



**DEPARTAMENTO:** *Ingeniería e Investigaciones Tecnológicas*

**ASIGNATURA:** Programación avanzada

**Ingeniería en Informática**  
**2013 cuatrimestral**

### **1. OBJETIVOS**

- **Objetivos Generales:**

- Capacidad de aplicar los conocimientos en la práctica
- Capacidad para organizar y planificar el trabajo
- Capacidad para analizar, plantear y resolver problemas
- Capacidad de trabajo en equipo
- Compromiso para realizar el trabajo con Calidad
- Aceptar el uso de INTERNET como un medio habitual para la realización de consultas, bajar librerías y software.

- **Objetivos Específicos:**

- Proporcionar una comprensión sólida de los conceptos fundamentales del modelo de objetos.
- Lograr que el alumno tenga una visión abstracta y concreta de cada concepto, es decir, que además de entender el concepto en sí, sepa de sus posibles aplicaciones y de su implantación computacional.
- Mejorar la técnica de diseño de algoritmos y su implantación como programas.
- Aceptar a las metodologías iterativas, incremental y en espiral, para el desarrollo del software como parte de la cultura del buen programador.
- Aceptar algunos principios de la programación XP (programación Extrema) como parte de la cultura del buen programador.
- Aceptar algunos principios de PSP (Personal Software Process) como



parte de la cultura del buen programador.

- Reconocer e incluir a la Verificación y Validación de Software como parte fundamental en la actividad del programador.
- Saber realizar buenas estimaciones para el desarrollo del trabajo en base a las métricas personales y grupales para el desarrollo del software.
- Resolver problemas del tipo Nivel 1, 2 y parcialmente del nivel 3 de la OIA (Olimpiadas Informáticas Argentinas)
- Dominio avanzado de algún lenguaje que soporte la programación orientada a objetos
- Conocimientos básicos de la programación funcional y lógica

## 2. CONTENIDOS

### PROGRAMA ANALÍTICO. CONTENIDOS TEÓRICOS Y PRÁCTICOS:

#### **Unidad I. Una Metodología del desarrollo de software (8 horas)**

Metodologías iterativas, incremental, espiralada. Procesos Ágiles: Programación extrema (XP) y Scrum. Métricas del software. Principios de la programación XP; Conceptos de PSP (Personal Software Process). Una metodología propuesta por la cátedra para el desarrollo de software. La preparación del lote de prueba. Documentación del Lote de Pruebas. El programa probador. El programa verificador del Input. Ventajas de preparar la prueba antes de comenzar la programación. Pruebas del software: de caja negra, de caja blanca, inspecciones. Ventajas de la inspección del código fuente.

#### **UNIDAD II. Paradigmas, estilos y Lenguajes de Programación ( 4 horas)**

Paradigmas de programación: Imperativo, Funcional y lógico. Estilos de programación en la programación Imperativa: Estructurada por procedimientos, modular, con objetos y orientada a objetos. Apoyo de un lenguaje para cada estilo de programación. Programación orientada a objetos y lenguajes de programación orientados a objetos.

#### **UNIDAD III El Lenguajes de Programación JAVA ( 28 horas)**

Tipos de datos abstractos. Estructuras de control. Implantación de Estructuras de Datos OO en Java. Sistema de tipos. Datos y funciones miembros. Miembros sta-



tic. Constructores. Las clases complejo, vectorMath y matrizMath. Implantación de las clases correspondiente a los TDA básicos: Clase lista, pila y cola. Resolución de problemas de Nivel 1 y 2 OIA. Recursividad. Eventos. Excepciones.

#### **UNIDAD IV Herencia Y Polimorfismo (8 horas)**

Jerarquías de clases: Generalización y Especialización (“es un”). Composición y Agregación. (“tiene un” y “es parte de”). Herencia simple, clases derivadas. Accesibilidad. Clases y métodos abstractos. Sobrecarga y definición. Aplicación de métodos polimórficos. Niveles de polimorfismo. Diagrama de clases usando UML. Diseño Orientado a Objetos.

#### **UNIDAD V COMPLEJIDAD COMPUTACIONAL. ( 12 horas)**

Medición del tiempo de ejecución de un algoritmo. Análisis de complejidad de algoritmos: Análisis asintótico, comportamiento en el mejor caso, caso promedio y peor caso. Balance entre tiempo y espacio en los algoritmos. La función O. Efectos al aumentar la velocidad del computador y/o al aumentar el tamaño del problema. Comparaciones de los distintos algoritmos desarrollados en los TTPP.

#### **UNIDAD VI Algoritmos (38 horas)**

Algoritmos de ordenamientos y sus implantaciones. Comparaciones de los distintos algoritmos en función de sus operaciones básicas. Diseño y Análisis de Algoritmos. Estabilidad y sensibilidad al input de un algoritmo de ordenamiento. Utilización de estructuras estáticas y dinámicas. Algoritmos de búsqueda. Algoritmos de Grafos: Dijkstra, Floyd, Warshall, Prim, Kruskal, Búsqueda en amplitud, Búsqueda en profundidad y coloreo de grafos. Resolución de problemas reales y ficticios. Concurrencia y Paralelismo. Algoritmos concurrentes, distribuidos y paralelos. Verificación de Algoritmos. Técnicas de diseño de algoritmos.

#### **UNIDAD VII Taller Programación Básica en Java con WebSphere/Rational (32 horas)**

Conceptos básicos de Java. Conceptos y características de J2ME, J2SE, J2EE. Arquitectura de los distintos frameworks. Implementación de un cliente pesado en Java. Implementación de un JavaBean. Acceso a datos.

#### **UNIDAD VIII Programación Lógica ( 16 horas )**

Lógica de predicados. argumentos, interpretación, reglas, preguntas, variables, ámbito de una variable, corte. Nociones básicas de semántica Formal. Estructura de un programa Prolog. Listas y secuencias finitas. Aplicaciones en grafos. Resolución de caminos mínimos. Equivalencia entre el Prolog y los operadores del TDA conjunto. Equivalencia entre Prolog y consultas SQL.



### UNIDAD IIX Programación Funcional ( 4 horas)

Introducción a la programación funcional. Haskell & Hugs. Tipos y clases. Funciones y operadores.

### 3. BIBLIOGRAFÍA

Título	Autor(es)	Editorial	Año Edición	Ejemplares disponibles en UNLaM
Core Java Volumen I – Fundamentos 7ma Edición	Horstmann Cornell	Pearson Prentice Hill	2005	1
Core Java Volumen I – Fundamentos 7ma Edición	Horstmann Cornell	Pearson Prentice Hill	2005	2 de la 2da edición y 7 de la 4ta edición
Java, cómo programar. 7ma Edición	Deitel -Deitel	Pearson Prentice Hill. Séptima edición	2008	2 de la 2da edición y 7 de la 4ta edición
Manual de Java	Naughton P	McGraw –Hill	2002	1
Estructuras de datos y algoritmos.	Aho, Hopcroft, j Ullman.	Addison-Wesley Iberoamericana	1983	5
Algoritmos en C++	Sedgewick	Addison-Wesley	1995	5
Introducción a la Programación en Java	D. Arnow-G. Weiss.	Addison Wesley	2001	0
Orientación a Objetos con Java y UML	FONTELA CARLOS	NUEVA LIBRERIA	2003	
<a href="http://www.youtube.com/playlist?p=PLAA2F26B4E3985B9E">http://www.youtube.com/playlist?p=PLAA2F26B4E3985B9E</a>	G. Dejean		2011	
<a href="http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf">http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf</a>	Henrik Kniberg		2007	