



UNIVERSIDAD NACIONAL DE RÍO CUARTO

FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y
NATURALES

DEPARTAMENTO DE COMPUTACIÓN

CARRERA/S: Analista en Computación – Profesorado en Ciencias de la Computación – Licenciatura en Ciencias de la Computación.

PLAN DE ESTUDIOS: 1999

ASIGNATURA: INTRODUCCIÓN A LA ALGORÍTMICA Y PROGRAMACIÓN

CÓDIGO: 3300

DOCENTE RESPONSABLE: Lic. Ariel Ferreira Szpiniak.

EQUIPO DOCENTE:

Ing. Guillermo Rojo
Lic. Luis Chavez
Prof. Jorge Guazzone

AÑO ACADÉMICO: 2020

REGIMEN DE LA ASIGNATURA: Anual

RÉGIMEN DE CORRELATIVIDADES: (para cursado)

<i>Aprobada</i>	<i>Regular</i>
-	-

CARGA HORARIA TOTAL

TEÓRICO: 98 hs **PRÁCTICAS:** 112 hs **LABORATORIO:** 45 (taller)

CARÁCTER DE LA ASIGNATURA: Obligatoria

A. CONTEXTUALIZACIÓN DE LA ASIGNATURA

La asignatura es de carácter anual y se desarrolla en el primer año de las carreras de Analista en Computación, Profesorado en Ciencias de la Computación, y Licenciatura en Ciencias de la Computación.

B. OBJETIVOS PROPUESTOS

- Capacidad para encontrar soluciones informáticas a problemas mediante la modelización disciplinada de soluciones y descomposición en módulos.
- Capacidad para desarrollar algoritmos que resuelvan situaciones problemáticas reales.
- Capacidad para traducir eficientemente algoritmos al lenguaje de programación C.
- Habilidad en el uso del lenguaje C.
- Habilidad en el uso de buenos hábitos de programación.
- Capacidad para documentar técnicamente los programas desarrollados (análisis, diseño, implementación, prueba, manuales para el usuario, etc).
- Habilidad en el uso de herramientas básicas de desarrollo de software.
- Capacidad para trabajar en grupo.

C. CONTENIDOS BÁSICOS DEL PROGRAMA A DESARROLLAR

En la asignatura se trabaja sobre los siguientes ejes temáticos:

- Resolución de problemas. Algoritmos. Notación
- Paradigma de programación imperativa. Tipos de composición: secuencial, condicional e iterativa
- Metodología de desarrollo de programas: análisis, diseño, implementación y prueba.
- Abstracciones: acciones, funciones, bibliotecas. Parametrización
- Secuencias: Modelos de acceso secuencial
- Implementación de algoritmos: traducción de notación algorítmica a lenguaje C
- Estructuras de datos
- Almacenamiento persistente
- Recursividad
- Métodos de búsqueda y ordenamiento
- Tipos Abstractos de Datos

Los contenidos se especifican en el programa analítico.

D. FUNDAMENTACIÓN DE LOS CONTENIDOS

Los contenidos propuestos están orientados a que el alumno adquiera capacidad para encontrar soluciones informáticas a problemas que, fundamentalmente, requieran representación de secuencias de datos y su posterior procesamiento. A su vez, los contenidos pretenden brindar herramientas para la construcción de algoritmos, y para traducirlos eficientemente al lenguaje de programación C. Se pretende que mediante estos contenidos el alumno sea capaz de utilizar buenos

hábitos de programación, y de documentar el proceso de desarrollo de las soluciones.

E. ACTIVIDADES A DESARROLLAR

CLASES TEÓRICAS: Presencial en aula, 98 hs, totales. Se requerirá asistir al 80% de las clases durante el primer cuatrimestre y el 80% durante el segundo.

CLASES PRÁCTICAS: Presencial en aula, 112 hs totales. Se requerirá asistir al 80% de las clases durante el primer cuatrimestre y el 80% durante el segundo.

CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO: Presencial en laboratorio de informática, 45 hs totales. Se requerirá asistir al 80% de las clases durante el primer cuatrimestre y el 80% durante el segundo.

Se trabajará con clases teóricas y clases prácticas semanales. Para las clases prácticas los alumnos serán distribuidos en comisiones. En ambos casos se contará con clases de consulta de dos (2) horas de duración por semana. Además habrá una clase de laboratorio de dos (2) horas de duración por semana. Para la actividad práctica se dispondrá de cuatro (4) horas por semana reservadas exclusivamente para el acceso al laboratorio de informática fuera de los horarios de clase.

Cada trabajo práctico tendrá un ejercicio que podrá ser entregado para su corrección.

Habrá un ejercicio integrador evaluable (que puede involucrar conceptos de uno o más TP) que será desarrollado de manera individual en el Laboratorio. Su nota aportará un 10% de la valoración total del parcial dentro del cual se encuadran los temas evaluados.

Se realizarán reuniones semanales entre el docente de teórico y los de práctico. Los docentes de práctico realizarán reuniones semanales con los ayudantes alumnos.

Todos los docentes participarán en la elaboración y análisis del material didáctico, la elaboración de exámenes, en la definición de los criterios de evaluación y en el seguimiento de las clases prácticas para detectar problemáticas.

Se utilizará un aula virtual disponible en la Campus Virtual de la UNRC para brindar información y materiales a los alumnos y e-mail para la comunicación diaria entre el equipo docente.

F. NÓMINA DE TRABAJOS PRÁCTICOS

1. TP 1 Introducción
2. TP 2 Composición secuencial.
3. TP 3 Composición condicional
4. TP 4 Modularización: Funciones
5. TP 5 Modularización: Acciones
6. TP 6 Composición Iterativa

7. TP 7 Arreglos
8. TP 8 Registros y Arreglos de registros
9. TP 9 Especificación y tratamiento de secuencias con marca final
10. TP 10 Estructuras de datos dinámicas y su tratamiento como secuencias con marca final. LSE y LDE.
11. TP 11 Especificación y tratamiento de secuencias con marca inicial. Uso de Arreglos y estructuras dinámicas.
12. TP 12 Archivos.
13. TP 13 Recursividad
14. TP 14 Algoritmos de Ordenamiento y Búsqueda
15. TP 15 Tipos Abstractos de Datos: uso e implementación de Pilas y Colas

G. HORARIOS DE CLASES:

HORARIO DE CLASES DE CONSULTAS:

- A. Ferreira Lunes 18 hs (aula Teoría) y Miércoles 12 hs oficina A13-F Dpto de Computación.
- L. Chavez Martes 12 hs Dpto de Computación
- G. Rojo Lunes de 12 hs y Miércoles de 12 hs Dpto Matemáticas Oficina A8
- J. Guazzone Jueves 13 hs Dpto de Computación

NOTA: antes de asistir, verificar que los días no coincidan con feriados o con turnos de exámen o evaluaciones parciales. Para situaciones excepcionales pueden enviar un e-mail al docente y preguntar si se puede tener una consulta fuera de los horarios antes pautados.

H. MODALIDAD DE EVALUACIÓN:

- EVALUACIONES PARCIALES:** cuatro exámenes parciales escritos sobre el práctico de la materia y una pregunta que relacione teoría-prácticas, con sus respectivos recuperatorios y un proyecto final.
- EVALUACIÓN FINAL:** Los exámenes finales son orales e individuales sobre la teoría y práctica de la materia. Los alumnos podrán optar por realizar un taller pre-final por turno a los efectos lograr una mejor adquisición de los conocimientos y una adecuada preparación y exposición de los temas abordados en la materia.
Alumnos libres: realización de un proyecto individual de complejidad y características similares al proyecto de finalización del segundo cuatrimestre pero más acotado en funcionalidades. Dicho proyecto será entregado para su realización dos semanas antes del último llamado de cada turno. De aprobar el proyecto el alumno deberá rendir un examen práctico escrito que será tomado tres días antes del último llamado de cada turno. De aprobar el examen práctico podrá rendir el examen final en las mismas condiciones que los alumnos regulares.
- CONDICIONES DE REGULARIDAD:** Aprobar los cuatro exámenes prácticos, el práctico integrador, y el proyecto final.
- CONDICIONES DE PROMOCIÓN:** No tiene.

PROGRAMA ANALÍTICO

A. CONTENIDOS

Unidad 1:

Esquema funcional de una computadora. La problemática del desarrollo de software: características de un buen programa, fracasos notables, la crisis del software, métodos de desarrollo de programas. Pasos básicos para la solución de problemas. Definiciones de acción, proceso, algoritmo, programa, información, estado, léxico.

Unidad 2

Paradigma de programación imperativa. Notación algorítmica: acciones básicas. Objetos constantes y variables, tipos simples y valores. Tipos de composición: secuencial, condicional e iterativa. Composición secuencial. Análisis por casos y composición condicional o alternativa. La forma "según", "si entonces sino", "si entonces".

Unidad 3

Metodología de desarrollo de programas. Análisis: pre y poscondición, especificación, notación. Diseño: notaciones, método descendente, partición, refinamiento.

Unidad 4

Abstracciones: acciones y funciones. Parametrización: parámetros formales y actuales, pasajes de parámetros. Acciones: noción intuitiva, acciones parametrizadas, Funciones: noción intuitiva, dominio, rango, descripción de funciones por algoritmo (*intensional*) o por tabulación (*extensional*), composición de funciones. Reglas de alcance. Efectos colaterales.

Unidad 5

Composición iterativa: "mientras", "repetir" y "para". Riesgos. Estados intermedios de un ciclo. Invariante de un ciclo.

Unidad 6

Secuencias: Características de una secuencia. Notación y operaciones sobre secuencias. Especificación. Modelos de acceso secuencial: Modelo 1 o de marca final y Modelo 2 o de marca inicial.

Unidad 7

Modelo 1. Especificación del Modelo 1. Esquemas de tratamiento y búsqueda secuencial: R1, R2, R3, RP y B. Máquinas de tratamiento de secuencias.

Modelo 2. Especificación del Modelo 2. Esquemas de tratamiento y búsqueda secuencial: R1, R2, R3, RP y B. Máquinas de tratamiento de secuencias.

Unidad 8

Implementación de algoritmos: traducción de notación algorítmica a lenguaje C, funciones y tipos. Reglas y convenciones para la escritura de código (notación, indentación, comentarios, etc.).

Unidad 9

Estructuras de datos: tipos estructurados. Tipos homogéneos: arreglos unidimensionales y bidimensionales, conjuntos. Tipos heterogéneos: registros.

Almacenamiento persistente: archivos, tipos de archivos. Acceso secuencial. Acceso directo. Gestión de archivos (ABML).

Unidad 10

Recursividad: Introducción. Concepto. Funciones y acciones recursivas en el paradigma imperativo. Recursión directa e indirecta. Profundidad. Algoritmos recursivos.

Unidad 11

Métodos de búsqueda: algoritmos de búsqueda secuencial y búsqueda dicotómica o binaria. Métodos de ordenamiento: ordenamiento por selección, por intercambio, por inserción.

Unidad 12

Abstracción: creación de distintos niveles de abstracción para la solución de problemas complejos. Relación entre máquinas abstractas y Tipos Abstractos de Datos. Ventajas de la utilización de TADs, TADs más usuales: pilas, colas, listas. Implementación de TADs: almacenamiento contiguo, colas circulares, estructuras encadenadas, estructuras doblemente encadenadas, utilización de un elemento ficticio en la cabeza de la estructura. Implementación de TADs usando lenguaje C.

B. CRONOGRAMA DE CLASES Y PARCIALES

Tentativo

Semana	Teóricos	Prácticos	Laboratorios	Parciales / Recuperatorios
1	Introducción	1		
1	Algoritmo	2	Linux	
2	Composición secuencial	2		
2	Composición secuencial	3	2	
3	Composición secuencial	3	2	
4	Introducción y Algoritmos. Composición de Algoritmos	3		
5	Introducción y Algoritmos. Composición de Algoritmos	4		
5	Datos simples. Pasos para solucionar problemas. Composición de Algoritmos	4	4	I

6	Datos simples. Pasos para solucionar problemas. Composición de Algoritmos	4		
6				Primer Parcial
7	Composición de Algoritmos	5		
7	Composición de Algoritmos	5	5	
9	Análisis y Diseño. Abstracción, acciones	5	5	
9	-	6		
10	Pasaje de parámetro, funciones	6	5	
11	Reglas de alcance	6	6	
11	Arreglos	6		
12	Arreglos	6	6	
12				Segundo Parcial
12	Arreglos	6	6	
13	Registros	7	7	
13	Registros	7		
14	Registros y arreglos	7	7	
14				Recupera torio Primer Parcial
15	Registros y arreglos	7	7	
15				Recupera torio Segundo Parcial
16	Bibliotecas	8		

* El cronograma del segundo cuatrimestre será presentado al finalizar el primero para coordinar con las demás materias que no son anuales.

C. BIBLIOGRAFÍA

□ Obligatoria

- Scholl, P. y J.-P. Peyrin, "Esquemas Algorítmicos Fundamentales: Secuencias e iteración", Barcelona, Ed. Masson, 1991.
- Lucas, M., J.-P. Peyrin y P. Scholl, "Algorítmica y Representación de Datos. Tomo 1: Secuencia, Autómata de estados finitos", Barcelona, Ed. Masson, 1985.
- Wirth, N. "Algoritmos más estructuras de datos = programas", Ediciones del Castillo, 1984.
- Watt, David, "Programming Language Concepts and Paradigms", Prentice-Hall International Series in Computer Science (1990).

- Biondi, J. y G. Clavel, "Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes", 2° ed., Barcelona: Masson, 1985.
- Clavel, G. y Biondi, J., "Introducción a la Programación. Tomo 2: Estructuras de Datos", 2° ed., Barcelona: Masson, 1985.
- De Guisti, A. "Algoritmos, datos y programas. Con aplicaciones en Pascal, Delphi y Visual Da Vinci. Prentice Hall.
- Aho, A., J. Hopcroft and J. Ullman, "Data Structures and Algorithms", Reading MA, Addison-Wesley, 1987 (traducción "Estructuras de Datos y Algoritmos", Addison-Wesley, 1988).

□ **Optativa**

- Brookshear, J. G., "Introducción a las Ciencias de la Computación", 4° ed., Addison-Wesley, 1995.

□ **Aulas Virtuales**

□ **Pública: para acceso universal**

<http://www.evelia.unrc.edu.ar>

Ciencias Exactas – Grado -> Introducción a la Algorítmica y Programación (3300)

- **Cursantes 2020: para acceso, autenticado, de los cursantes 2020. El aula contiene foros, materiales teóricos y prácticos, software, enlaces de interés, noticias, pizarrón, grupos, correo, y actividades (publicación, recepción de resoluciones, calificaciones y devoluciones).**