



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

FORMULARIO PARA LA PRESENTACIÓN DE PROGRAMAS DE ASIGNATURAS

Año Lectivo: 2025

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

CARRERA/S: Licenciatura en Ciencias de la Computación

PLAN DE ESTUDIOS: 2024.

ASIGNATURA: Algoritmos y Estructuras de Datos II

CÓDIGO: 3380.

MODALIDAD DE CURSADO: Presencial

DOCENTE RESPONSABLE: Pablo Ponzio, Profesor Adjunto, Dedicación Exclusiva.

EQUIPO DOCENTE: Lic. Sonia Permigiani, Lic. Gastón Scilingo, Prof. Ernesto Cerdá.

RÉGIMEN DE LA ASIGNATURA: Cuatrimestral

UBICACIÓN EN EL PLAN DE ESTUDIO: 2do Año – 1er Cuatrimestre

RÉGIMEN DE CORRELATIVIDADES:

Para cursar:

- Asignaturas aprobadas: Introducción a los Algoritmos (3375).
- Asignaturas regulares: Algoritmos y Estructuras de Datos I (3378), Matemática Discreta (3379).

Para rendir:

- Asignaturas aprobadas: Introducción a los Algoritmos (3375). Algoritmos y Estructuras de Datos I (3378), Matemática Discreta (3379).

CARÁCTER DE LA ASIGNATURA: Obligatoria

CARGA HORARIA TOTAL: 112 horas horas

Teóricas:	56 hs	Prácticas:	28hs	Teóricas -Prácticas: hs	Laboratorio:	28 hs
------------------	--------------	-------------------	-------------	---------------------------------	----------------	---------------------	--------------

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

Teóricas:	4 hs	Prácticas:	2 hs	Teóricas -Prácticas: hs	Laboratorio:	2 hs
------------------	-------------	-------------------	-------------	---------------------------------	----------------	---------------------	-------------



1. CONTEXTUALIZACIÓN DE LA ASIGNATURA

La asignatura está en el segundo año de las carreras citadas, el material abordado en la materia provee las herramientas básicas para poder abordar los cursos más avanzados de computación.

2. OBJETIVOS PROPUESTOS

El objetivo esencial de la materia es lograr que los alumnos se familiaricen con las técnicas más importantes de representación de datos, que comprendan los mecanismos de los lenguajes de programación modernos para su implementación, y sepan identificar los problemas prácticos en los cuales pueden emplearse cada una de las representaciones de datos estudiadas.

Otro objetivo relevante de la materia es que los estudiantes aprendan herramientas para analizar el tiempo de ejecución de los programas, y comprendan la importancia de implementar algoritmos eficientes en la práctica.

Finalmente, se espera que los alumnos se familiaricen con los conceptos fundamentales de la programación orientada a objetos, y realicen sus primeras actividades prácticas en relación al diseño y la implementación de programas orientados a objetos.

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

3.1. Contenidos mínimos

Contenidos mínimos: Medidas de tiempo de ejecución de programas. Análisis de tiempo de ejecución en el peor caso de programas iterativos y recursivos. Resolución de ecuaciones de recurrencias. Tasas de crecimiento de funciones de tiempo de ejecución. Algoritmos de ordenamiento y búsqueda avanzados. Representación de datos: funciones de abstracción e invariantes de representación. Especificación de tipos abstractos de datos. Formalización de tipos abstractos de datos como estructuras algebraicas. Encapsulamiento y abstracción. Estructuras de datos avanzadas: árboles binarios de búsqueda, árboles balanceados y tablas hash. Grafos: representación y algoritmos fundamentales.

Objetivos de aprendizaje fundamentales: Con el desarrollo de esta asignatura se espera que el estudiante pueda apropiarse de los conocimientos, y desarrollar habilidades y aptitudes necesarias para: Explicar el uso de las notaciones O grande, ω y θ , para describir la cantidad de trabajo asociado a un algoritmo. Utilizar las notaciones O grande, ω y θ , para dar asintóticas superiores e inferiores, y cotas ajustadas, para el tiempo y el espacio correspondientes a la complejidad de los algoritmos. Determinar la complejidad en tiempo y espacio de algoritmos simples. Deducir relaciones de recurrencia que permitan describir la complejidad de tiempo de algoritmos recursivos. Resolver relaciones de recurrencia elementales. Razonar sobre la correcta implementación de tipos abstractos de datos. Utilizar adecuadamente estructuras de datos simples y avanzadas en la implementación de tipos de datos. Comprender las características de diferentes alternativas en la implementación de tipos de datos, y poder elegir la que mejor se ajuste a cada instancia de aplicación.



3.2. Ejes temáticos o unidades

Introducción a la programación orientada a objetos. Clases y objetos. Relaciones de agregación y uso. Herencia y polimorfismo. Genericidad. Introducción al diseño orientado a objetos. Conceptos de cohesión y acoplamiento. Principio abierto-cerrado.

La noción de tipos abstractos de datos. Representación de datos. Funciones de abstracción e invariantes de representación. Implementación de tipos abstractos de datos en lenguajes orientados a objetos. Especificaciones básicas y jerárquica. Corrección y completitud. Revisión de implementaciones clásicas de tipos abstractos de datos estructurados lineales. Listas, pilas y colas, y sus implementaciones clásicas. Comparación de las implementaciones clásicas de tipos abstractos de datos lineales.

Medidas de tiempo de ejecución de programas. Análisis de tiempo de ejecución en caso promedio y peor caso. Análisis de tiempo de ejecución de algoritmos iterativos y recursivos. Resolución de ecuaciones de recurrencias. Tasas de crecimiento de funciones de tiempo de ejecución. Análisis empírico del tiempo de ejecución de programas.

Tipos abstractos de datos más avanzados. Los tipos abstractos de datos Conjunto y Diccionario. Diccionarios y su implementación. Árboles como estructuras de datos. Recorridos sobre árboles (inorder, preorder, postorder). Árboles binarios de búsqueda. AVLs y árboles 2-3. Comparación de las implementaciones clásicas de diccionarios de acuerdo a la eficiencia de sus operaciones. Estructuras de datos avanzadas. Heaps y sus aplicaciones. Hashing.

El tipo abstracto de datos Grafo. Algoritmos de recorrido/visita de grafos: recorridos "primero en profundidad" y "primero a lo ancho". Implementaciones clásicas de grafos (matrices de adyacencias y listas de adyacencias). Grafos dirigidos y no dirigidos. Grafos con costos. Aplicaciones del tipo abstracto de datos Grafo. Algoritmos clásicos sobre grafos. Cálculo de camino de mínimo costo: Algoritmos de Dijkstra y Floyd. Clausura transitiva: Algoritmo de Warshall. El problema del árbol abarcador de costo mínimo: Algoritmos de Prim y Kruskal.

4. ACTIVIDADES A DESARROLLAR

CLASES TEÓRICAS: 4 horas semanales de clases teóricas, divididas en dos clases de dos horas cada una; en estas clases se les enseña a los estudiantes la teoría básica de la materia.

CLASES PRÁCTICAS: 2 horas semanales de clases prácticas y de laboratorio, estas clases permiten a los estudiantes afianzar los conocimientos teóricos mediante la aplicación de la teoría a problemas medianamente complejos.

CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO: 2 horas semanales de laboratorios, en estas clases los estudiantes implementan en lenguajes de programación los conceptos vistos en los teóricos y prácticos.

HORARIOS DE CONSULTA: Cada docente provee horas de consulta adicionales para los prácticos, teóricos y laboratorios, estas son dos horas semanales adicionales.



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

5. PROGRAMAS Y/O PROYECTOS PEDAGÓGICOS INNOVADORES E INCLUSIVOS

6. CRONOGRAMA TENTATIVO DE CLASES E INSTANCIAS EVALUATIVAS

Semana	2025	Exámenes	Día	Mes
1	Escuela RIO		11	3
1	Escuela RIO		12	3
1	Escuela RIO		13	3
1	Escuela RIO		14	3
2	Teórico: Introducción a la materia		18	3
2	Teórico: Especificación e implementación de tipos de datos en Java		19	3
2	Práctico: Especificación e implementación de tipos de datos en Java		20	3
2	Práctico: Especificación e implementación de tipos de datos en Java		21	3
3	Teórico: Testing y debugging		25	3
3	Teórico: Abstracción y Modularización		26	3
3	Práctico: Testing - Abstracción y Modularización		27	3
3	Práctico: Testing - Abstracción y Modularización		28	3
4	Teórico: Uso de colecciones I (listas)		1	4
4	Feriado		2	4
4	Práctico: Uso de colecciones		3	4
4	Práctico: Uso de colecciones		4	4
5	Teórico: Uso de colecciones II (sets, maps)		8	4
5	Teórico: Herencia y polimorfismo - Manejo de errores		9	4
5	Práctico: Uso de colecciones		10	4
5	Práctico: Herencia y manejo de errores		11	4
6	Teórico: TADs básicos en Java - Función de abstracción - Invariante de representación		15	4
6	Teórico: Haskell - TADs en Haskell		16	4
6	Semana santa		17	4
6	Semana santa		18	4
7	Teórico: Teoría de TADs: Álgebras - Función de abstracción		22	4
7	Teórico: Teoría de TADs: Invariante de representación		23	4
7	Práctico: Haskell - TADs en Haskell		24	4
7	Práctico: Teoría de TADs		25	4
8	Teórico: Algoritmos avanzados de ordenamiento y búsqueda		29	4
8	Teórico: Tiempo de Ejecución Notación Big-Oh		30	4
8	Feriado		1	5
8	Feriado puente		2	5
9	Teórico: Tiempo de Ejecución para Programas recursivos		6	5



9	Teórico: Árboles binarios, ABBs		7	5
9	Práctico: Ordenamiento y tiempo de ejecución		8	5
9	Práctico: Tiempo de ejecución para programas recursivos	Primer parcial	9	5
10	Teórico: AVLs		13	5
10	Teórico: Árboles 2-3		14	5
10	Práctico: Árboles binarios, ABBs		15	5
10	Práctico: AVLs, Árboles 2-3	Recup. primer parcial	16	5
11	Teórico: Heaps		20	5
11	Teórico: Hashing		21	5
11	Práctico: Heaps		22	5
11	Práctico: Hashing		23	5
12	Teórico: Grafos: Implementaciones Básicas		27	5
12	Teórico: Grafos: Algoritmos de recorridos	Presentación TP	28	5
12	Práctico: Grafos		29	5
12	Práctico: Grafos	Segundo parcial	30	5
13	Teórico: Algoritmos de Warshall y Algoritmo de Dijkstra		3	6
13	Teórico: Spanning Trees, Algoritmo de Kruskal y Algoritmo de Prim		4	6
13	Práctico: Algoritmos sobre grafos		5	6
13	Práctico: Algoritmos sobre grafos	Recup. segundo parcial	6	6
14	Consultas TP		10	6
14	Consultas TP		11	6
14	Consultas TP		12	6
14		Coloquio promoción	13	6
15		Entrega y Defensas TP	17	6

*Teóricos, teóricos-prácticos, trabajos de laboratorios, salidas a campo, seminarios, talleres, coloquios, instancias evaluativas, consultas grupales y/o individuales, otras.

7. BIBLIOGRAFÍA

7.1. Bibliografía obligatoria y de consulta

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

D. Barnes, M. Kölling. "Objects First with Java A Practical Introduction using BlueJ". Sixth Edition. Pearson. 2016.

Robert Sedgewick, Kevin Wayne. "Algorithms (Fourth edition)". Addison-Wesley 2016.



T.Cormen, C. Leiserson, R.Rivest. C.Stein, "Introduction to Algorithms, Fourth Edition", The MIT Press, 2022.

Mark Allen Weiss, " Data Structures and Algorithm Analysis in Java", Addison-Wesley, 2010.

7.2. Plataformas/herramientas virtuales; materiales audiovisuales, enlaces, otros.

Se utilizará Google Classroom como repositorio de materiales, apuntes, presentaciones, videos, etc. Se utilizará la aplicación de mensajería instantánea Slack, y los repositorios para código de programación GitHub. Por el cual, se dejará disponible código de programación para que los alumnos puedan acceder al mismo, y modificarlo.

8. DÍA Y HORARIOS DE CLASES

Clases Teóricas: Martes de 16 a 18 hs. y Miércoles de 14 a 16 hs.

Clases Prácticas/Laboratorio:

Comisión 1: Martes de 10 a 12 hs. y Jueves de 8 a 10 hs.

Comisión 2: Miércoles de 12 a 14 hs. y Jueves de 14 a 16 hs.

Comisión 3: Martes de 8 a 10 hs. y Jueves de 12 a 14 hs.

9. DÍA Y HORARIO DE CLASES DE CONSULTAS

Serán acordadas con los estudiantes.

10. REQUISITOS PARA OBTENER LA REGULARIDAD Y LA PROMOCIÓN

Para regularizar la materia se deben aprobar los dos parciales o sus respectivos recuperatorios. Además, se deberá aprobar el trabajo práctico dado durante la materia.

Para promocionar la materia se deben aprobar los dos parciales o sus respectivos recuperatorios con nota mayor o igual a 7, el trabajo práctico con nota mayor o igual a 7, y el coloquio de promoción con nota mayor o igual a 7.

11. CARACTERÍSTICAS, MODALIDAD Y CRITERIOS DE LAS INSTANCIAS EVALUATIVAS

Los parciales se desarrollarán de manera presencial. De la misma forma se desarrollarán los recuperatorios, y el coloquio para la promoción. Además, se dará un trabajo práctico que los estudiantes deberán entregar en los plazos establecidos (15 días).

El examen final para alumnos regulares se llevará a cabo de manera escrita y presencial. El examen final abarca la totalidad de los contenidos de la asignatura.

La materia se puede rendir como libre, para para ello se solicitará un proyecto de programación que deberá ser entregado antes del final. Una vez aprobado el proyecto, se tomará un examen escrito que abarca la totalidad de los contenidos de la asignatura.



CREER...CREAR...CRECER

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

Firma Profesor/a Responsable
Pablo Ponzo

Firma Secretario/a Académico/a