

LABORATORIO DE SOFTWARE**Año 2017****Carrera/ Plan:***Licenciatura en Informática*

Plan 2003-07/ Plan 2012/2015

Licenciatura en Sistemas

Plan 2003-07/ Plan 2012/2015

Año: 4°(Lic en Informática)

4° y 5° (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 - Jorge Rosso***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/la estudiante 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 de Experiencia de 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.

OBJETIVOS GENERALES

Trabajo integrador que signifique para el/la estudiante 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 MÍNIMOS

Se desarrollará una aplicación específica que integra conocimientos teóricos y herramientas conocidas por el estudiantes. 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 Standard 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.

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

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 aplicaciones de escritorio con interfaces gráficas de usuario.

Usar JNLP (Java Network Launch Protocol) como mecanismo unificado para el despliegue de aplicaciones enriquecidas (RIAs).

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. 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
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.

Diseño de la capa de persistencia aplicando el patrón DAO

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. La interface Executor y sus subinterfaces. El factory Executors.

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.

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 y otras aplicaciones.

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.

Persistencia de datos de la aplicación: tecnología SQLite.

Preferencias de usuario.

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, etc ubicando a los/as estudiantes en un ambiente profesional actual. Esto permite consolidar la formación experimental del estudiante.

EVALUACIÓN



Para la aprobación de la asignatura, el/la estudiante deberá desarrollar a lo largo de la cursada un proyecto cuyo producto final es una aplicación móvil destinada a dispositivos Android. El desarrollo del proyecto contempla 5 entregas con fechas pautadas. Cada una de estas entregas se utilizará para seguimiento del proyecto y serán evaluadas con calificación.

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.

El régimen de aprobación de la materia es de promoción directa: los estudiantes deberán obtener al menos 6 puntos de promedio entre las cinco instancias de evaluación parcial y en cada una de las mismas deberán obtener al menos 4 puntos. La quinta entrega consiste en integrar los diferentes módulos que componen el proyecto y su puesta en funcionamiento.

Los estudiantes que no alcancen los requisitos de la promoción directa y hayan cumplido con todas las entregas habiendo obtenido al menos 4 puntos en cada una de ellas, aprobarán la cursada. Para aprobar la asignatura deberán rendir un examen final escrito en una de las mesas de examen contemplada en el calendario académico.

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

CRONOGRAMA DE CLASES Y EVALUACIONES

Fecha	Contenidos	Actividades	Evaluaciones previstas
Clase 1: Miércoles 16/8	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 y la herencia. Práctica 1: especificadores de acceso, constructores y clases abstractas	
Clase 2: Miércoles 23/8	Unidad 2: Interfaces y Clases Abstractas 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. Práctica 2: Interfaces y Polimorfismo. TALLER "Bots Conquistadores" Objetivo del taller: desarrollar una estrategia ganadora.	
Clase 3: Miércoles 30/8	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. TALLER "INTERFACES Y CLASES ANIDADAS" Objetivo del taller: implementar el estrategia usando un strategy basado en clases internas e interfaces.	
Clase 4: Miércoles 6/9	Unidad 2: Tipos Enumerativos Unidad 7: Construcción de Interfaces de Usuario en JAVA	Presentación de tipos Enumerativos y de la API para construcción de interfaces de usuario gráficas. Manejo de eventos de interfaz de usuario.	



		Práctica 3: clases anidadas e internas. Clases anónimas y Tipos Enumerativos. TALLER "Singleton Charly García"	
Clase 5: Miércoles 13/9		Consultas y orientaciones sobre el desarrollo y entrega de la evaluación.	Primera instancia de evaluación práctica: evaluación de las unidades 2, 4 y 5
Clase 6: Miércoles 20/9	Unidad 7: Construcción de Interfaces de Usuario en JAVA (continuación) Unidad 10: Acceso a BD (JDBC). Patrón DAO.	Práctica 4: Modelo de Delegación de Eventos. Acceso a Recursos (imágenes, sonidos). Aplicaciones de escritorio. Acceso a BD. Explicación de la herramienta Java Web Start (JWS). TALLER "Compositor de melodías musicales" (se comienza a trabajar)	
Clase 7: Miércoles 27/9	Unidad 6: "Genéricos"	Presentación del framework de colecciones. Se continúa con la práctica 4. Entrega del "Compositor de melodías musicales"	
Clase 8: Miércoles 4/10	Unidad 6: El framework de Colecciones	Presentación y desarrollo de manejo de errores en Java: Excepciones. Práctica 5: Arreglos de primitivos y objetos, clases de la API para manejar colecciones de objetos y tipos Genéricos.	
Clase 9: Miércoles 11/10	Unidad 8: Manejo de errores mediante excepciones	Presentación y desarrollo de Anotaciones.	Segunda instancia de evaluación práctica: evaluación de las unidades 3, 6, 7 y 10. Entregar con el JAR FUNCIONANDO



<p>Clase 10: Miércoles 18/10</p>	<p>Unidad 11: Concurrencia: multithreading en Java</p>	<p>Presentación y desarrollo de aplicaciones concurrentes en JAVA.</p> <p>Práctica 6: Conceptos y uso de excepciones. Las cláusulas try/catch/finally. Definir Excepciones. Conceptos y uso de anotaciones. Definir anotaciones.</p> <p>Práctica 7: Threads. Ciclo de vida. Sincronización de threads. Ejecutores.</p>	
<p>Clase 11: Miércoles 25/10</p>	<p>Unidad 9: Anotaciones</p> <p>Explicación sobre programación de sockets en JAVA. El paquete java.net. Explicación de la api de Reflection y su uso con anotaciones.</p>	<p>Práctica de anotaciones</p>	
<p>Clase 12: Miércoles 1/11</p>	<p>Unidad 12: Aplicaciones móviles (primera parte)</p>	<p>Presentación de desarrollo de aplicaciones móviles con tecnologías Android. La arquitectura de Android. Las componentes de las aplicaciones.</p>	<p>Tercera instancia de evaluación práctica: evaluación de las unidades 8, 9 y 11.</p>
<p>Clase 13: Miércoles 8/11</p>	<p>Unidad 12: Aplicaciones móviles (segunda parte)</p>	<p>Taller sobre Android Studio (entorno de desarrollo de Android)</p> <p>Práctica 8: Ambiente de desarrollo Android. Aplicación, Activities, Views, resources y Layouts. Diseño declarativo de interfaz gráfica.</p> <p>Explicación y publicación del enunciado de la Cuarta Entrega (Trabajo Final)</p>	
<p>Clase 14: Miércoles 15/11</p>		<p>Consultas y guías en el desarrollo de la cuarta entrega.</p>	
<p>Clase 15: Miércoles 22/11</p>		<p>Desarrollo de ejemplos durante la clase práctica. Action Bar -Menu- Drag and Drop</p> <p>Explicación y publicación del enunciado de la Quinta Entrega (Trabajo Final)</p>	<p>Cuarta instancia de evaluación práctica: evaluación de la unidad 12 (16:30)</p>
<p>Clase 16: Miércoles 29/12</p>		<p>Desarrollo de ejemplos de BD SQLite.</p>	<p>Evaluación Flotante para los estudiantes que adeudan entregas o no llegaron a la</p>



			nota de la promoción
VACACIONES			
Febrero		Quinta Entrega: Entrega Final (se desdobra en Febrero)	

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

Profesores: Claudia Queiruga (claudiaq@info.unlp.edu.ar) y Jorge Rosso (jrosso@info.unlp.edu.ar)

Jefes de Trabajos Prácticos: Pablo Iuliano (piuliano@info.unlp.edu.ar) y Diego Bellante (diegobellante@gmail.com)

Plataforma virtual: <https://catedras.info.unlp.edu.ar/> (categoría "Cursos 2017")

Firmas del/los profesores responsables: