



VISTO

El Informe de Evaluación enviado por los pares evaluadores de la CONEAU, en el marco de la acreditación de la Licenciatura en Ciencias de la Computación;

CONSIDERANDO

Que en dicho informe se presenta una lista de ocho requerimientos a fin de que la carrera cumpla los estándares establecidos por la Resolución Ministerial N° 786/09 para la Licenciatura en Ciencias de la Computación;

Que la Comisión de Autoevaluación conjuntamente con los miembros de la Comisión ad-hoc del plan de estudios de la carrera han elaborado una respuesta a los requerimientos;

Que dentro de los requerimientos se solicita la incorporación de los objetivos, metodología

Que es necesario responder a la vista de la CONEAU informando las acciones a llevar a cabo para subsanar los déficits detectados;

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

R E S U E L V E

ARTÍCULO 1º: Aprobar la respuesta a la vista de acuerdo al Anexo I que forma parte de la presente resolución.

ARTÍCULO 2º: Establecer los objetivos, contenidos, metodología y evaluación de todas las asignaturas de la Licenciatura en Ciencias de la Computación de acuerdo a lo establecido en el Anexo II que forma parte de la presente resolución.

ARTÍCULO 3º: Comuníquese y archívese.

DADA EN LA SALA DE SESIONES DEL HONORABLE CONSEJO DIRECTIVO DE LA FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA, A LOS VEINTINUEVE DIAS DEL MES DE AGOSTO DE DOS MIL ONCE.

pk

  
Dra. NOEMÍ PATRICIA KISBYE  
SECRETARIA ACADÉMICA  
Fa.M.A.F. - U.N.C.

  
Dr. FRANCISCO A. TAMARIT  
DECANO  
Fa.M.A.F.



ANEXO I

RESPUESTA A LAS VISTAS

La Comisión de Autoevaluación ha considerado cada uno de los déficits señalados en el Informe de Evaluación y los correspondientes requerimientos.

En algunos casos se ha elaborado una respuesta aclarando la situación de la Unidad Académica y en otros se ha agregado un plan de mejoramiento a fin de subsanar el déficit señalado.

**REQUERIMIENTO 1**

Formalizar la carga horaria total del plan de estudios vigente indicando la carga horaria asignada a todas las actividades curriculares que se dictan

**REQUERIMIENTO 4**

Incluir en el plan de estudios un pronunciamiento sobre el grado de dominio de idioma inglés exigido a los alumnos para alcanzar la titulación.

**RESPUESTA**

Se anexa el texto ordenado del plan de estudios de la Licenciatura en Ciencias de la Computación, Resolución HCD N° 204/2011 y Resolución HCS N° 687/2011 formalizando la inclusión del Curso de Nivelación, la carga horaria total del plan de estudios de la carrera en su dictado actual y la carga horaria asignada a cada una de las asignaturas.

Asimismo, se ha agregado en el plan de estudios el requerimiento de un nivel de conocimientos de idioma inglés para alcanzar la titulación.



**REQUERIMIENTO 2**

Asegurar el dictado con un tratamiento adecuado de los siguientes Contenidos Curriculares Básicos listados en el Anexo I-1 de la Resolución ME N° 786/09.

- a. Del área de Ciencias Básicas, geometría analítica.
- b. Del área de Teoría de la Computación, fundamentos de inteligencia artificial simbólica y no simbólica.
- c. Del área Arquitectura, Sistemas Operativos y Redes,
  - i. subárea Arquitectura, contenidos de arquitectura y organización de computadoras; error (con respecto a la representación de los datos a nivel máquina); jerarquía de memoria y organización funcional; máquinas algorítmicas, procesadores de alta prestación; arquitecturas no von Neumann; arquitecturas multiprocesadores; conceptos de arquitecturas Grid; conceptos de arquitecturas reconfigurables; arquitecturas basadas en servicios.
  - ii. subárea Sistemas Operativos, sistemas operativos de tiempo real; memoria compartida distribuida; control de concurrencia en sistemas distribuidos, transacciones distribuidas, seguridad en sistemas distribuidos.
- d. En el área de Ingeniería de Software, Bases de Datos y Sistemas de Información, subárea Ingeniería de Software, ingeniería de software de sistemas de tiempo real y diseño centrado en el usuario; subárea Sistemas de Información, nociones de auditoría y peritaje y nociones de sistemas colaborativos.

**RESPUESTA**

- a. Se incluye en el programa de la asignatura Álgebra, correspondiente al núcleo de Ciencias Básicas, contenidos de geometría analítica.
- b. Con respecto a los *fundamentos de inteligencia artificial simbólica y no simbólica*, el plan de la carrera ya incluye los siguientes temas considerados troncales: fundamentos de lógica (inferencia, teorías, completitud, decidibilidad, etc.) tanto en lógicas clásicas como no clásicas, estrategias de resolución de problemas, SAT solving y máquinas abstractas y unificación. Concretamente, temas de lógica clásica se cubren en las materias "Introducción a la Lógica y la Computación" y "Lógica". Lógicas no clásicas se tratan en la Unidad V de "Ingeniería de Software II". Las estrategias de resolución de problemas (algoritmos voraces, divide y vencerás, programación dinámica y backtracking) se abordan en la Unidad III de "Algoritmos y Estructuras de Datos II". En las Unidades VII y IX de "Ingeniería del Software II" se tratan temas de SAT solving. Finalmente máquinas abstractas y unificación es visto en la materia "Paradigmas de Programación", Unidades II y III.



Siguiendo las indicaciones del informe de evaluación, y a fin de cubrir conceptos sobre *fundamentos de inteligencia artificial no simbólica*, se incluye la noción de lógicas de Horn y su aplicación a las inferencias realizadas en los lenguajes de programación en la materia "Ingeniería de Software II", Unidad IX. Asimismo, se incorporan contenidos de algoritmos de búsqueda y algoritmos genéticos a la materia Matemática Discreta II.

Cabe mencionar que se dictan regularmente materias optativas con temas directamente relevantes al área de inteligencia artificial simbólica y no simbólica como: Inteligencia Artificial -con contenidos avanzados en el tema-, Redes Neuronales, Procesamiento de Lenguaje Natural,

c.

i.

Con respecto a los contenidos de *arquitectura y organización de computadoras* se señala que los mismos son actualmente abordados en las asignaturas "Organización del computador" y "Arquitectura de computadoras". Particularmente, en el programa de "Organización del computador", en la Unidad 5, puntos 5.1 y 5.2, se hace referencia a conceptos fundamentales sobre memorias (incluyendo explícitamente en el punto 5.2 memorias cache), lo cual cubre los contenidos curriculares de *jerarquía de memoria y organización funcional*. La unidad VI del programa Matemática Discreta II realiza un tratamiento de códigos de autocorrección, de manera que queda cubierta la temática sobre *error (con respecto a la representación de los datos a nivel máquina)*.

Asimismo la Sección 2 del programa de "Arquitectura de Computadoras" se refiere a arquitecturas no convencionales, lo cual equivale a las *arquitecturas no von Neumann*. A su vez, la Unidad 1 en su totalidad cubre temas sobre *arquitecturas multiprocesadores*, mientras que la Unidad 4 referida al lenguaje VHDL abarca *conceptos de arquitecturas reconfigurables*.

Para dar un mejor tratamiento a los contenidos del subárea Arquitectura, se ha considerado pertinente, como ha sido requerido en el Informe de Evaluación, actualizar la bibliografía utilizada. En la respuesta a los requerimientos 6. y 7. se detalla cómo se ha resuelto el déficit de la bibliografía.

ii.

Con respecto a *sistemas operativos de tiempo real*, se dicta en la asignatura "Sistemas operativos" en lo que corresponde a la Unidad I, punto 3: Variedad de sistemas operativos.

En las Unidades VI y VII de "Redes y sistemas distribuidos" se desarrollan contenidos sobre *seguridad en sistemas distribuidos*. Finalmente, en la Unidad II de Sistemas Operativos y la Unidad II de "Ingeniería del Software II" se hace un



tratamiento de *control de concurrencia en sistemas distribuidos*.

Particularmente en “Redes y Sistemas Distribuidos” se incluyen aspectos de conectividad y seguridad en sistemas distribuidos, y en “Sistemas Operativos” se presentan aspectos arquitectónicos.

A fin de cumplimentar con el requerimiento, se ha agregado en la Unidad IV de “Sistemas Operativos” contenidos sobre *memoria compartida distribuida*, y sobre *transacciones distribuidas* en la Unidad III de Ingeniería del Software II.

d.

Con respecto a los contenidos *ingeniería de software de sistemas de tiempo real* se señala que la asignatura Ingeniería del Software II realiza un tratamiento de estos temas. La materia abarca una gama más amplia de sistemas incluyendo diversos tipos de sistemas reactivos (entre los que se cuentan los sistemas de tiempo real). Se analizan los diversos problemas que surgen en estos sistemas y las propiedades que se espera que estos cumplan (categorizadas en propiedades de alcanzabilidad, safety, liveness y fairness en la Unidad III). Se enseñan diversas técnicas para la especificación, modelado y verificación de estos sistemas, que se expanden a lo largo del programa y que incluyen, entre otros aspectos, model checkers específicos para sistema de tiempo real (como lo son Uppaal, Kronos y Zeus, ver Unidad VI). En la unidad VIII se utiliza la herramienta Alloy para resolver problemas con restricciones (Constraint Solving). La derivación de schedulers en sistemas de tiempo real es un caso particular de esta clase de problema.

Las *nociones de sistemas colaborativos* se dictan en la Unidad II de “Ingeniería del Software I” conjuntamente con los diversos procesos asociados al proceso del software. En particular los sistemas colaborativos son centrales en los temas de Administración del proyecto, la Administración de configuración y la Administración de cambios. No obstante se ha incluido este tópico en la Unidad II para hacerlo explícito en el programa.

A fin de cumplimentar con los requerimientos, se han agregado temas sobre *diseño centrado en el usuario* en la Unidad VII de “Ingeniería del Software I”, y *nociones de auditoría y peritaje* en la Unidad II de “Ingeniería del Software I”.



**REQUERIMIENTO 3**

Revisar y adecuar la bibliografía utilizada para el dictado de los contenidos de arquitectura y organización de computadoras, lenguaje ensamblador, circuitos combinatorios y secuenciales.

**REQUERIMIENTO 6**

Adecuar la bibliografía utilizada para el dictado de los contenidos incluidos en Organización del Computador.

**RESPUESTA**

Se han agregado a la bibliografía de Arquitectura de Computadoras y Organización del computador los siguientes libros:

Patterson, David y Hennesy, John. *Estructura y diseño de computadores. La interfaz hardware/software*. Editorial Reverté. 4ta. Edición. Año 2011. (adquirido con fondos específicos para la Autoevaluación de carreras de informática)  
<http://biblio.famaf.unc.edu.ar>

Patterson, David y Hennesy, John. *Computer organization and design : the hardware / software interface*. Editorial Morgan Kaufmann, c2009.

<http://biblio.famaf.unc.edu.ar/cgi-bin/koha/opac-detail.pl?biblionumber=16147>

William Stallings. *Organización y arquitectura de computadores*. Prentice Hall, 2007.

<http://biblio.famaf.unc.edu.ar/cgi-bin/koha/opac-detail.pl?biblionumber=15413>

Patterson, David y Hennesy, John. *Computer architecture : a quantitative approach*. 4th ed., Morgan Kaufmann, 2007.

<http://biblio.famaf.unc.edu.ar/cgi-bin/koha/opac-detail.pl?biblionumber=14366>



RESOLUCIÓN HCD N° 232/11

**REQUERIMIENTO 5**

Explicitar en los programas de todas las asignaturas objetivos, descripción de actividades teóricas y prácticas, metodologías y modalidad de evaluación.

**RESPUESTA**

Se adjuntan los programas de las asignaturas aprobados por el Consejo Directivo de la Facultad, Anexo II de la presente resolución, incluyendo objetivos, descripción de actividades teóricas y prácticas, metodologías y modalidad de evaluación.

Se adjunta además la reglamentación de régimen de alumnos recientemente aprobada por el Consejo Directivo, Ordenanza HCD N° 4/2011, en la cual se establece

*Artículo 45º: Los programas de las materias serán aprobados por el Consejo Directivo a propuesta del Consejo de Grado y deberán:*

- a) explicitar con detalle los contenidos de la materia subdivididos en unidades temáticas,*
- b) contener fundamentación, objetivos, bibliografía, carga horaria, ubicación en el plan de estudios, metodología de trabajo y evaluación.*

Esta reglamentación garantiza que los programas de las asignaturas expliciten los objetivos, contenidos, metodologías y modalidades de enseñanza y evaluación.

**REQUERIMIENTO 7**

Revisar las exigencias de los sistemas de evaluación en función de asegurar que los alumnos sean evaluados de acuerdo a la amplitud y profundidad de contenidos que figuran en los programas de las asignaturas.

**RESPUESTA**

Particularmente, la nueva reglamentación de alumnos señala las posibles condiciones para otorgar la regularidad y promoción en las distintas asignaturas. Asimismo el Consejo de Grado tiene acceso al resultado estadístico de las encuestas, las cuales consultan a los estudiantes acerca de la pertinencia de los contenidos evaluados.

Por otro lado, la inclusión de las formas de evaluación en los programas analíticos permite realizar un seguimiento de los métodos de evaluación.

El Consejo de Grado ha comenzado a realizar un seguimiento sobre los sistemas de evaluación en las diferentes asignaturas, a través del seguimiento de las encuestas y el control de los programas de las asignaturas.



RESOLUCIÓN HCD N° 232/11

**REQUERIMIENTO 8**

Aclarar la información referida a la cantidad de egresados.

**RESPUESTA**

El cuadro detallado en el Informe de Evaluación refleja la situación del número de egresados a octubre de 2010, según lo reflejado por el Sistema de Gestión de Alumnos Guaraní. En este aspecto se aclara que los datos de egresados son cargados en el sistema una vez que el estudiante a cumplido con todos los requisitos administrativos, como por ejemplo el libre deuda en Biblioteca, y por ello no se habían contabilizado formalmente los egresados en el formulario electrónico. En el Informe de Autoevaluación se ha omitido involuntariamente la aclaración de esta situación.

Los egresados 2010 corresponden a aquellos que rindieron su última materia entre julio de 2010 y marzo de 2011, inclusive. El número de egresados 2010 para la carrera de Licenciatura en Ciencias de la Computación es 13 (trece). Esto particularmente demuestra que el número de egresados ha ido en aumento en los últimos año, un 62.5% del año 2008 al año 2010, y muestra un impacto positivo a las medidas de apoyo académico implementadas en la Facultad.

Se ha corregido este dato en el Formulario Electrónico.

**RECOMENDACIONES**

1. Realizar un monitoreo continuo de los resultados de las medidas de apoyo académico implementadas a fin de mejorar su impacto en la carrera.

2. Revisar la denominación de las asignaturas a fin de que reflejen los contenidos que dictan.

1. En el año 2010 se ha creado la Oficina de Inclusión Educativa, Resolución HCD N° 273/2010, y por Resolución Decanal N° 225 se ha designado un profesional psicólogo responsable de la misma. Junto con la Dirección de Asuntos Estudiantiles se realiza un seguimiento de los estudiantes, talleres de integración con estudiantes, talleres sobre atención a la discapacidad con docentes, y consultorio de atención psicológica personal.

2. Respecto de este punto, la Comisión de Plan de Estudios no ha tomado una decisión unánime hasta el momento.

  
Dra. NOEMÍ PATRICIA KISBYE  
SECRETARIA ACADÉMICA  
Fa.M.A.F. - U.N.C.

  
Dr. FRANCISCO A. TAMARIT  
DECANO  
Fa.M.A.F.



**ANEXO II**  
**PROGRAMA DE ASIGNATURA**

ASIGNATURA: Álgebra

CARÁCTER: Obligatoria

**OBJETIVOS**

Objetivo general:

- Adquirir prestancia en los conocimientos y las técnicas del Álgebra Lineal

Objetivos específicos

- Introducir conceptos fundamentales del álgebra matricial.
- Relacionar la representación y resolución de sistemas de ecuaciones con el álgebra de matrices.
- Introducir nociones de espacios vectoriales y transformaciones lineales.
- Desarrollar conceptos de geometría analítica

**CONTENIDO**

Unidad I: Sistemas de Ecuaciones lineales y Matrices.

Sistemas de Ecuaciones Lineales. Resolución por el método de eliminación de variables. Matrices. Representación matricial de un sistema de ecuaciones lineales. Sistemas Homogéneos y no homogéneos. Operaciones Elementales por Filas (OEFs). Matrices Equivalentes por filas. Reducción por filas. Teorema: *Sistemas correspondientes a matrices (ampliadas) equivalentes por filas tienen las mismas soluciones.* Métodos de Gauss y Gauss-Jordán. Matriz Escalon Reducida por Filas. (MERF).

Conceptos de geometría analítica: ecuaciones de rectas y planos. Relación entre sistemas de ecuaciones e intersecciones de rectas y planos.

Unidad II: Operaciones con Matrices y VITs (Very Important Theorems).

Operaciones con matrices (suma, multiplicación). Matriz Identidad. Propiedades de la multiplicación y suma de matrices. Teorema: *Toda matriz es equivalente por filas a una sola MERF.* Definición: Rango de una matriz=numero de pivotes de la



única MERF a la cual es equivalente. VIT!:El sistema homogéneo  $AX = 0$  tiene solo la solución nula si y solo si el rango de  $A$  es igual al número de columnas de  $A$ . VIT $\exists$ : Los sistemas no homogéneos  $AX = b$  tienen solución para todo  $b$  si y solo si el rango de  $A$  es igual al número de filas de  $A$ . Corolarios: Un sistema homogéneo con más incógnitas que ecuaciones tiene soluciones no nulas. Si  $A$  tiene más filas que columnas existen  $b$ s tales que el sistema  $AX = b$  no tiene solución. Matrices cuadradas. Inversa de una matriz, inversa a izquierda, a derecha. Unicidad. VIT $^{-1}$  : Si  $A$  es  $n \times n$ , las siguientes afirmaciones son equivalentes: 1)  $A$  es inversible 2)  $A$  tiene inversa a izquierda 3)  $A$  tiene inversa a derecha 4)  $AX = 0 \Rightarrow X = 0$  5)  $\forall b \ n \times 1 \exists X \ n \times 1 : AX = b$  6)  $\text{Rango}(A) = n$  7)  $A \sim I$  8)  $\forall b \ n \times 1 \exists !X \ n \times 1 : AX = b$ . Consecuencia de la prueba: Algoritmo para encontrar la inversa de una matriz. Propiedad: Un producto de matrices cuadradas es inversible si y solo si cada una lo es. Matrices Elementales. VIT  $\Pi$ :  $A$  inversible  $\Leftrightarrow A \sim I \Leftrightarrow A$  es producto de elementales.

### Unidad III:Espacios Vectoriales sobre un cuerpo IK

Definición de Espacio Vectorial. Ejemplos. Propiedades básicas. Subespacios: definición y caracterización Transformaciones lineales. Núcleo e Imagen de una transformación lineal como subespacios. Dependencia e independencia lineal. Combinaciones lineales. Intersección de subespacios vectoriales. Conjuntos generadores. Bases. Dos bases cualesquiera tienen la misma cardinalidad (probado para el caso finito). Dimensión. Todo conjunto LI en un EV de dimensión finita puede extenderse a una base. Espacio-fila y rango-fila. Espacio-columna y rango-columna. Matrices equivalentes por filas tienen el mismo espacio fila. Propiedad: Las filas no nulas de una MERF son base de su espacio-fila. Corolario: Rango-fila=Rango. Cálculos relativos a subespacios (caracterización, extracción de una base, etc). Bases Ordenadas. Coordenadas de un vector relativo a una base ordenada.

### Unidad IV:Transformaciones Lineales

Transformaciones no singulares. Isomorfismos. Ejemplos. Teorema:  $T : V \rightarrow W$  lineal,  $\dim V = n \Rightarrow \dim \text{Nu}T + \dim \text{Im}T = n$ . Corolarios. Teorema: rango-columna=rango-fila=rango. Definición de suma de subespacios. Espacio producto. Teorema: $W_i$  supespacios de dimensión finita de  $V$  entonces  $\dim (W_1 + W_2) = \dim W_1 + \dim W_2 - \dim(W_1 \cap W_2)$ . Matriz de una transformación lineal respecto de bases ordenadas. Teorema:  $T : V_1 \rightarrow V_2$  lineal,  $B_i$  base ordenada de  $V_i$ , entonces  $[T \alpha]B_2 = [T ]B_1 \cdot B_2 [\alpha]B_1$  Corolario: La matriz de la composición de transformaciones lineales es el producto de las matrices de las transformaciones. Matriz de Cambio de Base. Matrices semejantes. Ejemplo de aplicación: el modelo predador-presa. El espacio dual. El doble dual. Transpuesta de una transformación lineal.

### Unidad V: Determinantes.



Definición de determinantes para los casos  $n = 1, 2, 3$ . Permutaciones. Signo de una permutación. Definición de determinantes en el caso general. Propiedades elementales. Multilinealidad. Propiedad: el determinante es alternante, i.e., si  $A$  tiene dos filas iguales entonces su determinante es nulo. Comportamiento del determinante bajo OEFs. Teorema: 1)  $A$  es invertible si y solo si  $\det A \neq 0$  2)  $\det(AB) = \det(A)\det(B)$ . Unicidad del determinante. Expansión por filas/columnas. Adjunta de una matriz. Inversa en términos de la adjunta. Regla de Cramer.

Unidad VI: Espacios con producto interno (real y complejo)

Definición. Propiedades básicas de un producto interno. Ejemplos. Desigualdad de Cauchy-Schwartz.

Desigualdad triangular. Norma de un vector. Espacio ortogonal. Conjuntos ortogonales y ortonormales. Procedimiento de Gram-Schmidt. Mejor aproximación a un vector por vectores de un subespacio. Proyección ortogonal. Teorema:  $W$  subespacio de dimensión finita de  $V$  entonces  $V = W \oplus W^\perp$ . Producto cruz de dos vectores en  $\mathbb{R}^3$ . Propiedades.

Unidad VII: Autovalores y autovectores.

Definición. Polinomio característico. Ejemplos y cálculos. Independencia de autovectores asociados a autovalores distintos. Matrices y transformaciones lineales diagonalizables. Teorema: Si  $V$  un espacio vectorial de dimensión finita  $n$ ,  $T : V \rightarrow V$  es una transformación lineal,  $c_1, c_2, \dots, c_r$  los autovalores distintos de  $T$ ,  $d_i = \dim \text{Nu}(T - c_i I)$ ,  $p(x) =$  polinomio característico de  $T$ . Entonces las siguientes afirmaciones son equivalentes: 1)  $T$  es diagonalizable. 2)  $p(x) = (x - c_1)^{d_1} (x - c_2)^{d_2} \dots (x - c_r)^{d_r}$  3)  $n = d_1 + \dots + d_r$ . Teorema de Cayley-Hamilton (sin prueba).

#### MODALIDAD DE ENSEÑANZA

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.



## BIBLIOGRAFÍA

### BIBLIOGRAFÍA BÁSICA

- ALGEBRA LINEAL. Hoffman and Kunze.

### BIBLIOGRAFÍA COMPLEMENTARIA

- ALGEBRA LINEAL. Stanley Grossman McGraw-Hill. 1997.
- APPLIED LINEAR ALGEBRA. Ben Noble-James Daniel, Prentice Hall. 1988

## EVALUACIÓN

### Requisitos para obtener la regularidad

- Aprobación de dos de los tres parciales.

### Requisitos para la aprobación:

- La evaluación consta de dos partes: teórica y práctica. Cada una de estas partes aporta hasta un 50% de la nota final.

### Promoción:

- No se prevé régimen de promoción.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Algoritmos y estructura de datos I

CARÁCTER: Obligatoria

#### 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 de programas: especificaciones, ejemplos.

Unidad III: Técnicas elementales para la programación: 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 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.

#### MODALIDAD DE ENSEÑANZA

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.

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

#### 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.



## 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Algoritmos y estructura de datos II

CARÁCTER: Obligatoria

#### 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.

###### *Notación complementaria*

La notaciones  $\Omega$  y  $\Theta$ . Definición formal. Equivalencia entre expresiones con  $\Omega$ ,  $\Theta$  y O. Distinción entre  $\Omega$ ,  $\Theta$  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  $i$ -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. 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

*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

MODALIDAD DE ENSEÑANZA

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.

## 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
- Kaldewaij. Programming: the Derivation of Algorithms.

## 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

- 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Análisis Matemático I

CARÁCTER: Obligatoria

#### OBJETIVOS

Que el estudiante:

1. maneje con solvencia las propiedades de los números reales
2. reconozca las propiedades fundamentales de las funciones elementales
3. adquiera nociones básicas del cálculo diferencial en una variable

#### CONTENIDO

Unidad I:

Números reales, propiedades. Valor absoluto. Definición de función. Ejemplos. Funciones trigonométricas. Límite de una función. Límites laterales. Límites notables.

Unidad II:

Funciones continuas, propiedades locales. Los teoremas fuertes: aplicaciones.

Unidad III:

La derivada de una función. Reglas de derivación, la regla de la cadena. Recta tangente. Función monótona creciente y decreciente. El teorema del valor medio. Derivadas de orden superior. Función cóncava y convexa. Estudios de máximos y mínimos. Gráficos de funciones.

Unidad IV:

Función inversa. El logaritmo y la exponencial. Propiedades. Aplicaciones de la Unidad III, a funciones logarítmicas y exponenciales. La regla de L'Hopital.

Unidad V:

La integral indefinida de una función continua. Área. Suma de Riemann. Teorema fundamental del cálculo. Propiedades básicas de la integral.

Unidad VI:

Técnicas de integración: método de sustitución, integración por partes. Aplicaciones al cálculo de áreas y volúmenes.



### MODALIDAD DE ENSEÑANZA

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.

### BIBLIOGRAFÍA

#### BIBLIOGRAFÍA BÁSICA

- “Cálculo”, de Serge Lang

#### BIBLIOGRAFÍA COMPLEMENTARIA

- “Calculus”, de M. Spivak.

### EVALUACIÓN

#### Requisitos para obtener la regularidad

- Aprobación de dos de los tres parciales.

#### Requisitos para la aprobación:

- La evaluación consta de dos partes: teórica y práctica. Cada una de estas partes aporta hasta un 50% de la nota final.

#### Promoción:

- No se prevé régimen de promoción.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Análisis Matemático II

CARÁCTER: Obligatoria

#### 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.

Primitiva o integral definida de una función. Dos primitivas de una función en un intervalo difieren en una constante. Tabla de primitivas. La integral definida.

Sumas superiores e inferiores de Riemann para una función acotada 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. Integración de funciones racionales por el método de fracciones simples. Integración de funciones racionales de funciones trigonométricas.

Integrales impropias. Criterio de comparación para la convergencia de integrales impropias.

Aplicaciones. Cálculo de área entre curvas, y cálculo de volúmenes dado por un sólido de revolución.

Unidad II: Polinomio de Taylor

Aproximación mediante funciones polinómicas. Definición del Polinomio de Taylor de grado  $n$  de una función  $f$  de una variable. Criterio para encontrar máximos, mínimos y puntos de inflexión usando el polinomio de Taylor. Fórmula del resto. Unicidad del polinomio de Taylor. Teorema de Taylor.



### Unidad III: Series Numéricas

Series numéricas. Definición de convergencia y convergencia absoluta. La serie geométrica. Criterios de convergencia: Criterio de comparación. Criterio del cociente, y de la raíz. Series alternantes. Criterio de Liebnitz. Criterio de la integral para series de términos positivos.

Series de potencias. Radio e intervalos de convergencias. Series de potencias de una función  $f$ .

### Unidad IV: Funciones de varias variables.

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  $R^n$ . Definición de Plano en  $R^3$ . Ecuaciones vectorial, implícita y normal del plano. Concepto de planos paralelos y ángulo entre planos y rectas.

Funciones de varias variables a valores vectoriales. Definición de límite y continuidad. Propiedades. Derivadas parciales de una función en un punto. La diferencial de una función en un punto. Relación de la diferencial con la matriz jacobiana. Continuidad de una función diferenciable. Continuidad de las derivadas parciales implica existencia de la diferencial. 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. Igualdad de las derivadas cruzadas. La matriz Hessiana. Criterios para determinar máximos, mínimos (locales) y puntos de silla, de funciones de varias variables a valores reales.

### Unidad V: Integrales en varias variables

Integrales múltiples en un dominio regular. Definición y propiedades. Integribilidad de funciones continuas. Cálculo de integrales múltiples por integración iterada.

Coordenadas polares (cambio de variables).



### MODALIDAD DE ENSEÑANZA

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.

### BIBLIOGRAFÍA

#### BIBLIOGRAFÍA BÁSICA

- James Steward, Calculus.
- James Steward, Cálculo multivariable.

#### BIBLIOGRAFÍA COMPLEMENTARIA

- Michael Spivak, Calculus
- Richard Williamson, Richard Crowell, Hale Trotter, Cálculo de funciones vectoriales.

### EVALUACIÓN

#### Requisitos para obtener la regularidad

- Aprobación de dos de los tres parciales
- Asistencia a al menos 20 clases prácticas

#### Requisitos para la aprobación:

- La evaluación consta de dos partes: teórica y práctica. Cada una de estas partes aporta hasta un 50% de la nota final.

#### Promoción:

- No se prevé régimen de promoción.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Análisis Numérico

CARÁCTER: Obligatoria

#### OBJETIVOS

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

#### CONTENIDO

- Interpolación numérica

Interpolación polinomial. Diferencias divididas. Interpolación de Hermite, spline.

- Integración numérica

Integración numérica basada en interpolación, reglas gaussianas.

- Solución numérica de Ecuaciones diferenciales

Método de Euler, Euler mejorado, método de Runge Kutta.

- Solución de ecuaciones no lineales

Bisección, Newton Secante, métodos de punto fijo

- Aproximación de funciones

Teoría de cuadrados mínimos

- Álgebra lineal numérica

Métodos iterativos de Jacobi y Gauss Sidel.

#### DESCRIPCIÓN DE ACTIVIDADES TEÓRICAS Y PRÁCTICAS

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 máquina una lista de problemas seleccionados.

#### BIBLIOGRAFÍA

- Análisis Numérico, Kincaid – Cheney Addison – Wesley (1994)

#### METODOLOGÍA DE ENSEÑANZA

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.

#### FORMAS DE EVALUACIÓN

##### Requisitos para obtener la regularidad

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

##### Requisitos para la aprobación:

- Examen escrito y prueba de laboratorio.

##### Promoción:

- No se prevé régimen de promoción.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Arquitectura de computadoras

CARÁCTER: Obligatoria

#### OBJETIVOS

- Que el alumno sea capaz de interpretar el funcionamiento de los bloques "internos" asociados a Sistemas de Computación.
- Que el alumno entienda las diferentes opciones de las implementaciones de los principales bloques constitutivos de un sistema de computación moderno.
- Que el alumno identifique los balances y compromisos involucrados en estas opciones.

#### CONTENIDO

##### -SECCIÓN 1: ARQUITECTURAS CONVENCIONALES

-Unidad 1: Unidad Central de Procesos - CPU ("Central Processor Unit")

1.1.-Arquitecturas Von Neumann y CISC ("Complex Instruction Set Computer").

1.2.-Microprocesadores 8086/88 de Intel. Principios de funcionamiento. Líneas de datos, direccionamiento y control.

1.3.-Sus registros internos.

1.4-Modos de direccionamientos.

1.5.-Instrucciones en lenguajes nemónico ("Assembly") y de máquina ("Hexadecimal").

1.6.-Interrupciones. El controlador 8259 - PIC - ("Programmable Interrupt Controller"). Nociones sobre acceso directo a memoria - DMA - ("Direct Memory Access"). Controlador 8237. Temporizador programable 8253.

1.7.-Coprocesador matemático 8087 - NPX - ("Numerical Processor



Extension").

1.8.- Introducción a las arquitecturas Harvard y RISC ("Reduced Instruction Set Computer"). Conceptos sobre "PIPELINE".

-Unidad 2: Memorias Semiconductoras

2.1.-Memorias de lectura solamente: ROM - ("Read Only Memory"), PROM ("Programmable ROM"), EPROM ("Erasable PROM"), EEPROM ("Electrically EPROM"), "Flash", PLA ("Programmable Logic Array"), PAL ("Programmable Array Logic") y PLDs ("Programmable Logic Devices").

2.2.-Memorias de lectura y escritura: SRAM ("Static Random Access Memory"), DRAM ("Dynamic RAM") y "Cache".

2.3.- Principios de funcionamiento, configuración interna y estructuración de bancos ("Memory Mapped").

-Unidad 3: Interfases Programables de Entrada y Salida.

3.1.-Estructuración de dispositivos de entrada y salida ("I/O Mapped").

3.2.-Norma de comunicación "Centronics".

3.3.- Normas de comunicaciones RS-232C y por lazo de corriente.

3.4.-Interfaz programable serie 8250 - UART - ("Universal Asynchronous Receiver/Transmitter").

## -SECCIÓN 2: ARQUITECTURAS NO CONVENCIONALES

-Unidad 4: Lenguaje de Descripción de "Hardware" (VHDL)

4.1.-Conceptos generales, historia y utilidad.

4.2.-Abstracción, modularidad y jerarquía. Niveles de abstracción.

4.3.-Elementos estructurales. Paquetes y librerías. Identificadores, palabras reservadas y símbolos utilizados. Constantes, variables y señales.



4.4.-Tipos de datos. Conversión de tipos. Subtipos.

4.5.-Sentencias secuenciales y sentencias concurrentes.

4.6.-Conceptos generales sobre síntesis.

4.7.-Estudio de casos: multiplexores, "flip-flops", "buffer tri-state", unidad aritmética lógica y contadores.

4.8.-Introducción a las plataformas de prototipado rápido: dispositivos lógicos programables complejos (CPLD) y arreglos lógicos programables en campo (FPGA).

-Unidad 5: Computadoras de Segmentación Encausada

5.1.-Conceptos sobre procesamiento paralelo.

5.2.-Segmentación encausada y procesamiento vectorial.

5.3.-Aplicaciones varias.

-Unidad 6: Procesadores Matriciales

6.1.-Organización de las computadoras matriciales.

6.2.-Redes de interconexión.

6.3.-Algoritmos para procesadores matriciales.

-Unidad 7: Sistemas Multiprocesadores

7.1.-Arquitectura de los multiprocesadores.

7.2.-Su programación.

7.3.-Ejemplos varios.



#### MODALIDAD DE ENSEÑANZA

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.

#### BIBLIOGRAFÍA

##### BIBLIOGRAFÍA BÁSICA

- Patterson, David y Hennesy, John. *Estructura y diseño de computadores. La interfaz hardware/software*. Editorial Reverté. 4ta. Edición. Año 2011.
- Computer architecture : a quantitative approach. John L. Hennessy, David A. Patterson. 4th ed., 2006.
- The 80X86 Family- Design, Programming and Interfacing. Second Edition. John Uffenbeck. Prentice Hall (1998).
- Terés, Y. Torroja, S. Olcoz, E. Villar. VHDL Lenguaje estándar de diseño electrónico. Mc Graw Hill, Madrid, 1998.
- Arquitectura de Computadoras y Procesamiento Paralelo. Kai Hwang y Faye A. Briggs. Mc Graw-Hill (1988).

##### BIBLIOGRAFÍA COMPLEMENTARIA

- iAPX 86/88, 186/188 User's Manual. Programmer's Reference. Intel - Reward Books (1985).
- Microsystem Components Handbook. Microprocessors and Peripherals (Volume I) Intel (1985).
- Memory Components Handbook. Intel (1983).
- Parallel Port Complete. Programming, Interfacing, & Using de PC's Parallel Printer Port. Jan Axelson. Lakeview Research (1997).



- Serial Port Complete – Programming and Circuits for RS-232 and RS-485 Links and Networks. Jan Axelson. Lakeview Research (1998).
- RS-232 made easy: Connecting computers, printers, terminals and modems. Martin D. Seyer. Prentice Hall, Inc. (1984).
- Microprocessors and Peripheral Circuits. Mitsubishi Electric (1985).
- The Indispensable Pentium Book. Hans - Peter Messmer. Addison Wesley (1995).
- Los Microprocesadores Intel 8086/88, 80186, 80286/386/486. Arquitectura, Programación e Interfaces. Tercera Edición. Prentice Hall.
- U. Heinkel y varios. The VHDL Reference. A Practical Guide to Computer-Aided Integrated Circuit Design Including VHDL-AMS. John Wiley & Sons, Ltd, West Sussex, England, 2000.
- 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/Coole. 2004.

## EVALUACIÓN

### Condiciones para la regularidad:

Aprobar ambos parciales, o un parcial y un recuperatorio.

Entregar y aprobar los informes correspondientes a los trabajos prácticos.

### Condiciones para la aprobación:

Aprobación de un examen final escrito.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Bases de Datos

CARÁCTER: Obligatoria

#### 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.

Abstracciones de datos. Ejemplares y Esquemas. Modelos de los datos. Lenguajes de bases de datos.

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

Unidad II: Modelo de Entidad-Relación.

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

Atributos. Cardinalidad 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.

#### MODALIDAD DE ENSEÑANZA

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.

#### 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).



## 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.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Curso de Nivelación

CARÁCTER: Obligatoria

#### 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.

#### CONTENIDOS

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.
- 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.

### MODALIDAD DE ENSEÑANZA

- 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.

### BIBLIOGRAFÍA

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



## EVALUACIÓN

### Requisitos para obtener la regularidad

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

### Requisitos para la aprobación:

- Aprobación de un examen final escrito.

### Promoción:

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



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Física

CARÁCTER: Obligatoria

#### 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: Electrostatica.

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

MODALIDAD DE ENSEÑANZA

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.

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.



### 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Ingeniería del Software I

CARÁCTER: Obligatoria

#### 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.
- El desafío de la Ingeniería del Software.
- El enfoque de la Ingeniería del Software.

##### 2 - El proceso del software:

- Procesos. Modelo de procesos. Componentes. Enfoque ETVX.
- Características deseadas del proceso del software: Predecible y repetible, Tolerante a cambios, Testeable y Mantenable.
- Proceso de desarrollo del software, etapas fundamentales.
- 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.
- Nociones de auditoría y peritaje.

##### 3 - Análisis y especificación de los requerimientos del software:

- 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.
  - 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.
- 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

#### 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.

MODALIDAD DE ENSEÑANZA

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.

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/](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.

## 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Ingeniería del Software II

CARÁCTER: Obligatoria

#### 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.

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.

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  $\omega$ -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 correccion 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.



#### MODALIDAD DE ENSEÑANZA

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.

#### BIBLIOGRAFÍA

- [1] J. Magee y J. Kramer. Concurrency: State Models & Java Programs, 2<sup>nd</sup> 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.

## 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Introducción a la Lógica y la Computación

CARÁCTER: Obligatoria

#### OBJETIVOS

Lograr que el alumno sea capaz de comprender aspectos de las estructuras algebraicas, lógicas y computacionales que proveen el marco teórico desde el cual fundamentar las prácticas profesionales en el campo de las aplicaciones, los desarrollos y la investigación.

#### CONTENIDO

Unidad I: Relaciones de equivalencia y orden.  
Noción de Relación. Relaciones de Equivalencia y Particiones. Relaciones de orden. Órdenes 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. Complementos. Reticulados distributivos y su caracterización. Álgebras de Boole y sus propiedades. Representación de reticulados distributivos finitos. Representación de las álgebras de Boole finitas. 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-Pravits. 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 movimientos "ε". Expresiones 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 gramática regular desde una expresión regular.

#### MODALIDAD DE ENSEÑANZA

El desarrollo de los contenidos se basa en la modalidad de clase magistral, 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.



## 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, Berlin

## EVALUACIÓN

Se toman tres evaluaciones parciales, uno para cada módulo principal de la materia: temas de álgebra, de lógica y de lenguajes formales.  
Los exámenes evalúan contenidos teóricos y prácticos.

### Requisitos para obtener la regularidad

- 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Introducción a los algoritmos

CARÁCTER: Obligatoria

#### 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. Ejemplos. Solución de problemas de ingenio.

Unidad II: Cálculo proposicional.

Operadores booleanos: equivalencia, disyunción, conjunción, implicación, negación, discrepancia. Tablas de verdad. Leyes para manipularlos. Propiedades. Ejemplos de aplicación.

Unidad III: Aplicaciones del cálculo 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 de listas. Operaciones sobre listas: concatenar, calcular longitud, tomar  $n$ , tirar  $n$ , indexar. Propiedades de las operaciones sobre las listas. Ejemplos.



Unidad V: Cálculo de predicados.

Noción de predicado. Variables libres y ligadas. Cuantificador universal. Cuantificador existencial. Interpretación y leyes.

Unidad VI: Especificaciones.

Concepto de especificación de un problema. Importancia de una buena especificación para la derivación de programas correctos. Ejemplos.

#### MODALIDAD DE ENSEÑANZA

La materia se articula en dos sesiones semanales de 4 (cuatro) horas, en las que se llevan a cabo el dictado de 2 (dos) horas de clase teórica con todo el grupo, seguidas de dos horas de resolución práctica de problemas, con el grupo dividido en comisiones.

Los problemas a resolver son ejemplos de los conceptos presentados en las clases teóricas.

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.

#### 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.



## EVALUACIÓN

### Condiciones para la regularidad

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

### Condiciones para la aprobación

- Aprobación de un examen final escrito.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Lenguajes Formales y Computabilidad

CARÁCTER: Obligatoria

#### OBJETIVOS

Que los alumnos sean capaces de comprender

- los aspectos básicos de las gramáticas libres de contexto y de los autómatas a pila,
- el concepto de computabilidad efectiva
- los paradigmas de computabilidad: funciones recursivas, lenguaje imperativo.

#### CONTENIDO

##### Capítulo 1

Autómatas finitos. Equivalencia del caso determinístico con el no determinístico con  $\varepsilon$ -movies. El pumping lema. Expresiones regulares. Teorema de Kleene. Gramáticas lineales. Equivalencia entre lenguajes regulares y generados por gramáticas lineales.

##### Capítulo 2

Gramáticas libre de contexto. Árboles de derivación y derivaciones lefmost. Forma normal de Chomsky. Autómatas a pila. Equivalencia de lenguajes aceptados por vaciamiento de pila por alcance de estado final. Equivalencia entre lenguajes libres de contexto y los lenguajes aceptados por la autómatas de pila.

##### Capítulo 3

Funciones  $\Sigma$ -mixtas. Identificación entre  $\Sigma^*$  y  $\omega$  para un orden total sobre  $\Sigma$ . Codificación de elementos de  $\omega^{|\mathbb{N}|}$  con números naturales. Codificación de elementos de  $\Sigma^{*\mathbb{N}}$  con palabras. Funciones  $\Sigma$ -recursivas y  $\Sigma$ -recursivas primitivas. Conjuntos  $\Sigma$ -recursivos y  $\Sigma$ -recursivos primitivos. Lema de división por casos. Iteración de funciones  $\Sigma$ -recursivas primitivas. Cuantificación acotada de predicados  $\Sigma$ -recursivos primitivos. Minimización acotada de predicados  $\Sigma$ -recursivos primitivos. Recursión de parámetro numérico sobre valores anteriores. Lema de independencia del alfabeto.



#### Capítulo 4

El lenguaje  $\Sigma^*$ . Sintaxis y semántica. Macros. Equivalencia entre funciones  $\Sigma$ -computables y  $\Sigma$ -recursivas. Forma normal de Kleene. Programas universales. El halting problem. Caracterización de los conjuntos  $\Sigma$ -recursivamente enumerables.

#### Capítulo 5

Maquinas de Turing. Equivalencia entre funciones  $\Sigma$ -Turing computables y  $\Sigma$ -recursivas y entre lenguajes  $\Sigma$ -recursivamente enumerables y lenguajes aceptados por maquinas de Turing.

#### MODALIDAD DE ENSEÑANZA

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.

#### BIBLIOGRAFÍA

- 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.

#### EVALUACIÓN

##### Requisitos para obtener la regularidad

- Aprobación de dos de los tres parciales.

##### Requisitos para la aprobación:



Universidad Nacional de Córdoba  
FACULTAD DE MATEMÁTICA, ASTRONOMÍA Y FÍSICA

---

- Evaluación escrita de contenidos teóricos y prácticos de la materia.

Promoción: (sólo de la parte práctica)

- Aprobación de los tres parciales, con promedio mayor o igual a 7 (siete).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Lenguajes y Compiladores

CARÁCTER: Obligatoria

#### OBJETIVOS

Lograr que el alumno sea capaz de comprender aspectos centrales del diseño de los lenguajes de programación a través del uso de herramientas teóricas como la semántica denotacional, la teoría de dominios, las nociones de ejecución y evaluación.

#### CONTENIDO

Unidad I: Sintaxis y Semántica Elemental

Lógica de Predicados

Sintaxis abstracta, frase abstracta, gramática abstracta. Sintaxis abstracta de la lógica de predicados.

Frases enteras y booleanas. Composición de la gramática abstracta, producciones, no terminales predefinidos, patrones diferentes. Diferencia entre gramática libre de contexto (BNF) y gramática abstracta. Representación simple (parentizada) de una sintaxis abstracta. Abstracción: carriers (contenedores), constructores y condiciones de abstracción. Construcción de los carriers como uniones infinitas. Ejemplos: notación parentizada.

Notación prefija. Notación por medio de árboles sintácticos. Notación en programación funcional. Semántica denotacional de la Lógica de Predicados. Dominios de interpretación para frases enteras y booleanas. Enteros, booleanos y estados. Modificación de una función en un punto. Funciones de interpretación. Metalenguaje. Diferencia entre variable y metavariante. Metavariante que corren sobre variables. Obtención del patrón a partir de la producción de la gramática abstracta. Ecuaciones Semánticas. Ecuaciones Dirigidas por Sintaxis. Existencia de una única solución al sistema de ecuaciones dirigidas por sintaxis. Semántica Composicional. Metacircularidad. Peligros que presenta.

Validez de una frase booleana en un estado. Validez de una frase booleana. Validez de  $p$  implica  $q$ . Reglas de inferencia (validez de  $p_0, \dots, p_{n-1}$  implica validez de  $q$ ). Axioma, esquema de axioma, regla de inferencia con una o más premisas. Corrección de la generalización como regla de inferencia, pero no como esquema



de axioma. Instancia de un regla. Prueba formal. Como secuencia y como árbol. Ligadores. Ocurrencias ligadoras. Alcance. Ocurrencias ligadas y libres. Frase cerrada. Variables libre de una frase. Teorema de Coincidencia para la lógica de predicados. Inducción estructural. Sustitución. Problema de la captura. Sustitución simultánea. Definición. Condición de variable fresca. Sustitución identidad. Sustitución finita.

Teorema de Coincidencia para sustituciones. Neutralidad de la sustitución identidad. Variables libres de una expresión a la que se aplicó una sustitución. Teorema de la Sustitución. Teorema de la Sustitución Finita. Teorema del Renombre.

Lenguaje Imperativo Simple (sin ciclos ni variables locales)

Sintaxis. Frases de tipo comando. Semántica Denotacional. Comandos como modificadores del estado. Ejemplos: asociatividad de la secuencia y semántica de una secuencia de asignaciones. Variables libres y variables asignables.. Éstas como subconjunto de aquellas. Teorema de Coincidencia. Sustituciones. Necesidad de restringirlas a sustituir variables por variables. Teorema de la Sustitución, necesidad de considerar sólo sustituciones inyectivas. Generalización. Demostración.

Semántica operacional o de transiciones. Configuraciones. Configuración terminal y configuración no terminal. Relación de transición. Ejecución. Relación determinística y relación que no se traba. Función de transición. Ejecución maximal. Semántica de transiciones del lenguaje imperativo simple (sin ciclos ni variables locales).  $n$  pasos de transición. Clausura reflexiva y transitiva de la relación de transición. Correctitud de la semántica de transiciones.

Variables locales

Sintaxis. Semántica denotacional. Variables libres. Variables asignables. Teorema de Coincidencia. Semántica de transiciones. Correctitud de la semántica de transiciones.

Teorema de Sustitución. Teorema del Renombre.

Ciclos

Sintaxis del while. Educación intuitiva. ¿Es dirigida por sintaxis? Existencia de infinitas soluciones. Ejemplo. Semántica de Transiciones. Ejecuciones infinitas. Notación para la no terminación ( $\perp$ ).

Unidad II: Dominios, Puntos Fijos y Lenguajes Imperativos

Dominios

Orden parcial, cadena, supremo. Cadenas interesantes y no interesantes. Predominio. Elemento mínimo. Dominio. Ejemplos. Orden discreto. Lifting. Dominio plano. Dominio vertical. Funciones monótonas. Funciones continuas. Funciones estrictas. Continuidad implica monotonicidad. Condición suficiente para que una función monótona sea continua. Observaciones elementales sobre continuidad en predomios sencillos. Funciones elementales que son continuas (constantes, identidad, composición de continuas, composición a izquierda y a derecha).

Teorema de Punto Fijo

Predominio de las funciones entre predomios. Orden parcial entre funciones, supremo de una cadena de funciones. Predominio de las funciones continuas entre



predominios.

Extensión estricta de una función entre predominios. Extensión estricta de una función de un predominio a un dominio. Unicidad de estas extensiones. Propiedades de la extensión de una composición. Continuidad de las extensiones y de los operadores de extensión. Teorema del Menor Punto Fijo.

Lenguaje Imperativo Simple (con ciclos)

Lifting del estado. Semántica del lenguaje imperativo simple con no terminación. Semántica de la secuencia, de la variable local y del ciclo while. Demostración de la continuidad del funcional correspondiente.

Productos y Sumas de (pre-)Dominios

Producto de predominios es predominio. Orden parcial, supremo. Producto de dominios es dominio.

Elemento mínimo. Continuidad de las funciones elementales sobre productos. Suma o unión disjunta de predominios es predominio. Orden parcial, supremo. Suma de dominios no es dominio. Continuidad de las funciones elementales sobre sumas.

Cálculo del menor punto fijo

Ejemplo del cálculo de una solución a una ecuación. Menor solución intuitiva. Otras soluciones intuitivas. Cálculo de aproximaciones. Propuesta de aproximación  $k$ -ésima. Demostración de la correctitud de la aproximación  $k$ -ésima. Propuesta de supremo. Demostración del supremo.

Cálculo de la semántica de un programa con while.

Teoremas

Variables libres y asignables del while. Sustitución. Teoremas de Coincidencia, de Sustitución y de Renombre. Equivalencia entre la semántica operacional y denotacional.

Dominios Recursivos y Extensiones al Lenguaje Imperativo Simple

Programas con fallas. Extensión del conjunto de resultados para incluir fallas. Semántica de la falla. Extensiones  $f_*$  y  $f_+$ . Semántica del catch. Output. Dominio Recursivo. Isomorfismo. Ejemplo de solución. Notación abstracta,  $\Omega$  y  $I_{term}$ ,  $I_{abort}$ ,  $I_{out}$  y  $\perp_{\Omega}$ . Representación gráfica y definición.

Propiedades de  $I_{term}$ ,  $I_{abort}$ ,  $I_{out}$  y  $\perp_{\Omega}$ . "Inyectividad" y "suryectividad". Programas con output.

Semántica denotacional del lenguaje. Extensiones  $f_*$  y  $f_+$ . Semántica del output. Input. Nuevo  $\Omega$ ,  $I_{in}$ .

Programas con input. Semántica denotacional del lenguaje. Extensiones  $f_*$  y  $f_+$ . Semántica del input.

Semántica Denotacional Inversa o de Continuaciones

Problema de la propagación del comportamiento. Alternativa: utilización de un argumento extra (continuación) que se aplicará al estado final. Semántica de continuaciones del lenguaje imperativo con ciclos, input y output. Correspondencia entre la semántica denotacional inversa y la directa.

Semántica de continuaciones del lenguaje imperativo con fallas. Semántica de la falla y del catch. Correspondencia entre la semántica denotacional inversa y la directa.



### Semántica de Transiciones de las Extensiones

Semántica de transiciones del lenguaje con falla. Extensión del conjunto de configuraciones terminales. Extensión de la relación de transición. Equivalencia con la semántica denotacional directa. Semántica de transiciones. Semántica de la falla y del catch. Semántica de transiciones del lenguaje con input y output. Conjunto de rótulos. Transiciones silenciosas. Reciclado de las reglas sin

rótulos como reglas silenciosas. Semántica del input y del output. Pérdida de la condición de ser función. Caracterización de las transiciones posibles en toda configuración no terminal. Ejecución silenciosa maximal. Caracterización de las ejecuciones silenciosas maximales posibles en toda configuración. Equivalencia con la semántica denotacional directa.

Demostración detallada de la correspondencia entre la semántica denotacional de continuaciones y la directa par el lenguaje imperativo simple con white, input y output.

Demostración detallada de la buena definición y continuidad de la semántica de continuaciones. Semántica de continuaciones del comando break con una continuación generalizada.

Semántica de continuaciones del lenguaje con break y fallas con 3 continuaciones.

### Unidad III: Semántica de Lenguajes Aplicativos

#### Cálculo Lambda Puro

Sintaxis abstracta, términos lambda, variable, aplicación (operador y operando), abstracción o expresión lambda. Precedencia de la aplicación sobre la abstracción, asociatividad a izquierda de la aplicación. Ocurrencia ligadora, alcance, variables libres, sustitución. Propiedades de la sustitución. Renombre o cambio de variable,  $\alpha$ -conversión,  $\alpha$ -equivalencia. Semántica de transiciones o de reducciones. Contracción, reducción, secuencia de reducciones, redex, "contrae a". Reglas de contracción ( $\beta$ -reducción, renombre, clausura contextual). Reglas de reducción (clausura reflexiva (salvo renombre) y transitiva). Configuración terminal, forma normal. No determinismo de la reducción. Confluencia o Church-Rosser. Unicidad de la forma normal (si existe). Ejemplos de ejecuciones infinitas. Operador de punto fijo. Redex externo (outermost). Reducción en orden normal. Redex externo izquierdo (leftmost outermost). Teorema de Estandarización.

Evaluación. Valor o forma canónica. Expresiones cerradas. Formas canónicas del cálculo lambda puro: abstracciones. Aplicación cerrada no es forma normal. Forma normal cerrada es forma canónica. Forma canónica (cerrada o no) no necesariamente es formal normal. Evaluación normal. Notación. Semántica natural o big-step. Reglas. Método de evaluación. Manera de desplegar. Ejemplo. Evaluación Eager (ansiosa, impaciente, entusiasta). Ejemplo motivador. Reglas. Método de evaluación. Manera de desplegar. Comparación de las evaluaciones normal e eager. Comentarios sobre call by name, call by value y evaluación lazy. Semántica denotacional del cálculo lambda puro. Dominio  $D_{\infty}$ . Isomorfismo. Ambientes. Ecuaciones semánticas. Teoremas de Coincidencia, sustitución, renombre. Correctitud de la  $\beta$ -contracción.

Semántica denotacional normal del cálculo lambda puro. Diferencia entre  $\Delta\Delta$  y



#### $\lambda x.\Delta\Delta$ . Denotaciones

de formas canónicas y denotaciones de resultados. Dominios. Ecuaciones semánticas. Validez de los teoremas. Semántica denotacional eager del cálculo lambda puro. Dominios. Ecuaciones semánticas.

#### Lenguajes Aplicativos

Lenguaje aplicativo eager. Sintaxis. Formas canónicas. Reglas de evaluación. División por cero y otros errores. Semántica denotacional directa del lenguaje aplicativo eager. Dominios. Funciones auxiliares. Ecuaciones semánticas.

Definiciones locales y recursión. Reglas de evaluación. Ecuaciones semánticas. Semántica denotacional de continuaciones del aplicativo eager. Dominios. Ecuaciones semánticas. Semántica del letrec.

Continuaciones como valores (callcc y throw). Ejemplo ilustrativo. Ecuaciones semánticas.

Lenguaje aplicativo normal. Sintaxis. Formas canónicas. Reglas de Evaluación. Semántica denotacional directa del lenguaje aplicativo normal. Dominios. Ecuaciones semánticas.

Recursión en el lenguaje aplicativo normal. El operador rec. Ejemplos, evaluación y ecuación semántica. Semántica de continuaciones del lenguaje aplicativo normal.

#### Lenguaje ISWIN

Lenguaje aplicativo eager con referencias y asignaciones. Definición de secuencia, variable local, while. Semántica operacional y denotacional.

#### Trabajos prácticos especiales

Desarrollo de un evaluador para un lenguaje elemental y otro para el lenguaje imperativo simple con fallas y entrada y salida, semántica de continuaciones.

#### MODALIDAD DE ENSEÑANZA

El desarrollo de los contenidos se basa en la modalidad de clase magistral, 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.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA BÁSICA

- John C. Reynolds. Theories of Programming Languages, 1998.

### BIBLIOGRAFÍA COMPLEMENTARIA

- Robert (Bob) Tennent. Semantics of Programming Languages, 1991. (también en castellano)
- Glynn Winskel. The Formal Semantics of Programming Languages: An Introduction, 1993.
- Joseph Stoy. The Scott-Strachey Approach to Programming Language Theory, 1977.
- Hanne Riis Nielson, Flemming Nielson. Semantics with Applications: A Formal Introduction, 1992, 1999.

## EVALUACIÓN

Se toman tres evaluaciones parciales, uno para cada módulo principal de la materia: temas de álgebra, de lógica y de lenguajes formales.  
Los exámenes evalúan contenidos teóricos y prácticos.

### Requisitos para obtener la regularidad

- 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).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Lógica

CARÁCTER: Obligatoria

#### OBJETIVOS

Que el alumno sea capaz de:

- Comprender los aspectos básicos de la lógica de primer orden.
- Diferenciar semántica de sintaxis.
- Comprender conceptos de completitud e incompletitud.
- Quedar en condiciones de abordar con facilidad el estudio de cálculos similares al de primer orden.

#### 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. Homomorfismo y congruencias de reticulados complementados. El teorema del filtro primo. Lema de Rasiova y Sikorki.

##### Capítulo 2

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

##### Capítulo 3

Estructuras de tipo  $\tau$ . 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. Álgebras. Subuniversos y subálgebras. Producto directo de dos 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 standard. Existencia de modelos no standard. Análisis de recursividad del lenguaje de primer orden: los teoremas forman un conjunto recursivamente enumerable. Funciones representables. La función  $\beta$  de Godel. Toda función primitiva recursiva es representable. Teorema de incompletitud de Godel.

#### MODALIDAD DE ENSEÑANZA

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.

#### BIBLIOGRAFÍA

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

#### EVALUACIÓN

##### Requisitos para obtener la regularidad

- Aprobación de dos de los tres parciales.



Requisitos para la aprobación:

- Evaluación escrita de contenidos teóricos y prácticos de la materia.

Promoción: (sólo de la parte práctica)

- Aprobación de los tres parciales, con promedio mayor o igual a 7 (siete).

*MPKisp*      *Jouf*



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Matemática Discreta I

CARÁCTER: Obligatoria

#### OBJETIVOS

Que los alumnos puedan:

- conocer la estructura algebraica de los números naturales, enteros, reales y complejos.
- Demostrar utilizando conceptos y técnicas establecidas.
- Utilizar conceptos de combinatoria para el conteo.
- Reconocer y distinguir nociones básicas de 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.

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

Unidad III: Divisibilidad.

Números enteros: definición y propiedades básicas. Algoritmo de la división. Desarrollos en base  $b$ ,  $b \geq 2$ . Máximo común divisor y mínimo común múltiplo. Números primos. Teorema fundamental de la Aritmética. Aplicaciones.

Unidad IV: Congruencia.

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. Sistemas de ecuaciones de congruencia. El teorema chino del resto. El pequeño teorema de Fermat. El teorema de Wilson.



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 (la expresión  $e^{i\theta}$ ). Cálculo de raíces de números complejos. Raíces de la unidad.

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: propiedades. Coloración de grafos. Número cromático. Algoritmo de Greedy. Grafos bipartitos: equivalencias.

MODALIDAD DE ENSEÑANZA

Las clases teóricas son con modalidad de clase magistral, que incluye interacción con los estudiantes para su participación en clase.

Las clases prácticas son con modalidad de taller donde los estudiantes resuelven los trabajos prácticos y el docente actúa como orientador respondiendo a las dificultades particulares que cada estudiante presente.

Los ejercicios con dificultades particulares se resuelven al frente en el pizarrón.

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.



## EVALUACIÓN

### Condiciones para la regularidad:

- Asistencia al 80% de las clases.
- Aprobación de dos de las tres evaluaciones parciales.

### Condiciones para la aprobación

- Aprobación de un examen final escrito, con contenidos teóricos y prácticos.



## PROGRAMA DE ASIGNATURA

ASIGNATURA: Matemática Discreta II

CARÁCTER: Obligatoria

### 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. 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.

### MODALIDAD DE ENSEÑANZA

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.

### 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, 2<sup>nd</sup> Ed., 1984.
- Network Flows: Theory, Algorithms and Applications. Ahoja-Magnani-Orlin, 1993, Prentice Hall.
- Combinatorial Optimization: Algorithms and Complexity. Papadimitriou-Steiglitz, 1998, Dover Publications.



## 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.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Modelos y Simulación

CARÁCTER: Obligatoria

#### 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.

MODALIDAD DE ENSEÑANZA

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.

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>

## 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.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Organización del Computador

CARÁCTER: Obligatoria

#### 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 tabulares y numéricos.

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

1.3-Ejemplos de aplicación utilizando HDL ("Hardware Description Language").

##### Unidad 2: Circuitos Lógicos Secuenciales

2.1-Celdas básicas de memoria ("Flip-Flops").

2.2-Circuitos sincrónicos y asincrónicos.

2.3-Autómatas de Moore y Mealy.

2.4-Introducción a los circuitos lógicos secuenciales programables.

2.5-Ejemplos de aplicación utilizando HDL ("Hardware Description Language").



### Unidad 3: Aritmética Binaria

- 3.1-Sistemas binarios de numeración.
- 3.2-Representación de números negativos.
- 3.3-Puntos fijo y flotante.
- 3.4-Operaciones aritméticas implementadas mediante circuitos combinacionales y secuenciales.
- 3.5-Ejemplos de aplicación utilizando HDL ("Hardware Description Language").

### Sección 2: "Estructura del Computador"

### Unidad 4: Microprocesadores

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

### Unidad 5: Memorias Semiconductoras

- 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). Memoria "CACHE".
- 5.3-Estructuración o decodificado de bancos de memorias ("Memory Mapped").



#### Unidad 6: Interfases Programables

6.1-Interfaz paralelo. Su constitución, decodificado ("I/O Mapped") y programación. Aplicaciones.

6.2-Interfaz serie. Su constitución, decodificado ("I/O Mapped") y programación. Aplicaciones.

#### MODALIDAD DE ENSEÑANZA

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.

#### 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.
- Mandado, E. : "Sistemas Electrónicos Digitales". Marcombo Boixareu Editores, última edición.
- Morris Mano, M.: "Ingeniería Computacional, diseño del hardware". Prentice Hall Hispanoamericana S.A., 1992.
- Thomas C. Bartee: "Fundamentos de Computadoras Digitales". Mc. Graw Hill, quinta edición (Primera en castellano).
- Mandado E. y Tassis E. : "Diseño de Sistemas Digitales con Microprocesadores". Marcombo Boixareu Editores, última edición.
- Mandado E.: "Procesadores Programables: El Microprocesador". Marcombo Boixareu Editores, última edición.



- Tocci R. J.: "Sistemas Digitales, Principios y Aplicaciones". Prentice Hall, tercera edición.
- Tanenbaum, A. S.: "Organización de Computadoras, un enfoque estructurado". Prentice Hall Hispanoamericana S. A., 1992.
- Antonio J. Gil Padilla: "Electrónica General: 1. Dispositivos y Sistemas Digitales". Mc. Graw Hill, última edición.
- Stephen Brown and Zvonko Vranesic: "Fundamentals of Digital Logic with VHDL Design" Mc. Graw Hill, 2000.(ISBN 0-07-012591-0).

#### EVALUACIÓN

##### Condiciones para la regularidad:

Aprobar ambos parciales, o un parcial y un recuperatorio.

Entregar y aprobar los informes correspondientes a los trabajos prácticos.

##### Condiciones para la aprobación:

Aprobación de un examen final escrito.



## PROGRAMA DE ASIGNATURA

ASIGNATURA: Paradigmas de la Programación

CARÁCTER: Obligatoria

### 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
  - Introduction and Oz Tutorial
- 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
  - Threads, Streams, Ejecución Lazy.
  - 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.

#### MODALIDAD DE ENSEÑANZA

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.

#### BIBLIOGRAFÍA

##### BIBLIOGRAFÍA BÁSICA

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

#### 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)



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Probabilidad y Estadística

CARÁCTER: Obligatoria

#### OBJETIVOS

- Proporcionar una base sólida a nivel universitario de la teoría estadística.
- Destacar su importancia en la resolución de problemas de diversas disciplinas.
- Sentar las bases para el cursado de la asignatura Modelos y Simulación.

#### CONTENIDO

Unidad I: Estadística descriptiva.

Modelos matemáticos; modelos determinísticos y aleatorios. Población, censo 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 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,  $\sigma$ -álgebra de eventos, función de probabilidad. Propiedades. Probabilidad de unión de eventos. Espacios finitos equiprobables. Probabilidad condicional. Propiedades. Fórmula multiplicativa, fórmula de 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 o función de probabilidad de masa. Función de distribución acumulada de una variable aleatoria. Propiedades. Esperanza, valor esperado o media de v. aleatorias discretas. Valor esperado de funciones de v. aleatorias discretas.



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.

Variable aleatoria absolutamente continua. Función densidad de probabilidad. Función de distribución acumulada. Percentil de una v.a. con densidad. Valor esperado o valor de media 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. Distribución normal estándar. Media y varianza. 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 cálculo de probabilidades. 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 de una distribución normal con 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 de una distribución normal con varianza desconocida. Tests para la varianza de una 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 grandes. Pruebas  $t$  e Intervalos de Confianza para la diferencia de dos medias poblacionales normales.

#### MODALIDAD DE ENSEÑANZA

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.

#### BIBLIOGRAFÍA

##### BIBLIOGRAFÍA BÁSICA

- Jay L. Devore. Probabilidad y Estadística para ingeniería y 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 Reverté , 2007.

### EVALUACIÓN

#### Condiciones para la regularidad

- Aprobación de dos de los tres parciales.

#### Condiciones para la aprobación

- Aprobación de un examen final escrito

#### Promoción

- Aprobación de los tres parciales con promedio mínimo 6(seis).



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Redes y Sistemas Distribuidos

CARÁCTER: Obligatoria

#### 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. Software 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, 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.

#### MODALIDAD DE ENSEÑANZA

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.

## BIBLIOGRAFÍA

### BIBLIOGRAFÍA BÁSICA

- Andrew S. Tanenbaum. Redes de Computadoras. Cuarta Edición, Prentice Hall, 2003.

### BIBLIOGRAFÍA COMPLEMENTARIA

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

## 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.

Dra. NOEMÍ PATRICIA KISBYE  
SECRETARÍA ACADÉMICA  
Fa.M.A.F. - U.N.C.

Dr. FRANCISCO A. TAMARIT  
DECANO  
Fa.M.A.F.



### PROGRAMA DE ASIGNATURA

ASIGNATURA: Sistemas Operativos

CARÁCTER: Obligatoria

#### 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.

#### CONTENIDOS

Unidad I: Introducción.

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

Unidad II: Procesos e Hilos.

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

Unidad III: 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) Implementación de los sistemas de archivos. (4) 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.

#### MODALIDAD DE ENSEÑANZA

- 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.



### 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.

### 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 alienta el régimen de promoción.