



Programa de la Asignatura:

Algoritmos y Programación I



Código: 01

Carrera: **Ingeniería en Computación** Plan: **2013** Carácter: **Obligatoria**
Unidad Académica: **Secretaría Académica** Curso: **Primer Año – Primer Cuatrimestre**
Departamento: **Ingeniería** Carga horaria total: **128 hs.** Carga horaria semanal: **8 hs.**
Resolución de Problemas de Ingeniería: **40 %** Formación teórica: **50 %**
Actividades de Proyecto y Diseño: **00 %** Formación experimental: **10 %**
Práctica Supervisada: **00 %**

Materias Correlativas Obligatorias

- Ninguna

Cuerpo Docente

Mg. Ing. Diego Fontdevila
Ing. Lucas Videla

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. 3
- Contenidos mínimos pág. 3
- Programa analítico pág. 3
- Bibliografía básica pág. 4
- Bibliografía de consulta pág. 5
- Metodología del aprendizaje pág. 5
 - Desarrollo de la asignatura pág. 5
 - Dinámica del dictado de las clases pág. 5
 - Trabajos prácticos pág. 5
- Metodología de evaluación pág. 6
- Planificación pág. 6
- Información de versiones pág. 7

AÑO ACADÉMICO 2013

ÚLTIMA REVISIÓN 02/07/2013

Firma Docente

Firma Coordinador

1. FUNDAMENTACION

La asignatura está incluida entre las materias que brindarán a los alumnos los conocimientos necesarios para poder desarrollar soluciones programáticas, utilizando un lenguaje de programación de alto nivel para delinear la forma de la resolución de los problemas planteados.

El avance del mercado y del estado del arte ha hecho indispensable el conocimiento por parte del alumno de métodos para programar computadoras, comunicándole instrucciones que luego se convertirán en algoritmos, y más tarde en sistemas completos que solucionarán los problemas humanos que tienen que ver con el manejo de la información, su obtención, tratamiento y distribución, automatización de procesos, control, entre otros.

Es por ello que esta materia proporciona una base sólida y a su vez aplicable, de tal modo de poder verter conceptos no solamente académicos sino también presentes en la industria. Así se ha optado por tomar el enfoque que la materia presenta, articulado con las materias posteriores.

Dicho enfoque toma como base la necesidad de comprender en primera instancia los pasos para la resolución de problemas, la identificación de los elementos componentes, y la posibilidad de coordinarlos e instruirlos de modo que cooperen (por medio de la programación proporcionada) para encontrar la solución al problema establecido en primera instancia. Se adopta la programación orientada a objetos como el inicio del camino que se desempeñará a lo largo de los dos primeros años de la carrera. Esto hace referencia al Currículum Invertido que propone Bertrand Meyer por el cual se programa desde afuera hacia adentro, comenzando por la utilización de componentes y con el correr del tiempo se llega a especificaciones de más bajo nivel.

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 comenzarán a adquirir experiencia en el área de la programación.

Requieren nociones de los conceptos de lógica.

Esta materia no presenta correlatividades.

3. Articulación Horizontal

En cuanto a la articulación horizontal se ha planeado que esta asignatura se curse en simultáneo con Álgebra I, Análisis Matemático I y Física I, de modo que el alumno pueda integrar una perspectiva de usos inmediatos y aplicaciones de los conocimientos que va desarrollando, pudiendo programar sus propias soluciones a problemas de las otras asignaturas.

4. Articulación Vertical

Esta asignatura brinda las bases para comenzar a plasmar soluciones de programación en forma de software, por lo que verticalmente no se articula con ninguna asignatura previa.

A esta materia le seguirá en un curso superior la asignatura Algoritmos y Programación II que proveerá al alumno de los conocimientos más avanzados y complejos de la programación orientada a objetos, algoritmia y estructuras de datos.

5. OBJETIVOS

Objetivo General

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

“Lograr que los alumnos conozcan los principios básicos de la resolución de problemas, el funcionamiento de un lenguaje de programación; adicionalmente se espera que reconozcan y puedan emplear los conceptos básicos de la programación orientada a objetos, y que comiencen a asir el vocabulario de la profesión”.

Objetivos Específicos

Luego de cursar esta asignatura el alumno deberá dominar los siguientes temas:

- Preparar los esquemas mentales para la resolución de problemas
- Conocer conceptos básicos de programación
- Aprender un lenguaje de programación de alto nivel
- Identificar cuestiones de ingeniería de software que aplican a grandes sistemas
- Iniciarse y mantener una cultura de trabajo en equipo

6. CONTENIDOS MÍNIMOS

Sistema como conjunto de algoritmos. Datos e información. Algoritmo. Paradigmas. Abstracción y Modelos. Resolución de problemas. Refinamientos sucesivos. Soluciones por medio de procesos, objetivos y estrategias. Validación de soluciones. Composición y Descomposición. Tipos de datos atómicos y estructurados. Tipos de Datos Abstractos. Objetos. Uso de objetos. Construcción de objetos. Modelado de problemas con objetos. Programación basada en objetos como caso superador de la programación estructurada. Clases y objetos. Identificación de objetos por su comportamiento. Tipos de datos estructurados asociativos (objetos) y calculables (vectores). Estructuras de control de flujo de datos para cada secuencia: secuencia, iteración, selección por decisión y selección múltiple. Refinamientos sucesivos. Modularización. Generalización del concepto de módulo: conceptos de programa y sistema. Concepto de módulo a través de la historia. Implantación de los módulos: paquete, clase, procedimiento, función, subrutina y subprograma. Interfaces entre módulos. Globalidad y localidad. Organización de la memoria. Atributos, Parámetros y argumentos. Encapsulamiento. Acoplamiento y cohesión. Referencias. Introducción al concepto de selección de lenguajes de programación. Flexibilidad del software. Claridad del código. Corrección del software. Diseño por contrato. Precondiciones y Postcondiciones. Invariantes de representación. Axiomas. Manejo de excepciones. Pruebas unitarias de software.

7. PROGRAMA ANALÍTICO

Unidad 1: Introducción a la resolución de problemas

Hardware. Software. Rol del profesional de la programación. El software como construcción intelectual. Contexto del software. Usuarios. Computadora. Tareas que desarrollan: Almacenamiento, recuperación, operaciones e intercambio. Máquina de Von Neumann. Memorias. Procesador. Dispositivos de comunicación. Resolución de problemas. Soluciones: Analogía, Contraposición, Experimentación. Validación. Comunicación de las soluciones

Unidad 2: Introducción a la Programación con Objetos

Datos e Información. El lenguaje de la solución. Secuencia de pasos. Algoritmos. Lenguaje de programación: Qué es, Sintaxis, Semántica, Sentencias. Código fuente. Conceptos de Compilador e Intérprete.

Definición de Objetos. Tipos de objetos. Primeros pasos. Variables. Componentes: Nombre, Valor, Tipo. Declaración. Asignación. Semántica de la asignación. Métodos: Sintaxis y Semántica.

Unidad 3: Algoritmos

Concepto. Algoritmos vs Recetas. Ejemplos. Características. Definición. Reglas. Variables. Tipo de dato. Ejemplos. Uso de objetos. Métodos de consulta. Sintaxis. Semántica. Parámetros. Sintaxis. Semántica.

Unidad 4: Clases

Concepto. Objetos y clases. Declaración. Inicialización. Construcción. Estructura. Sintaxis.

Unidad 5: Lógica

Concepto. Verdad, Creencia, Justificación. Valores de verdad. Lógica proposicional. Operaciones básicas (AND, OR, XOR, NOT). Propositiones categóricas. Álgebra de Boole. Leyes de De Morgan. La lógica en un lenguaje de programación. Ejemplos.

Unidad 6: Estructuras de Control

Introducción. Secuencia. Repetición. Condiciones. Ejemplos. Continuidad y ruptura de secuencia. Adyacencia en las sentencias. Proximidad temporal. Condición, estructura Condicional. Iteración indefinida, definida.

Unidad 7: Tipos de Variables

Atributos. Parámetros. Variables locales. Explicación. Ejemplos.

Unidad 8: POO

Modelado de problemas. Objetos. Clasificación. Tipos predefinidos. Tipos personalizados. Relación de uso entre clases. Miembros: Métodos (Consultas y Comandos), Atributos, Constructores.

Unidad 9: POO II

Ocultamiento de la Información. Encapsulamiento.

Unidad 10: Programación Orientada al Contrato

Modularidad. Programación sobre interfaces. Concepto. Comentarios. Métodos y Atributos. Interface e Implementación. Diseño por Contrato. Composición. "this". Modelo Eiffel. Modelo Java. Rupturas de contrato.

Unidad 11: Pruebas

Calidad. Pruebas: Concepto. Roles para la confección y ejecución de pruebas. Error. Defecto. Falla. Detalle de pruebas. Tipos de pruebas. Conveniencias. JUnit: Historia, usos. Ejemplo.

Unidad 12: Desarrollo Guiado por Pruebas (TDD)

Concepto. Espíritu. Contraposición al paradigma actual. Ejemplo.

Unidad 13: Arreglos

Datos atómicos. Datos estructurados: Calculables, Asociativos. Masividad del término. Arreglos. Definición. Acceso y recorrido.

Unidad 14: Arreglos multidimensionales

Arreglos. Generalización. Regulares. Irregulares. Inicialización. Recorrido.

Unidad 15: Memoria

Tareas. Modelo de Von Neumann. Componentes. Datos. Información. Datos y Programas. Tipos de Memoria. Volátil. No Volátil. Central. Secundaria. Segmentos de Memoria: CS, DS, SS, ES. Memoria dinámica. Asignación. El operador new. Liberación. Garbage Collection. Esquema de Memoria.

8. BIBLIOGRAFÍA BÁSICA

- MEYER , BERTRAND, Construcción de Software Orientada a Objetos, Prentice-Hall, 1985, 2da. Edición 1997.
- MEYER, BERTRAND, Touch of class, Springer, 2009.

9. BIBLIOGRAFÍA DE CONSULTA

- KNUTH, DONALD, The Art of Computer Programming, volumen 3, Sorting and Searching, 1973
- AHO, ALFRED, Estructura de datos y algoritmos, Addison-Wesley, 1988
- ARNOLD, GOSLING, HOLMES, Holmes, El lenguaje de programación Java, Addison-Wesley, 3ra. Edición 2002.
- ECKEL, BRUCE, Pensando en Java, Prentice-Hall, 2006

10. METODOLOGÍA DEL APRENDIZAJE

8.a DESARROLLO DE LA ASIGNATURA

Inicialmente se tratará de familiarizar al alumno con los conceptos básicos que se deben dominar para poder acceder al entendimiento de las bases de la programación. Para ello es necesario que el alumno adquiera una primera aproximación a la resolución de problemas, especialmente a aquellos del tipo algorítmico.

Se utilizará un “juego didáctico” basado en naves espaciales y comandos básicos que ilustran las bases de la programación orientada a objetos. El mismo tiene como objetivo la correcta asimilación de los métodos de composición de algoritmos.

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 (que a estas alturas, es totalmente nueva para el alumno).

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.

Asimismo habrá 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

Habrán dos trabajos prácticos durante el cuatrimestre, los cuales tendrán como objetivo enfatizar los conceptos de resolución de problemas, establecimiento de vocabulario adecuado, formulación de algoritmos y explicación de soluciones.

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

11. 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 recursar la materia.
- Los alumnos que obtengan una nota superior a siete puntos, se les tomará la materia como promocionada, eximiéndolos del examen final.

12. PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Unidad 1
Semana 2	Unidad 2
Semana 3	Unidad 3
Semana 4	Unidad 4 y 5
Semana 5	Unidad 6 y 7
Semana 6	Unidad 8 y 9
Semana 7	Unidad 10
Semana 8	Primer Parcial
Semana 9	Unidad 11 y 12
Semana 10	Unidad 13
Semana 11	Unidad 14
Semana 12	Unidad 15
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 I.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