



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area I: Datos

(Programa del año 2026)
(Programa en trámite de aprobación)
(Presentado el 20/08/2026 20:41:13)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
ESTRUCTURA DE DATOS Y ALGORITMOS	ING. INFORM.	026/1	2-	2026 2° cuatrimestre
		08/15		

II - Equipo Docente

Docente	Función	Cargo	Dedicación
REYES, NORA SUSANA	Prof. Responsable	P.Asoc Exc	40 Hs
KASIAN, FERNANDO ANDRES	Prof. Co-Responsable	P.Adj Exc	40 Hs
AZAR, ELIANA PAOLA	Responsable de Práctico	JTP Exc	40 Hs
SOSA TORANZO, CECILIA LORENA	Responsable de Práctico	JTP Exc	40 Hs
JOFRE, ANA MARIA	Auxiliar de Práctico	A.1ra Exc	40 Hs
LEIVA OLGUIN, GASTON EMANUEL	Auxiliar de Práctico	A.1ra Simp	10 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
Hs	2 Hs	1 Hs	3 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
03/08/2026	13/11/2026	15	90

IV - Fundamentación

Se toma en este curso el diseño de estructuras de datos considerando principalmente las evocaciones asociativas con respuesta no múltiple y algunos tipos de evocaciones no asociativas, como las extremales.

Además, se inicia al alumno en el análisis de algoritmos, con el fin de que aprenda a comparar distintos algoritmos y a qué es fundamental para diseñar buenos algoritmos.

En esta asignatura se sientan las primeras bases para construir una base sólida en las disciplinas de Algoritmos, Estructuras de Datos y Bases de Datos, de forma tal que se tenga la idoneidad suficiente en la temática, contando con los conceptos, principios y teorías que constituyen el ámbito de competencia.

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se pretende que el alumno sea capaz de:

- Manejar con idoneidad los conceptos que involucran el diseño de estructuras de datos y algoritmos.
- Conocer algunos de los principales algoritmos y estructuras de datos, incluyendo el análisis de su desempeño.

- Analizar y diseñar algoritmos.
- Desarrollar una actitud crítica frente al uso de las estructuras de datos y algoritmos con los que se pueda enfrentar.
- Desarrollar la actitud de búsqueda de nuevas soluciones.
- Reconocer, modificar, innovar y aplicar los distintos tipos de algoritmos.
- Frente a una aplicación o problema particular, poder brindar una solución eficiente utilizando los conceptos vistos sobre diseño de estructuras de datos y algoritmos, introducir herramientas de diseño de algoritmos y la ingeniería algorítmica como selección de las estructuras de datos y de las técnicas algorítmicas más adecuadas y además, utilizar el análisis de los algoritmos para evaluar y justificar la eficiencia de la solución elegida.

VI - Contenidos

Contenidos Mínimos (Ing. en Informática OCD 26/12)

Teoría, Propiedades, demostraciones y representaciones de grafos. Evaluación de algoritmos: función de costo, medidas en tiempo y espacio, notaciones asintóticas, complejidad, clases de complejidad. Listas, pilas y colas: representación y operaciones sobre cada tipo de estructura, análisis de costos. Direccionamiento directo. Árboles computacionales. Distribución pseudo-aleatoria de datos. Técnicas de diseño de algoritmos. Laboratorio: Uso de las distintas estructuras de datos en aplicaciones típicas. Corroboración de la complejidad teórica de algoritmos de ordenamiento y búsqueda por métodos experimentales, identificando diferencias entre los comportamientos del mejor, promedio y peor caso.

Durante el dictado de la asignatura se abordan los siguientes ejes transversales:

- Identificación, formulación y resolución de problemas de informática.
- Concepción, diseño y desarrollo de proyectos de informática.
- Gestión, planificación, ejecución y control de proyectos de informática.
- Utilización de técnicas y herramientas de aplicación en la informática.
- Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Fundamentos para el desempeño en equipos de trabajo.
- Fundamentos para la comunicación efectiva
- Fundamentos para la acción ética y responsable.
- Fundamentos para el aprendizaje continuo.

Unidad 1. Explicitación del objetivo de la materia:

Repaso de temas y homogeneización de terminología. Las estructuras. Dato e información. Doble acepción de estructuras de datos. Visión relacional. Fundamento de la evocación asociativa. Concepto de Servicio en una estructura de datos. Operaciones. Datos de un problema de estructuras de datos. Algoritmos.

Unidad 2. Evaluación de algoritmos:

Planteo. Problema. Función de costo. Necesidad de una función de evaluación. Propiedades. Medidas en tiempo y espacio. Elección de la función de evaluación. Familias de problemas. Solución parametrizada. Notaciones asintóticas. Propiedades. Complejidad. Clases de Complejidad.

Unidad 3. Listas, Pilas y Colas:

Listas vinculadas y secuenciales. Incidencia del orden, Incidencia del tipo de recurrencia que la define. Búsqueda binaria. Lista Invertida. Lista de 2 niveles. Pilas y colas. Skip Lists. Operaciones sobre cada tipo de estructura. Aplicaciones.

Unidad 4. Propiedades, demostraciones y representación de grafos:

Definiciones. Propiedades. Representaciones básicas de grafos. Algoritmos sobre grafos

Unidad 5. Árboles computacionales:

Definición. Representación computacional. Árbol completo, lleno y balanceado. Árboles Binarios de Búsqueda. Árboles balanceados en altura. Parva. Operaciones básicas sobre cada tipo de árbol. Aplicaciones.

Unidad 6. Distribución pseudo-aleatoria de datos:

Motivación. Funciones de pseudo-aleatorización. El problema del rebalse. Distintas propuestas para su manejo.

Direccionamiento Directo, condiciones para su aplicación, relajación de la exigencia de totalidad. Aplicaciones.

Unidad 7. Técnicas de diseño de algoritmos:

Técnicas ávidas, programación dinámica, dividir para vencer. Ejemplos de aplicación sobre métodos de Ordenamiento. Ejemplos de aplicación en algoritmos sobre Grafos. Otros ejemplos de aplicación de las distintas técnicas.

VII - Plan de Trabajos Prácticos

Mediante la realización de práctica, tanto en papel como en la computadora, se promueve la integración de conceptos teóricos, su afianzamiento y su aplicación en la resolución de diferentes problemas planteados.

Las clases prácticas consistirán principalmente en la resolución de prácticos de laboratorio que implementan los conceptos teóricos desarrollados en la asignatura, de manera que pueda poner en práctica los conocimientos adquiridos.

Se proponen diferentes situaciones de la vida real que permiten al estudiante identificar problemas y abordar su posible solución, aplicando tanto técnicas presentadas en teoría como su ingenio e investigación. Los planteos presentados, por medio de los prácticos, incrementan gradualmente su complejidad, de manera que el

Estudiante afiance conceptos más simples antes de abordar situaciones más complejas.

Este proceso se evalúa de manera formativa, mediante la entrega de ejercicios representativos e integradores que son corregidos y comentados posteriormente en clase, permitiendo una devolución general y otra personal a los estudiantes.

Las prácticas de laboratorio, además de presentar la posibilidad de diseñar y desarrollar un proyecto informático, contribuye a que los estudiantes planifiquen las diferentes etapas del mismo; trabajen de manera colaborativa a través de la formación de grupos de trabajo; debatan y analicen diferentes soluciones a los problemas planteados, para seleccionar la más eficiente y efectiva; utilicen herramientas adecuadas (por ej. un lenguaje de programación imperativo, un entorno de desarrollo, etc.); mejoren su comunicación a través de la presentación de informes, orales y/o escritos, sobre lo desarrollado en cada proyecto; y demuestren su ética, responsabilidad profesional al cumplir con los plazos establecidos al principio del curso.

Este proceso se evalúa de forma sumativa mediante la entrega y aprobación de los proyectos diseñados, presentando su código fuente, en las fechas predeterminadas en el cronograma de la asignatura, y realizando una descripción oral del mismo, en la que todos los integrantes del grupo de trabajo participen.

Además, se propone material extra disponible en la web sobre charlas y temas de interés para que los estudiantes investiguen sobre estructuras y/o algoritmos particularmente novedosos, impulsando su curiosidad, la posibilidad de indagar más allá de lo visto en clase y el aprendizaje autónomo.

Finalmente, lo trabajado durante el cuatrimestre es evaluado de manera sumativa por medio de un examen con sus respectivas recuperaciones. En cada instancia de evaluación, se realiza la devolución de los resultados mediante la corrección y debate sobre posibles soluciones a cada uno de los ejercicios planteados (corrección informada).

De esta manera se aporta a los ejes:

- Identificación, formulación y resolución de problemas de informática.
- Concepción, diseño y desarrollo de proyectos de informática.
- Gestión, planificación, ejecución y control de proyectos de informática.
- Utilización de técnicas y herramientas de aplicación en la informática.
- Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Fundamentos para el desempeño en equipos de trabajo.
- Fundamentos para la comunicación efectiva.
- Fundamentos para la acción ética y responsable.
- Fundamentos para el aprendizaje continuo.

Prácticos de aula:

Los mismos están orientados a afianzar los conceptos teóricos sobre las diferentes estructuras y algoritmos presentados en la asignatura y a estimular el planteo de soluciones ingeniosas a las diferentes problemáticas presentadas. Para un seguimiento formativo, se deberán presentar ejercicios representativos de cada práctico, los que serán corregidos para una posterior devolución general y personal.

1. Repaso de relaciones y funciones.
2. Desarrollo de algoritmos para realizar búsqueda de elementos en las distintas representaciones computacionales posibles

para un conjunto.

3. Funciones de evaluación. Notaciones asintóticas. Utilización para expresar tiempos de ejecución de programas.

4. Diseño de los algoritmos necesarios para operar cada una de las distintas implementaciones de listas y cálculos de esfuerzos. Pilas y Colas.

5. Árboles como estructuras de información. Barridos. Árbol binario de búsqueda. Árbol binario balanceado en altura. Colas de prioridad. Implementación y algoritmos necesarios para su manejo. Aplicaciones.

6. Lista de 2 niveles. Optimización de parámetros para distintos casos. Direccionamiento directo. Skip list. Distribución pseudo-aleatoria de datos.

7. Teoría de grafos. Representación. Aplicación de los distintos conceptos teóricos a ejemplos. Algoritmos sobre grafos.

8. Técnicas de diseño de algoritmos. Análisis de los algoritmos vistos para cada técnica de diseño. Diseño de algoritmos usando las técnicas estudiadas.

9. Otros tipos de Árboles de búsqueda. Trie y Árboles Patricia. Árboles de búsqueda auto-ajustables.

Laboratorios:

En todos los laboratorios o prácticos de máquina se estudian situaciones reales a resolver y cuya solución se deba implementar en algún lenguaje de programación, teniendo en cuenta los conceptos que gradualmente se vayan incorporando en los contenidos teóricos y prácticos de aula. Para el desarrollo de los laboratorios se deberán conformar grupos de trabajo y se codificará con un lenguaje de programación determinado, de los vistos en materias anteriores, que se considere apropiado a los fines de lo que se desea evaluar en la práctica. El grupo deberá diseñar una solución adecuada al problema y deberá entregar el código fuente correspondiente, correctamente documentado, y un informe escrito que describa y justifique la solución planteada, los detalles de implementación, las elecciones realizadas, los costos involucrados, etc; además, deberá entregar los resultados obtenidos con la ejecución de su trabajo y el análisis de los mismos. Se espera así que los estudiantes realicen una autoevaluación sobre lo aprendido al asegurarse de que su solución funcione adecuadamente y logre el desempeño deseado, lo que les permite comprender más profundamente los conceptos teóricos y prácticos involucrados, facilitando la comparación de las soluciones propuestas frente a problemas similares.

Cada entrega posee su respectiva recuperación.

Además, los grupos realizarán una presentación oral detallando las decisiones de diseño tomadas, resultados obtenidos y especificaciones de implementación de alguno de los laboratorios aprobados.

A continuación se describe cómo se aborda y evalúa cada uno de los ejes transversales en el Plan de Trabajos Prácticos:

1. Identificación, formulación y resolución de problemas de informática

Abordaje: Se proponen situaciones de la vida real con complejidad gradual para que el estudiante identifique problemas y diseñe soluciones aplicando técnicas teóricas, ingenio e investigación. Los estudiantes deben debatir y analizar diferentes soluciones para seleccionar la más eficiente y efectiva.

Evaluación: Se realiza de forma formativa mediante la entrega de ejercicios representativos que son corregidos y comentados en clase. También se evalúa de forma sumativa a través del examen parcial y sus recuperaciones.

2. Concepción, diseño y desarrollo de proyectos de informática

Abordaje: Se utilizan prácticas de laboratorio para que los estudiantes diseñen y desarrollen proyectos integradores que implementan los conceptos teóricos. Los estudiantes deben diseñar una solución adecuada, codificarla y documentarla.

Evaluación: Evaluación sumativa mediante la entrega y aprobación de los proyectos diseñados, incluyendo el código fuente y los resultados obtenidos de su ejecución.

3. Gestión, planificación, ejecución y control de proyectos de informática

Abordaje: Las prácticas de laboratorio contribuyen a que los estudiantes planifiquen las diferentes etapas del desarrollo de un proyecto informático.

Evaluación: Se evalúa mediante el cumplimiento de las entregas en las fechas predeterminadas en el cronograma de la asignatura.

4. Utilización de técnicas y herramientas de aplicación en la informática

Abordaje: Los estudiantes emplean herramientas adecuadas como lenguajes de programación y entornos de desarrollo en sus prácticas de laboratorio. Se utiliza un lenguaje determinado, de los aprendidos en materias anteriores, que sea apropiado para los fines evaluativos.

Evaluación: Se verifica a través del análisis del código fuente entregado y de la justificación de las elecciones técnicas

realizadas en los informes.

5. Contribuir a la generación de desarrollos tecnológicos y/o innovaciones tecnológicas

Abordaje: Se promueve mediante el planteo de soluciones ingeniosas a problemáticas presentadas y la investigación de estructuras o algoritmos novedosos.

Evaluación: Se refleja en la calidad y originalidad de la solución planteada en los laboratorios y su posterior defensa.

6. Fundamentos para el desempeño en equipos de trabajo

Abordaje: Se fomenta el trabajo colaborativo a través de la formación de grupos de trabajo obligatorios para el desarrollo de los laboratorios.

Evaluación: Se evalúa mediante la observación del trabajo grupal y la participación de todos los integrantes en la descripción oral del proyecto.

7. Fundamentos para la comunicación efectiva

Abordaje: Se trabaja mediante la preparación y redacción de informes escritos (que describen y justifican la solución, costos y detalles de implementación).

Evaluación: Se evalúa de forma sumativa a través de una presentación detallada sobre las decisiones de diseño y resultados de los laboratorios, además de la corrección del informe escrito.

8. Fundamentos para la acción ética y responsable

Abordaje: Se enfoca en la responsabilidad profesional vinculada al cumplimiento de las normas y fechas límites propuestas en el curso.

Evaluación: Se demuestra a través del cumplimiento de los plazos establecidos al principio del curso, el respeto por el cronograma de entregas y el cumplimiento de las normas de trabajo para el grupo.

9. Fundamentos para el aprendizaje continuo

Abordaje: Se impulsa la curiosidad y el aprendizaje autónomo mediante material extra en la web (charlas y temas de interés) sobre algoritmos novedosos, incentivando al estudiante a indagar más allá de lo visto en clase. También se promueve la autoevaluación al verificar que sus soluciones funcionen adecuadamente.

Evaluación: Se fomenta mediante la investigación requerida para resolver los problemas planteados y el análisis de resultados en los laboratorios.

VIII - Regimen de Aprobación

Régimen para Estudiantes Regulares:

* Asistencia a práctico: 70%

* Asistencia a teoría: 70%

* Entregar los ejercicios requeridos de cada práctico de aula.

* Presentación y aprobación de los prácticos de laboratorio propuestos. Para ello deben entregar el código fuente y un informe en el que se justifique la solución propuesta y se brinden detalles del trabajo realizado. Los códigos fuente deben poder compilarse y resolver las situaciones planteadas. Cada práctico de laboratorio tendrá una posibilidad de recuperación para su aprobación.

* Aprobar la evaluación final o sus recuperaciones con el 70%. Se tomará una evaluación, la cual tendrá dos posibles instancias de recuperación.

- Modalidad de examen final: El examen final podrá ser oral y/o escrito.

- Examen Libre: No se admiten alumnos libres dado que los prácticos de máquina y aula se desarrollan de manera incremental desde el comienzo del cuatrimestre; por consiguiente, no es posible en un examen poder evaluar correctamente este proceso.

- No es una materia promocional.

IX - Bibliografía Básica

[1] Introduction to Algorithms. Cormen, Leiserson, Rivest, Stein. The MIT Press, 4a edición. ISBN: 026204630X. (2022).

(3a edición , 2009 - Ubicación en biblioteca: 519.254 C811)

[2] Algorithms. Jeff Erickson. Editorial independiente. ISBN-13: 978-1792644832. (2019). Disponible desde: <https://jeffe.cs.illinois.edu/teaching/algorithms/book/Algorithms-JeffE.pdf> (último acceso 16 de abril de 2026).

[3] The Algorithm Design Manual. Steven S. Skiena. Editorial Springer, 3a Edición. ISBN-13: 978-3030542559. (2020)

[4] Algorithms;. Christos Papadimitriou, Sanjoy Dasgupta. 1ST Edition, Editorial McGraw-Hill India. ISBN-13: 978-9355325525. (2023)

[5] Fundamentos de Algoritmia. Brassard, Gilles y Bratley, Paul. Prentice Hall, 1a edición. ISBN: 84-89660-00-X. (2000). Ubicación en Biblioteca: 004.021.B823f.

[6] Compared to What?: An Introduction To The Analysis Of Algorithms. Rawlins, Gregory. W. H. Freeman. ISBN: 071678243X. (1991)

[7] The Art of Computer Programming (Vol 1 Y 3). Knuth, Donald E. Addison-Wesley Professional. (1998). Vol 1: 3a edición, 1997, Vol 3: 2a edición, 1998, Ubicación en biblioteca: 681.3.06 K74.

[8] Algorithms in C (Parts 1-4). Sedgewick, Robert. Addison-Wesley Professional, 3a edición. ISBN 0-201-31452-5. (1997). Ubicación en Biblioteca: 004.422.63 S448a3 I.

[9] Graphes et Hypergraphes. Berge, C. North Holland. ISBN-13: 978-0444876034. (1989). Ubicación en biblioteca: 519.28 B495.

[10] An Introduction to the Analysis of Algorithms. Sedgewick, Robert and Flajolet, Philippe. Addison-Wesley Professional. ISBN: 020140009X. (1995). Ubicación en biblioteca: 004.021 S448.

[11] Algorithmics: Theory and Practice. Brassard, Gilles and Bratley, Paul. Prentice-Hall. ISBN-13: 978-0130232434. Primera edición, (1888). Ubicación en biblioteca: 519.681 B823.

[12] Data Structures and Program Design In C. Kruse, Robert L.; Leung, Bruce P.; Tondo, Clovis L. Prentice Hall, 2a edición. ISBN: 0-13-725649-3. (1996). Ubicación en biblioteca: 519.682.2681.3.06 K94.

[13] Apuntes de la Cátedra: Durante el dictado se entregarán apuntes confeccionados por la cátedra sobre algunos de los temas. Estos apuntes se revisan y se actualizan todos los años. *Repaso de Conjuntos, Relaciones y Funciones, *Introducción a las Estructuras de Datos. *Descripción Informática de Conjuntos. *Pertenencia en Conjuntos Computacionales. *Operaciones en Conjuntos Computacionales. *Evaluación de Algoritmos. *Teoría de Grafos (Parte I). *Teoría de Grafos (Parte II). *Árboles Binarios Ordenados. *Costos de Búsqueda en Árboles Binarios de Búsqueda. *Árboles de Expresión. *Lista de 2 Niveles. *Distribución Pseudo-aleatoria de Datos. *Diseño de Funciones de Enumeración. *Deducción de algunos esfuerzos para distribución pseudo- aleatoria de datos. *Diseño de Funciones de pseudo-azar. *Estructuras de Datos Aleatorias: Skip Lists. *Árboles Digitales: Trie y Patricia.

[14] Se cuenta con toda la bibliografía en formato pdf.

X - Bibliografía Complementaria

[1] COMPUTER ALGORITHMS: INTRODUCTION TO DESIGN AND ANALYSIS, Autor: Baase, Sara, ISBN: 0-201-06035-3, Addison Wesley, 3a edición, 1999, Ubicación en biblioteca: 519.682.4 B111c2

[2] DATA STRUCTURES AND THEIR ALGORITHMS, Autor: Lewis, Harry R., Denenberg, Larry, ISBN: 0-673-39736-X, Addison Wesley, 1991, Ubicación en biblioteca: 519.683.2 L674.

[3] ESTRUCTURA DE DATOS y ALGORITMOS, Autor: Mark Weiss, ISBN: 0-201-62571-7, Addison Wesley Longman, 2000.

[4] DATA STRUCTURES AND ALGORITHMS, Autor: Aho, Hopcroft, Ullman, ISBN: 0-201-00023-7, Addison Wesley, 1983, Ubicación en biblioteca: 519.683.2 A286

[5] CONCRETE MATHEMATICS, Autor: Graham, Ronald L., Knuth, Donald E. , Patashnik, Oren, ISBN: 0-201-55802-5, Addison-Wesley Professional, 2 edición, 1994, Ubicación en biblioteca: 511.333 G741c2

[6] THE DESIGN AND ANALYSIS OF COMPUTER ALGORITHMS, Autor: Aho, Hopcroft, Ullman, ISBN: 0-201-00029-6, Addison-Wesley, 1974, Ubicación en biblioteca: 519.683.2 A286

[7] DATA STRUCTURES & PROGRAM DESIGN, Autor: Kruse, Robert, ISBN: 0-132-08182-2, Prentice-Hall, 2a edición, 1994, Ubicación en biblioteca: 681.3.06 K94D2

[8] DATA STRUCTURES TECHNIQUES, Autor: Standish, T., ISBN: 0-201-07256-4, Addison-Wesley, 1980, Ubicación en biblioteca: 681.3.0651.S785

[9] COMPUTER ALGORITHMS: KEY SEARCH STRATEGIES, IEEE Computer Society Technology Series, ISBN: 0-818-69123-9, IEEE Computer Society, 1990, Ubicación en biblioteca: 519.681.5519.878681.3.06 A638

[10] MATHEMATICS FOR THE ANALYSIS OF ALGORITHMS, Autor: Greene, Daniel , Knuth, Donald, ISBN: 0-8176-3515-7 (Birkhäuser), ISBN: 3-7643-3515-7 (Boston-Basel-Berlín), Birkhäuser Boston; 3a edición Rev., 2007.

- [11] PUGH, WILLIAM: Skip Lists: A Probabilistic Alternative to Balanced Trees. Communications of the ACM, 33,1990, pág. 668-676.
- [12] LEWIS, T y COOK, C: Hashing for Dynamic and Static Internal Tables. IEEE Comp. Oct.1988.
- [13] LARSON, P: Dynamic Hash Tables. C. ACM, Vol 31 N0 4, Abril 1988.
- [14] SLEATOR D. D. y TARJAN R. E.: Self-adjusting binary search trees Journal of ACM, ISSN 0004-5411, Vol 32, Nro.3, pág. 652-686, Julio 1985.

XI - Resumen de Objetivos

Al finalizar el curso se pretende que el alumno sea capaz de:

- Manejar los conceptos que involucran el diseño de estructuras de datos.
- Conocer los principales algoritmos y estructuras de datos y el análisis de su desempeño.
- Analizar y diseñar algoritmos.
- Frente a una aplicación o problema particular, poder brindar una solución eficiente utilizando los conceptos vistos sobre diseño de estructuras de datos y algoritmos, introducir herramientas de diseño de algoritmos y la ingeniería algorítmica como selección de las estructuras de datos y de las técnicas algorítmicas más adecuadas y, además, utilizar el análisis de los algoritmos para evaluar y justificar la eficiencia de la solución elegida.

XII - Resumen del Programa

1. Explicitación del objetivo de la materia: Las estructuras. Dato e información. Datos de un problema de estructuras de datos. Algoritmos.
2. Evaluación de algoritmos: Planteo. Problema. Medidas en tiempo y espacio. Familias de problemas. Notaciones asintóticas. Clases de Complejidad.
3. Listas, Pilas y Colas: Listas vinculadas y secuenciales. Búsqueda binaria. Lista Invertida. Lista de 2 niveles. Pilas y colas. Skip Lists. Operaciones sobre cada tipo de estructura. Aplicaciones.
4. Propiedades, demostraciones y representación de grafos: Definiciones. Propiedades. Representaciones básicas de grafos. Algoritmos sobre grafos.
5. Árboles computacionales: Definición. Representación computacional. Árboles Binarios de Búsqueda. Árboles balanceados en altura. Parva. Operaciones básicas sobre cada tipo de árbol. Aplicaciones.
6. Distribución pseudo-aleatoria de datos: Motivación. El problema del rebalse. Distintas propuestas para su manejo. Direccionamiento Directo. Aplicaciones.
7. Técnicas de diseño de algoritmos: Técnicas ávidas, programación dinámica, dividir para vencer. Ejemplos de aplicación de las distintas técnicas.

XIII - Imprevistos

La modalidad de dictado es PRESENCIAL, tanto para las clases teóricas como para las prácticas. En caso de presentarse alguna dificultad con respecto al modo de dictado, por favor comunicarse de inmediato con la cátedra.

XIV - Otros

Contacto con la cátedra:

Por mail: eda@unsl.edu.ar

- Por formulario electrónico a través de la página web: <http://eda.dirinfo.unsl.edu.ar/>

- Personalmente: Segundo Bloque - box 7, box 24 - Primer piso.

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	