



Programa de la Asignatura:

Algoritmos y Programación III



Código: 15

Carrera: **Ingeniería en Computación** Plan: **2013** Carácter: **Obligatoria**
Unidad Académica: **Secretaría Académica** Curso: **Segundo Año – Segundo Cuatrimestre**
Departamento: **Ingeniería** Carga horaria total: **64 hs.** Carga horaria semanal: **4 hs.**
Resolución de Problemas de Ingeniería: **50 %** Formación teórica: **50 %**
Actividades de Proyecto y Diseño: **00 %** Formación experimental: **00 %**
Práctica Supervisada: **00 %**

Materias Correlativas Obligatorias

- Estructura de Datos (10)

Cuerpo Docente

Azurra, Diego
Henrión, Guillermo

Índice

- Fundamentación pág. 2
- Encuadre y articulación de la asignatura pág. 2
 - Encuadre dentro del Plan de Estudios pág. 2
 - Articulación Horizontal pág. 2
 - Articulación Vertical pág. 2
- Objetivos pág. 2
 - Objetivo General pág. 2
 - Objetivos Específicos pág. 2
- Contenidos mínimos pág. 3
- Programa analítico pág. 3
- Bibliografía básica pág. 4
- Bibliografía de consulta pág. 4
- Metodología del aprendizaje pág. 4
 - Desarrollo de la asignatura pág. 4
 - Dinámica del dictado de las clases pág. 4
 - Trabajos prácticos pág. 5
- Metodología de evaluación pág. 5
- Planificación pág. 5
- Información de versiones pág. 6

AÑO ACADÉMICO 2013

ÚLTIMA REVISIÓN 02/07/2013

Firma Docente

Firma Coordinador

1. FUNDAMENTACION

La asignatura está incluida entre las materias que brindará a los alumnos los conocimientos necesarios para poder desarrollar soluciones programáticas, utilizando un lenguaje de programación de bajo nivel para delinear la forma de la resolución de los problemas planteados. Las materias anteriores utilizaron lenguajes de alto nivel para enfocarse en la resolución de problemas, esta aporta una mirada enfocada en la implementación subyacente, sus ventajas y desventajas. Siguiendo con el planteo de Algoritmos y Programación I, se aplica el Currículum Invertido de Meyer, en este caso bajando el nivel de abstracción.

2. ENCUADRE Y ARTICULACIÓN DE LA ASIGNATURA

La asignatura está ubicada en la currícula de la carrera en la parte inicial de la misma que corresponde al "Ciclo Profesional".

En esa etapa, los alumnos profundizarán su experiencia en el área de la programación.

Esta materia tiene como correlativa a Estructura de Datos.

Articulación Vertical

Articula con las materias similares de cuatrimestre anteriores (Algoritmos y Programación I y Algoritmos y Programación II) y cerrará un ciclo de aprendizaje en las disciplinas de Algoritmia y Programación con la asignatura Programación Avanzada.

3. OBJETIVOS

Objetivo General

La cátedra se ha fijado como "objetivo cognoscitivo" de esta materia:

"Lograr que los alumnos conozcan las implementaciones subyacentes al código que programan, comparando y evaluando la eficiencia temporal y espacial de diferentes alternativas disponibles".

Objetivos Específicos

Luego de cursar esta asignatura el alumno podrá:

- Conocer la implementación y organización de la memoria
- Implementar estructuras dinámicas de datos.
- Administrar memoria y comprender las ventajas y desventajas de la administración por parte del programador.
- Desarrollar código reusable en términos de su estructura interna y representación.
- Analizar un sistema en tiempo de ejecución.

4. CONTENIDOS MÍNIMOS

Solución de problemas en lenguajes de bajo nivel. Representación física de la memoria. Punteros y Referencias. Representación de bajo nivel de tipos de datos atómicos y estructurados. Código independiente de la posición (PIC). Bibliotecas y herramientas de reuso.

5. PROGRAMA ANALÍTICO

UNIDAD 1: Introducción

Representación de bajo nivel de datos y algoritmos. Almacenamiento de tipos de datos. Representación y organización física de la memoria. Representación de bajo nivel de tipos de datos atómicos y estructurados.

UNIDAD 2: Punteros

Referencias y punteros. Administración de la memoria. Primitivas. Modos de administración de la memoria. Comparación con la recolección de basura. Aritmética de punteros. Errores típicos en uso de memoria.

UNIDAD 3: Algoritmia

Representación de algoritmos. Estructuras de control de bajo nivel. Llamadas a funciones. Saltos. Ligadura temprana y tardía. Implementaciones alternativas. Optimizaciones.

UNIDAD 4: Reutilización de código

Condiciones de la reutilización de código. Código independiente de la posición. Bibliotecas y herramientas de reutilización. Empaquetamiento y documentación de código.

UNIDAD 5: Análisis en tiempo de ejecución

Transformación de tipos. Uso de información de tipo en tiempo de ejecución. Reflexión. Ventajas y riesgos. Objetos estáticos y referencias a objetos. Inicialización y liberación de memoria. Copia y clonación. Igualdad e identidad de objetos. Profiling y evaluación en tiempo de ejecución.

6. BIBLIOGRAFÍA BÁSICA

- Stroustrup, Bjarne, "El lenguaje de programación C++", edición especial, Addison-Wesley, 2009.
- AHO, ALFRED, Estructura de datos y algoritmos, Addison-Wesley, 1988

7. BIBLIOGRAFÍA DE CONSULTA

- MEYER , BERTRAND, Construcción de Software Orientada a Objetos, Prentice-Hall, 1985, 2da. Edición 1997.
- ESAKOV, JEFFREY, WEISS, TOM, Data structures, An advanced approach using C, Prentice-Hall, 1989

8. METODOLOGÍA DEL APRENDIZAJE

8.a DESARROLLO DE LA ASIGNATURA

Inicialmente, se introducirá al alumno a la representación de bajo nivel (cercana al hardware) de los conceptos que maneja desde materias anteriores. En esa misma línea, se desarrollarán los conceptos de punteros y manejo de memoria por el programador, comparándolos con los modelos de referencias y recolección de basura vistos en materias anteriores.

Finalmente, se estudiarán representación de algoritmos y estructuras de control de bajo nivel, reutilización de código y análisis en tiempo de ejecución del comportamiento de un sistema.

8.b DINÁMICA DEL DICTADO DE LAS CLASES

Para poder llevar adelante los objetivos de la cátedra, se hace un gran énfasis hacia la carga horaria de las aproximaciones a problemas, dando tanto peso a éstas como a la teoría.

El Profesor a cargo del curso se ocupará en forma personal y semanal del dictado de aquellos temas con un fuerte contenido teórico y que significan conceptos básicos y poco volátiles en la especialidad. Procederá a describir técnicas, características y pondrá ejemplos. Éste generará un ámbito de reflexión y discusión de los temas presentados, para que mediante la intervención de los alumnos, se puedan aclarar aquellos aspectos que el docente puede captar

a través de las consultas recibidas, como los que han resultado de más compleja comprensión. También deberá discutir las distintas soluciones que se presentan en muchos casos, y mostrar ventajas y desventajas.

Habrà una parte de la clase dedicada a la aplicación de los conceptos vistos durante la parte teórica. Se desarrollarán problemas con creciente nivel de dificultad.

8.c TRABAJOS PRÁCTICOS

Habrà dos trabajos prácticos durante el cuatrimestre, los cuales tendrán como objetivo resolver un problema algorítmico novedoso, para el cual los alumnos deberán profundizar sobre los conceptos vistos.

Los mismos serán de consigna variante según el año en curso, lo cual ayuda a darle flexibilidad a la cátedra.

• METODOLOGÍA DE EVALUACIÓN

9.a NORMAS DE EVALUACIÓN.

- El criterio es que la evaluación del alumno es permanente.
- Se tomarán dos exámenes parciales teórico/prácticos pudiendo acceder a un recuperatorio.
- Las notas de los parciales representan los resultados de la evaluación teórico/práctica.
- Los exámenes parciales y sus recuperatorios serán escritos.

9.b RÉGIMEN DE APROBACIÓN DE LA MATERIA.

- Para la aprobación de la materia los alumnos deberán tener los dos parciales aprobados, teniendo la posibilidad de recuperar cada uno de ellos en una única oportunidad adicional, en la fecha acordada con los docentes.
- Además los alumnos deberán aprobar los trabajos prácticos, como condición para la aprobación de la materia.
- Los alumnos que obtengan una nota inferior a cuatro puntos se les asignará la nota insuficiente y deberán recurrir a la materia.

• PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Unidad 1
Semana 2	Unidad 2
Semana 3	Unidad 2
Semana 4	Unidad 2
Semana 5	Unidad 3
Semana 6	Unidad 3
Semana 7	Unidad 3
Semana 8	Primer Parcial
Semana 9	Unidad 4
Semana 10	Unidad 4
Semana 11	Unidad 5
Semana 12	Unidad 5
Semana 13	Segundo Parcial
Semana 14	Repaso

Semana 15	Recuperatorios
Semana 16	Cierre de la materia
Del al de	FINAL

Información de Versiones	
Nombre del Documento:	Ficha Académica de la asignatura
Nombre del Archivo	Plan 2013 – Algoritmos y Programación III.docx
Documento origen:	
Elaborado por:	Diego Fontdevila
Revisado por:	Alejandro Oliveros
Aprobado por:	Alejandro Oliveros
Fecha de Elaboración:	02/07/2013
Fecha de Revisión:	03/07/2013
Fecha de aprobación	03/07/2013
Versión:	1.0