

**PROGRAMA DE LA ASIGNATURA:**  
Algorítmica y Programación I (IF003)**CÓDIGO:** IF003  
**AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:**  
1 año  
**FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:**  
2019-04-29  
**CARRERA/S:** Licenciatura en Sistemas 049/2017,  
Analista Universitario de Sistemas 050/2017,**CARÁCTER:** CUATRIMESTRAL (2do)  
**TIPO:** OBLIGATORIA  
**NIVEL:** GRADO  
**MODALIDAD DEL DICTADO:** PRESENCIAL (EN LÍNEA)  
**MODALIDAD PROMOCION DIRECTA:** SI  
**CARGA HORARIA SEMANAL:** 8 HS  
**CARGA HORARIA TOTAL:** 120 HS**EQUIPO DOCENTE**

Nombre y Apellido	Cargo	e-mail
Daniel Aguil Mallea	profesor adjunto	daguil@untdf.edu.ar
Luciana Lens	Asistente principal	lsLens@untdf.edu.ar
Antonio Retamar	Asistente principal	alretamar@untdf.edu.ar

## 1. FUNDAMENTACION

La asignatura se dicta en el primer año (segundo cuatrimestre) de ambas carreras. Dentro del plan de estudios pertenece al área de Algoritmos y Lenguajes. Mediante el cursado de la asignatura se espera que el alumno adquiera competencias que le permitan Analizar, Diseñar e Implementar la solución de distintos problemas de la vida real, con complejidad creciente, utilizando una computadora.

Dada la importancia de los temas que se abordan en la misma y teniendo en cuenta el régimen de correlativas establecido en el plan de estudios, se considera que es una de las asignaturas troncales de la carrera.

Articula verticalmente con la asignatura Expresión de Problemas y Algoritmos, que es la materia correlativa para el cursado. Así mismo, para lograr una articulación horizontal, con las asignaturas del área de matemática, de primer año, se incluyen entre los ejercicios a resolver por los alumnos, problemas que deben hacer uso de algunos conceptos matemáticos fundamentales vistos en las asignaturas Álgebra, Análisis Matemático y Elementos de Lógica y Matemática Discreta. Es la base para poder abordar los contenidos de Algorítmica y Programación II.

## 2. OBJETIVOS

### a) OBJETIVOS GENERALES

Potenciar la capacidad para la resolución de problemas y la expresión de soluciones mediante un lenguaje de programación estructurado, siguiendo la metodología top down a través de :

- La generación de un espacio colectivo que permita potenciar los procesos de razonamiento implicados en la resolución de problemas y su posterior programación, lo que implica que dado un problema real el alumno logre:

- 1) Alcanzar alguna representación mental del mismo
- 2) Convertirla en un modelo computable.
- 3) Analizar alternativas de solución
- 4) Decidir la secuencia de ejecución más eficaz y eficiente para cumplir con una función específica de un programa.

- El desarrollo individual de las capacidades cognitivas generales que le permitan incorporar conocimientos de manera autónoma, confrontar y sintetizar el Saber y el Hacer.

Reconocer y resolver utilizando el lenguaje de programación, problemas de naturaleza recursiva

Se considera en primera instancia el rol de la asignatura en el proceso de enseñanza y de aprendizaje de la programación presente en el plan de estudio de las carreras donde se dicta. Partiendo de que la capacidad de resolución de problemas es una aptitud cognitiva que caracteriza una de las actividades humanas más inteligentes, y que la programación de algoritmos es una tarea intelectual exigente y reflexiva, se pone énfasis en la forma correcta de abordar los problemas a resolver y los pasos a seguir hasta obtener la solución buscada a través de un lenguaje de programación.

La metodología que se aplica se sustenta en que los conocimientos nuevos se introducen a partir de los conocimientos previos de los alumnos y de las estrategias básicas de razonamiento adquiridas (aprendizaje significativo). Por otra parte se reconoce que el saber no se limita a la información, que

lo importante es saber - hacer, y que esto sólo se aprende haciendo.

Por esta razón el alumno es el centro sobre el que giran los procesos de enseñanza y de aprendizaje. Se motiva al alumno para que tenga un rol activo: debe ser observador, formulador de hipótesis y defensor de sus propios criterios. Se fomenta una actitud crítica y reflexiva, la que le será útil a lo

largo de su carrera y en su futura vida profesional.

En el proceso de enseñanza se han tomado 4 ejes fundamentales:

- La especificación y expresión de algoritmos, analizando su corrección y eficiencia.
- La modularización de los problemas
- El tratamiento de estructuras de datos simples y compuestas.
- La contextualización de la expresión de algoritmos en el ámbito más amplio del desarrollo de sistemas.

Tratando de considerar los intereses y motivaciones que traen los alumnos, que generalmente está más relacionado con el campo ocupacional que con un área del conocimiento, se seleccionan problemas reales, simples, que se van complejizando de manera que los nuevos conceptos surjan

como una necesidad para la resolución de los nuevos problemas planteados.

A lo largo del curso el alumno deberá resolver diferentes clases de problemas eligiendo distintos algoritmos y tipos de datos (a medida que estos crezcan en complejidad), buscando que asimile el concepto de que un especialista en Informática debe ser un analista del mundo real capacitado para

elegir:

- Alternativas de solución por computadora, en función del problema a resolver, de los recursos disponibles y de las restricciones de contexto existentes.
- Trabajar en forma colaborativa y cooperativa.

Partiendo de la consigna que la docencia se hace con pedagogía y didáctica, además de la transmisión de los conocimientos específicos se ha puesto énfasis en la forma de abordar cada una de las clases a impartir en la asignatura, de tal forma de hacer interesantes tanto la enseñanza como el aprendizaje.

A pesar de la división entre la teoría y la práctica (explícita en el plan de estudios), no se

considera a las mismas como entidades separadas, sino que forman parte de un proceso único. Por ello a veces se parte de cuerpos teóricos para abordar luego cuestiones más cercanas a la práctica, y otras veces, partiendo de la práctica, se le requieren a la teoría razones y fundamentos. El cuerpo docente a cargo de la asignatura está preparado para abordar la tarea de enseñanza respetando estas pautas.

Por las características de la asignatura el aprendizaje concreto se obtiene con trabajos teórico - prácticos.

De cualquier forma no están ausentes las clases magistrales, ya que estas también son necesarias cuando se requieren introducir los conceptos generales de algunos de los temas que se abordan. Se propicia tanto el trabajo grupal como el individual, según las tareas que se planteen en cada una de las clases.

Cabe remarcar que el alumno es informado, desde el primer día de clase, que su participación en las mismas debe ser activa y no pasiva, crítica y no receptiva. En definitiva: que debe aprender a pensar por su propia cabeza.

Las soluciones de algunos de los problemas planteados en las guías de trabajos son expuestas en forma individual o por grupo (según corresponda a la consigna de la tarea), y se cotejan entre ellas, descubriendo así los conceptos de eficiencia a los que alude el programa de la asignatura.

Cada uno

debe poder encontrar las bondades y falencias de las soluciones presentadas.

Con el fin de que el alumno comprenda que el aprendizaje profundo se logra resolviendo problemas por sí mismo, se solicitará al alumno entregar un ejercicio resuelto de cada una de las guías de trabajos prácticos. El ejercicio será seleccionado por la cátedra. Posteriormente, y con el fin de que

todos los alumnos puedan cotejar las soluciones realizadas por sus compañeros, se habilitará un espacio en el entorno virtual de enseñanza y de aprendizaje, donde las soluciones quedarán almacenadas, y estarán a disposición de los alumnos. La cátedra, por su parte, dejará una solución posible al mismo ejercicio. Se espera que, esta sea una alternativa más que ayude a los alumnos a alcanzar y profundizar los conceptos enseñados.

## **b) OBJETIVOS ESPECIFICOS**

Seguir adquiriendo nuevos hábitos de programación en el paradigma imperativo, profundizando los conceptos básicos de la Ingeniería de Software que incluyen: Análisis de Algoritmos, Modularización, Técnicas de Diseño Descendente, Encapsulamiento, Verificación, etc.

- Reforzar los conceptos fundamentales del "QUE" y del "COMO" en el desarrollo del software.
- Potenciar la capacidad de abstracción, en base al planteo de problemas y casos de estudio.
- Profundizar conocimientos y mejorar la capacidad para programar en el paradigma imperativo.
- Adquirir los conceptos básicos de recursividad, y su aplicación en la Ciencia que estudian.
- Promover la integración y participación a fin de afianzar y valorar el aprendizaje colaborativo.
- Generar capacidades que le permitan reflexionar potenciando el pensamiento crítico.
- Promover acciones que despierten interés en investigar tópicos relacionados con la disciplina.

COMPETENCIAS A DESARROLLAR:

- Analizar, comprender y resolver problemas en forma eficiente, utilizando un lenguaje de programación.
- Trabajar colaborativamente.
- Incorporar el interés por investigar temas de la disciplina, como una herramienta fundamental en su futuro desarrollo profesional.

## **3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA**

Partiendo de la base que la calidad de la relación pedagógica se expresa también en los

momentos de la evaluación de lo aprendido, y que el objetivo de la evaluación es el mejoramiento, tanto de los procesos de enseñanza como el de aprendizaje, se espera que la misma sirva como instrumento potente para el diagnóstico, la valoración y el cambio, lo que la constituye en una instancia más de aprendizaje, es decir de producción de conocimientos. Se evaluará con intencionalidad pedagógica, y no sólo para control. Por esta razón y dado que en nuestra Sede la relación docentes- alumnos así lo permite, se llevará a cabo una evaluación permanente tanto de la tarea de enseñanza como la de aprendizaje.

Se espera que el alumno sienta que estudia para saber buscar la verdad y encarar debidamente los problemas, o sea que estudia para aprender y no solo para aprobar la asignatura. Alcanzar la auto evaluación será entonces otro de los objetivos planteados. Para poder realizar en parte esta evaluación los alumnos deberán presentar ejercicios prácticos previamente pautados, los que serán corregidos con el mismo criterio que luego se utilizará para la corrección de los exámenes parciales y finales. Esto les permitirá conocer de antemano los criterios y forma de evaluaciones que se usarán para la acreditación, además de permitirles ver sus aciertos y sus errores. Por su parte los docentes ayudarán a encontrar estrategias más aptas que las ya enseñadas para la rectificación de los mismos.

#### Acreditación:

Esta responde a una lógica institucional, explícita en el Reglamento de Estudios de Grado y de Pos Grado de la Universidad, para certificar conocimientos alcanzados. El criterio utilizado es que el alumno haya alcanzado los objetivos (generales y específicos) citados anteriormente. Para ello se prevén tanto exámenes (parciales y finales), trabajos integradores, entrega de ejercicios y régimen de promoción sin examen final. Se tendrá en cuenta, además de los resultados cuantitativos, el desempeño y las actitudes que el alumno ha demostrado durante los meses del cursado de la asignatura.

#### Condiciones para la aprobación de la asignatura:

Para aprobar el cursado de la asignatura los alumnos deberán aprobar dos parciales con contenido netamente práctico, y haber entregado 100 % de los ejercicios seleccionados de las guías de trabajos prácticos. Los ejercicios de las guías, que así lo requieran, deberán ser desarrollados utilizando la computadora y el lenguaje de programación establecido.

La nota requerida para la aprobación de los es de 4 o superior. Para alcanzar esta nota, el alumno deberá en todos los casos, haber desarrollado correctamente al menos el 60% de lo requerido en ellos. Cada parcial tendrá su respectivo recuperatorio.

Para los alumnos que cursen por régimen con examen final, una vez obtenida la cursada y aprobadas las correlativas estarán en condiciones de rendir el examen final en algunas de las fechas establecidas en el Calendario Académico, y por el período establecido en el Reglamento de Estudios de Grado y Pos Grado.

Para rendir en calidad de alumno libre, el alumno deberá aprobar, en primera instancia, el ejercicio práctico, que se plantee. Aprobado el mismo, deberá rendir un examen teórico práctico, en las condiciones que se establecen en el Reglamento de Estudios de Grado y Pos Grado. El contenido del examen se basará en los contenidos del programa de la asignatura vigente para ese ciclo lectivo.

#### Régimen de aprobación por Promoción:

Los alumnos que tengan las asignaturas correlativas aprobadas y que así lo deseen podrán hacer uso del régimen de promoción (sin examen final). Para aprobar la asignatura bajo este régimen el alumno deberá:

- 1- Aprobar los dos parciales prácticos con nota equivalente a 7 o superior (no se tendrán en cuenta las fechas de los exámenes recuperatorios para la opción de promoción).
- 2.- Haber entregado al 100% de los ejercicios de las guías de prácticas.

Para aquellos alumnos que cumplan con lo estipulado anteriormente, se les tomarán dos

parciales integradores con contenidos de fundamento teórico. Estos parciales no poseen recuperatorio, y deben ser aprobados con nota equivalente a 7 o superior, lo que equivale al desarrollo correcto de al menos el 80% de los temas planteados. Los alumnos que no aprueben alguno de estos parciales dejan automáticamente la promoción y se incorporan al régimen regular.

## 4. CONTENIDOS DE LA ASIGNATURA

Contenidos Mínimos :

- Análisis y diseño de algoritmos.
- Tipos de datos y estructuras de datos elementales. Representación en memoria.
- Estructuras de control. Programación estructurada.
- Procedimientos y funciones.
- Recursividad.
- Algoritmos fundamentales: recorrido, búsqueda, ordenamiento, actualización.

Programa Analítico:

### 1) Principios de Programación

01.01 Programación - Definiciones de conceptos básicos

01.02 Modelización de problemas del mundo real.

01.03 Algoritmos- Definición y clasificación.

01.04 Etapas en la resolución de problemas con computadora.

### 2) Diseño de Algoritmos y Elementos de Programación

02.01 Algoritmos, programas y lenguajes de programación.

02.02 Programación Estructurada.

02.03 Datos, tipos de datos simples y operadores.

02.04 Variables, constantes y expresiones.

02.05 La operación Asignación.

02.06 Funciones internas.

02.07 Entrada. Salida

02.08 Herramientas básicas para construir programas: Estructuras Selectivas e Iterativas.

02.09 Estructuras adicionales control.

02.10 Resolución de programas en un lenguaje de programación estructurado y modular.

### 3) Programación Modular

03.01 Modularización- Definición- Importancia de la misma- Refinamientos sucesivos.

03.02 El diseño descendente.

03.03 Subalgoritmos: Funciones y procedimientos

03.04 Parámetros y argumentos.

03.05 Pasaje de parámetros.

03.06 Variables locales y globales.

### 4) Programación con Recursividad

04.01 Principios de la recursión.

04.02 Funcionamiento de la recursión.

04.03 Funciones y procedimientos recursivos.

04.04 Ventajas y desventajas de los algoritmos recursivos.

### 5) Estructuras de Datos

05.01 Definición - Necesidades de uso de estos tipos de datos.

05.02 Registros: declaración y uso de los mismos.

05.03 Arreglos- Definición- Necesidades de uso de estos tipos de datos.

05.04 Vectores- Definición- Operaciones con vectores.

05.05 Matrices- Definición- Operaciones con matrices.

05.06 Distinto tipo de matrices (cuadradas, rectangulares, triangulares, ralas, etc.)

05.07 Operaciones con matrices.

05.08 Arreglos n- dimensionales

6) Análisis de Algoritmos. Conceptos de Eficiencia

06.01 Concepto de eficiencia.

06.02 Análisis de eficiencia de un algoritmo.

06.03 Algoritmos de ordenamiento elementales (Selección, Inserción e Intercambio) y avanzados (Quick Sort y Mezcla)

06,04 Ordenamiento utilizando mezcla y estrategia indirecta.

06.05 Análisis de los métodos de ordenación elementales y avanzados

06.06 Algoritmos de búsqueda elementales (Secuencial y Binaria)

06.07 Análisis de los métodos de búsquedas elementales

## 5. RECURSOS NECESARIOS

- Proyector
- Pc
- Laboratorio Informatica

## 6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
semana 1	Presentación de la asignatura. Formas de evaluación y acreditación. Conceptos de la Unidad I	Comprender el alcance de la materia. Conocer el nivel del grupo. Comprender y aplicar los conceptos desarrollados	
Semana 2	Unidad II	Comprender y aplicar los conceptos desarrollados en la Unidad	
Semana 3	Unidad II	Comprender y aplicar los conceptos desarrollados en la Unidad	
Semana 4	Unidad III	Comprender y aplicar los conceptos desarrollados en la Unidad	
Semana 5	Unidad IV	Comprender y aplicar los conceptos desarrollados	
Semana 6	Unidad IV	Comprender y aplicar los conceptos desarrollados	
Semana 7	Primer Parcial (incluye las unidades I..IV) Se desarrollan contenidos de la Unidad VI.	Acreditar conocimientos Comprender y aplicar los conceptos desarrollados	
Semana 8	Unidad V	Comprender y aplicar los conceptos desarrollados en las Unidades	
Semana 9	Clases de repaso. Recuperatorio primer parcial. Parcial integrador	Acreditar conocimientos Comprender y aplicar los conceptos desarrollados	
Semana 10	Unidad V	Comprender y aplicar los conceptos desarrollados en las Unidades	
Semana 11	Unidad VI	Comprender y aplicar los conceptos desarrollados en las Unidades	
Semana 12	Segundo Parcial (incluye las unidades V y VI)	Acreditar conocimientos	

## 7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
De Giusti- Madoz y otros	2001	Algoritmos, Datos y Programas	-		Prentice Hall
Dale- Lilly	1989	Pascal y Estructuras de Datos	-		M. Graw Hill
Berlanga LLavori R. - García Sevilla P. y otros	2000	Introducción a la Programación con Pascal	-		UNIVERSITAT JAUME -I
Braunstein -Gioia	1990	Introducción a la Programación y Est. de Datos	-		Eudeba
Joyanes Aguilar	1997	Fundamentos de Programación	-		M.Graw Hill
O. Cairó - S. Guardati	2007	Estructuras de Datos – Seg. Edición	-		Mc Graw Hill
Desarrollo de Algoritmos y Técnicas de Programación en Pascal	1997	Pareja . M. Ojeda - A. Andeyro - C. Rossi	-		RA-Ma
Fundamentos de Programación	2007	J. Carretero - F. García - J. Lobato - J. Muñoz - A. mateos	-		Thomsom Paraninfo
Introducción a las Ciencias de las Computadoras	1980	Tremblay- Bunt	-		Mc Graw Hill
Introducción a la Programación Sistemática	1982	N.Wird	-		El Ateneo
Algoritmos y Programación	2007	M. Perez Berro	-		Nueva Librería

-----  
Firma del docente-investigador responsable

<b>VISADO</b>		
<b>COORDINADOR DE LA CARRERA</b>	<b>DIRECTOR DEL INSTITUTO</b>	<b>SECRETARIO ACADEMICO UNTDF</b>
Fecha :	Fecha :	