



Laboratorio de Computación II

PROGRAMA

1) OBJETIVOS DE LA ASIGNATURA

- Que el alumno conozca entornos integrados de programación.
- Que el alumno conozca y aplique herramientas complementarias para el desarrollo, documentación y prueba de componentes de *software*.
- Que el alumno conozca las diferentes modalidades de programación, de acuerdo a la naturaleza del problema planteado, y los recursos tecnológicos existentes.
- Que se logre una integración vertical y horizontal con materias afines.
- Que se promueva la participación del alumno en trabajos de investigación y desarrollo.

2) CONTENIDOS MÍNIMOS

- Edición y depuración de programas.
- Ejecución de programas codificados en C y Java.
- Funciones y archivos.
- Paradigmas de programación.
- Pruebas del *software*. Introducción a Calidad de *Software*.
- Documentación de programación.



3) PROGRAMA ANALÍTICO

UNIDAD TEMÁTICA	CONTENIDOS
1	LABORATORIO DE SOFTWARE FACTORY (SF) 1.1 Concepto. Roles dentro de una SF. Integración del programador en el proceso de desarrollo de <i>software</i> . 1.2 Proyecto de <i>software</i> . Estudio de Factibilidad. 1.3 Actividades en una SF: Análisis y Diseño, Edición, Ejecución y Depuración. <i>Testing</i> . Versionado. Documentación. Otras actividades.
2	PARADIGMAS DE PROGRAMACIÓN 2.1 Concepto de Paradigma de Programación. 2.2 Características del Paradigma Orientado a Objetos. Estructura de un programa OO. 2.3 Características del Paradigma Funcional: Estructura de un programa funcional 2.4 Características del Paradigma Lógico: Estructura de un programa lógico. 2.5 Programación paralela: concepto. Ambientes de desarrollo. Conceptos básicos de programación concurrente. <u>Bibliografía / Guías de estudio:</u> • Java 2: Lenguajes y Aplicaciones. F. J. Ceballos. • Cómo programar en JAVA. Deitel & Deitel. • Programación Orientada a Objetos. Técnicas Avanzadas de Programación. Carlos Fontela. • Fundamentos de Programación. Piensa en C. Osvaldo Cairó. • Java 2. Manual del Usuario y Tutorial. Agustín Froufe. • Guía de estudio 2: Paradigmas de programación (Monetti). • Conceptos de Cómputo Paralelo. José Torres Jiménez & Eduardo Arturo Rodríguez Tello.



3	ALMACENAMIENTO DE DATOS 3.1 El fichero de datos como complemento a la aplicación ejecutable. 3.2 Organización de ficheros. 3.3 Introducción a la seguridad sobre los ficheros de datos. Bloqueos y permisos. 3.4 Introducción a las bases de datos. La base de datos como complemento a la aplicación ejecutable. 3.5 Otras fuentes de datos: <i>sockets</i> y llamadas a servicios. <u>Bibliografía / Guías de estudio:</u> • Essential System Administration. Eleen Frisch. • Java 2: Lenguajes y Aplicaciones. F. J. Ceballos. • Fundamentos de Programación. Piensa en C. Osvaldo Cairó. • Guía de Estudio 1º. Fuentes de datos
4	EDICIÓN Y EJECUCIÓN DE PROGRAMAS 4.1 Edición de código fuente. IDEs más comunes (de consola y gráfico). Características de los principales IDEs. 4.2 Ejemplos de utilización. 4.3 Administración de un proyecto de <i>software</i> Java. 4.4 Mejoramiento del código: <i>Refactoring</i>. Ejemplos y utilización del refactorizador. 4.5 Proyectos Java-Maven. <u>Bibliografía / Guías de estudio:</u> • Essential System Administration. Eleen Frisch. • Medición y Estimación del Software: Técnicas y Métodos para Mejorar la Calidad y la Productividad. Mario Piattini Velthuis y otros. Alfaomega. • LINUX Desmitificado. Manuel Alberto Ricart. • Integración Linux-Windows. Mike Mc Cune. • Fundamentos de Programación. Piensa en C. Osvaldo Cairó.
5	VERSIONADO 5.1 Conceptos Básicos de Versionado. 5.2 <i>Subversion</i> SVN. 5.3 Configuración de un proyecto con <i>Subversion</i> SVN. 5.4 Otras herramientas de versionado para trabajo colaborativo en proyectos de <i>software</i>. 5.5 Comandos y operaciones básicas. Conflictos. <u>Bibliografía / Guías de estudio:</u> • Pro Netbeans 6 IDE. Rich Client Platform Edition. Adam Myatt. Apress.



6	PRUEBAS DE SOFTWARE 6.1 Prueba de <i>Software</i>: Fallos, defectos y errores. Tipos de defectos. Conceptos técnicos y de uso. Alcance de una prueba de <i>software</i>. 6.2 Prueba Unitaria. Prueba de Integración. Prueba de validación y aceptación. Prueba del sistema. 6.3 Diseño de prueba de escritorio. Diseño de archivos de bitácora. 6.4 Técnicas y Estrategias para la prueba de <i>software</i>. 6.5 Organización para las pruebas de <i>software</i>. Verificación y validación de programas. 6.6 Uso de JUnit: clases de pruebas. Aserciones. Métodos de la clase de prueba. <u>Bibliografía / Guías de estudio:</u> • Ingeniería de Software, un enfoque práctico. Roger Pressman. 6ta edición. • Ingeniería de Software Orientada a Objetos con UML, Java E Internet. Alfredo Weitzenfeld.
7	DEPURACIÓN DE PROGRAMAS 7.1 Depuración (<i>debugging</i>): Conceptos. 7.2 Herramientas de depuración: Trazas y Aserciones, Depuradores. 7.3 Depuradores de memoria, seguidores de llamadas, 7.4 Tipos de <i>breakpoints</i>, <i>step-over</i>, <i>step-into</i>, <i>step-out</i>, <i>move-to-cursor</i>, 7.6 Inspección de variables. 7.6 Otra información provista por <i>debuggers</i>. 7.7 Uso de Log4J. <u>Bibliografía / Guías de estudio:</u> • Essential System Administration. Eleen Frisch. • Medición y Estimación del Software: Técnicas y Métodos para Mejorar la Calidad y la Productividad. Mario Piattini Velthuis y otros. Alfaomega. • LINUX Desmitificado. Manuel Alberto Ricart. • Integración Linux-Windows. Mike Mc Cune. • Fundamentos de Programación. Piensa en C. Osvaldo Cairó.



8	ENTORNOS DE EJECUCIÓN JAVA 8.1 Programas ejecutables: <i>standalone</i> , <i>GUI</i> , <i>librería</i> . 8.2 Concepto de arquitectura cliente-servidor. 8.3 La Máquina Virtual de Java: Su uso: y posibilidades de configuración. 8.4 Formas alternativas de ejecución: Los <i>Applets</i> y su entorno de ejecución. 8.5 <i>WebServices</i> : entorno de trabajo y consumo de servicios. <i>Servlets</i> y su entorno de ejecución. 8.6 Performance. Uso del <i>profiler</i> para el análisis de performance. <u>Bibliografía / Guías de estudio:</u> • Java 2: Lenguajes y Aplicaciones. F. J. Ceballos. • Cómo programar en JAVA. Deitel & Deitel. • Programación Orientada a Objetos. Técnicas Avanzadas de Programación. Carlos Fontela. • Fundamentos de Programación. Piensa en C. Osvaldo Cairó. Java 2. Manual del Usuario y Tutorial. Agustín Froufe.
9	LA MÁQUINA VIRTUAL DE JAVA 9.1 Arquitectura. Modelo de Memoria. 9.2 Uso de la memoria. Proceso de recolección de basura. 9.3 Uso de la JVM y posibilidades de configuración. Tasas de permanencia, localización y promoción. 9.4 <i>Tunning</i> de la máquina virtual de Java (JVM). 9.5 Análisis de procesos de la máquina virtual de Java. 9.6 Administración de hilos de ejecución. 9.7 <i>Profiling</i> . Posibilidades de perfilado en Netbeans.



10	INTRODUCCIÓN A LA CALIDAD DE SOFTWARE Y DOCUMENTACION EN PROGRAMACIÓN 10.1 Definición de calidad, fiabilidad de <i>software</i>. 10.2 Relación de la ingeniería de <i>software</i> con el proceso de desarrollo. 10.3 Métodos y Herramientas. Normas y estándares. 10.4 Introducción a las métricas de programación. 10.5 Métricas para el código fuente y diseño de la aplicación. 10.6 Documentación de procesos. Documentación de base de datos. 10.7 Diseño de manuales de programación: destinatarios, componentes. 10.8 Diseño del manual del usuario. <u>Bibliografía / Guías de estudio:</u> - Ingeniería de Software, un enfoque práctico. Roger Pressman. 6ta edición. - Ingeniería de Software Orientada a Objetos con UML, Java E Internet. Alfredo Weitzenfeld.
----	--

4) PROGRAMA DE EXAMEN

El examen final está compuesto por un conjunto de preguntas representativas de cada bolilla.

Bolilla de examen nº 1: 1
Bolilla de examen nº 2: 2
Bolilla de examen nº 3: 3
Bolilla de examen nº 4: 4
Bolilla de examen nº 5: 5

Bolilla de examen nº 6: 6
Bolilla de examen nº 7: 7
Bolilla de examen nº 8: 8
Bolilla de examen nº 9: 9
Bolilla de examen nº 10: 10

Lugar y fecha: Mendoza, Febrero de 2019	PROFESOR/ES A CARGO Apellido y nombre: Julio César MONETTI N° de Legajo: 35459
	Firma: