor esperado o valor medio de una v.a. continua. Valor esperado de funciones de v. aleatorias discretas. Varianza y desviación estándar. Ejemplos de distribuciones de v.a. continuas. Distribución uniforme y normal. Media y varianza. Distribución normal estándar. Uso de tablas normales. Cálculo de percentiles de una distribución normal en términos de la distribución normal estándar. Distribución Gamma. Casos particulares: Distribución Exponencial y Distribución Chi-cuadrado. Distribución lognormal. Distribución de Weibull. Distribución Beta. Media y varianza de todas las variables mencionadas.

Unidad V: Distribución de probabilidad conjunta.

Distribución de probabilidad conjunta. Función de probabilidad de masa conjunta de dos v.a. discretas. Caso continuo: Función de densidad de probabilidad conjunta. Funciones de densidad de probabilidad marginales. Variables aleatorias independientes. Caracterización en términos de la factorización de la Función de densidad de probabilidad conjunta o de la función de probabilidad de masa conjunta. Cálculo de esperanza usando distribución de probabilidad conjunta. Covarianza. Coeficiente de correlación. Propiedades.

Unidad VI: Distribución de muestreo y estimación puntual.

Estadísticos. Muestra aleatoria. Media muestral. Distribución en el caso normal. Enunciado del Teorema Central del Límite. Ejemplos. Aproximación normal a la binomial. Esperanza, varianza y covarianza de combinaciones lineales de v.a. Caso de muestra aleatoria de una distribución normal. Estimación puntual. Parámetros de una población o distribución. Estimadores insesgados. Error estándar estimado. Métodos de estimación puntual: Método de los Momentos y Método de Máxima Verosimilitud (EMV). Propiedad de Invarianza para el EMV.



Unidad VII: Intervalos de confianza basados en una sola muestra.

Intervalos de confianza. Nivel de confianza. Intervalo de confianza para la media de una distribución normal con varianza conocida. Longitud del intervalo de confianza. Intervalo de confianza con muestras grandes para la media poblacional y proporción poblacional. Selección del tamaño muestral para lograr una longitud especificada. Intervalo de confianza para la media de una distribución normal con varianza desconocida. Distribución t de Student con n grados de libertad. Uso de tablas de la distribución t de Student para el cálculo de probabilidades. Uso de tablas de la distribución chi cuadrado con v grados de libertad. Intervalo de confianza para la varianza de la distribución normal.

Unidad VIII: Pruebas o tests de hipótesis.

Pruebas o tests de hipótesis. Elementos de un test de hipótesis: hipótesis nula y alternativa, estadístico de prueba, región de rechazo. Error Tipo I y II. Nivel y potencia del test. Tests unilaterales y bilaterales. Pruebas para la media para una m. a. con distribución normal y varianza conocida. Función de potencia. Determinación de tamaño muestral para conseguir una potencia prefijada en una alternativa fija. Tests de nivel aproximado para muestras grandes. Tests para la media para una m.a. con distribución normal con varianza desconocida. Tests para la varianza para una m.a. con distribución normal. Tests de muestras grandes para proporción desconocida. Determinación de potencia y tamaño muestral. P-valores. Relación entre tests bilaterales e intervalos de confianza.

Unidad IX: Inferencia basadas en dos muestras.

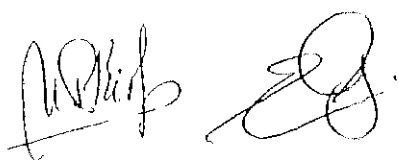
Pruebas Z e Intervalos de Confianza para la diferencia de dos medias poblacionales, para dos muestras aleatorias de tamaños muestrales grandes. Pruebas t e Intervalos de Confianza para la diferencia de dos medias para dos m.a. con distribuciones poblacionales normales.

BIBLIOGRAFÍA**BIBLIOGRAFÍA BÁSICA**

- Jay L. Devore. Probabilidad y Estadística para ingeniería ciencias. Thomson, sexta edición (2005).

BIBLIOGRAFÍA COMPLEMENTARIA

- * Dennis Wackerly, William Mendenhall III, Richard L. Scheaffer. Estadística Matemática con Aplicaciones. Editorial Thomson, 2002.
- * Paul Hoel, Sidney Port and Charles Stone. Introduction to Probability Theory. Houghton Mifflin College, Boston, 1971.
- * Sheldon Ross. Introducción a la Estadística. Editorial Revertè, 2007.





UNC

Universidad
Nacional
de Córdoba1613-2013
400
AÑOSFAMA F
Facultad de Matemática,
Astronomía y Física

METODOLOGÍA DE TRABAJO

Los alumnos tendrán 4 horas semanales de teórico y 4 horas de prácticos. Para el teórico se emplea una clase magistral mientras que las actividades prácticas son grupales o individuales, de acuerdo a las necesidades de los alumnos. Al finalizar el tiempo estimado para la realización de cada guía, se realiza un cierre en la pizarra con la resolución de ejercicios que presentaron mayor dificultad. No se considera necesario notas de clase pues la bibliografía recomendada cumple en gran parte las metas y nivel deseado. Las Guías prácticas están disponibles en la fotocopiadora y página de la facultad, con las tablas necesarias para su resolución. En la página de la facultad (el alumno puede encontrar información de todas las novedades de la materia (como ser fecha de los parciales, horarios de consultas, etc).

EVALUACIÓN

FORMAS DE EVALUACIÓN

Tres evaluaciones parciales.

CONDICIONES PARA OBTENER LA REGULARIDAD

- Tener por lo menos un 70% de asistencia a los Prácticos.
- Aprobación de por lo menos dos de los tres parciales con 4 (cuatro).

Sistema de evaluación en el examen final

- Para el alumno regular:
Para su aprobación deberá tener por lo menos un 40% del examen correcto. Si el alumno tiene el examen final aprobado y además aprobó los TRES parciales, entonces puede ser acreedor de un premio en la forma de puntos extra, conforme a las notas de sus parciales. Dicho premio consistirá en:
 - 1 punto si la suma de los tres parciales es de por lo menos 21.
 - 0.6 puntos si la suma de los tres parciales pertenece al intervalo [18,21)
 - 0.3 puntos si a suma de los tres parciales pertenece al intervalo [15,18).

Además este premio SOLO SERÁ VALIDO PARA LOS TURNOS PREVIOS A QUE SE VUELVA A DICTAR LA MATERIA.

- Para el alumno libre:
Para su aprobación deberá tener por lo menos un 50% del examen correcto.



UNC
Universidad
Nacional
de Córdoba



FAMAF
Facultad de Matemática,
Astronomía y Física

PROGRAMA DE ASIGNATURA

ASIGNATURA: Redes y Sistemas Distribuidos	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Tercer año – Primer cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Que el alumno sea capaz de:

- Comprender los conceptos de las distintas capas de sistemas operativos de redes
- Resolver problemas prácticos concernientes a los protocolos.
- Programar sistemas distribuidos que usan primitivas de APIs de comunicación de redes.

CONTENIDO

Unidad I: Introducción.

Usos de redes de computadores. Hardware de red. Sistemas operativos de red. Modelos de referencia. Software libre.

Unidad II: La Capa Física.

Bases teóricas de comunicación de datos. Análisis de Fourier. Niquist y Shannon. Medios de transmisión guiados y no guiados. Sistema telefónico.

Unidad III: La Capa de Enlace de Datos.

Funciones de la capa de enlace de datos. Tramas. Servicios provistos a la capa de red. Entramado. Control de errores. Control de flujo. Detección y corrección de errores. Protocolos elementales de enlace de datos. Protocolos de ventana corrediza. Modelado mediante autómatas finitos.

Unidad IV: La Subcapa de Control de Acceso al Medio.

El problema de la asignación del canal. Asignación de canal estático. Asignación de canal dinámico. Protocolos de acceso múltiple: ALOHA puro, ALOHA ranurado,

Anexo Res. CD N° 136/12

protocolos de acceso múltiple con detección de portadora (CSMA persistente-1, CSMA no persistente, CSMA persistente-p, CSMA/CD), protocolos libres de colisiones (protocolo de mapa de bits, protocolo de conteo descendente binario), protocolos de contención limitada (protocolo de recorrido de árbol adaptable). Ethernet: cableado Ethernet, codificación Manchester, el protocolo de subcapa MAC de Ethernet, algoritmo de retroceso exponencial binario, Ethernet conmutada. Fast Ethernet. Gigabit Ethernet. Estandar Ethernet 802.2: control lógico del enlace. Redes inalámbricas. Switching en la capa MAC.

Unidad V: La Capa de Red

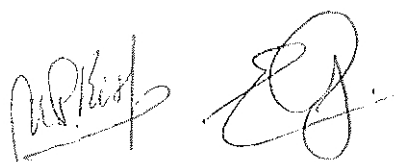
Aspectos de diseño de la capa de red. Conmutación de paquetes de almacenamiento y reenvío, servicios proporcionados a la capa de transporte, implementación del servicio no orientado a la conexión, implementación del servicio orientado a la conexión. Algoritmos de enrutamiento: principio de optimización, enrutamiento de ruta más corta, inundación, enrutamiento de vector de distancia, enrutamiento por estado del enlace, enrutamiento jerárquico, enrutamiento por difusión, enrutamiento por multidifusión. Algoritmos de control de congestión: principios generales del control de congestión, políticas de prevención de congestión, control de congestión en subredes de circuitos virtuales, control de congestión en subredes de datagramas, desprendimiento de carga. Control de fluctuación. Interconectividad: cómo difieren las redes, conectando redes, circuitos virtuales concatenados, interconectividad no orientada a la conexión, entunelamiento, enrutamiento entre redes, fragmentación. Capa de red de Internet (IP): protocolo, direccionamiento, protocolos de control, Ipv6.

Unidad VI: La Capa de Transporte

Primitivas y sockets. Elementos de los protocolos de transporte: direccionamiento, establecimiento, finalización, control de flujo, multiplexado y recuperación de fallas. Protocolo simple de ejemplo de transporte. Protocolos UDP, RTMP, y RPC. Protocolo TCP: modelos de servicio, protocolo, encabezado, establecimiento y fin de conexión, políticas de transmisión y control de flujo. Seguridad de protocolos. Seguridad en redes: autenticación y confidencialidad, passwords, protocolos criptográficos, modelos de seguridad, Needham Schroeder, TLS.

Unidad VII: La Capa de Aplicación

El sistema de nombres de dominio (DNS): el espacio de nombres del DNS, registros de recursos, servidores de nombres. Correo Electrónico: arquitectura y servicios, el agente de usuario, formatos de mensaje, transferencia de mensajes, entrega final. Word Wide Web: panorama de la arquitectura, el lado del cliente, el lado del servidor, protocolos, cookies, documentos web estáticos (HTML), documentos web dinámicos, generación de páginas web del lado del cliente, protocolo de transferencia de hipertexto (HTTP). Seguridad: Javascript XSS. Arquitecturas basadas en servicios. Administración de redes.



Unidad VIII: Nociones de Sistemas Distribuidos

Sincronización: sincronización de relojes, algoritmos de elección. Memoria compartida distribuida. Transacciones distribuidas. Sistemas de archivos distribuidos. Conceptos de tolerancia a falla. Seguridad.

BIBLIOGRAFÍA

BIBLIOGRAFÍA BÁSICA

- Andrew S. Tanenbaum. Redes de Computadoras. Cuarta Edición, Prentice Hall, 2003.
- Andrew S. Tanenbaum and Maarten Van Steen. Distributed Systems: Principles and Paradigms (2nd Edition). Prentice Hall, 2006.

BIBLIOGRAFÍA COMPLEMENTARIA

- Seguridad: slides y apuntes complementarias provistas en clase por el profesor (y disponibles en un grupo de Google).

METODOLOGÍA DE TRABAJO

La asignatura se organiza en clases teóricas y prácticas, de cuatro horas reloj cada una.

Las clases teóricas son expositivas, y en las clases prácticas los alumnos resuelven de manera independiente o grupal ejercicios prácticos, bajo la supervisión y acompañamiento de los docentes.

EVALUACIÓN

Requisitos para obtener la regularidad


- Aprobación de los dos parciales, o sus correspondientes recuperatorios.
- Aprobación del laboratorio.

Requisitos para la aprobación:

- Aprobación de un examen final escrito.

Promoción:

- Obtener la regularidad, con nota superior a 8 (ocho) en cada uno de los parciales.



PROGRAMA DE ASIGNATURA

ASIGNATURA: Sistemas Operativos	AÑO: 2012
CARÁCTER: Obligatoria	
CARRERA: Licenciatura en Ciencias de la Computación	
RÉGIMEN: cuatrimestral	CARGA HORARIA: 120 hs.
UBICACIÓN en la CARRERA: Segundo año – Segundo cuatrimestre	

FUNDAMENTACIÓN Y OBJETIVOS

Que el estudiante pueda:

- conocer la estructura y el funcionamiento del Sistema Operativo incluyendo estructuras de datos y algoritmos de los subsistemas.
- Comprender, diseñar y programar algoritmos concurrentes de baja complejidad.
- Evolucionar sobre la práctica de programación.

CONTENIDO

Unidad I: Introducción.

(1) Historia de la Computación. (2) ¿Qué es un sistema operativo? (3) Historia de los sistemas operativos. (4) Variedad de sistemas operativos: de tiempo real, embebidos, distribuidos. (5) Conceptos de sistemas operativos. (6) Llamadas a sistemas. (7) Estructura de los sistemas operativos.

Unidad II: Procesos e Hilos.


(1) Concepto de proceso. (2) Hilos. (3) Comunicación entre procesos. (4) Problemas clásicos de la comunicación entre procesos. (5) Planificación de procesos (Scheduling).

Unidad III: Concurrencia de ejecución. Interbloqueos (*deadlocks*).

(1) Introducción a *deadlocks* y recursos. (2) Detección y recuperación de *deadlocks*. (3) Técnicas para evitar los *deadlocks*. (4) Técnicas para prevenir los *deadlocks*.

Unidad IV: Administración de Memoria.

(1) Administración básica de memoria. (2) Swapping. (3) Memoria Virtual. (4)



Algoritmos para reemplazo de página. (5) Tópicos de diseño para sistemas de paginado. (6) Tópicos de implementación. (7) Segmentado. (8) Memoria compartida distribuida.

Unidad V: Entrada/Salida.

(1) Principios del hardware de entrada/salida. (2) Principios del software de entrada/salida. (3) Organización en capas del software de entrada/salida. (4) Discos. (5) Otros dispositivos.

Unidad VI: Sistema de Archivos:

(1) Archivos. (2) Directorios. (3) Protección. (4) Implementación de los sistemas de archivos. (5) Ejemplos de sistemas de archivos.

Unidad VII: Seguridad.

(1) Amenazas: Confidencialidad, Integridad y Disponibilidad. (2) Ataques desde fuera y dentro del sistema. (3) Mecanismos de protección. (4) Sistemas Confiables. (5) Intrusos. (6) Conceptos básicos de Criptografía. (7) Autenticación de usuarios.

Unidad VIII: Sistemas operativos de usos específicos

(1) Sistemas operativos para sistemas de tiempo real. (2) Planificación para sistemas de tiempo real. (3) Prioridades. (4) Sistemas operativos para sistemas embebidos. (5) Sistemas operativos para dispositivos móviles.

BIBLIOGRAFÍA

- [1] Andrew S. Tanenbaum. Sistemas Operativos Modernos, Tercera Edición. Prentice Hall, 2009.
- [2] Abraham Silberschatz. Operating System Concepts, Sixth Edition. John Wiley & Sons, 2001.
- [3] Raphael Finkel. An operating systems Vade Mecum, Segunda Edición. Prentice Hall, 1988.
- [4] Jonathan Corbet, Alessandro Rubini, and Greg Kroah-Hartman. Linux Device Drivers, Third Edition. O'Reilly, 2005.
- [5] Gary J. Nutt. Kernel Projects for Linux, First Edition. Addison-Wesley Pub Co, 2001.

METODOLOGÍA DE TRABAJO

- Dictado de clases teóricas interactivas.
- Trabajo individual sobre:
 - resolución de guías de estudios



- resolución de problemas teórico – prácticos
- trabajo grupal (hasta 3 estudiantes)
 - realización de tres proyectos de desarrollo en laboratorio más un proyecto de investigación experimental.
 - Seguimiento y asistencia continuos sobre estos proyectos por parte de los docentes.
- Uso de aula virtual para
 - hacer disponible electrónicamente toda la información de la materia incluyendo material didáctico.
 - Uso activo de foros de discusión para información y consulta de dudas a los docentes y entre pares.

EVALUACIÓN

Requisitos para obtener la regularidad

- Aprobación de dos parciales
- Aprobación de los cuatro proyectos de laboratorio.

Requisitos para la aprobación:

- Aprobación de un examen final escrito.

Promoción:

- No se allenta el régimen de promoción.


Dra. NOEMI PATRICIA KISBYE
SECRETARIA ACADEMICA
Fa.M.A.F. UNC


Dra. ESTHER GALINA
VICE DECANA
Fa.M.A.F.