



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 y Profesorado en Ciencias de la Computación (Estructuras de Datos y Algoritmos), Licenciatura en Ciencias de la Computación (Algoritmos I)

PLAN DE ESTUDIOS: 1999

ASIGNATURA: Estructura de Datos y Algoritmos/Algoritmos I

CÓDIGO: 3325/3301

MODALIDAD DE CURSADO: Presencial

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

EQUIPO DOCENTE: Lic. Sonia Permigiani, Lic. Gastón Scilingo, Lic. Agustín Borda

RÉGIMEN DE LA ASIGNATURA: Cuatrimestral

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

RÉGIMEN DE CORRELATIVIDADES:

Asignaturas aprobadas: Introducción a la algorítmica y Programación

Asignaturas regulares: Programación Avanzada, Álgebra (Algoritmos I)

CARÁCTER DE LA ASIGNATURA: Obligatoria

CARGA HORARIA TOTAL: 112 horas horas

<b>Teóricas:</b>	<b>56 hs</b>	<b>Prácticas:</b>	<b>28hs</b>	<b>Teóricas -Prácticas:</b>	<b>... hs</b>	<b>Laboratorio:</b>	<b>28 hs</b>
------------------	--------------	-------------------	-------------	---------------------------------	---------------	---------------------	--------------

**CARGA HORARIA SEMANAL:** 8 horas

<b>Teóricas:</b>	<b>4 hs</b>	<b>Prácticas:</b>	<b>2 hs</b>	<b>Teóricas -Prácticas:</b>	<b>... hs</b>	<b>Laboratorio:</b>	<b>2 hs</b>
------------------	-------------	-------------------	-------------	---------------------------------	---------------	---------------------	-------------



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

## 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.

Un objetivo importante de la asignatura es 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

Introducción a la programación orientada a objetos. La noción de tipos abstractos de datos. Revisión de implementaciones clásicas de tipos abstractos de datos estructurados lineales. Listas, pilas y colas, y sus implementaciones clásicas. 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. Tipos abstractos de datos más avanzados. 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. Estructuras de datos avanzadas. Heaps y sus aplicaciones.

### 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 jerárquicas. 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.



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

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.

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.

Estructuras de datos avanzadas. Heaps y sus aplicaciones. Diagramas binarios de decisión (BDDs) y aplicaciones. Hashing.

#### 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.

#### 5. PROGRAMAS Y/O PROYECTOS PEDAGÓGICOS INNOVADORES E INCLUSIVOS

#### 6. CRONOGRAMA TENTATIVO DE CLASES E INSTANCIAS EVALUATIVAS

Semana	Actividad: tipo y descripción*	Exámenes	Día	Mes
1	Teórico: Introducción a la materia		14	8
1	Práctico: Repaso		15	8



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

1	Teórico: Mecanismos de Reuso		16	8
2	Laboratorio: Introducción a Java		19	8
2	Teórico: Encapsulación		21	8
2	Práctico: Ejercicios básicos de OO		22	8
2	Teórico: Programación Orientada a Objetos		23	8
3	Laboratorio: Uso de Java		26	8
3	Teórico: Introducción a Java		28	8
3	Práctico: Ejercicios en Java, estructuras de datos simples		29	8
3	Teórico: Excepciones en Java		30	8
4	Laboratorio: Ejercicios con Excepciones		2	9
4	Teórico: TAD y Álgebras		4	9
4	Práctico: TADs y Álgebras		5	9
4	Teórico: Álgebras		6	9
5	Práctico: TADs y Álgebras		9	9
5	Teórico: Implementaciones de TADs en JAVA		11	9
5	Laboratorio: Implementaciones de TADs en JAVA		12	9
5	Teórico: Funciones de Abstracción		13	9
6	Práctico: Implementaciones de TADS en JAVA		16	9
6	Jupa (usan aulas)		18	9
6	Jupa (usan aulas)		19	9
6	Teórico: Tiempo de Ejecución Notación Big-Oh		20	9
7	Laboratorio: Tiempo de ejecución		23	9
7	Teórico: Sorting I		25	9
7	Laboratorio: Implementación de ejercicios con análisis de tiempo de ejecución		26	9
7	Teórico: Sorting II		27	9
8	Práctico: Ejercicios sobre Big-Oh		30	9
8	Teórico: Árboles I	Parcial 2/10	2	10
8	Laboratorio: Implementación de Algoritmos de Sorting		3	10
8	Teórico; Árboles II		4	10
9	Práctico: Análisis de tiempo de ejecución de sorting		7	10
9	Teórico: Diccionarios		9	10



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

9	Laboratorio: Implementación de Árboles en JAVA		10	10
10			11	10
10	Laboratorio: Implementación de Árboles en JAVA		14	10
10	Teórico: Implementaciones Básicas de Diccionarios, ABB	Recup. 16/10	16	10
10	Práctico: resolución de problemas sobre árboles		17	10
11	Teórico: Grafos, Introducción		18	10
11	Práctico: Problemas adiciones sobre Árboles.		21	10
11	Teórico: Implementaciones Básicas de Grafos		23	10
11	Práctico: Problemas adiciones sobre Árboles.		24	10
12	Teórico: Algoritmos de recorridos de grafos		25	10
12	Laboratorio: : Implementación de ABBs en JAVA	Tp(30-10 al 13-11)	28	10
12	Teórico: Algoritmos de Warshall y Algoritmo de Dijkstra		30	10
12	Laboratorio: Implementación de ABBs en JAVA		31	10
13	Teórico: AVL, Árboles 2-3		1	11
13	Laboratorio: Implementación de Grafos	Parcial 2 4/11	4	11
13	Teórico: Colas de Prioridad - Heaps		6	11
13	Práctico: Resolución de problemas sobre AVL y Árboles 2-3		7	11
14	Teórico: Funciones de Hashing		8	11
14	Laboratorio: Implementación de Grafos		11	11
14	Teórico: Spanning Trees, Algoritmo de Kruskal y Algoritmo de Prims		13	11
14	Defensas Tp		14	11
15	Recuperatorio 2do Parcial	Recup. 2 15/11	15	11

\*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



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

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: Miércoles de 14 a 16 y Viernes de 12 a 14

Clases Prácticas/Laboratorio:

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

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

## **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 parciales o sus respectivos recuperatorios con nota mayor o igual a 7, y el trabajo práctico 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. 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.



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

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.

*Pablo Ponzio*  
*Pablo Ponzio*

**Firma Profesor/a Responsable**

**Firma Secretario/a Académico/a**