



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

LABORATORIO DE SOFTWARE

Año 2014

Carrera/ Plan:

Licenciatura en Informática

Plan 2003-07/ Plan 2012

Licenciatura en Sistemas

Plan 2003-07/ Plan 2012

Año: 4º(Lic en Informática)

4º y 5º (Lic en Sistemas)

Régimen de Cursada: *Semestral*

Carácter:

Obligatoria (Lic en Informática)

Optativa Área Ing. de Software y Base de Datos (Lic en Sistemas)

Correlativas: *Proyecto de Software*

Profesor: *Claudia Queiruga*

Hs Semanales: 6 hs

FUNDAMENTACIÓN

Laboratorio de Software es una asignatura obligatoria de cuarto año de la carrera *Licenciatura en Informática* y de carácter optativo para cuarto y quinto año de la carrera *Licenciatura en Sistemas*.

El alumno que cursa **Laboratorio de Software** cuenta con los conocimientos fundamentales de informática en diferentes áreas y ha comenzado a entrenarse en el uso de herramientas de desarrollo de software.

Laboratorio de Software provee a los estudiantes de los conocimientos específicos sobre la construcción de aplicaciones orientadas a servicios, con acceso a bases de datos y aplicaciones nativas para dispositivos móviles inteligentes, utilizando tecnologías JAVA. El estudiante adquiere las habilidades necesarias para desarrollar un trabajo integrador que signifique la aplicación concreta de los conocimientos adquiridos hasta el momento en la carrera, integrando temas de lenguajes de programación, ingeniería de software y base de datos. Mediante este trabajo, el estudiante se enfrenta con problemas reales y con la utilización de tecnologías de desarrollo de software actuales.

Esta asignatura articula en forma vertical con **Algoritmos y Estructura de Datos**, asignatura de segundo año e introductoria sobre desarrollo en lenguaje JAVA y, con materias de quinto año relacionadas a tecnologías de desarrollo: **Java y Aplicaciones Avanzadas sobre Internet**, **Diseño Centrado en el Usuario** y **Interfaces Adaptadas para Dispositivos Móviles**.

Laboratorio de Software consolida la formación experimental y profesional del estudiante, ubicándolo en un entorno de trabajo similar al real y cotidiano.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

OBJETIVOS GENERALES

Trabajo integrador que signifique para el alumno una aplicación concreta de los conocimientos adquiridos hasta el momento (cuarto año), en particular como una evolución de lo que ya han visto en Proyecto de Software.

CONTENIDOS MINIMOS

Se realizará un desarrollo específico integrando conocimientos teóricos y herramientas conocidas por el alumno. El enfoque podrá ajustarse e incluso dividir la temática experimental de la asignatura según las pautas que la cátedra considere de interés anualmente.

PROGRAMA ANALÍTICO

Unidad 1: La Plataforma JAVA

Objetivos:

Profundizar en la universalidad de la plataforma JAVA y su impacto en el desarrollo de software.

Conocer la importancia de la comunidad de especificaciones de las tecnologías JAVA que transforma a JAVA en una tecnología estándar.

El desafío de la plataforma universal JAVA. La plataforma de desarrollo Java o J2SE (Java 2 Standar Edition). La plataforma de ejecución Java o JRE (Java Run-time Environment). IDEs (Integrated Development Environment) para desarrollo en JAVA. Software Libre y JAVA. La comunidad de especificaciones JAVA, el Java Community Process.

Unidad 2: Clases, Espacios de Nombres, Especificadores de Acceso, Interfaces y Tipos Enumerativos

Objetivos:

Profundizar en los conceptos de orientación a objetos del desarrollo de software en JAVA.

Relacionar los especificadores de acceso y el control de visibilidad de los miembros de las clases con las características propias de los paradigmas de orientación a objetos como ocultamiento y encapsulamiento.

Definir tipos de datos usando interfaces y tipos enumerativos.

Usar el formato de empaquetado y compresión de archivos (.class y otros recursos) disponible para distribución de aplicaciones JAVA.

Usar la documentación en línea de la API JAVA.

Definición de clases. Miembros de una clase: atributos y métodos. Creación e inicialización de objetos.

Constructores. La palabra clave this y this(). Especialización de clases.

Especificadores de acceso: private, protected, default y public.

Especificadores de calificación: static, final y abstract.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Interfaces y clases abstractas. Tipos Enumerativos. Paquetes, como espacio de nombres: la palabra clave package. El formato JAR (Java ARchive). La variable de entorno CLASSPATH
Documentación de la API de Java: ¿dónde está disponibles? y ¿cómo usarla?

Unidad 3: Clases Básicas y Arreglos

Objetivos:

Examinar la característica de autoboxing/unboxing de JAVA: usar tipos primitivos, transformarlos en objetos y viceversa, automáticamente.
Profundizar en el buen uso de Strings.

Las clases wrappers: Integer, Short, Long, Byte, Character, Boolean, Float, Double. Boxing, Unboxing. La clase String y StringBuffer. Arreglos en JAVA

Unidad 4: Herencia y Polimorfismo en JAVA

Objetivos:

Profundizar en el mecanismo de herencia simple implementado en JAVA.
Definir interfaces JAVA y su relación con la herencia múltiple.
Comprender la potencia del concepto de upcasting automático a clases e interfaces.
Usar buenas prácticas de programación para creación de objetos en JAVA.
Analizar cómo impacta en la herencia el control de acceso protected.

Creación de objetos. Encadenamiento de constructores. Bloques de inicialización. La palabra clave super y super()

La clase Object: los métodos toString(), equals() y hashCode()

La herencia y la accesibilidad de atributos y métodos. Relación entre el especificador de acceso protected y la herencia.

Sobreescritura de métodos y ocultamiento de atributos. Polimorfismo.

Upcasting

Comparación entre interfaces y clases abstractas. Relación entre interfaces y herencia múltiple. Interfaces y polimorfismo.

Unidad 5: Clases Anidadas y Clases Internas

Objetivos:

Agrupar clases relacionadas y controlar su visibilidad.
Relacionar la herencia múltiple y las clases internas.

Clases como miembros de otra clase. El acceso irrestricto a los miembros de la clase contenedora. Resolución de la ambigüedad de nombres de miembros. Creación de objetos de la clase anidada. Clases anidadas locales. Clases anónimas y el uso de bloques de inicialización. Clases internas estáticas.



Unidad 6: El framework de Colecciones y “Genéricos”

Objetivos:

Analizar las características de los tipos y métodos genéricos de JAVA y su impacto en la programación segura.

Analizar la arquitectura del framework de colecciones.

Promover el uso de colecciones genéricas.

Tipos Genéricos y Parametrizados. Comodines. Métodos Genéricos

Arquitectura del framework de Colecciones

Interfaces core: Collections, Set, List, SortedSet, Map, SortedMap

Interfaces para ordenación de objetos: Comparable y Comparator

Interfaces para iterar: Iterator, ListIterator.

Implementaciones: HashSet, HashMap, Hashtable, ArrayList, Vector, TreeSet, TreeMap, LinkedList.

Algoritmos polimórficos para ordenación, búsqueda, manipulación de datos

Unidad 7: Construcción de Interfaces de Usuario en JAVA y Applets

Objetivos:

Analizar la arquitectura MVC (Model View Controller) de las componentes de interfaces gráficas de usuario en JAVA. Sus beneficios.

Analizar el mecanismo de manejo eventos de las interfaces gráficas de usuario en JAVA.

Usar la librería de clases del JFC (Java Foundation Classes) y las librerías de clases relacionadas al desarrollo de interfaces de usuario gráficas.

Construir applets y aplicaciones de escritorio con interfaces gráficas de usuario.

Usar JNLP (Java Network Launch Protocol) como mecanismo unificado para el despliegue de aplicaciones enriquecidas (RIAs – applets y aplicaciones Java Web Start).

La JFC (Java Foundation Classes). AWT (Abstract Window Toolkit) es un mundo de componentes de interfaz de usuario. Componentes Swing. Manejo de eventos basado en delegación. Applets, su ciclo de vida y las restricciones de seguridad. La herramienta de deployment Java Web Start.

Unidad 8: Manejo de errores mediante excepciones

Objetivos:

Simplificar la creación de programas confiables mediante el tratamiento de excepciones.

Analizar el mecanismo de tratamiento de excepciones impuesto por el compilador.

Destacar el valor fundamental de las excepciones para “informar errores”.

Tipos de Excepciones: Chequeables y no-chequeables en compilación.

Los objetos Throwable: las clases java.lang.Exception y java.lang.RuntimeException.

Manejadores de excepciones: los bloques try, catch y finally.

Propagación de excepciones: las cláusulas throws y throw



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Excepciones customizadas.
Sobreescritura de métodos que disparan excepciones

Unidad 9: Anotaciones

Objetivos:

Usar y definir anotaciones que agregan meta-información a los programas Java para usar en compilación o ejecución.
Construir procesadores de anotaciones.
Conceptos y terminología de Anotaciones. Anotaciones estándares. Definición de Anotaciones. Anotaciones y Reflection.

Unidad 10: Acceso a Bases de Datos

Objetivos:

Describir cómo se accede y consulta una base de datos desde un programa JAVA en forma independiente del motor de base de datos utilizado.
Escribir aplicaciones JAVA que accedan a bases de datos relacionales.

La API JDBC (Java DataBase Connectivity) para acceso universal a múltiples fuentes de datos. Tipos de Drivers JDBC. Establecimiento de una Conexión. Ejecución de Sentencias SQL. Las clases Statement, PreparedStatement y CallableStatement. DataSource

Unidad 11: Concurrencia: multithreading en Java

Objetivos:

Comprender los fundamentos de la programación concurrente en JAVA.
Escribir programas multithread de una complejidad razonable.
Analizar las mejoras introducidas en las versiones actuales de la plataforma JAVA para programación concurrente.

Creación y gerenciamiento de threads: la clase Thread y la interface Runnable. El ciclo de vida de un Thread
Métodos de la clase Thread: run(), sleep(), join(), interrupt() y yield().
Abstracción del gerenciamiento de threads: Ejecutores
Objetos compartidos y sincronización: bloque y métodos synchronized.
Los métodos wait(), notify() y notifyAll()

Unidad 12: Aplicaciones móviles

Objetivos:

Introducir a los estudiantes en el ciclo de vida del desarrollo de aplicaciones móviles usando tecnologías JAVA para dispositivos móviles.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Desarrollar aplicaciones móviles novedosas destinadas a dispositivos Android, enriquecidas con la ubicación del usuario, con información de contexto que se combinan con otros dispositivos móviles (robots del proyecto “Programando con Robots”: <http://robots.linti.unlp.edu.ar/inicio>).

La tecnología Android y la apertura del mundo móvil. La arquitectura del sistema operativo Android. Ciclo de vida y componentes de una aplicación Android: Activities, Intents, Services, Content Providers, BroadcastReceiver. El ambiente de desarrollo de aplicaciones Android. Interfaces de usuario gráficas. Modalidades de desarrollo de interfaces gráficas: declarativo y programático. Geolocalización y Sensores.

METODOLOGÍA DE ENSEÑANZA

Las clases están organizadas en actividades teóricas y prácticas de carácter presencial y se utiliza la plataforma de gestión de cursos en línea Moodle (<http://catedras.info.unlp.edu.ar>) como mecanismo de apoyo adicional. La asignatura es de tipo taller; teoría y práctica se encuentran estrechamente vinculadas.

Las estrategias empleadas para dictar la asignatura combinan la exposición oral para el desarrollo de la teoría y la resolución de trabajos prácticos para las clases prácticas.

En las clases teóricas se trabajan contenidos conceptuales que son vistos en forma aplicada durante las prácticas. Se preparan clases en las que se utiliza una PC y un cañón óptico, de manera tal de poder mostrar en el aula, ejemplos que pueden aplicarse a las clases prácticas. Estas explicaciones y guías se ponen a disposición de los estudiantes y docentes de la asignatura, mediante su publicación en la plataforma virtual de la asignatura.

Las clases prácticas se desarrollan en las salas de PC de la Facultad que cuentan con máquinas con doble *boot* permitiendo trabajar tanto en sistemas Microsoft Windows como GNU/Linux.

En las clases prácticas los estudiantes desarrollan un proyecto cuyo producto final es una aplicación móvil destinada a dispositivos Android, que permite integrar los contenidos desarrollados a lo largo de la cursada. Los estudiantes desarrollan el proyecto supervisados por los docentes.

Para la implementación de los trabajos prácticos se utilizan herramientas de soporte y desarrollo típicas en la comunidad de software libre y en ambientes de desarrollo profesional como ser la utilización de un sistema de versionado de código, IDEs de desarrollo, virtualización, emuladores de dispositivos móviles, servidores web, sistemas de versionado, etc ubicando a los alumnos en un ambiente profesional actual. Esto permite consolidar la formación experimental del estudiante.

EVALUACIÓN

Para la aprobación de la materia se usa un sistema de promoción que consiste en una evaluación incremental que permite evaluar el conocimiento adquirido por los estudiantes en forma modular.

A lo largo de la cursada se realizarán 4 instancias de evaluación que consistirán en la resolución y entrega de diferentes módulos en los que se divide el proyecto o de trabajos prácticos. Estas entregas se realizan en la sala de PC en los horarios de práctica de manera de permitir a los estudiantes recibir orientaciones sobre el desarrollo de las mismas.



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Para aprobar la asignatura bajo el régimen de promoción, los estudiantes deberán obtener al menos 6 puntos de promedio entre las cuatro instancias de evaluación parcial y en cada una de las mismas deberán obtener al menos 4 puntos. Asimismo, los estudiantes que se encuentran bajo este régimen, deberán realizar una quinta entrega que consiste en integrar los diferentes módulos que componen el proyecto y su puesta en funcionamiento.

Para aprobar solamente la cursada los estudiantes deberán obtener al menos 5 puntos de promedio entre las cuatro instancias de evaluación parcial y en cada una de ellas al menos 4 puntos, no accediendo al régimen de promoción y debiendo realizar una prueba escrita para aprobar la materia.

BIBLIOGRAFÍA

Effective Java™ - Second Edition. Joshua Bloch. Addison Wesley

Thinking in JAVA - Fourth Edition. Bruce Eckel. Prentice Hall

API-Documentación: <http://download.oracle.com/javase/7/docs/api/>

FAQ sobre Tipos Genéricos: <http://www.angelikalanger.com/GenericsFAQ/JavaGenericsFAQ.html>

Hello, Android (3rd edition): Introducing Google's Mobile Development Platform; Ed Burnette. Pragmatic Bookshelf

Android in Action – Third Edition. W. Frank Ableson. Robi Sen. Chris King.C. Enrique Ortiz

Professional Android™ 4 Application Development, Reto Meier. Wiley Publishing



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

CRONOGRAMA DE CLASES Y EVALUACIONES

	Contenidos	Actividades	Evaluaciones previstas
Semana del 11/8	Jueves 14/8 Unidad 1: La Plataforma JAVA Unidad 2: Clases, Espacios de Nombres, Especificadores de Acceso	Presentación de la materia	
		Presentación de la plataforma JAVA y de la iniciativa Android Presentación del tema espacios de nombre y especificadores de acceso en JAVA y su relación con el ocultamiento de información	
Semana del 18/8	Miércoles 20/8	Práctica 1: especificadores de acceso, constructores y clases abstractas	
	Jueves 21/8 Unidad 2: Interfaces y Tipos Enumerativos Unidad 4: Herencia y Polimorfismo en JAVA	Presentación y desarrollo de interfaces JAVA y su relación con la herencia.	
		Presentación de herencia y polimorfismo en JAVA.	
Semana del 25/8	Miércoles 27/8	Práctica 2: Interfaces y Polimorfismo Desarrollo y entrega durante la clase práctica de un ejercicio sobre los temas dados. Esta actividad está planteada como una actividad de seguimiento.	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

	Jueves 28/8 Unidad 5: Clases Anidadas y Clases Internas	Presentación y desarrollo de clases internas, anidadas, su relación con el ocultamiento de información y con la herencia.	
Semana del 1/9	Miércoles 3/9	Práctica 3: clases anidadas e internas. Clases anónimas y Tipos Enumerativos	
	Jueves 4/9 Unidad 5: Clases Anidadas y Clases Internas Unidad 2: Interfaces y Tipos Enumerativos	Se continúa con el tema clases anidadas e internas y se presentan los tipos enumerados en JAVA.	
Semana del 8/9	Miércoles 10/9	Se continúa trabajando con la práctica 3. Devolución del trabajo práctico entregado el miércoles 27/8. Explicación y publicación de los alcances de la primera entrega evaluatoria.	
	Jueves 11/9 Asueto por el día del "Trabajador Docente de la UNLP"		
Semana del 15/9	Miércoles 17/9	Consultas y orientaciones sobre el desarrollo y entrega de la evaluación.	Primera instancia de evaluación práctica: evaluación de las unidades 2, 4 y 5
	Jueves 18/9 Unidad 7: Construcción de Interfaces de Usuario en JAVA y Applets	Presentación y desarrollo de applets JAVA.	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Semana del 22/9	Miércoles 24/9	Práctica 4: Applets (Definición, ciclo de vida). Applet Swing. Modelo de Delegación de Eventos. Acceso a Recursos (imágenes, sonidos). Aplicaciones de escritorio. Acceso a bases de datos mediante JDBC.	
	Jueves 25/9 Unidad 7: Construcción de Interfaces de Usuario en JAVA y Applets	Presentación de la API para construcción de interfaces de usuario gráficas en JAVA. Manejo de eventos de interfaz de usuario	
Semana del 29/9	Miércoles 1/10	Se continúa con la práctica 4. Presentación de un ejemplo de construcción de GUI sin persistencia.	
	Jueves 2/10 Unidad 6: El framework de Colecciones y "Genéricos"	Presentación y desarrollo del framework de colecciones y de tipos genéricos.	
Semana del 6/10	Miércoles 8/10	Explicación de la herramienta Java Web Start (JWS). Explicación de Java e integración con Bases de Datos (JDBC) Explicación del uso del patrón DAO	
	Jueves 9/10 Unidad 8: Manejo de errores mediante excepciones	Presentación y desarrollo de manejo de errores en Java: Excepciones.	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Semana del 13/10	Miércoles 15/10	<p>Práctica 5: Arreglos de primitivos y objetos, clases de la API para manejar colecciones de objetos y tipos Genéricos.</p> <p>Explicación sobre programación de sockets en JAVA. El paquete java.net.</p> <p>Explicación y publicación de los alcances de la segunda entrega.</p>	
	Jueves 16/10 Unidad 9: Anotaciones	Presentación y desarrollo de Anotaciones.	
Semana del 20/10	Miércoles 22/10	<p>Práctica 6: Conceptos y uso de excepciones. Las cláusulas try/catch/finally. Definir Excepciones. Conceptos y uso de anotaciones. Definir anotaciones.</p> <p>Consultas y guías en el desarrollo de la segunda entrega.</p>	
	Jueves 23/10 Unidad 11: Concurrencia: multithreading en Java	Presentación y desarrollo de aplicaciones concurrentes en JAVA.	
Semana del 27/10	Miércoles 29/10	Consultas y orientaciones sobre el desarrollo y entrega de la evaluación.	Segunda instancia de evaluación práctica: evaluación de las unidades 3, 6, 7 y 10.
Semana del 3/11	Miércoles 5/11	Explicación de la api de Reflection y su uso con	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

		<p>anotaciones</p> <p>Explicación de flujos de Entrada/Salida (java.io).</p> <p>Se continúa trabajando en la práctica 6.</p>	
Semana del 10/11	Miércoles 12/11	<p>Práctica 7: Threads. Ciclo de vida. Sincronización de threads. Ejecutores</p> <p>Explicación y publicación de los alcances de la tercera entrega.</p>	
	Jueves 13/11		
	Unidad 12: Aplicaciones móviles (primera parte)	Explicación y desarrollo de aplicaciones móviles en Android.	
Semana del 17/11	Miércoles 19/11	Consultas y guías en el desarrollo de la tercera entrega.	
Semana del 24/11	Miércoles 26/11	Consultas y orientaciones sobre el desarrollo y entrega de la evaluación.	Tercera instancia de evaluación práctica: evaluación de las unidades 8, 9 y 11.
	Jueves 27/11		
	Unidad 12: Aplicaciones móviles (segunda parte)	Explicación y desarrollo de aplicaciones móviles en Android enriquecidas con la ubicación del usuario y de información de contexto	
Semana del 1/12	Miércoles 3/12	TALLER DE ANDROID -PRIMERA PARTE-	
		Práctica 8: Ambiente de desarrollo Android. Aplicación, Activities, Views, resources y Layouts. Diseño declarativo de interfaz gráfica.	



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

		Desarrollo de ejemplos durante la clase práctica	
Semana del 8/12	Miércoles 10/12	TALLER DE ANDROID – Segunda Parte – Explicación y publicación de los alcances de la cuarta entrega. Explicación de conectividad en Android: permisos, asyncTask, sockets y servicios REST. Ejemplos de Drag&Drop en Android. Consultas y guías en el desarrollo de la cuarta entrega.	
Semana del 15/12	Miércoles 17/12	Consultas y orientaciones sobre el desarrollo y entrega de la evaluación.	Cuarta instancia de evaluación práctica: evaluación de la unidad 12.

La fecha de la quinta entrega del proyecto se acordará con los estudiantes.

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):

Prof. Claudia Queiruga: claudiaq@info.unlp.edu.ar

Jefes de Trabajos Prácticos: Jorge Rosso (jrosso@info.unlp.edu.ar) y Pablo Iuliano (piuliano@info.unlp.edu.ar)

Plataforma virtual: <https://catedras.info.unlp.edu.ar/> (categoría "Cursos 2014")



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMÁTICA

Firmas del/los profesores responsables: