



Programa de la Asignatura:

# Estructuras de Datos I



Código: 743

Carrera: <b>Ingeniería en Computación</b>	Plan: <b>2008</b>	Carácter: <b>Obligatoria</b>
Unidad Académica: <b>Secretaría Académica</b>	Curso: <b>Primer Año – Primer cuatrimestre</b>	
Departamento: <b>Ingeniería</b>	Carga horaria total: <b>60 hs.</b>	Carga horaria semanal: <b>4 hs.</b>
Formación Experimental: <b>10 %</b>	Formación teórica: <b>50 %</b>	Formación práctica: <b>40 %</b>

### Materias Correlativas Obligatorias

- -----
- -----
- -----

### Cuerpo Docente

Fontdevila, Diego  
Videla, Lucas

### Índice

• Fundamentación	pág. 2
• Encuadre y articulación de la asignatura	pág. 2
➤ Encuadre dentro del Plan de Estudios	pág. 2
➤ Articulación Horizontal	pág. 3
➤ Articulación Vertical	pág. 3
• Objetivos	pág. 3
➤ Objetivo General	pág. 3
➤ Objetivos Específicos	pág. 3
• Contenidos mínimos	pág. 3
• Programa analítico	pág. 3
• Bibliografía básica	pág. 5
• Bibliografía de consulta	pág. 5
• Metodología del aprendizaje	pág. 5
➤ Desarrollo de la asignatura	pág. 5
➤ Dinámica del dictado de las clases	pág. 6
➤ Trabajos prácticos	pág. 7
• Metodología de evaluación	pág. 7
• Planificación	pág. 8
• Información de versiones	pág. 8

AÑO ACADÉMICO 2013

ÚLTIMA REVISIÓN 07/06/2013

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 (en este caso, Java) 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 completos sistemas que solucionarán los problemas humanos que tienen que ver con el manejo de la información, su obtención, tratamiento y distribución, entre otros.

Es por ello que debe proporcionarse una base sólida y a su vez aplicable, de tal modo que se puedan 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**

### **Encuadre dentro del Plan de Estudios**

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.

### **Articulación Horizontal**

En cuanto a la articulación horizontal se ha planeado que esta asignatura se curse en simultáneo con Álgebra I, Matemática Discreta 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.

### **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. Por otra parte, a esta materia le seguirá en un curso superior la asignatura Lenguaje de Programación I que proveerá al alumno de los conocimientos más avanzados y complejos de lo que hace a la práctica y escritura de algoritmos; también articula con Estructura de Datos II, que profundiza en la confección y estudio de los tipos de dato más ampliamente reconocidos, basándose fuertemente en los lineamientos marcados por Estructura de Datos I.

### 3. 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: Java
- Identificar cuestiones de ingeniería de software que aplican a grandes sistemas
- Iniciarse y mantener una cultura de trabajo en equipo

### 4. CONTENIDOS MÍNIMOS

Conceptos básicos de construcción de algoritmos. Soluciones informales. Seudocódigo normalizado y lenguajes. Lógica. Estructuras de control de flujo de datos. Modularización. Tipos de datos abstractos (TDA).

### 5. 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). Proposiciones 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. Comentarios. Métodos y Atributos. Interface e Implementación. Diseño por Contrato. Composición. "this".

## **6. BIBLIOGRAFÍA BÁSICA**

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

## **7. BIBLIOGRAFÍA DE CONSULTA**

- ARNOLD, GOSLING, HOLMES, Holmes, El lenguaje de programación Java, Addison-Wesley, 3ra. Edición 2002.
- ECKEL, BRUCE, Pensando en Java, Prentice-Hall, 2006

## **8. 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.

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

## 10. PLANIFICACIÓN

CALENDARIO DE CLASES Y EVALUACIONES	
Semana 1	Unidad 1
Semana 2	Unidad 1 – Práctica
Semana 3	Unidad 2
Semana 4	Unidad 3
Semana 5	Unidad 4
Semana 6	Unidad 5
Semana 7	Unidad 6
Semana 8	Primer Parcial
Semana 9	Unidad 7
Semana 10	Unidad 8

Semana 11	Unidad 9
Semana 12	Unidad 10
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 Estructuras de Datos I
Nombre del Archivo	Estructuras de Datos I.docx
Documento origen:	
Elaborado por:	Diego Fontdevila
Revisado por:	Lucas Videla
Aprobado por:	Diego Fontdevila
Fecha de Elaboración:	01-02-2013
Fecha de Revisión:	07-06-2013
Fecha de aprobación:	10-06-2013
Versión:	1.0