

## MODALIDAD ACADÉMICA

<b>Asignatura</b>	Diseño de Lenguajes de Consulta – Electiva Cuarto Nivel	
<b>Carrera</b>	<b>INGENIERÍA EN SISTEMAS DE INFORMACIÓN</b>	
<b>Ciclo Lectivo</b>	2021	
<b>Vigencia del programa</b>	Desde el ciclo lectivo 2014	
<b>Plan</b>	2008	
<b>Nivel</b>	<input type="checkbox"/> 1er. Nivel <input type="checkbox"/> 2do. Nivel <input type="checkbox"/> 3er. Nivel <input checked="" type="checkbox"/> 4to. Nivel <input type="checkbox"/> 5to. Nivel	
<b>Coordinador de la Cátedra</b>	Ing. Valerio Frittelli	
<b>Área de Conocimiento</b>	<input checked="" type="checkbox"/> Programación <input type="checkbox"/> Computación <input type="checkbox"/> Sistemas de Información <input type="checkbox"/> Gestión Ingenieril <input type="checkbox"/> Modelos <input type="checkbox"/> Complementaria	
<b>Carga horaria semanal</b>	8 horas cátedra.	
<b>Anual/ cuatrimestral</b>	Cuatrimestral	
<b>Contenidos Mínimos, según Diseño Curricular-Ordenanza 1150 (sólo para asignaturas curriculares, no electivas)</b>	No aplica (asignatura electiva).	
<b>Correlativas para cursar</b> (según Diseño Curricular-Ordenanza 1150)	Regulares (sugeridas)	Regulares (sugeridas)
	<ul style="list-style-type: none"> <li>Tecnología de Software de Base [Electiva Tercer Nivel]</li> </ul>	<ul style="list-style-type: none"> <li>Tecnología de Software de Base [Electiva Tercer Nivel]</li> </ul>
<b>Correlativas para rendir</b> (según Diseño Curricular-Ordenanza 1150)	Regulares (sugeridas)	Regulares (sugeridas)
	<ul style="list-style-type: none"> <li>Tecnología de Software de Base [Electiva Tercer Nivel]</li> </ul>	<ul style="list-style-type: none"> <li>Tecnología de Software de Base [Electiva Tercer Nivel]</li> </ul>
<b>Objetivos generales de la Asignatura</b>	<p>Al finalizar el ciclo lectivo, se espera que el estudiante logre:</p> <ul style="list-style-type: none"> <li>Comprender, definir y aplicar estructuras de datos avanzadas, junto con los algoritmos asociados a ellas, para la resolución de problemas y situaciones complejas, en un contexto propio de estudiantes avanzados de la carrera.</li> <li>Dominar elementos de análisis y diseño de algoritmos, para poder medir la eficiencia relativa de distintos algoritmos que pudieran servir para el mismo problema, considerando situaciones de aplicación profesional real.</li> <li>Implementar eficientemente las soluciones planteadas, aplicando buenas prácticas y patrones de diseño, mediante una plataforma de programación de uso profesional actual, considerando la futura inserción del estudiante en el mundo laboral.</li> <li>Incorporar 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</li> </ul>	

*búsqueda para esos contextos, considerando un nivel de diseño y desarrollo a nivel de prototipo.*

**Programa Analítico**

**Unidad Nº 1: Recuperación de Información: Motores de Búsqueda.**

**Resultados de Aprendizaje:**

- Dominar los principales modelos existentes para la búsqueda y recuperación de información en ambientes de datos no estructurados, y la forma de diseñar y desarrollar un motor de búsqueda para esos contextos, considerando un detalle de diseño y desarrollo a nivel de prototipo.

**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: vocabulario, listas de posteo, índices.
- Técnicas de búsqueda en un índice ya generado: ranking y recuperación de documentos que encajen con una consulta. Detalles de implementación.
- Otros modelos: el modelo PageRank.

**Bibliografía:**

- 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]**
- Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 **[disponible en biblioteca del Departamento de Sistemas]**
- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. **[disponible en biblioteca del Departamento de Sistemas]**
- 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]**
- 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]**
- Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 **[disponible en biblioteca central]**
- 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 el trabajo práctico único y en los cuestionarios 1 y 2.

**Unidad Nº 2: Estrategias de Resolución de Problemas.**

**Resultados de Aprendizaje:**

- Diseñar algoritmos basados en diversas estrategias de resolución para los problemas estudiados, de

manera de poder plantear soluciones a diversos problemas de ingeniería, considerando que esa habilidad de diseño consolide en cada alumno la capacidad de razonamiento, acreciente la confianza en sí mismo, y fomente su independencia de criterio.

- Analizar la eficiencia de un algoritmo, ya sea en términos de tiempo de ejecución, consumo de memoria o complejidad aparente de código fuente, para poder plantear soluciones profesionales y eficientes a diversos problemas del campo de la ingeniería, teniendo en cuenta que esa capacidad de análisis les permita diferenciarse como programadores y desarrolladores, además de convertirse en referentes sólidos y confiables.

**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:**

- 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]
- Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [disponible en biblioteca del Departamento de Sistemas]
- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [disponible en biblioteca del Departamento de Sistemas]
- Frittelli, V. (2013). "Fichas de Clase para TSB". Córdoba: UTN FRC [disponible en el aula virtual de la asignatura]
- 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]
- Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [disponible en biblioteca central]
- 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]
- Documentación oficial de la plataforma Java (Javadocs).

**Evaluación:**

Evaluación de contenidos y prácticas en los cuestionarios 3 y 4 más los desafíos de programación 1 y 2.

**Unidad Nº 3: Grafos.**

**Resultados de Aprendizaje:**

- Mostrar capacidad para detectar situaciones de procesamiento de datos que presentan algún tipo de relación explícitas, de modo de usar en forma correcta estructuras tipo grafo para el modelado de esas relaciones, en el contexto de un avance sólido en el futuro desarrollo profesional.
- Diseñar algoritmos basados en grafos para problemas que impliquen procesamiento datos que presentan algún tipo de relación explícita, para poder plantear soluciones a de ingeniería, considerando que esa habilidad de diseño consolide en cada alumno la capacidad de razonamiento, acreciente la confianza en sí

mismo, y fomente su independencia de criterio.

**Contenidos:**

- Concepto de Grafo. Vértices y aristas. Tipos de grafos: dirigido, no dirigidos y ponderados.
- Implementación matricial e implementación enlazada.
- 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:**

- 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]
- Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [disponible en biblioteca del Departamento de Sistemas]
- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [disponible en biblioteca del Departamento de Sistemas]
- Frittelli, V. (2013). "Fichas de Clase para TSB". Córdoba: UTN FRC [disponible en el aula virtual de la asignatura]
- 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]
- Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [disponible en biblioteca central]
- 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]
- Documentación oficial de la plataforma Java (Javadocs).

**Evaluación:**

Evaluación de contenidos y prácticas en cuestionarios 5, 6, 7 y 8; los desafíos de programación 2 y 3; y el parcial único.

**Unidad Nº 4: Compresión de Archivos.**

**Resultados de Aprendizaje:**

- Dominar diversas técnicas generales de representación de información, de forma de poder diseñar sistemas de almacenamiento que permitan reducir la cantidad de espacio ocupado, en el contexto de una formación diferenciada en el futuro desarrollo profesional.

**Contenidos:**

- Algoritmos Avidos.
- 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.
- Gestión de archivos binarios.
- Desarrollo de un prototipo de suite de compresión – descompresión.

#### **Bibliografía:**

- 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]**
- Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 **[disponible en biblioteca del Departamento de Sistemas]**
- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. **[disponible en biblioteca del Departamento de Sistemas]**
- 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]**
- 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]**
- Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 **[disponible en biblioteca central]**
- 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 en el cuestionario 9 y en el parcial único.

#### **Unidad Nº 5: Programación Dinámica.**

#### **Resultados de Aprendizaje:**

- Diseñar algoritmos basados en una estrategia de almacenamiento de resultados de subproblemas previamente detectados en la resolución de problemas, de manera plantear soluciones eficientes para diversos problemas de ingeniería, considerando que esa habilidad de diseño consolide en cada alumno la capacidad de razonamiento, acreciente la confianza en sí mismo, fomente su independencia de criterio y lo diferencie en el plano profesional.

#### **Contenidos:**

- Estrategia de resolución de problemas basada en Programación Dinámica.
- Análisis de subproblemas y determinación de fórmula de recurrencia.
- Pruebas de validez.
- 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.

**Bibliografía:**

- 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]**
- Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 **[disponible en biblioteca del Departamento de Sistemas]**
- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. **[disponible en biblioteca del Departamento de Sistemas]**
- 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]**
- 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]**
- Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 **[disponible en biblioteca central]**
- 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 en el cuestionario 10 y en el desafío 4.

**Unidad Nº 6: Complejidad Computacional.**

**Resultados de Aprendizaje:**

- Demostrar conocimientos generales en el campo de la Teoría de la Complejidad, para poder distinguir en forma esencial entre problemas tratables e intratables, considerando que esta habilidad oriente a los estudiantes en el aprendizaje futuro de nuevas técnicas para enfrentar problemas en el límite de la computabilidad.

**Contenidos:**

- Algoritmos de tiempo de ejecución de orden polinomial. Algoritmos intratables.
- Reducción de un problema a otro. Reducción polinómica.
- Nociones de máquina algorítmica determinista y de máquina algorítmica no determinista.
- Problemas de decisión y problemas de optimización.
- Clases de Complejidad. Las clases P, NP y NP Completo. Propiedades de un problema NP Completo.
- Aproximación conceptual al Teorema de Cook-Levin.
- Los problemas del milenio y la cuestión P vs. NP en la actualidad.

**Bibliografía:**

- 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]**
- Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 **[disponible en biblioteca del Departamento de Sistemas]**
- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN:

9788489660342. **[disponible en biblioteca del Departamento de Sistemas]**

- 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]**
- 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]**
- Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 **[disponible en biblioteca central]**
- 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 el cuestionario 11.

#### **Unidad Nro 7: Fundamentos de Acceso a Bases de Datos y Aplicaciones Empresariales.**

#### **Resultados de Aprendizaje:**

- Dominar conceptos y técnicas del campo de las aplicaciones empresariales, para poder diseñar sistemas que operen en forma eficiente en ese campo, considerando que esta habilidad permita a los futuros profesionales diferenciarse en el mercado de la producción de software.
- Demostrar dominio técnico en el campo del acceso y gestión de bases de datos, de forma de incorporar ese dominio en el diseño de aplicaciones empresariales, considerando un contexto profesional muy competitivo en ese requerimiento.

#### **Contenidos:**

- Consideraciones acerca de la persistencia en las aplicaciones. Tipos de DBMS actuales.
- Introducción a JDBC. Esquema de Conectores J. Servidores de Base de Datos frente a Bases de Datos Embebidas.
- Carga del Driver o Conector J. Establecimiento de la Conexión. Ejecución de una sentencia simple. Alternativa de ejecución mediante PreparedStatement. Alternativa de ejecución mediante CallableStatement. Procesamiento de los resultados. Sentencias de manipulación de datos. Manejo de errores. Administración de transacciones. Utilización de archivos .properties para configuración.
- 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:**

- Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. **[disponible en biblioteca del Departamento de Sistemas]**
- 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]**
- 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]**
- Stelting, S. – Maasen, O. (2003 aunque existe edición posterior). "Patrones de diseño aplicados a Java".

Madrid: Pearson Educación. ISBN: 9788420538396

- The Java EE 6 Tutorial. [<http://docs.oracle.com/javaee/6/tutorial/doc/index.html>]

**Evaluación:**

Evaluación de contenidos y prácticas en el trabajo práctico único.

<p><b>Metodología de enseñanza y aprendizaje</b> (Planificar estrategias centradas en el aprendizaje activo del estudiante)</p>	<ul style="list-style-type: none"> <li>✓ En cada clase se desarrolla un tema central, y el mismo se ejemplifica y analiza con modelos de programas presentados por los profesores.</li> <li>✓ Los alumnos realizan modificaciones y variantes sobre los modelos, y realizan ejercicios nuevos en base a los temas tratados.</li> <li>✓ El trabajo práctico único y las tareas semanales (cuestionarios y desafíos) 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.</li> <li>✓ Todos los materiales del curso (materiales del estudio y evaluaciones de tipos diversos) están siempre disponibles en el aula virtual de la asignatura. El esquema de trabajo al que se orienta toda la Cátedra (desde 2015 en adelante) es el de <i>Clase Invertida</i>, con variaciones que son propias del hecho de contar con un muy elevado número de estudiantes.</li> </ul>
<p><b>Sistema de evaluación</b> (Nombrar y describir cada una de las diferentes instancias de evaluación, pensando en la Evaluación como proceso continuo de recolección de evidencias)</p>	<ul style="list-style-type: none"> <li>✓ La materia se evalúa con un parcial único (que se puede recuperar), un trabajo práctico único, y una nota promedio entre diversas actividades prácticas (como cuestionarios y tareas especiales de programación designadas como "desafíos de programación").</li> <li>✓ 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.</li> </ul>
<p><b>Criterios de evaluación</b> (los cuales serán tenidos en cuenta en las correcciones)</p>	<ul style="list-style-type: none"> <li>• El <i>trabajo práctico único</i> se entrega a través del aula virtual, y es revisado y calificado por los docentes de la Cátedra.</li> <li>• Los <i>cuestionarios y desafíos</i> son de corrección automática.</li> <li>• En casos de <i>parciales</i> y el <i>trabajo práctico único</i>, los alumnos deben presentar a través del aula virtual proyectos desarrollados en Java, que incluyan los archivos fuente implementados.</li> <li>• Cuando se trate de <i>cuestionarios y desafíos de programación</i>, los alumnos simplemente deben subir sus respuestas mediante los recursos disponibles en el aula virtual.</li> <li>• El enunciado del <i>trabajo práctico único</i> incluye consignas a cumplir y recomendaciones adicionales, además de criterios de evaluación tales como: compilación, completitud de consignas cubiertas, diseño de clases, lógica aplicada en la solución de cada ítem, diseño de interfaz de usuario y elementos de control de carga e integridad. Cada criterio se evalúa y aporta un peso a la calificación final. De acuerdo a la suma de esos pesos, se aplica la escala de notas que ya fue indicada en secciones anteriores de este mismo documento (y que forma parte de la consigna mostrada a los estudiantes).</li> <li>• Todo lo expresado en el ítem anterior para los trabajos prácticos, vale</li> </ul>

	<p>también para los parciales.</p> <ul style="list-style-type: none"> <li>• Esencialmente, se aplica un esquema de evaluación continua, con reglas de trabajo firmes pero claras.</li> <li>• En todo momento los estudiantes saben cuál será el esquema de evaluaciones, cómo serán evaluados, y cuándo les serán entregadas sus devoluciones y calificaciones.</li> <li>• El soporte del aula virtual única y centralizada para todos los cursos, con exactamente el mismo y único modelo de intervención para todos los cursos, facilita el proceso de avance y de control.</li> </ul>
<p><b>Regularidad: condiciones</b> (Describir las condiciones necesarias para regularizar. Se sugiere incluir la aclaración que el estudiante en condición de regular puede rendir en el plazo de un ciclo lectivo sin control de correlativas aprobadas)</p>	<p>A partir de 2018 sólo se aplica la escala con aprobación desde 6(seis) (que es la misma que corresponde a la evaluación de exámenes finales). En consecuencia, para aprobar un parcial, un práctico o cualquier otra evaluación normal del ciclo de cursado, el estudiante deberá alcanzar como mínimo una nota de 6(seis).</p> <p>Todo alumno que obtenga al menos una calificación de 6(seis) en cada una de las tres evaluaciones generales: el parcial, el práctico integrador y el promedio de actividades prácticas de aula, obtendrá la <b>aprobación directa</b>: no deberá rendir examen final. Simplemente deberá inscribirse en cualquier turno en que tenga ocasión de hacerlo, y pasar el día del examen a hacer firmar la libreta con su nota final, que en todos los casos será igual al promedio simple y redondeado entre las tres notas de regularidad.</p> <p>Sólo se puede recuperar el parcial único, si la nota en ese parcial fuese menor que 6(seis). En ese caso, el alumno deberá hacer un parcial de recuperación. En caso de aprobar este recuperatorio, <u>la nota máxima del mismo será 6(seis)</u> (esto se hace por causa de estricta justicia, ya que la nota del recuperatorio <b>reemplazará</b> a la nota original del parcial y se procederá luego en forma normal a calcular el promedio final para aprobación directa de la materia). Pero si el alumno obtiene nota menor a 6(seis) en este recuperatorio, quedará <b>libre</b>.</p> <p>También quedará <b>libre</b> el alumno que no llegue al 6(seis) en el trabajo práctico único y/o en el promedio de las notas por actividades prácticas. Estas actividades no cuentan con instancia de recuperatorio: el trabajo práctico se entrega tantas veces como sea necesario hasta aprobarlo, y el promedio de las actividades prácticas abarca un conjunto suficientemente grande de calificaciones como para que cada una de ellas actúe como recuperatorio en el promedio.</p> <p>A partir de 2017, todo alumno puede inscribirse a rendir una asignatura dentro de un período de ciclo lectivo completo a partir del momento de la regularización, sin que afecte el cumplimiento de las correlatividades previas. A partir de vencido ese plazo de un ciclo completo, para poder inscribirse a examen deberá cumplir primero con las correlatividades previas.</p> <p>A los fines estadísticos, se considerará que un alumno <b>abandonó</b> el cursado si <i>no llegó a realizar todas las evaluaciones previstas</i>. Si hizo todas esas evaluaciones y sus calificaciones no alcanzaron a cubrir las exigencias de aprobación, el alumno será considerado libre.</p>

	<p>Escala de notas aplicable DURANTE el cursado (a partir de 2018, igual a la usada en el examen final) (*)</p> <table border="1" data-bbox="754 353 1332 763"> <thead> <tr> <th>NOTA</th> <th>PORCENTAJE</th> <th>CALIFICACIÓN</th> </tr> </thead> <tbody> <tr><td>1</td><td></td><td>Insuficiente</td></tr> <tr><td>2</td><td></td><td>Insuficiente</td></tr> <tr><td>3</td><td></td><td>Insuficiente</td></tr> <tr><td>4</td><td></td><td>Insuficiente</td></tr> <tr><td>5</td><td></td><td>Insuficiente</td></tr> <tr><td>6</td><td>60% a 68%</td><td>Aprobado</td></tr> <tr><td>7</td><td>69% a 77%</td><td>Bueno</td></tr> <tr><td>8</td><td>78% a 86%</td><td>Muy Bueno</td></tr> <tr><td>9</td><td>87% a 95%</td><td>Distinguido</td></tr> <tr><td>10</td><td>96% a 100%</td><td>Sobresaliente</td></tr> </tbody> </table> <p>(*) Escala acordada en reunión de Docentes Coordinadores</p>	NOTA	PORCENTAJE	CALIFICACIÓN	1		Insuficiente	2		Insuficiente	3		Insuficiente	4		Insuficiente	5		Insuficiente	6	60% a 68%	Aprobado	7	69% a 77%	Bueno	8	78% a 86%	Muy Bueno	9	87% a 95%	Distinguido	10	96% a 100%	Sobresaliente
NOTA	PORCENTAJE	CALIFICACIÓN																																
1		Insuficiente																																
2		Insuficiente																																
3		Insuficiente																																
4		Insuficiente																																
5		Insuficiente																																
6	60% a 68%	Aprobado																																
7	69% a 77%	Bueno																																
8	78% a 86%	Muy Bueno																																
9	87% a 95%	Distinguido																																
10	96% a 100%	Sobresaliente																																
<p><b>Promoción: condiciones</b> (Aclarar si hubiera promoción de alguna parte de la asignatura, las condiciones y si tiene duración, con el mayor detalle posible)</p>	<p>No aplica.</p>																																	
<p><b>Aprobación Directa: condiciones.</b> (la calificación será la nota registrada como Nota Final en Autogestión) (Se sugiere incluir la aclaración que el estudiante, en esta condición, puede registrar su nota en examen en el plazo de un ciclo lectivo, sin control de correlativas aprobadas, y después de ello se le exigirán correlativas aprobadas)</p>	<p>Los alumnos que hayan obtenido la <b>aprobación directa</b> no deberán rendir examen final. Simplemente deberán inscribirse en cualquier turno en que tengan ocasión de hacerlo, y pasar el día del examen a hacer firmar la libreta con la nota final, que en todos los casos será igual al promedio simple y redondeado entre las tres notas de regularidad. La condición de <b>aprobación directa</b> no tiene vencimiento.</p> <p>A partir de 2017, todo alumno puede inscribirse a rendir una asignatura dentro de un período de ciclo lectivo completo a partir del momento de la regularización, sin que afecte el cumplimiento de las correlatividades previas. A partir de vencido ese plazo de un ciclo completo, para poder inscribirse a examen deberá cumplir primero con las correlatividades previas.</p>																																	
<p><b>Modalidad de examen final</b> (Describir las características metodológicas del examen final para los distintos estados del estudiante)</p>	<p>Como se indicó, no hay examen final: los alumnos que regularicen tienen aprobada la materia con nota final igual al promedio entre sus notas de regularidad.</p> <p><b>Escala de Notas para Examen Final (*)</b></p>																																	

		NOTA	PORCENTAJE	CALIFICACIÓN
		1		Insuficiente
		2		Insuficiente
		3		Insuficiente
		4		Insuficiente
		5		Insuficiente
		6	60% a 68%	Aprobado
		7	69% a 77%	Bueno
		8	78% a 86%	Muy Bueno
		9	87% a 95%	Distinguido
		10	96% a 100%	Sobresaliente
	(*) Escala acordada en reunión de Docentes Coordinadores			
<b>Actividades en laboratorio</b>	<ul style="list-style-type: none"> <li>✓ Todas las clases de la materia se desarrollan en el laboratorio de informática (si se retorna a la modalidad presencial). Si el cursado es a distancia (por razón de la pandemia COVID-19), la clases se desarrollan por medio de salas de teleconferencia (palataforma Zoom).</li> <li>✓ 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.</li> <li>✓ Es bienvenida la utilización por parte de los alumnos de sus propias notebooks, en el transcurso de cada clase.</li> <li>✓ En varias clases se prevé la realización de trabajos y tareas pautadas para ser teminadas (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.</li> </ul>			
<b>Cantidad de horas prácticas totales</b> (en el aula)	64 horas.			
<b>Cantidad de horas teóricas totales</b> (en el aula)	64 horas.			
<b>Cantidad de horas estimadas totales de trabajo</b> (extra áulicas).	64 horas.			
<b>Horas/año totales de la asignatura</b> (en el aula).	128 horas.			
<b>Tipo de formación práctica</b> (sólo si es asignatura curricular -no electiva-)	No aplica.			
<b>Cantidad de horas cátedras afectadas a la formación práctica indicada en el punto anterior</b> (sólo si es asignatura curricular -no electiva-)	No aplica.			
<b>Descripción de los prácticos</b>	<ul style="list-style-type: none"> <li>✓ Desarrollo de programas basados en diversas estructuras de datos estudiadas y en diversas estrategias de resolución de problemas.</li> <li>✓ Desarrollo de programas que apliquen algoritmos para procesamientos de grafos.</li> </ul>			

	✓ Desarrollo de programas que apliquen técnicas básicas de diseño y desarrollo de aplicaciones empresariales.
<b>Cronograma de actividades de la asignatura</b> (contemplando las fechas del calendario 2019 y para cada unidad)	Ver archivo CRDLC2021.doc anexo a esta presentación.
<b>Propuesta para la atención de consultas y mail de contacto.</b>	Email de los docentes: Ing. Valerio Frittelli: <a href="mailto:vfrittelli@gmail.com">vfrittelli@gmail.com</a> Ing. Felipe Steffolani: <a href="mailto:fsteffolani@gmail.com">fsteffolani@gmail.com</a> Ing. Diego Scarafia: <a href="mailto:scarafia@gmail.com">scarafia@gmail.com</a> Ing. Diego Serrano: <a href="mailto:diegojserrano@gmail.com">diegojserrano@gmail.com</a> 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 ( <a href="http://uv.frc.utn.edu.ar">http://uv.frc.utn.edu.ar</a> ) 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.
<b>Plan de integración con otras asignaturas</b>	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.
<b>Bibliografía Obligatoria</b>	<ol style="list-style-type: none"> <li>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]</li> <li>2.) Drozdek, A. (2007). "Estructura de Datos y Algoritmos en Java". México D.F.: Thomson. ISBN: 9789706866110 [disponible en biblioteca del Departamento de Sistemas]</li> <li>3.) Eckel, B. (2002 aunque existe edición posterior). "Piensa en Java". Madrid: Pearson Educación. ISBN: 9788489660342. [disponible en biblioteca del Departamento de Sistemas]</li> <li>4.) Frittelli, V. (2001). "Algoritmos y Estructuras de Datos". Córdoba: Universitas. ISBN: 9879496 [disponible en biblioteca central]</li> <li>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]</li> <li>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]</li> <li>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]</li> <li>8.) Sedgewick, Robert (1995). "Algoritmos en C++". Reading: Addison Wesley – Díaz de Santos. ISBN: 978-0-201-62574-5 [disponible en biblioteca central]</li> </ol>
<b>Bibliografía Complementaria</b>	1.) Folk, M., Zoellick, B., Riccardi, G. (1998). "File Structures – An Object

	<p>Oriented Approach with C++". Reading: Addison Wesley. ISBN: 0-201-87401-6 [disponible en biblioteca central]</p> <p>2.) Hekmatpour, S. (1992). "C++: Guía para programadores en C". México: Prentice Hall Hispanoamericana. ISBN: 968-880-257-3 [disponible en biblioteca central]</p> <p>3.) Stelting, S. – Maasen, O. (2003 aunque existe edición posterior). "Patrones de diseño aplicados a Java". Madrid: Pearson Educación. ISBN: 9788420538396</p>																		
<b>Distribución de docentes</b>	<table border="1"> <thead> <tr> <th><i>Curso</i></th> <th><i>Turno</i></th> <th><i>Día y Horas</i></th> <th><i>Profesor</i></th> <th><i>JTP</i></th> <th><i>Ayudante</i></th> </tr> </thead> <tbody> <tr> <td>4K4</td> <td>noche</td> <td>Lun 3-4-5-6 Mie 3-4-5-6</td> <td>Steffolani</td> <td>Scarafia</td> <td></td> </tr> <tr> <td>4K6</td> <td>mañana</td> <td>Mar 3-4-5-6 Vie 1-2-3-4</td> <td>Frittelli</td> <td>Serrano</td> <td></td> </tr> </tbody> </table>	<i>Curso</i>	<i>Turno</i>	<i>Día y Horas</i>	<i>Profesor</i>	<i>JTP</i>	<i>Ayudante</i>	4K4	noche	Lun 3-4-5-6 Mie 3-4-5-6	Steffolani	Scarafia		4K6	mañana	Mar 3-4-5-6 Vie 1-2-3-4	Frittelli	Serrano	
<i>Curso</i>	<i>Turno</i>	<i>Día y Horas</i>	<i>Profesor</i>	<i>JTP</i>	<i>Ayudante</i>														
4K4	noche	Lun 3-4-5-6 Mie 3-4-5-6	Steffolani	Scarafia															
4K6	mañana	Mar 3-4-5-6 Vie 1-2-3-4	Frittelli	Serrano															

Firma: ..... 

Aclaración: Ing. Valerio Frittelli