



MODALIDAD ACADÉMICA

Asignatura	Diseño de Lenguajes de Consulta – Electiva Cuarto Nivel	
Carrera	INGENIERÍA EN SISTEMAS DE INFORMACIÓN	
Ciclo Lectivo	2014	
Vigencia del programa	Desde el ciclo lectivo 2014	
Plan	2008	
Nivel	☐ 3er. Nivel ☒ 4to. Nivel ☐ 5to. Nivel	
Coordinador de Cátedra	Ing. Valerio Frittelli	
Área de Conocimiento	☒ Programación ☐ Computación ☐ Sistemas de Información ☐ Gestión Ingenieril ☐ Modelos ☐ Complementaria	
Carga horaria semanal	8 horas cátedra.	
Anual/ cuatrimestral	Cuatrimestral	
Modalidad de Dictado	Presencial, con fuerte uso de recursos de aula virtual. Todas las clases se llevan a cabo en los laboratorios de informática de la Facultad.	
Correlativas para cursarla	Regulares	Aprobadas
	Tecnología de Software de Base [Electiva Tercer Nivel]	Paradigmas de Programación
Correlativas para rendirla	Regulares	Aprobadas
	Tecnología de Software de Base [Electiva Tercer Nivel]	Tecnología de Software de Base [Electiva Tercer Nivel]
Fundamentación y Relación con el Perfil Profesional	De acuerdo a lo expresado por la Ordenanza 1150 que reglamenta la carrera, en la sección 2. Perfil Profesional se indica que "el Ingeniero de Sistemas de Información es un profesional de sólida formación analítica que le permite la interpretación y resolución de problemas mediante el empleo de metodologías de sistemas y tecnologías de procesamiento de información". En ese sentido, la asignatura es la correlativa natural de Tecnología de Software de Base [Electiva Tercer Nivel] y aporta nuevos elementos sólidos y profundos sobre análisis y diseño de algoritmos, diseño y desarrollo de estructuras de datos avanzadas, estudio y aplicación de patrones de diseño, y formación práctica en una plataforma de desarrollo profesional.	
Objetivos de la Asignatura	Que el estudiante: - Comprenda, defina y aplique estructuras de datos avanzadas, junto con los algoritmos asociados a ellas, para la resolución de problemas y situaciones complejas. - Incorpore elementos de análisis y diseño de algoritmos, para medir la eficiencia relativa de distintos algoritmos que pudieran servir para el mismo problema. - Implemente eficientemente las soluciones planteadas, aplicando buenas prácticas y patrones de diseño, mediante una plataforma de programación de uso profesional actual, para facilitar la futura	



inserción del estudiante en el mundo laboral.

Programa Analítico

Unidad N° 1: Estrategias de Resolución de Problemas.

Objetivos específicos:

Que el estudiante adquiera dominio en técnicas generales de resolución de problemas, así como en el análisis de la eficiencia de los algoritmos que apliquen esas técnicas.

Contenidos:

Introducción a los fundamentos de la estrategias de fuerza bruta, divide y vencerás, algoritmos ávidos, programación dinámica y algoritmos de base aleatoria. Aplicaciones basadas en la estrategia Divide y Vencerás: el algoritmo Merge Sort. Relaciones de recurrencia. El Teorema Maestro del Análisis de Algoritmos y sus aplicaciones.

Bibliografía:

- 1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar" . México: Prentice Hall. ISBN: 970-26-0518-0 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 4.) Frittelli, V. (2013). "Fichas de Clase para TSB". Córdoba: UTN FRC [\[disponible en el aula virtual de la asignatura\]](#)
- 5.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [\[disponible en biblioteca central\]](#)
- 6.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [\[disponible en biblioteca central\]](#)
- 7.) Weiss, M. A. (2000). "Estructuras de Datos en Java – Compatible con Java 2". Madrid: Addison Wesley. ISBN: 84-7829-035-4 [\[disponible en biblioteca central\]](#)
- 8.) Documentación oficial de la plataforma Java (Javadocs).

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.

Unidad N° 2: Grafos

Objetivos específicos:

Que el estudiante adquiera dominio sobre las principales aplicaciones sobre grafos, la formas de implementarlos y los algoritmos más comunes aplicables sobre ellos.

**Contenidos:**

Concepto de Grafo. Vértices y aristas. Tipos de grafos: dirigido, no dirigidos y ponderados. Implementación matricial e implementación enlazada. Problemas básicos: cierre transitivo (algoritmo de Warshall), determinación de existencia de ciclos, recorrido en profundidad y en amplitud, recorrido de componentes conexas, determinación del árbol de expansión mínimo (Algoritmo de Prim), camino más corto de un nodo a cada uno de los otros nodos (Algoritmo de Dijkstra) y recorrido de componentes fuertemente conexas (algoritmo de Kosaraju). Estructuras de Unión-Pertenencia (Union-Find) y su aplicación en el problema del árbol de expansión mínimo (algoritmo de Kruskal). Aplicación de algoritmos de Base Aleatoria: el algoritmo de Karger para el problema del Corte Mínimo en un Grafo no dirigido.

Bibliografía:

- 1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar" . México: Prentice Hall. ISBN: 970-26-0518-0 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 4.) Frittelli, V. (2013). "Fichas de Clase para TSB". Córdoba: UTN FRC [\[disponible en el aula virtual de la asignatura\]](#)
- 5.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [\[disponible en biblioteca central\]](#)
- 6.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [\[disponible en biblioteca central\]](#)
- 7.) Weiss, M. A. (2000). "Estructuras de Datos en Java – Compatible con Java 2". Madrid: Addison Wesley. ISBN: 84-7829-035-4 [\[disponible en biblioteca central\]](#)
- 8.) Documentación oficial de la plataforma Java (Javadocs).

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.

Unidad N° 3: Gestion de Archivos Binarios en Java.**Objetivos específicos:**

Que el estudiante adquiera dominio en técnicas de acceso directo a contenidos de archivos.

Que el estudiante conozca la forma de aplicar técnicas de búsqueda sobre datos en disco.

Que el estudiante adquiera dominio en el diseño de sistemas de archivos adaptables a diversas situaciones y necesidades de persistencia.

**Contenidos:**

Archivos de texto y Archivos binarios. Estructura de un Archivo Binario. Modos de apertura. Lectura y grabación. Posicionamiento directo (seeking). Gestión ABM: bajas lógicas, bajas físicas, altas, modificaciones. Búsqueda en archivos. Archivos con estructura hash. Diseño de un marco de trabajo integral para gestión de archivos que incluya acceso directo, hashing, ordenamiento y otras operaciones generales.

Bibliografía:

- 1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar" . México: Prentice Hall. ISBN: 970-26-0518-0 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 4.) Folk, M., Zoellick, B., Riccardi, G. (1998). "File Structures – An Object Oriented Approach with C++". Reading: Addison Wesley. ISBN: 0-201-87401-6 [\[disponible en biblioteca central\]](#)
- 5.) Horstmann, C., y Cornell G. (2000). "Core Java 2 – Volume I: Fundamentals". (Disponible en español) Upper Saddle River: Prentice Hall. ISBN: 84-205-4832-4 [\[disponible en biblioteca central\]](#)
- 6.) Horstmann, C., y Cornell G. (2001). "Core Java 2 – Volume II: Advanced Features". (Disponible en español) Palo Alto: Prentice Hall. ISBN: 84-8322-310-4 [\[disponible en biblioteca central\]](#)
- 7.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [\[disponible en biblioteca central\]](#)
- 8.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [\[disponible en biblioteca central\]](#)
- 9.) Weiss, M. A. (2000). "Estructuras de Datos en Java – Compatible con Java 2". Madrid: Addison Wesley. ISBN: 84-7829-035-4 [\[disponible en biblioteca central\]](#)

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.

Unidad N° 4: Compresión de Archivos.**Objetivos específicos:**

Que el estudiante adquiera dominio de las principales técnicas de compresión existentes.

Que el estudiante comprenda y desarrolle aplicaciones de la estrategia de resolución de problemas basada en algoritmos ávidos.

Contenidos:

Algoritmos Ávidos. Aplicación: el problema de la minimización de la longitud promedio de la



representación binaria de los símbolos de un mensaje. Desventajas de los sistemas de codificación de longitud constante. Sistemas de codificación de longitud variable: el Código de Huffman. Árboles Estrictamente Binarios y sus propiedades. Árbol de Huffman. Algoritmo de Huffman para compresión de datos. Aproximación a otras técnicas modernas para compresión de datos: Huffman bytes. Huffman sobre palabras. Ziv – Lempel. Gestión de bits en Java: operadores de nivel de bits y sus aplicaciones.

Bibliografía:

- 1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar" . México: Prentice Hall. ISBN: 970-26-0518-0 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 4.) Horstmann, C., y Cornell G. (2000). "Core Java 2 – Volume I: Fundamentals". (Disponible en español) Upper Saddle River: Prentice Hall. ISBN: 84-205-4832-4 [\[disponible en biblioteca central\]](#)
- 5.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [\[disponible en biblioteca central\]](#)
- 6.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [\[disponible en biblioteca central\]](#)
- 7.) Weiss, M. A. (2000). "Estructuras de Datos en Java – Compatible con Java 2". Madrid: Addison Wesley. ISBN: 84-7829-035-4 [\[disponible en biblioteca central\]](#)

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.

Unidad N° 5: Recuperación de Información: Motores de Búsqueda.

Objetivos específicos:

Que el estudiante adquiera conocimiento y dominio sobre los principales modelos existentes para la búsqueda y recuperación de información en ambientes de datos no estructurados, y sobre la forma de diseñar y desarrollar un motor de búsqueda para esos contextos.

Contenidos:

Recuperación de Datos vs. Recuperación de Información. Modelos matemáticos de soporte para la Recuperación de Información: el modelo Booleano y el modelo Vectorial. Conceptos del modelo Vectorial: términos y documentos, factor de peso de un término, fórmulas aceptadas para el factor de peso. Estructuras de datos para implementar el modelo vectorial en un motor de búsqueda: diccionario, listas de posteo, índices. Técnicas de búsqueda en un índice ya generado: ranking y recuperación de documentos que encajen con una consulta.



Bibliografía:

- 1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar" . México: Prentice Hall. ISBN: 970-26-0518-0 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 4.) Horstmann, C., y Cornell G. (2000). "Core Java 2 – Volume I: Fundamentals". (Disponible en español) Upper Saddle River: Prentice Hall. ISBN: 84-205-4832-4 [\[disponible en biblioteca central\]](#)
- 5.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [\[disponible en biblioteca central\]](#)
- 6.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [\[disponible en biblioteca central\]](#)
- 7.) Weiss, M. A. (2000). "Estructuras de Datos en Java – Compatible con Java 2". Madrid: Addison Wesley. ISBN: 84-7829-035-4 [\[disponible en biblioteca central\]](#)

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.

Unidad N° 6: Elementos Avanzados.

Objetivos específicos:

Que el estudiante adquiera conocimiento sobre técnicas de resolución de problemas basadas en programación dinámica.

Que el estudiante conozca los fundamentos sobre complejidad computacional y clases de complejidad, para comprender la importancia de los problemas que se encuentran en el límite de la intratabilidad computacional.

Contenidos:

Estrategia de resolución de problemas basada en Programación Dinámica. Aplicaciones: el problema del subconjunto independiente de máximo peso en un grafo de camino único. El problema de la mochila: una solución basada en programación dinámica. Fundamentos de Complejidad Computacional. Problemas resolubles en tiempo polinómico. Reducción polinómica. Noción de completitud y clases de complejidad. Intratabilidad. Las clases P y NP. Problemas NP-Complejos. Aproximaciones algorítmicas prácticas para tratar problemas NP-Complejos.

Bibliografía:

- 1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar" . México: Prentice Hall. ISBN: 970-26-0518-0 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson



Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)

- 4.) Horstmann, C., y Cornell G. (2000). "Core Java 2 – Volume I: Fundamentals". (Disponible en español) Upper Saddle River: Prentice Hall. ISBN: 84-205-4832-4 [\[disponible en biblioteca central\]](#)
- 5.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [\[disponible en biblioteca central\]](#)
- 6.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [\[disponible en biblioteca central\]](#)
- 7.) Weiss, M. A. (2000). "Estructuras de Datos en Java – Compatible con Java 2". Madrid: Addison Wesley. ISBN: 84-7829-035-4 [\[disponible en biblioteca central\]](#)

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.

Unidad Nro 7: Fundamentos de Aplicaciones Empresariales.

Objetivos Específicos:

Que el estudiante adquiera conceptos y prácticas básicas de diseño de sistemas para la Web.

Contenidos:

Conceptos de aplicaciones empresariales. Introducción a Java EE. Componentes del contenedor Web. Patrones de diseño para la construcción de la capa de negocio. Consideraciones de la capa de acceso a datos, patrón DAO y JPA.

Bibliografía:

- 1.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [\[disponible en biblioteca del Departamento de Sistemas\]](#)
- 2.) Horstmann, C., y Cornell G. (2000 aunque existe edición posterior). "Core Java 2 – Volume I: Fundamentals". (Disponible en español) Upper Saddle River: Prentice Hall. ISBN: 84-205-4832-4 [\[disponible en biblioteca central\]](#)
- 3.) Horstmann, C., y Cornell G. (2001 aunque existe edición posterior)). "Core Java 2 – Volume II: Advanced Features". (Disponible en español) Palo Alto: Prentice Hall. ISBN: 84-8322-310-4 [\[disponible en biblioteca central\]](#)
- 4.) Stelting, S. – Maasen, O. (2003 aunque existe edición posterior). "Patrones de diseño aplicados a Java". Madrid: Pearson Educación. ISBN: 9788420538396
- 5.) The Java EE 6 Tutorial. [<http://docs.oracle.com/javaee/6/tutorial/doc/index.html>]

Evaluación:

Evaluación de contenidos y prácticas en cuestionarios, desafíos de programación, trabajos prácticos y parciales.



Metodología de enseñanza y aprendizaje	✓ Cada clase se realiza en el laboratorio de informática. ✓ En cada clase se dicta un tema central, y el mismo se ejemplifica y analiza con modelos de programas presentados por los profesores. ✓ Los alumnos realizan modificaciones y variantes sobre los modelos, y realizan ejercicios nuevos en base a los temas tratados. ✓ Los trabajos prácticos y las tareas semanales integran los conocimientos de las distintas unidades, y parte del tiempo de clase se usa para analizar dudas y elementos relevantes referidos a esos trabajos.
Sistema de evaluación	✓ La materia se evalúa con un parcial (que se puede recuperar), un trabajo práctico integrador, y una nota promedio entre diversas actividades prácticas (como cuestionarios y tareas especiales de programación designadas como "desafíos de programación"). ✓ El tema de cada actividad evaluable integra conocimientos de las distintas unidades del programa, y tiene complejidad y volumen de trabajo acorde a la necesidad de evaluar a los estudiantes.
Condiciones de regularidad	La condición de cada alumno surgirá del siguiente esquema: i. Será Promocionado el alumno que obtenga una calificación mayor o igual a 8(ocho) en cada una de las tres evaluaciones, sin pasar por el parcial recuperatorio. Los alumnos promocionados quedarán eximidos de rendir el examen final, y sólo deberán inscribirse y presentarse en el turno que se hayan inscripto para proceder a la firma de libreta y completar las actas. La nota en el acta de examen final, será el promedio directo y redondeado de las tres evaluaciones de regularidad. ii. Será Regular el alumno que no sea promocionado y que obtenga una calificación de 4(cuatro) o más en cada una de las evaluaciones (con eventual aprobación del parcial recuperatorio). iii. Será Libre el alumno que no llegue a la condición mínima de regular, y deberá recursar la materia.
Modalidad de examen final	Los alumnos regulares deberán rendir el examen final, que consistirá en el desarrollo de una aplicación sobre un tema que oportunamente será informado y acordado con los alumnos, pero que se prevé de tal forma que puedan desarrollarlo en un lapso de uno o dos meses. El examen final es de naturaleza individual (no grupal).
Actividades en laboratorio	✓ Todas las clases de la materia se desarrollan en el laboratorio de informática. ✓ Cada alumno debe instalar en su computadora personal las



	herramientas de software requeridas por la materia, y realizar en forma personal prácticas y pruebas sobre ellas para lograr dominio pleno de su uso. ✓ Es bienvenida la utilización por parte de los alumnos de sus propias notebooks, en el transcurso de cada clase. ✓ En varias clases se prevé la realización de trabajos y tareas pautadas para ser terminadas (en la medida de lo posible) en el transcurso de la misma clase. Esos trabajos se suben al aula virtual para su revisión por parte de los jefes de trabajos prácticos y auxiliares del curso.
Horas/año totales de la asignatura	128 horas cátedra en total, todas en el segundo cuatrimestre (16 semanas de clase, 8 horas cátedra por semana).
Cantidad de horas prácticas totales	64 horas.
Cantidad de horas teóricas totales	64 horas.
Tipo de formación práctica (marque la que corresponde si es asignatura curricular -no electiva-)	☐ Formación experimental ☒ Resolución de problemas de ingeniería ☐ Actividades de proyecto y diseño ☐ Prácticas supervisadas en los sectores productivos y/o de servicios
Cantidad de horas afectadas a la formación práctica indicada en el punto anterior	Al menos la mitad de las horas asignadas a la materia se dedican al desarrollo de aplicaciones y trabajos específicos. Estimación: 4 horas por semana (de las 8 que tiene la asignatura)
Descripción de los prácticos	✓ Desarrollo de programas basados en diversas estructuras de datos estudiadas y en diversas estrategias de resolución de problemas. ✓ Desarrollo de programas que apliquen algoritmos para procesamiento de grafos. ✓ Desarrollo de programas que apliquen técnicas de procesamiento de sistemas de archivos.
Criterios de evaluación de los prácticos	Los trabajos se entregan a través del aula virtual, y son revisados por los docentes de la cátedra. También se plantean cuestionarios y desafíos de corrección automática. El enunciado de cada trabajo incluye consignas a cumplir y recomendaciones adicionales, además de criterios de evaluación (completitud, diseño de clases, diseño de interfaz de usuario, elementos de control, etc.) Cada criterio se evalúa y aporta un peso a la calificación final.
Descripción de la presentación de los prácticos	Los alumnos deben presentar a través del aula virtual proyectos desarrollado en plataforma Java, que incluya los archivos fuente de cada clase implementada, más los paquetes externos que pudieran haber usado. Cuando se trate de cuestionarios y desafíos de programación, los alumnos simplemente deben subir sus respuestas mediante los recursos disponibles en el aula virtual.
Cronograma de actividades de la asignatura, incluyendo semanas previstas para cada unidad	Ver archivo CRDLC2014.doc anexo a esta presentación.
Descripción de metodología propuesta de consultas y cronograma de consultas	Email de los docentes: Ing. Valerio Frittelli: vfrittelli@gmail.com Ing. Felipe Steffolani: fsteffolani@gmail.com



	Ing. Scarafía: scarafia@gmail.com Ing. Romina Teicher: rteicher@gmail.com En caso de requerir horarios de consulta, serán pactados en el momento con los alumnos que lo requieran. TODO el trabajo de la cátedra está permanentemente disponible para los alumnos en el espacio virtual de la cátedra (http://uv.frc.utn.edu.ar) Desde esa aula virtual se canalizan todas las comunicaciones, trabajos prácticos, subida y bajada de archivos, apuntes, notas de clases, foros, planillas de calificaciones, etc.
Plan de integración con otras asignaturas	La asignatura se basa en dos previas: Paradigmas de Programación (de segundo año) y Tecnología de Software de Bases (electiva de tercer año). De ambas, el alumno debe traer un adecuado dominio de la programación de objetos y fundamentos de Java, además de conocimientos sólidos de las estructuras de datos tocadas en esas asignaturas. Está en estudio la preparación de una nueva asignatura electiva para quinto año, basada en Arquitectura de Sistemas para la Web. La materia DLC sería la base para el posterior cursado de la nueva asignatura propuesta.
Bibliografía Obligatoria	1.) Deitel, H., Deitel, P. (2005 o posterior). "Java Cómo Programar". México: Prentice Hall. ISBN: 970-26-0518-0 [disponible en biblioteca del Departamento de Sistemas] 2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [disponible en biblioteca del Departamento de Sistemas] 3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [disponible en biblioteca del Departamento de Sistemas] 4.) Frittelli, V. (2001). "Algoritmos y Estructuras de Datos". Córdoba: Universitas. ISBN: 9879496 [disponible en biblioteca central] 5.) Horstmann, C., y Cornell G. (2000). "Core Java 2 – Volume I: Fundamentals". (Disponible en español) Upper Saddle River: Prentice Hall. ISBN: 84-205-4832-4 [disponible en biblioteca central] 6.) Horstmann, C., y Cornell G. (2001). "Core Java 2 – Volume II: Advanced Features". (Disponible en español) Palo Alto: Prentice Hall. ISBN: 84-8322-310-4 [disponible en biblioteca central] 7.) Langsam, Y., Augenstein, M., y Tenenbaum, A. (1997). "Estructura de Datos con C y C++ (2da. Edición)". México: Prentice Hall. ISBN: 968-880-798-2 [disponible en biblioteca central] 8.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [disponible en biblioteca central]
Bibliografía Complementaria	1.) Folk, M., Zoellick, B., Riccardi, G. (1998). "File Structures – An Object Oriented Approach with C++". Reading: Addison Wesley.



	ISBN: 0-201-87401-6 [disponible en biblioteca central] 2.) Hekmatpour, S. (1992). "C++: Guía para programadores en C". México: Prentice Hall Hispanoamericana. ISBN: 968-880-257-3 [disponible en biblioteca central] 3.) Stelting, S. – Maasen, O. (2003 aunque existe edición posterior). "Patrones de diseño aplicados a Java". Madrid: Pearson Educación. ISBN: 9788420538396
--	--

Firma:

Aclaración: Ing. Valerio Frittelli