

PROGRAMA DE LA ASIGNATURA:
Algorítmica y Programación II (IF006)

CÓDIGO: IF006
AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:
2 año
FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:
2017-03-08
CARRERA/S: Licenciatura en Sistemas , Analista Universitario de Sistemas

CARÁCTER: CUATRIMESTRAL (1ro)
TIPO: OBLIGATORIA
NIVEL: GRADO
MODALIDAD DEL DICTADO: PRESENCIAL
MODALIDAD PROMOCION DIRECTA: SI
CARGA HORARIA SEMANAL: 8.00 HS
CARGA HORARIA TOTAL: 120.00 HS

EQUIPO DOCENTE

Nombre y Apellido	Cargo	e-mail
Beatriz Depetris	Profesor Titular	bdepetris@untdf.edu.ar
Daniel Aguil Mallea	Profesor Adjunto	daguil@untdf.edu.ar

1. FUNDAMENTACION

La asignatura se dicta en el segundo año (primer 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 pequeños sistemas de software. 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.

Articulación Vertical y Horizontal:

En la asignatura se profundizan temas abordados en Algorítmica y Programación I y se utilizan conceptos abordados en Elementos de Lógica y Matemática Discreta. (Materias correlativas) Por otra parte, el alumno incorporará una serie de conceptos nuevos, que luego utilizará, y en muchos casos profundizará, en otras materias de la carrera, como Bases de Datos I y II, Laboratorio de Programación y Lenguajes, entre otras.

2. OBJETIVOS

a) OBJETIVOS GENERALES

- Profundizar y adquirir nuevos conocimientos que le permitan Analizar, Diseñar e implementar pequeños sistemas de software.
- 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.

- Profundizar los conceptos de recursividad, tanto en lo referido a algoritmos como a tipos de datos.
- 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.

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.
- Conocer las estructuras de datos fundamentales y su presentación como Tipo Abstracto de Dato, las implementaciones estáticas y dinámicas de los mismos y evaluar cuando resulta conveniente una u otra implementación, enfatizando la relación existente entre las estructuras de datos y los algoritmos.
- Implementar soluciones recursivas, analizando ventajas y desventajas de su aplicación.
- Desarrollar e implementar un sistema de software de mediana complejidad, a partir de las especificaciones dadas por la cátedra.

3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

Para aprobar el cursado de la asignatura los alumnos deberán aprobar dos parciales con contenido netamente práctico, dos trabajos integradores, y haber entregado el 100 % de los ejercicios seleccionados, por el docente, de las guías de trabajos prácticos. Tanto los trabajos integradores como los ejercicios de las guías deberán ser desarrollados utilizando la computadora y el lenguaje de programación establecido. Lo harán en forma individual, en el caso de los ejercicios, y en un grupo de 2 o 3 alumnos en el caso de los trabajos integradores.

La nota requerida para la aprobación de los parciales y de los trabajos integradores 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.

La nota de cursada se obtiene como el promedio entre la nota de los parciales aprobados y la obtenida en cada uno de los trabajos integradores.

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 solicitar, con al menos 15 de anticipación la entrega de un trabajo Integrador, que deberá desarrollar correctamente utilizando el lenguaje de programación que se establezca. Aprobado este trabajo, en la fecha establecida para rendir el final, el alumno deberá aprobar el examen escrito, de carácter teórico práctico que se plantee, 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 antes del 1º parcial 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 (80% o más de desarrollo correcto de los ejercicios). No se tendrán en cuenta las instancias de recuperación.

2- Aprobar los trabajos integradores con nota 7 o superior.

3.- Haber entregado el 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 :

- Estrategias de diseño de algoritmos.
- Tipos abstractos de datos. Estructuras de datos: listas, pilas, colas, árboles, grafos. Tipos de datos recursivos.
- Programación con recursividad.
- Representación de datos en memoria. Estrategias de implementación. Manejo de memoria en ejecución.
- Métodos avanzados de ordenamiento y búsqueda.
- Análisis de complejidad de algoritmos.
- Tratamiento de errores y manejo de excepciones.
- Corrección, verificación y eficiencia de algoritmos.
- Algoritmos numéricos y propagación del error.

Programa Analítico:

Unidad 1: Conceptos de Metodologías en el desarrollo de Sistemas de Software

01.01 Ingeniería de Software

01.02 El ciclo de vida clásico de un sistema de software.

01.02.01 Análisis y Especificación de Requerimientos.

01.02.02 Diseño y Especificación del Sistema.

01.02.03 Codificación y Verificación de los módulos. Tipos de errores, tratamiento y propagación, manejo de excepciones

01.02.04 Integración y Prueba Global del Sistema.

01.02.05 Mantenimiento del Sistema.

01.02.06 Tiempos y costos requeridos por las etapas del Ciclo de Vida.

01.03 Especificación, codificación y prueba de algoritmos.

01.04 El método de refinamiento sucesivos.

01.04.01 Ocultación de la información.

01.04.02 Niveles de Abstracción.

01.05 Noción de paradigma de programación.

01.06 Aspectos importantes de los sistemas de software.

- 01.06.01 Corrección.
- 01.06.02 Verificabilidad.
- 01.06.03 Confiabilidad.
- 01.06.04 Eficiencia.
- 01.06.05 Interfaz amigable
- 01.06.06 Facilidad de mantenimiento
- 01.06.07 Reusabilidad.
- 01.06.08 Portabilidad.

Unidad 2: Tipos y Estructuras Básicas de Datos

- 02.01 Tipos Simples
 - 02.01.01- Tipo Entero (INTEGER) - Tipo -Real (REAL) - Tipo Carácter (CHAR) - Tipo Lógico (BOOLEAN).
 - 02.01.02 - Tipos ordinales
 - 02.01.03 Tipos definidos por el usuario
 - 02.01.04 Tipo Enumerado
 - 02.01.05 Tipo Subrango
 - 02.01.06 Tipo Conjunto
 - 02.01.07 Tipo String
- 02.02 Tipos Estructurados.
 - 02.02.01 Tipo Registro (RECORD)
 - 02.02.01.01 Registros con parte variante
 - 02.02.02 Tipo Arreglo (ARRAY)
 - 02.02.02.01 Vectores
 - 02.02.02.02 Matrices
 - 02.02.02.03 Arreglos multidimensionales
 - 02.03 Tipo Archivo (FILE)
 - 02.03.01 Conceptos Generales
 - 02.03.02 Administración de archivos
 - 02.03.03 Operaciones básicas sobre archivos.
 - 02.03.04 Técnicas de organización y Acceso a un archivo.

Unidad 3: Tipos Abstractos de Datos

- 03.01 Abstracción de Datos
- 03.02 Encapsulamiento de Datos
- 03.03 Concepto de Tipo Abstracto de Dato (TDA)
- 03.04 Diferencia entre tipo de dato y tipo abstracto de dato.
- 03.05 Requerimientos y diseño de un Tipo Abstracto de Dato (TDA)
- 03.06 Abstracción y Ocultamiento en Pascal.

Unidad 4: Estructuras de Datos Pila

- 04.01 El tipo Pila.
 - 04.01.01 Definición y ejemplos
 - 04.01.02 Definición formal del TDA "PILA"
 - 04.01.03 Implementación de una PILA
- 04.03.01 Implementación estática mediante tipos de datos estructurados.

Unidad 5: Estructuras de Datos Cola

- 05.01 El tipo Cola
 - 05.01.01 Definición y ejemplos
 - 05.01.02 Definición formal del TDA "COLA"
 - 05.01.03 Implementación de una Cola.

05.03.01 Implementación estática mediante tipos de datos estructurados.

Unidad 6: Punteros

06.01 Gestión dinámica de memoria

06.02 Concepto de variable dinámica

06.02.01 Implementación de variables Dinámicas

06.02.02 Generación y destrucción de variables Dinámicas

06.02.03 Operaciones básicas con Punteros

Unidad 7: Programación con Recursividad

07.01 Principios de la recursión.

07.02 Funcionamiento de la recursión.

07.03 Funciones y procedimientos recursivos.

07.04 Ventajas y desventajas de los algoritmos recursivos.

07.05 Tipos de Datos Recursivos

Unidad 8: Estructuras de Datos Lista

08.01 El tipo lista.

08.01.01 Definición y ejemplos.

08.01.02 Definición Formal del TDA "LISTA".

08.01.03 Implementación del tipo lista, estática y dinámica.

08.02 Implementación de Pilas y Colas utilizando lista enlazadas

08.03 Implementación de Listas ordenadas

08.04 Listas Doblemente enlazadas y circulares.

08.05 Multilistas.

Unidad 9: Árboles

09.01 Terminología Fundamental y definiciones.

09.02 El TDA Árbol.

09.03 Implementación de un Árbol Binario.

09.03.01 Implementación estática mediante tipo estructurado de datos.

09.03.02 Implementación dinámica mediante punteros.

09.04 Recorridos en un Árbol Binario.

09.05 Árboles Binarios de Búsqueda.

09.05.01 Búsqueda.

09.05.02 Inserciones.

09.05.03 Supresiones

0.05.04 Árboles AVL

09.06 Temas adicionales sobre árboles

Unidad 10: Grafos

10.01 Definiciones fundamentales

10.02 Representación y uso

10.03 Recorridos de un grafo

Unidad 11: Algoritmos de Ordenación y Búsqueda avanzados: Análisis de Complejidad y Consideraciones de Eficiencia

11.01 Métodos de Ordenación elementales: Inserción, Selección y Burbuja.

11.02 Complejidad y Orden de magnitud de un algoritmo.

11.03 Análisis de los métodos de ordenación elementales.

11.04 Métodos de ordenación eficientes.

11.04.01 Ordenamiento por mezcla.

- 11.04.02 Ordenamiento rápido.
- 11.05 Análisis de los métodos de ordenación eficientes.
- 11.06 Métodos de Búsqueda
- 11.06.01 Búsqueda Secuencial
- 11.06.02 Búsqueda Binaria.
- 11.06.03 Hashing.
- 11.06.03.01 Colisiones.
- 11.06.03.02 Funciones de Hashing.
- 11.06.03.03 Direccionamiento abierto y cerrado.
- 11.07 Análisis de los algoritmos de Búsqueda.

5. RECURSOS NECESARIOS

- Proyector
- Pc
- Laboratorio Informatica
-

6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
Semana 1	Unidad 1	Conceptos de Metodologías en el desarrollo de Sistemas de Software	Ver ítem 7
Semana 2	Unidad 2	Tipos y Estructuras Básicas de Datos (02.01 -02.02)	Ver ítem 7
Semana 3	Unidad 2	Tipos y Estructuras Básicas de Datos (02.03)	Ver ítem 7
Semana 4	Unidad 3	Tipos Abstractos de Datos	Ver ítem 7
Semana 5	Unidad 4	Estructuras de Datos Pila	Ver ítem 7
Semana 6	Unidad 5	Estructuras de Datos Cola	Ver ítem 7
Semana 7	1º Parcial (Unid. 1 - 5) y Unid 6	Punteros	Ver ítem 7
Semana 8	Unidades 6 y 7	Punteros y Programación con Recursividad	Ver ítem 7
Semana 9	Repaso, Rec 1º P y 1º P Teór.
Semana 10	Unidad 8	Estructuras de Datos Lista	Ver ítem 7
Semana 11	Unidad 9	Árboles	Ver ítem 7
Semana 12	Unidad 10	Grafos	Ver ítem 7
Semana 13	Unidad 11	Algoritmos de Ordenación y Búsqueda: Análisis de Complejidad y Consideraciones de Eficiencia	Ver ítem 7
Semana 14	Repaso y 2º Parcial (Uni. 6 a 11)
Semana 15	Rec 2º Parcial y 2º Parcial T.
Semana 16	Clases de Consulta Exa. Final

Semana 17	Informes finales de de los docentes
-----------	-------------------------------------	-------	-------

7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
De Giusti-Madoz y otros	2000	Algoritmos, Datos y Programas	1,2,3,6,7,11 (Básica)	Argentina	Prentice Hall
Cairó O. - Guardati S.	2010	Estructuras de datos - 3º ediciónMc Graw Hill	2 a 11 (Básica)	Argentina	Mc Graw Hill
Kruse Robert	1998	Estructuras de Datos y Diseño de Programas	2 a 11 (Básica)	Argentina	Prentice Hall
Weiss Mark	1995	Estructura de Datos y Algoritmos	2 a 11 (Básica)	Argentina	Addison-Wesley
: Helman P.- Veroff R.	1991	Intermediate Problem Solving and Data Structures	5 a 11 (Básica)	Estados Unidos	Walls and Mirror
Dalle N. - Lilly S.	1990	Pascal y Estructuras de Datos	1 a 11 (Básica)	Argentina	Mc Graw Hill
Aho -Hopcroft - Ulman	1998	Estructuras de Datos y Algoritmo	2 a 10 (Complement.)	España	Pearson-Addison Wesley
Joyanes Aguilar	1996	Programación en Turbo Pascal- (ver. 5.5.-6.0.7.0)	2 a 10 (Complement.)	Argentina	Mc Graw Hill

Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	