



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

FORMULARIO PARA LA PRESENTACIÓN DE PROGRAMAS DE ASIGNATURAS

Año Lectivo: 2024

UNIVERSIDAD NACIONAL DE RÍO CUARTO
FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES

DEPARTAMENTO DE COMPUTACIÓN

CARRERA/S: Analista en Computación, Licenciatura en Ciencias de la Computación

PLAN DE ESTUDIOS: 1999, versión 1

ASIGNATURA: Diseño de Algoritmos y Algoritmos II **CÓDIGO:** 3326/3302

MODALIDAD DE CURSADO: Presencial

DOCENTE RESPONSABLE: Valeria Bengolea

EQUIPO DOCENTE: Valeria Bengolea, Profesor Adjunto Dedicación Exclusiva (Interino)

Sonia Permigiani, JTP Dedicación Exclusiva. Gastón Scilingo, Ayudante de Primera Dedicación Exclusiva.

RÉGIMEN DE LA ASIGNATURA: Cuatrimestral

UBICACIÓN EN EL PLAN DE ESTUDIO: 1er. Cuatrimestre de 3er. año

RÉGIMEN DE CORRELATIVIDADES:

Diseño de Algoritmos, para cursar y rendir respectivamente

Para cursar

<i>Aprobada</i>	<i>Regular</i>
---	Estructura de Datos y Algoritmos (3325)

Para rendir

<i>Aprobada</i>
Estructura de Datos y Algoritmos (3325)

Algoritmos II, para cursar y rendir respectivamente



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

Para cursar

<i>Aprobada</i>	<i>Regular</i>
---	Algoritmos I (3301)

Para rendir

<i>Aprobada</i>
Algoritmos I (3301)

CARÁCTER DE LA ASIGNATURA: Obligatoria

CARGA HORARIA TOTAL: 112 horas

Teóricas:	56 hs	Prácticas:	56 hs	Teóricas -Prácticas:	... hs	Laboratorio:	... hs
------------------	--------------	-------------------	--------------	---------------------------------	---------------	---------------------	---------------

CARGA HORARIA SEMANAL: 8 horas (según el plan de estudio vigente)

Teóricas:	4 hs	Prácticas:	4 hs	Teóricas -Prácticas:	... hs	Laboratorio:	... hs
------------------	-------------	-------------------	-------------	---------------------------------	---------------	---------------------	---------------



1. CONTEXTUALIZACIÓN DE LA ASIGNATURA

Diseño de Algoritmos y Algoritmos II son asignaturas obligatorias de los planes de estudios de las carreras de Analista en Computación y Licenciatura en Ciencias de la Computación, respectivamente. Estas asignaturas, que se dictan en el tercer año de las respectivas carreras, constituyen la última materia del trayecto de algorítmica en las mencionadas carreras, que además constituye el núcleo central de las mismas. Estas materias dan un cierre al trayecto, completando así los conocimientos fundamentales de algorítmica: elementos básicos de la programación y la algorítmica (Intr. a la Algorítmica y Programación), técnicas de programación más avanzadas y herramientas para construir programas correctos (Programación Avanzada), la representación de datos en la programación (Estructuras de Datos y Algoritmos, Algoritmos I), y técnicas y estrategias de diseño de algoritmos (Diseño de Algoritmos/Algoritmos II).

2. OBJETIVOS PROPUESTOS

El objetivo esencial de las materias es lograr que los alumnos se familiaricen con las técnicas y estrategias de diseño de algoritmos más importantes, que sepan aplicarlas en la práctica, aprovechando los mecanismos disponibles para su implementación en lenguajes de programación particulares. Se espera sin embargo que los alumnos aprendan las técnicas de diseño de algoritmos fundamentales de manera independiente de lenguajes de programación específicos. Se espera además que los alumnos conozcan las diferentes clases de problemas, de acuerdo a su complejidad.

Un objetivo particular de estas materias es conseguir integrar los conocimientos obtenidos por los alumnos en todas las materias anteriores del área de algoritmos (Introducción a la Algorítmica y Programación, Programación Avanzada, Estructuras de Datos y Algoritmos, Algoritmos I). En esta materia se hará un fuerte énfasis en el uso adecuado de tipos abstractos de datos y su definición, y de anotaciones de programas tales como invariantes y contratos en términos de pre- y post-condición. Se utilizarán también las estructuras de datos más importantes, vistas en materias anteriores del área de algoritmos.

3. EJES TEMÁTICOS ESTRUCTURANTES DE LA ASIGNATURA Y ESPECIFICACIÓN DE CONTENIDOS

3.1. Contenidos mínimos

Análisis funcional de programas. Análisis de tiempo de ejecución de programas. La construcción de soluciones algorítmicas a problemas. Estrategias de diseño de algoritmos. Fuerza bruta. Decrease & Conquer. Divide & Conquer. Programación Dinámica. Algoritmos Greedy. Búsqueda no informada, búsqueda informada y búsqueda adversaria. Minimax. Poda alfa-beta. Algoritmos bioinspirados. Algoritmos Genéticos y redes neuronales. Introducción a la computabilidad y la complejidad. P vs NP. NP-Complejidad y NP-hardness. Modelos formales abstractos de software. Especificaciones formales. Expresividad, análisis y trade-offs.

3.2. Ejes temáticos o unidades

Revisión de conceptos previos. Anotaciones en programas. La importancia de las especificaciones. Pre- y post-condición. Invariantes de ciclos. Funciones variantes. Pruebas de



corrección de programas. Técnicas de transformación de programas. Pruebas de terminación de programas. Estructuras de datos fundamentales. Tipos abstractos de datos y su implementación. Análisis de algoritmos. Concepto de tiempo de ejecución. Reglas de análisis de programas imperativos y recursivos.

Introducción a la resolución algorítmica de problemas. Técnicas básicas de diseño de algoritmos. Fuerza bruta. Algoritmos basados directamente en definiciones de problemas. Búsqueda exhaustiva. Limitaciones de la técnica. Divide & Conquer. Relación entre Divide & Conquer y la recursión. Técnicas más específicas basadas en Divide & Conquer: Decrease & Conquer y Transform & Conquer. Algunos algoritmos importantes basados en estas estrategias. Aplicación de la técnica en la práctica.

Técnicas avanzadas de diseño de algoritmos. Programación Dinámica. La programación dinámica como una optimización a soluciones Divide & Conquer. Algunos algoritmos importantes que siguen esta estrategia. Aplicación de la técnica en la práctica. Greedy. Algunos ejemplos de Greedy como una técnica completa. Greedy como una estrategia de aproximación a soluciones. Algunos algoritmos importantes que siguen esta estrategia. Aplicación de la técnica en la práctica. Backtracking. Backtracking y su relación con la búsqueda exhaustiva. Búsqueda local. Branch & Bound. Backtracking en problemas de búsqueda convencionales. Búsqueda en problemas con adversarios. Su aplicación a juegos. Poda alfa-beta del árbol de juego. Aplicación de la técnica en la práctica.

Otras técnicas de diseño de algoritmos. Enfoques algorítmicos al aprendizaje. Introducción a las redes neuronales. La evolución como motivadora en la construcción de algoritmos: los algoritmos genéticos. Los algoritmos genéticos y su relación con la búsqueda heurística.

Clases de problemas de acuerdo a su complejidad. La relación entre complejidad espacial y temporal. P y NP. NP-Completeness. Problemas NP-hard. Ejemplos. Soluciones mediante aproximación para problemas NP-hard.

La importancia de las especificaciones de software. Especificaciones de requisitos y especificaciones de diseño. Lenguajes informales vs. lenguajes formales de especificación. La lógica en los lenguajes formales de especificaciones. Limitaciones en poder expresivo. Análisis de especificaciones mediante deducción. Ventajas y desventajas de la deducción como mecanismo de análisis. Técnicas de análisis automático. Limitaciones de las técnicas de análisis automático.

4. ACTIVIDADES A DESARROLLAR

Como una forma de simplificar las tareas de los alumnos, se elegirán como lenguajes para la implementación de algoritmos el lenguaje funcional Haskell y el lenguaje orientado a objetos Java. Los alumnos ya tienen experiencia en estos lenguajes, adquiridas en materias anteriores. Con el fin de afianzar las técnicas de diseño de algoritmos estudiadas en la asignatura, se pedirá en algunos casos que se implementen soluciones en dos paradigmas diferentes, el funcional y el imperativo/orientado a objetos.



Los dos trabajos prácticos obligatorios, cuya aprobación es requisito para la regularidad, tienen por objetivo lograr que los alumnos puedan aplicar la teoría aprendida en la resolución algorítmica de problemas mediante la aplicación de las distintas técnicas de diseño de algoritmos estudiadas. Además, permiten integrar los contenidos de esta asignatura a muchos otros aprendidos en otras materias, principalmente de programación y lenguajes (evaluando las diferencias entre los distintos paradigmas de programación, por ejemplo), y afianzar las capacidades adquiridas en las materias de lógica y matemática discreta. Se intentará utilizar ejemplos y problemas interesantes, en los cuales las soluciones más adecuadas sean difíciles de reconocer, intentando estimular al alumnado.

Se fomentará la lectura de material adicional y la auto organización de los alumnos en sus actividades. Además, se dejará en manos de los alumnos la instalación y manejo de las herramientas de software utilizadas en la asignatura, como una manera de estimular la práctica en cuestiones más técnicas (no necesariamente ligadas a los tópicos que la asignatura abarca), y la experiencia en la utilización de herramientas nuevas.

CLASES TEÓRICAS: 4 horas semanales

CLASES PRÁCTICAS: 4 horas semanales

CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO: Todas las clases prácticas serán de laboratorio.

5. PROGRAMAS Y/O PROYECTOS PEDAGÓGICOS INNOVADORES E INCLUSIVOS

No existen actividades de la asignatura enmarcadas en programas y/o proyectos pedagógicos.

6. CRONOGRAMA TENTATIVO DE CLASES E INSTANCIAS EVALUATIVAS

Semanas	días	Actividad Teórica	días	Actividad Práctica	Evaluaciones
1	11 al 15 de marzo	Introducción al Diseño de Algoritmos. Revisión de conceptos previos de algoritmos y representación de datos	14 al 19 de marzo	Repaso de conceptos previos de algoritmos y representación de datos	
2	18 al 22 de marzo	Técnica de Diseño Fuerza Bruta	21 al 25 de marzo	Práctico sobre fuerza bruta en la resolución de problemas computacionales	
3	28 al 31 de Marzo	Técnica de Diseño Divide & Conquer	28 al 31 de	Práctico sobre Divide & Conquer en la resolución de	



	y 1 de Abril		Marzo y 1 de Abril	problemas computacionales	
4	4 al 8 de abril	Técnica de Diseño Decrease & Conquer	4 al 8 de abril	Práctico sobre Decrease & Conquer en la resolución de problemas computacionales	
5	11 al 15 de abril	Técnica de Diseño Programación Dinámica	11 al 15 de abril	Práctico sobre Programación Dinámica en la resolución de problemas computacionales	
6	18 al 22 de abril	Técnica de Diseño Greedy	18 al 22 de abril	Práctico sobre Programación Dinámica en la resolución de problemas computacionales	19/4 Primer Parcial
7	25 al 29 de abril	Técnica de Diseño Búsqueda	25 al 29 de abril	Práctico sobre Greedy en la resolución de problemas computacionales	22/4. Entrega de enunciados del trabajo práctico obligatorio No. 1
8	2 al 6 de Mayo	Búsqueda Informada, Búsqueda en Problemas con adversarios	2 al 6 de Mayo	Práctico sobre Búsqueda en la resolución de problemas computacionales	3/5 Recuperatorio primer parcial 5/5 Entrega de los enunciados del trabajo práctico obligatorio No. 1
9	09 al 13 de Mayo	Técnicas de aprendizaje automático: redes neuronales	09 al 13 de Mayo	Práctico sobre Búsqueda informada y adversaria en la resolución de problemas computacionales	
10	16 al 20 de Mayo	Algoritmos Genéticos	16 al 20 de Mayo	Práctico sobre Búsqueda informada y adversaria en la resolución de problemas computacionales	16/5 Entrega de los enunciados del trabajo práctico obligatorio No. 2
11	23 al 27 de mayo	Revisión y comparación de técnicas estudiadas	23 al 27 de mayo	Práctico sobre Redes Neuronales en la resolución de problemas computacionales	
12	30 de mayo al 3 de Junio	Clases de problemas de acuerdo a su complejidad	30 de mayo al 3 de Junio	Práctico sobre Algoritmos Genéticos en la resolución de problemas computacionales	30/5 Entrega de las resoluciones del trabajo práctico obligatorio No. 2 3/6 Segundo parcial
13	6 al 10 de Junio	Introducción a los modelos formales de software	6 al 10 de Junio	Práctica sobre modelado formal en Alloy	Defensas orales individuales de proyectos



14	13 al 17 de Junio	Análisis automático de modelos formales de software	13 al 17 de Junio	Práctica sobre análisis automático de modelos Alloy	14/6 Recuperatorio segundo parcial
----	-------------------	---	-------------------	---	------------------------------------

7. BIBLIOGRAFÍA

7.1. Bibliografía obligatoria y de consulta

Obligatoria

A. Levitin, *"Introduction to the Design and Analysis of Algorithms"*, 3rd. Edition, Addison-Wesley, 2014.

S. Rusell & P. Norvig, *Artificial Intelligence. A Modern Approach*, Pearson, 2016.

D. Jackson, *"Software Abstractions"*, Revised Edition, MIT Press, 2012.

De consulta

A. Aho, J. Hopcroft y J. Ullman, *"Data Structures and Algorithms"*, Addison-Wesley, 1983.

T. Cormen, C. Leiserson, R. Rivest y C. Stein, *"Introduction to Algorithms"*, Third Edition, MIT Press, 2009.

P. Hudak, J. Peterson y J. Fasel, *"A Gentle Introduction to Haskell"*, 2000.

U. Manber, *"Introduction to Algorithms, A Creative Approach"*, Addison-Wesley, 1989.

S. Baase, *"Computer Algorithms, Introduction to Design and Analysis"*, second edition, Addison-Wesley, 1988.

R. Bird, *"Introduction to Functional Programming using Haskell"*, second edition, Prentice Hall, 1998.

B. Meyer, *"Object Oriented Software Construction"* second edition, Addison-Wesley, 2000.

B. Liskov, *"Program Development in Java, Abstraction, Specification and Object-Oriented Design"*, Addison-Wesley, 2001.

7.2. Otros: materiales audiovisuales, enlaces, otros.

Slack (comunicaciones, material, etc.)

Google Classroom (comunicaciones, material, videos grabados de las clases en vivo)

Github (plantillas de programación con tests para autocorrección, repositorio de código para interacción entre estudiantes y docentes)

Sistema de issues (comunicación y reporte de errores, etc.)

Screen (asistencia remota en problemas de compilación o generales de programación).



8. DÍA Y HORARIOS DE CLASES

Clases Prácticas:

Comisión 1: Lunes de 10 a 12 y Jueves de 8 a 10

Comisión 2: Lunes de 14 a 16 y Viernes de 14 a 16

Clases Teóricas: Martes de 18 a 20 y Jueves de 14 a 16

9. DÍA Y HORARIO DE CLASES DE CONSULTAS

La asignatura no tiene horarios fijos de consultas. Se fijan a pedido de los alumnos, bajo demanda, en horarios acordados cada semana, cuando corresponda o sea necesario.

10. REQUISITOS PARA OBTENER LA REGULARIDAD Y LA PROMOCIÓN

CONDICIONES DE REGULARIDAD: Las condiciones para la regularidad de la asignatura son las siguientes: Aprobación de los dos exámenes parciales y dos trabajos prácticos obligatorios con sus defensas (nota mayor o igual a 5).

CONDICIONES DE PROMOCIÓN: Aprobación de los dos exámenes parciales y dos trabajos prácticos obligatorios (con sus instancias de defensa) con 7 como nota promedio en estas instancias de evaluación, La nota de promoción se calculará como un promedio ponderado de las notas de trabajos prácticos y parciales.

11. CARACTERÍSTICAS, MODALIDAD Y CRITERIOS DE LAS INSTANCIAS EVALUATIVAS

- **Evaluaciones Parciales:** La materia tendrá dos exámenes parciales, cada uno con una recuperación. Los exámenes parciales serán teórico-prácticos, y abarcarán la primera y la segunda mitad de los contenidos de la asignatura, respectivamente. La evaluación del alumnado será complementada con trabajos prácticos obligatorios (ver cronograma).
- **Evaluación Final:** El examen final para alumnos regulares se llevará a cabo mediante evaluación oral, si el número de alumnos evaluados así lo permite. Abarcará la totalidad de los contenidos de la asignatura, verificando que los alumnos hayan adquirido los conocimientos teóricos y puedan aplicarlos en casos concretos en la práctica.

Por otra parte, el examen final para alumnos libres estará compuesto por una primera instancia escrita, en la cual se evaluarán las capacidades de los alumnos para diseñar algoritmos utilizando las técnicas de diseño de algoritmos aprendidas en la asignatura. La segunda instancia del examen final para alumnos libres será similar al examen final para alumnos regulares.



Universidad Nacional de Río Cuarto
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

Firma Profesor/a Responsable

Firma Secretario/a Académico/a