

# PROGRAMACIÓN DE LA ASIGNATURA

ASIGNATURA: Inteligencia Computacional

CARRERA: Ingeniería en Informática

PLAN DE ESTUDIOS: 1999 (Optativa) -2006 (Obligatoria)

ÁREA O DEPARTAMENTO: Departamento de Informática

DOCENTE RESPONSABLE: Diego Milone

CARÁCTER DE LA ASIGNATURA: Cuatrimestral

## DOCENTES

Leandro E. Di Persia: JTPI, SE, Lunes 13 a 16 hs y Miércoles 16 a 17 hs

Leonardo L. Giovanini: PAsI, S, Jueves 11 a 16 hs

Diego H. Milone: PTO, S, Jueves 11 a 16 hs

H. Leonardo Rufiner: PTO, S, Jueves 11 a 16 hs

Leandro D. Vignolo: ADI, S, Lunes 13 a 16 hs y Miércoles 16 a 17 hs

Se incluyen también los horarios de consultas personalizadas.

## CARGA HORARIA SEMANAL

TEORÍA: 3 hs

- Clases de discusión: jueves 13 a 14 hs.
- Clases teóricas: jueves de 14 a 16 hs.

(Se utilizará el horario de las clases de discusión y las clases teóricas para los dos exámenes parciales)

PRÁCTICA (total): 3 hs

- Formación experimental: 50 min
- Resolución de ejercicios prácticos: 50 min
- Resolución de problemas abiertos: 35 min
- Proyecto y diseño: 30 min
- Evaluación continua: » 15 minutos en el horario de las clases prácticas.

CARGA HORARIA TOTAL (s/Plan de Estudios): 90 hs

Cada docente estará disponible una vez por semana para consultas, en un horario a convenir con los alumnos al inicio de cada cursado.

Se puede estimar que los alumnos deberían dedicar fuera del horario de clases, y para cada actividad, el equivalente a la carga horaria semanal, de forma de poder complementar los estudios y trabajos en grupo requeridos.

Los alumnos deberán suscribirse al grupo electrónico (<http://groups.yahoo.com/group/icfich>), desde el cual podrán realizar consultas breves y recibir información de la asignatura. Se ha creado un Wiki (<http://icfich.wikidot.com>) para el fácil acceso al material de estudio, las guías de trabajos prácticos, transparencias de teoría, patrones de entrenamiento y prueba y otros documentos de utilidad para el cursado. Se recomienda a los alumnos prestar especial atención a los horarios de cursado y consultas, y asistir con puntualidad a cada actividad. Tanto para consultas técnicas como administrativas se solicita respetar los horarios acordados.

## **OBJETIVOS DE LA ASIGNATURA**

### ***Objetivos específicos***

Que el alumno:

- obtenga conocimientos generales acerca del área de la inteligencia computacional y una idea de su magnitud y diversidad,
- conozca las arquitecturas neuronales más utilizadas,
- comprenda las características dinámicas de redes recurrentes,
- aprenda los algoritmos de entrenamiento,
- implemente distintas arquitecturas neuronales y algoritmos básicos de entrenamiento,
- aprenda y utilice la teoría de conjuntos borrosos,
- aprenda los métodos de computación evolutiva y el diseño de la solución de problemas mediante éstos,
- implemente los algoritmos básicos de computación evolutiva y los aplique a problemas sencillos,
- adquiera independencia en el diseño de la solución de problemas mediante las técnicas tratadas,
- adquiera destrezas para aplicar los conocimientos adquiridos,
- conozca diversas aplicaciones directas de la inteligencia computacional.

### ***Objetivos generales***

Que el alumno:

- adquiera una nueva visión de la computación, desde la perspectiva del aprendizaje maquina,
- entienda los principios en que se basan muchas de las tecnologías con las que tiene un contacto permanente,
- incremente sus capacidades para el trabajo en grupo y la distribución de tareas y responsabilidades,

- incremente sus destrezas para la transmisión oral y escrita de conocimientos científicos y tecnológicos,
- desarrolle su capacidad de análisis aplicando diversas estrategias para resolución de problemas,
- incremente sus destrezas para aprender de forma independiente,
- realice trabajos experimentales que reflejen situaciones reales típicas,
- establezca contacto con publicaciones de nivel científico, pudiendo analizarlas, reproducirlas parcialmente y criticarlas,
- desarrolle su creatividad en la propuesta de nuevas técnicas o aplicaciones y mejoras de las ya conocidas,
- utilice correctamente la terminología técnica del área,
- aplique e incremente sus conocimientos de inglés técnico,

Además, entre otros objetivos de formación general, se espera que el alumno:

- valore la discusión abierta como una fuente de generación y difusión de conocimientos,
- valore los medios que la Universidad pone a su disposición y desarrolle sentimientos positivos hacia ella,
- se involucre más intensamente con la vida universitaria,
- conozca los valores y principios que sustentan a las instituciones académicas,
- se introduzca al pensamiento científico y tecnológico,
- se interese por formar parte en grupos de investigación,
- se interese por continuar su formación mediante estudios de postgrado.

## **PROGRAMA ANALÍTICO**

### ***Introducción general***

Unidad I: Breve revisión histórica de la inteligencia computacional. Áreas del conocimiento involucradas y su relación como parte de la inteligencia artificial. El cerebro humano y las limitaciones del cálculo computacional. El impacto y el amplio espectro de aplicaciones de la inteligencia computacional. Introducción conceptual a las tres técnicas fundamentales de la inteligencia computacional.

### ***Redes neuronales***

Unidad II: Introducción a las redes neuronales. La inspiración biológica: fisiología neuronal básica, redes de neuronas biológicas y escalas de organización estructural del cerebro. Modelos de neurona: la sinapsis, funciones de activación. Generalidades: características de las redes neuronales, clasificación de las arquitecturas neuronales, clasificación de los procesos de aprendizaje, espacio de soluciones, mínimos locales y globales, capacidad de generalización y técnicas de validación cruzada.

Unidad III: Redes neuronales estáticas. Perceptrón simple: hiperplanos para la separación de clases, entrenamiento y limitaciones. Perceptrón multicapa: formulación matemática del algoritmo de retropropagación, velocidad de aprendizaje y término de momento, inicialización y criterios de finalización, definición de la topología y los parámetros de entrenamiento. Redes neuronales con funciones de base radial: arquitectura, fronteras de decisión, algoritmos de entrenamiento. Mapas auto-organizativos: arquitecturas, algoritmo de entrenamiento, mapas topológicos, cuantización vectorial con aprendizaje.

Unidad IV: Redes neuronales dinámicas. Dinámica temporal en las redes neuronales: concepto y tipos de dinámicas temporales, clasificación general. Redes totalmente recurrentes: red de Hopfield, almacenamiento y recuperación de datos en una memoria asociativa, introducción al temple simulado y la máquina de Boltzmann. Redes parcialmente recurrentes: retropropagación a través del tiempo, redes neuronales con retardos en el tiempo, clasificación espacio-temporal, redes de Elman y Jordan.

## ***Sistemas borrosos***

Unidad V: Introducción a los sistemas borrosos. La borrosidad como multivalencia: incerteza versus aleatoriedad, teoría clásica de conjuntos y conjuntos borrosos, función de membresía. Representaciones del conocimiento basadas en reglas: variables lingüísticas, los sistemas expertos y la lógica borrosa. Características generales y campo de aplicaciones.

Unidad VI: Teoría de los conjuntos borrosos. Geometría de los conjuntos borrosos: representación gráfica, membresía, definición de subconjunto borroso. Definición e interpretación gráfica de los operadores borrosos: inclusión, igualdad, complemento, intersección, unión, suma disyuntiva, diferencia. El conjunto medio. Medidas de distancia y distancias relativas. Caracterización de conjuntos borrosos: conjuntos de nivel  $\alpha$ , conjunto convexo, conjunto normal, tamaño de un conjunto borroso. Entropía borrosa, teorema de la entropía borrosa, teorema del subconjunto, teorema entropía-subconjunto.

Unidad VII: Memorias asociativas borrosas. Generalidades: memorias asociativas borrosas como mapeos, reglas borrosas simples y compuestas, ejemplos. De las entradas escalares a los antecedentes borrosos. Codificación de reglas borrosas: discretización, memorias asociativas borrosas hebbianas, codificaciones por correlación-mínimo y correlación-producto, bidireccionalidad. Composición de reglas. De los consecuentes borrosos a las salidas escalares. Métodos de máximo y centroide borroso. Conjuntos de membresía continuos, representación y composición de varios antecedentes por consecuente. Comparación con la representación del conocimiento con redes neuronales y sistemas expertos.

## ***Computación evolutiva***

Unidad VIII: Introducción la computación evolutiva. La inspiración biológica: teoría de la evolución, la evolución darwiniana como un algoritmo, Darwin vs. Lamarck, genética básica. La estructura de un algoritmo evolutivo: iniciación de individuos, evaluación de aptitudes, selección

de progenitores, variación y reproducción. Generalidades: clasificación de los métodos de computación evolutiva, características generales, búsqueda y optimización de soluciones, algoritmos determinísticos y la importancia de las variables aleatorias, campos de aplicación.

Unidad IX: Diseño de un algoritmo evolutivo. Representación de los individuos: espacios del genotipo y fenotipo, representaciones binarias y otros esquemas más generales. Función de aptitud: características necesarias, mapeo del genotipo al fenotipo. Mecanismos de selección: importancia de las variables aleatorias, método de la ruleta, método de las ventanas o rangos, método de competencias, elitismo. Operadores elementales de variación: cruza simples y múltiples, mutaciones, el rol de cada operador en la búsqueda de soluciones. Reproducción: generación de la nueva población de individuos, brecha generacional. Parámetros de control para los algoritmos evolutivos.

Unidad X: Variantes de la computación evolutiva. Definiciones y análisis comparativo: algoritmos genéticos, programación genética, estrategias de evolución. Comparación con otras técnicas: métodos de búsqueda exhaustiva, métodos de gradiente descendiente, método de Monte Carlo y otros métodos de búsqueda aleatoria y heurística.

## ***Aplicaciones***

Unidad XI: Configuración del problema y aplicación de las técnicas de inteligencia computacional en: clasificación de patrones, agrupación de patrones, aproximación de funciones, optimización, búsqueda de soluciones, predicción de series temporales, control de procesos, identificación de sistemas, compresión de señales, memorias y recuperación de información. Interrelaciones entre las técnicas de inteligencia computacional: sistemas híbridos.

## **PROGRAMA DE ACTIVIDADES PRÁCTICAS**

GTP1: Perceptrón simple

GTP2: Perceptrón multicapa

GTP3: RBF-NN, SOM, Redes de Hopfield

GTP4: Sistemas borrosos

GTP5: Computación evolutiva

Ver detalles en el “Cronograma de Actividades”

## **BIBLIOGRAFÍA**

La mayor parte de la bibliografía está disponible en inglés, aunque algunos libros pueden conseguirse también en castellano. Se recomienda especialmente leer las ediciones en inglés, para incorporar la terminología técnica, practicar el idioma, y dado que las versiones originales contienen una menor cantidad de errores.

## Libros

Algunas aclaraciones en cuanto a los libros de la siguiente lista:

- Los que se encuentran disponibles en la Biblioteca de la Facultad fueron marcados con (b)
- Los que pueden pedirse prestados en el Laboratorio de Señales e Inteligencia Computacional se marcaron (i)
- En su mayor parte están disponibles en formato electrónico
- Al final de la presentación de cada clase teórica se detallan las secciones de consulta básica para cada unidad temática. Esta información también está disponible en el Wiki de la asignatura.

### Redes neuronales

- (i)S. Haykin, Neural Networks and Learning Machines, Prentice Hall, 2008.
- M. M. Gupta, L. Jin, N. Homma, Static and Dynamic Neural Networks From Fundamentals to Advanced Theory, John Wiley & Sons, 2003.
- (i)D. P. Mandic, J. Chambers, Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability, John Wiley & Sons, 2001.
- (i)C. Bishop, Neural Networks for Pattern Recognition, Oxford, 1999.
- R. M. Hristev, The ANN Book, 1998.
- (i)B. D. Ripley, Pattern Recognition and Neural Networks, Cambridge Univ. Press, 1996.
- (i)T. Masters, Neural, Novel & Hybrid Algorithms for Time Series Prediction, J. Wiley & Sons, 1995.
- (i)T. Kohonen, Self Organizing Maps, Springer Verlag, 1995.
- (ib)S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, 1994.
- (i)J. Freeman, Simulating Neural Networks, Addison Wesley, 1994.
- (i)J. Freeman y D. Skapura, Redes neuronales: algoritmos, aplicaciones y técnicas de programación, Addison Wesley, 1993.
- C. Lau, Neural Networks: Theoretical Foundations and Analysis, IEEE Press, 1992.

### Lógica borrosa

- W. Siler, J. J. Buckley, Fuzzy Expert Systems and Fuzzy Reasoning, John Wiley & Sons, 2005.
- T. Ross, Fuzzy Logic with Engineering Applications, Wiley, 2004.
- K. Tanaka, H. O. Wang, Fuzzy Control Systems Design and Analysis, John Wiley & Sons, 2001.
- O. Wolkenhauer, Data Engineering: Fuzzy Mathematics in Systems Theory and Data Analysis, John Wiley & Sons, 2001.
- (i)B. Kosko, Fuzzy Engineering, Prentice Hall, 1997.
- J-S. R. Jang, C-T. Sun, E. Mizutani, Neuro-Fuzzy And Soft Computing: A Computational Approach To Learning And Machine Intelligence, Prentice Hall, 1997.
- F. M. McNeill, E. Thro, Fuzzy Logic, A Practical Approach, Academic Press, 1994.

- (i)B. Kosko, Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, 1992.
- L. A. Zadeh y J. Kacprzyk, Fuzzy Logic for the Management of Uncertainty, John Wiley, 1992.
- J. C. Bezdek y S. K. Sankar, Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data, IEEE Press, 1992.

#### Computación evolutiva

- S. Sumathi, T. Hamsapriya, P. Surekha, Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab, Springer, 2008.
- C. A. Coello Coello, G. B. Lamont, A. A. Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems Series: Genetic and Evolutionary Computation, (2da Edición), Springer, 2007.
- R. L. Haupt, S. E. Haupt, Practical Genetic Algorithms, John Wiley & Sons, 2004.
- Menon, Frontiers of Evolutionary Computation, Kluwer Academic Publishers, 2004.
- L. D. Chambers, The Practical Handbook of Genetic Algorithms, Vols. I, II & III, CRC Press, 2000.
- T. Back, D. B. Fogel, Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators, IOP Publishing Ltd, 2000.
- T. Back, D. B. Fogel, Z. Michalewicz, Evolutionary Computation 2: Advanced Algorithms and Operators, IOP Publishing Ltd, 2000.
- M. Mitchell, An Introduction to Genetic Algorithms, MIT Press, 1999.
- T. Back, D. B. Fogel, Z. Michalewicz, Handbook of Evolutionary Computation, IOP Publishing Ltd and Oxford University Press, 1997.
- S. Pal y P. Wang, Genetic Algorithm for Pattern Recognition, CRC Press, 1996.
- von Altrock, Fuzzy Logic and Neurofuzzy Application Explained, Prentice Hall, 1995.
- (i)Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer Verlag, 1992.
- L. Davis (Ed.), Handbook of Genetic Algorithms, Van Nostrand Reinhold, 1991.
- (i)D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley, 1989.

#### Varios temas de la asignatura

- F. Emmert-Streib, M. Dehmer, Information Theory and Statistical Learning, Springer, 2009.
- (i)C. M. Bishop, Pattern Recognition and Machine Learning, Springer Series: Information Science and Statistics, 2006.
- L. Rutkowski, Computational Intelligence: Methods and Techniques, Springer, 2005
- (i)David J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.
- T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, (2da Edición), 2003.
- (i)A. P. Engelbrecht, Computational Intelligence: An Introduction, John Wiley & Sons, 2002.

- V. Kecman, Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models, MIT Press, 2001.
- (i)R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification (2da Edición), Wiley Interscience, 2000.
- Konar, Behavioral and Cognitive Modeling of the Human Brain Artificial Intelligence and Soft Computing, CRC Press, 2000.
- (i)V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 2000.
- V. Cherkassky y F. Mulier, Learning from Data: Concepts, Theory and Methods, Wiley International Science, 1998.

## ***Publicaciones periódicas***

Se sugieren las siguientes publicaciones como base para el trabajo final, tanto en el caso del análisis crítico de una publicación científica como para la búsqueda bibliográfica en el caso de una propuesta creativa. En estas publicaciones también se pueden encontrar revisiones y tutoriales muy completos sobre los temas centrales de la asignatura (varios de ellos se incluyen en el DVD preparado por la Cátedra y se citan al final de la clase teórica correspondiente y en el Wiki).

Proceedings of the IEEE.

IEEE Transactions on:

- Fuzzy Systems
- Evolutionary Computation
- Neural Networks
- Pattern Analysis and Machine Intelligence
- Systems, Man, and Cybernetics
- Robotics and Automation
- Image Processing
- Information Theory
- Knowledge and Data Engineering

Elsevier Science:

- Fuzzy Sets and Systems
- Intelligent Data Analysis
- International Journal of Neurocomputing
- Neural Networks
- Pattern Recognition
- Neurocomputing

Ablex Publishing: Journal of Artificial Neural Networks

World Scientific Publishing:

- International Journal on Artificial Intelligence Tools
- International Journal of Intelligent Control and Systems
- International Journal of Neural Systems
- International Journal of Pattern Recognition and Artificial Intelligence
- International Journal of Cooperative Intelligent Systems

- International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems

MIT Press:

- Neural Computation
- Evolutionary Computation
- Journal of Cognitive Neuroscience

Kluwer Academic Publishers:

- Machine Learning
- Neural Processing Letters
- Journal of Intelligent Systems (con Freund Publishing House Ltd.)

Springer Verlag: Neural Computing with Applications

John Wiley & Sons: International Journal of Intelligent Systems

Pergamom Press: Neural Networks

Blackwell Publishers: Computational Intelligence

International Neural Network Society: INNS Neural Networks Newsletter

Finance & Technology Publishing: Journal of Computational Intelligence in Finance

## CRONOGRAMA DE ACTIVIDADES

### Semana 1:

Clase de práctica (17/08/09): feriado.

Clase de discusión (20/08/09): planificación y condiciones de regularidad.

Clase de teoría (20/08/09): Unidades I-III (introducción a la asignatura, generalidades de redes neuronales, perceptrón simple).

### Semana 2:

Clase de práctica (24/08/09): comienza la GTP 1 (perceptrón simple).

Clase de teoría (27/08/09): Unidad III (perceptrón multicapa).

### Semana 3:

Clase de práctica (31/08/09): continúa la GTP 1.

Clase de teoría (03/09/09): Unidad III (redes con funciones de base radial).

### Semana 4:

**Evaluación** (07/09/09): GTP 1 ejercicios 1 y 2.

Clase de práctica (07/09/09): comienza GTP 2 (perceptrón multicapa).

Clase de teoría (10/09/09): Introducción al reconocimiento estadístico de formas (clase especial dictada por el Profesor César Martínez (invitado)).

### Semana 5:

**Evaluación** (14/09/09): GTP 1 ejercicios 3 y 4.

Clase de práctica (14/09/09): continúa GTP 2.

Clase de teoría (17/09/09): Unidad III (mapas auto-organizativos, cuantización vectorial con aprendizaje).

Semana 6:

Clase de práctica (21/09/09): feriado.

Clase de teoría (24/09/09): Unidad IV (redes dinámicas).

Semana 7:

**Evaluación** (28/09/09): GTP 2 ejercicios 1 y 2.

Clase de práctica (28/09/09): comienza GTP 3 (redes con funciones de base radial, mapas auto-organizativos y redes de Hopfield). Continúa GTP 2.

Clase de teoría (01/10/09): Unidades V y VI (sistemas y conjuntos borrosos).

Semana 8:

**Evaluación** (05/10/09): GTP 2 ejercicios 3 y 4.

Clase de práctica (05/10/09): continúa GTP 3.

Clase de teoría (08/10/09): Unidad VII (memorias asociativas borrosas).

Semana 9:

Clase de práctica (12/10/09): feriado.

Clase de teoría (15/10/09): **Parcial 1** (Unidades I a III).

Semana 10:

**Evaluación** (19/10/09): GTP 3 ejercicios 1, 2 y 3

Clase de práctica (19/10/09): comienza GTP 4 (control borroso).

Clase de teoría (22/10/09): Unidades VIII y IX (introducción a la computación evolutiva, algoritmos genéticos).

Semana 11:

**Evaluación** (26/10/09): GTP 3 ejercicios 4, 5 y 6.

Clase de práctica (26/10/09): continúa GTP 4.

Clase de teoría (29/10/09): Unidad X (variantes de la computación evolutiva). Unidad XI (distribución de trabajos de aplicación y discusión sobre los artículos o propuestas creativas para el trabajo final).

Semana 12:

Clase de práctica (02/11/09): comienza GTP 5 (algoritmos genéticos).

Clases de teoría (05/11/09): **Parcial 2** (Unidades IV a XI).

Semana 13:

**Evaluación** (09/11/09): GTP 4 (control borroso).

Clase de práctica (09/11/09): continúa GTP 5.

Clases de teoría (12/11/09): feriado.

Semana 14:

**Evaluación** (16/11/09): GTP 5 (algoritmos genéticos).

**Evaluación** (19/11/09): Introducción a los sistemas basados en conocimientos (Clase especial dictada por la Profesora Gabriela Henning (invitada)).

Semana 15:

**Evaluación** (23/11/09): Recuperación de prácticos y parciales. Presentación de los trabajos de aplicación.

**Evaluación** (30/11/09): Entrega de informes y presentación de los trabajos finales (Todos los alumnos deben estar presentes durante esta clase).

### ***Profesores a cargo de cada clase***

- Clases teóricas y de discusión de las semanas 1, 2, 3, 7 y 10: Diego H. Milone.
- Clases teóricas y de discusión de las semanas 5, 6 y 11: Hugo L. Rufiner.
- Clase teórica y de discusión de la semana 8: Leonardo L. Giovanini.
- Clases prácticas: Leandro E. Di Persia (a cargo) y Leandro D. Vignolo.
- Evaluaciones continuas: todo el equipo de cátedra.
- Evaluación parcial 1: Diego H. Milone.
- Evaluación parcial 2: Hugo L. Rufiner.
- Evaluación de trabajos finales: todo el equipo de cátedra.

## **REQUERIMIENTOS DE LA ASIGNATURA**

### ***Desarrollo del cursado***

*Clases de teoría:* se desarrollan como exposiciones de aproximadamente 2 horas semanales a cargo del responsable de la asignatura. En estas clases se introducen los conceptos teóricos de cada unidad temática. Estas introducciones deberán complementarse con la actividad en las clases de discusión, las guías de estudio y la bibliografía recomendada para cada tema. Este enfoque obliga a una participación activa del alumno en su propia formación aumentando su independencia y fortaleciendo sus capacidades para el autoaprendizaje.

*Clases de discusión:* se desarrollan durante aproximadamente 1 hora y tienen como objetivo profundizar los temas desarrollados teóricamente y trabajar sobre las dificultades e inquietudes conceptuales generadas a partir de la actividad práctica. Se propone una modalidad de trabajo más flexible y participativa, de forma que los alumnos planteen abiertamente sus dudas y discutan guiados por el responsable de la asignatura. Las discusiones están relacionadas con los objetivos específicos pero especialmente orientadas a tratar muchos de los aspectos mencionados en los objetivos generales.

*Clases de práctica:* la práctica de laboratorio consiste en la formulación de soluciones e implementación de las diferentes técnicas de inteligencia computacional. Durante los primeros

minutos de la clase se explican las características básicas del trabajo a realizar, asociado al tema desarrollado previamente en la clase teórica. A continuación se atiende a cada grupo individualmente y se realizan las evaluaciones correspondientes. Los prácticos son intensivos por lo que los alumnos recibirán una orientación y apoyo durante la clase pero deberán desarrollar también actividades en forma autónoma.

*Clases de consulta:* cada docente estará disponible en un horario semanal para consultas, de forma que se puedan evacuar las dudas en cualquiera de las actividades planteadas.

## ***Instancias de evaluación durante el cursado***

La evaluación durante el cursado se realiza en diferentes escalas temporales:

1. Evaluación continua: se realiza dentro de las 3 horas previstas para la actividad práctica y tiene por objetivo evaluar tanto los avances en la realización de trabajos prácticos con la incorporación de los conocimientos teóricos básicos relacionados. Esta instancia de evaluación es una de las más importantes del cursado ya que brinda la oportunidad de evaluar de cerca la evolución y sobre todo poder ajustar a tiempo los procesos de enseñanza y aprendizaje de acuerdo a las necesidades particulares de cada grupo de trabajo o alumno, recomendando lecturas adicionales, sugiriendo ampliaciones de algún ejercicio práctico, explicando en detalle algún tema en particular, etc. La evaluación continua se extiende entre 15 a 30 minutos, dependiendo de la modalidad elegida. En cada ocasión el responsable de la práctica podrá optar por:

- Evaluación oral grupal: defensa del trabajo práctico del tema correspondiente y frente a la computadora, con todo el código fuente y todos los ejercicios resueltos. Durante esta defensa se evalúan tanto los conocimientos prácticos como los teóricos. Si bien la evaluación es grupal, cada integrante del grupo debe estar en condiciones de responder correctamente a las preguntas<sup>1</sup>.
- Evaluación escrita individual: consistente en dos o tres preguntas (cortas) en relación al trabajo práctico y sus fundamentos teóricos.
- Evaluación escrita grupal: entrega de un informe técnico del trabajo práctico, donde deben presentarse adecuadamente los resultados, discusión y conclusiones para cada ejercicio de la guía<sup>2</sup>.

Los grupos de trabajo deberán estar conformados por 2 o 3 miembros, no se aceptarán trabajos individuales. Si bien cada grupo puede utilizar cualquier lenguaje o herramienta de programación, se recomienda utilizar un lenguaje compilado, dada la gran carga computacional de los cálculos involucrados. Vale aclarar que si bien el aprendizaje de una forma de implementación particular, lenguaje o entorno de programación no es parte de los objetivos de la asignatura<sup>3</sup>, la programación de todos los algoritmos desarrollados, sin la utilización de programas, toolboxes, librerías o funciones previamente desarrolladas por terceros, es una importante característica del enfoque pedagógico que damos a la asignatura. Por un lado, consideramos que el entendimiento acabado

1. Dependiendo de la cantidad de grupos, en esta instancia de evaluación/seguimiento suelen participar todos los integrantes de la cátedra.

2. En este caso se informará al alumno acerca de la modalidad con una semana de antelación a la entrega prevista en el cronograma, para que pueda preparar el informe escrito.

de todos los algoritmos clave se completa al programarlos desde cero, y más aún en la mente de nuestros estudiantes de informática. Por otro lado, este es un elemento motivacional muy importante para los alumnos, que sin dudas prefieren terminar de comprender los diferentes métodos invirtiendo más tiempo en programación que en la demostración de teoremas u otras actividades que claramente no les son gratas.

2. Evaluación parcial: se proveen dos exámenes parciales, involucrando aproximadamente la mitad de los temas del programa en cada uno (según se detalla en el cronograma.). Estas evaluaciones serán de aproximadamente 2:30 horas cada una, consistiendo en un examen escrito con especial énfasis en los aspectos teóricos pero incluyendo también problemas de aplicación. Para los problemas de aplicación no se requerirá el nivel de detalle de los trabajos prácticos pero si la formulación clara de las soluciones propuesta, utilizado diagramas en bloque, pseudocódigo y ecuaciones según corresponda.

Todos los exámenes parciales incluyen un problema abierto, que representa un desafío mayor del tipo de los últimos problemas de las guías de trabajos prácticos. No existe una solución única e incluso en muchos casos no existe una solución cerrada a este tipo de problemas. Por lo tanto, no se espera que el alumno “resuelva” el problema por completo sino que demuestre que puede integrar los conocimientos adquiridos para proponer una posible solución, utilizar adecuadamente las técnicas y conceptos desarrollados y formular una propuesta coherente, con claridad y precisión en el uso del lenguaje técnico del área. Este problema nunca supera los 15 puntos (sobre 100) y el tiempo máximo sugerido para su resolución se indica claramente en el enunciado, como para todos los ejercicios. De esta forma el alumno puede realizar un mejor manejo del tiempo durante el examen.

3. Trabajo final: se puede realizar individualmente o en grupos de hasta 2 alumnos. Este trabajo deberá ajustarse a una de las siguientes modalidades:

- Análisis crítico de una publicación científica: consistente en la lectura y análisis de un trabajo original de reciente publicación y en la temática de la asignatura. Se proveerán diversas publicaciones para realizar una selección pero también se aceptarán, previa revisión, artículos que los alumnos busquen en la hemeroteca de la Facultad o en otras fuentes.
- Propuesta creativa: es un trabajo original propuesto por el alumno. Este trabajo deberá ser acompañado por una búsqueda bibliográfica de antecedentes relacionados e implementado a partir de las herramientas computacionales que se proveen durante el cursado.

En ambos casos la aprobación consistirá en la entrega de un informe escrito y una presentación oral de 15 minutos, con defensa de 5 minutos. Las características del informe y la presentación se especificarán oportunamente durante el cursado. El estilo y estructura puede descargarse desde el Wiki de la Cátedra. La temática y el alcance debe ser acordado

---

3. Por lo tanto, los aspectos técnicos de la compilación y utilización de lenguajes y entornos de programación quedan bajo la responsabilidad de cada grupo de trabajo. Al momento de la entrega de los trabajos prácticos todos los programas deben funcionar correctamente y cada alumno debe ser capaz de explicar y justificar cada paso de la solución propuesta.

con el responsable de la asignatura antes de comenzar el trabajo. La presentación se realizará durante las últimas semanas del cursado, según se indica en el cronograma.

## **Modalidad de los exámenes finales**

Los exámenes finales serán individuales y se dividirán en:

Parte 1 *Evaluación de práctica*: examen a libro abierto y con computadora. Se formularán problemas que involucren varios temas de la asignatura y deberán ser resueltos en un plazo de 3 horas. Durante el examen el alumno podrá consultar toda la bibliografía con la que cuente en el aula, utilizar código fuente desarrollado previamente e incluso traer su propia computadora con todo el material que necesite. No se podrán realizar consultas a terceros y una vez resueltos los problemas el alumno deberá defender adecuadamente cada parte de la implementación según lo solicite el tribunal.

Parte 2 *Evaluación de teoría*: examen oral<sup>4</sup> de entre 15 y 30 minutos por alumno. Se evaluarán tres temas para los que el alumno podrá realizar un desarrollo preliminar en la pizarra y luego explicar oralmente, responder a las preguntas y realizar las ampliaciones que se le soliciten. El tribunal podrá agregar preguntas de otros temas para terminar de definir la calificación.

## **Aprobación de la asignatura**

Para la aprobación de la asignatura se consideran todas las instancias de evaluación durante el cursado, con la siguiente distribución relativa de puntos:

- Evaluación continua: 40 puntos (distribuidos por tema).
- Evaluaciones parciales: 40 puntos (distribuidos por parcial).
- Trabajo final: 20 puntos (10 en la presentación oral y 10 en el informe escrito)

Para aprobar la asignatura se contemplan las siguientes condiciones:

- Promoción completa: acumular 80 puntos o más en las evaluaciones durante el cursado. No rinde el examen final.
- Promoción de práctica: acumular 60 puntos o más en las evaluaciones durante el cursado. En el examen final se rinde sólo la Parte 2 (evaluación de teoría).
- Alumno libre: acumula menos de 60 puntos en las evaluaciones durante el cursado. Rinde el examen final completo y además debe aprobar previamente el trabajo final tal como se indicó más arriba. El alumno deberá informar al responsable de la asignatura 15 días antes de la fecha en que desea presentarse.

*Aclaración*: en ningún caso es válida la acumulación de puntos si en alguna de las instancias de evaluación se obtuviera menos del 40% de la puntuación total.

## **Exámenes de recuperación**

Se proveen dos instancias de recuperación de exámenes:

---

4. El tribunal podrá optar por un examen escrito si se presentan más de diez alumnos.

- En caso de no alcanzar este 40% en alguna de las instancias de evaluación continua se podrá recuperar un tema.
- Para aumentar la calificación en las evaluaciones parciales se podrá recuperar un examen a elección.

Los exámenes de recuperación serán individuales aunque la modalidad (escrito/oral) será dispuesta por el responsable de la asignatura independientemente de aquella con que se hubiese evaluado originalmente el tema. La nota final es el máximo entre el examen original y el examen de recuperación.

### ***Deshonestidad académica***

En el caso de que un alumno incurra en cualquier acto de deshonestidad académica quedará automáticamente *LIBRE* sin importar su condición previa en la asignatura. Además se elevará un pedido a la Secretaría Académica para que el alumno sea sancionado de acuerdo al caso. Se considerarán actos de deshonestidad académica: copiar exámenes (de cualquier tipo y en cualquier forma), copiar informes, copiar programas o ideas originales para la resolución de problemas. Todo trabajo presentado por el alumno se considera que ha sido el resultado de su proceso de aprendizaje y fruto de su propio esfuerzo. Como regla general, en un caso de copia a otros compañeros son culpables ambas partes, por lo que se sugiere a los alumnos cuidar sus informes, códigos fuente o cualquier otro objeto de una evaluación. Se considera también un acto deshonesto utilizar material (texto, figuras, etc.) de otras fuentes (Internet, libros, revistas, etc.) sin realizar la cita correspondiente. No es aceptable la copia textual en ningún caso, desde ninguna fuente.

Como es natural, no es posible enumerar todos los casos de deshonestidad académica por lo que la lista anterior no es exhaustiva y otros casos serán analizados oportunamente. Se invita a los alumnos a que en el caso de tener cualquier duda al respecto, consulten al responsable de la asignatura antes de realizar cualquier acción.

## **EVALUACIONES**

### ***Parciales***

TEMAS QUE INCLUYE: Ver detalles en el “Cronograma de Actividades”

ORAL/ESCRITO: Ver detalles en los “Requerimientos de la Asignatura”

### ***Recuperatorios***

TEMAS QUE INCLUYE: Ver detalles en el “Cronograma de Actividades”

ORAL/ESCRITO: Ver detalles en los “Requerimientos de la Asignatura”

### ***Coloquio Final Integrador***

DESCRIPCIÓN: Ver detalles en los “Requerimientos de la Asignatura”

ORAL/ESCRITO: Ver detalles en los “Requerimientos de la Asignatura”

### ***Información complementaria:***

Ver detalles en los “Requerimientos de la Asignatura”

## **RECURSOS REQUERIDOS**

Para el dictado de clases:

- Pizarra blanca y marcadores de color de borrado en seco.
- Transparencias y proyector de transparencias.
- Proyector de cañón con entrada SVGA
- Laboratorio de computación
- Computadoras tipo PC Pentium IV o superiores (al menos 1 cada 2 alumnos)
- Sistema operativo Linux o Windows
- Herramientas de programación: básicamente compiladores y entornos de desarrollo para los lenguajes que el alumno ha aprendido en los primeros años de la carrera y la disponibilidad para instalar otros entornos que eventualmente puedan elegir.
- Conexión a Internet
- Periféricos para el registro y reproducción de audio (micrófono y parlantes)\*
- Biblioteca y hemeroteca
- No se preveen gastos de movilidad.

Espacio físico para tareas de Cátedra (aprox. 10 m) con:

- 1 computadora PC Pentium IV o superior,
- 1 impresora láser,
- 2 escritorios,
- 1 armario,
- 1 archivero,
- 5 sillas,
- 1 pizarra,
- Consumibles varios: hojas, fibras, fotocopias, etc.

## **FIRMAS DE LOS DOCENTES**

Leandro E. Di Persia

Leonardo L. Giovanini

Diego H. Milone

H. Leonardo Rufiner

Leandro D. Vignolo