

Carrera/ Plan:**LABORATORIO DE SOFTWARE**

Licenciatura en Informática Plan 2015/Plan 2012/Plan 2003-07
Licenciatura en Sistemas Plan 2015/Plan 2012/Plan 2003-07

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)

Año 2019

Correlativas: *Proyecto de Software*

Profesor/es: *Claudia Queiruga y Jorge Rosso*

Hs. semanales: 6

FUNDAMENTACIÓN

Laboratorio de Software es una asignatura obligatoria de cuarto año de la carrera Licenciatura en Informática y de carácter optativa de cuarto y quinto año de la carrera Licenciatura en Sistemas. Las y los estudiantes que cursan Laboratorio de Software cuentan con los conocimientos fundamentales de la Informática en diferentes áreas y han 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 programación en lenguaje JAVA y, con materias de quinto año relacionadas a tecnologías de desarrollo: Java y Aplicaciones Avanzadas en Internet, Diseño de Experiencia de Usuario e 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

Desarrollar un trabajo integrador que signifique para las y los estudiantes la aplicación concreta de los conocimientos adquiridos hasta el momento, en particular como una evolución de lo que ya han trabajado en Proyecto de Software.

COMPETENCIAS

- CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.
- CGS2- Comunicarse con efectividad en forma oral y escrita.
- CGS3- Actuar con ética, responsabilidad profesional y compromiso social y ambiental, considerando el impacto económico, social y ambiental de su actividad en el contexto local, regional y global.
- CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.
- CGT2- Concebir, diseñar y desarrollar proyectos de Informática.

- CGT3- Gestionar, planificar, ejecutar y controlar proyectos de Informática.
- CGT8 Capacidad de interpretación y resolución de problemas multidisciplinarios, desde los conocimientos de la disciplina informática.
- LI- CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de especificación, diseño, implementación, verificación, validación, puesta a punto, mantenimiento y actualización para arquitecturas de sistemas de procesamiento de datos, con capacidad de incorporar aspectos emergentes del cambio tecnológico.
- LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LI- CE6 – Controlar las normas de calidad en el software o software integrado a otros componentes. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.
- LS-CE1- Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.
- LS-CE5- Establecer métricas y normas de calidad y seguridad de software, contralando las mismas a fin de tener un producto industrial que respete las normas nacionales e internacionales. Control de la especificación formal del producto, del proceso de diseño, desarrollo, implementación y mantenimiento. Establecimiento de métricas de validación y certificación de calidad. Capacidad de evaluación de performance de sistemas de software y sistemas que integren hardware y software.
- LS-CE9- Analizar y evaluar proyectos de especificación, diseño, implementación, puesta a punto, mantenimiento y actualización de sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico.

CONTENIDOS MÍNIMOS

Se desarrollará una aplicación específica que integra conocimientos teóricos y herramientas conocidas por el/la estudiante. 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 de aprendizaje:

- 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 de aprendizaje:

- Relacionar conceptos de orientación a objetos y su implementación en JAVA.
- Definir tipos de datos usando interfaces y tipos enumerativos.
- Usar el formato de empaquetado y compresión de archivos 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 de aprendizaje:

- Examinar la característica de autoboxing/unboxing de JAVA.
- 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 de aprendizaje:

- Profundizar en el concepto de herencia simple y su implementación en JAVA.
- Definir interfaces JAVA y su relación con la herencia múltiple.
- Comprender el concepto de upcasting automático a clases e interfaces.
- Usar buenas prácticas de programación para creación de objetos en JAVA.
- Relacionar el control de acceso con la herencia.

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 de aprendizaje:

- 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 de aprendizaje:

- Analizar las características de los tipos y métodos genéricos de JAVA y su relación con la programación segura.
- Analizar la arquitectura del framework de colecciones.
- Usar de colecciones de tipos 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 de aprendizaje:

- 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 en el desarrollo de interfaces de usuario gráficas.
- Construir aplicaciones de escritorio con interfaces gráficas de usuario.
- Usar JNLP (Java Network Launch Protocol) como protocolo unificado para el despliegue de aplicaciones enriquecidas (RIAs).

La JFC (Java Foundation Classes). AWT (Abstract Window Toolkit). Componentes Swing. Manejo de eventos basado en delegación. La herramienta de deployment Java Web Start.

Unidad 8: Manejo de errores mediante excepciones

Objetivos de aprendizaje:

- 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 de aprendizaje:

- 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 de aprendizaje:

- 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 de aprendizaje:

- 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: Lambda y Streams

Objetivos de aprendizaje:

- Introducirse en la programación funcional en JAVA: expresiones Lambda.
- Introducirse en la API Streams: el complemento de las Colecciones. La aplicación de Streams para resolver problemas de clasificación, identificar el mayor y menor elemento y reducir duplicados.
- Identificar cuándo usar Lambda y cuándo no.
- Mejorar la performance con streams paralelos.

De las clases anónimas a expresiones Lambda. Tipos de expresiones Lambda: Consumidores, Proveedores, Funciones y Predicados. Referencias a métodos. Uso de Collectors.

Unidad 13: Aplicaciones móviles

Objetivos de aprendizaje:

- Introducirse 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.

BIBLIOGRAFÍA

Effective Java™ 2nd Edition. Joshua Bloch. Editorial Addison Wesley, 2008. 9780321356680.

Piensa en JAVA, 4ta Edición. Bruce Eckel. Editorial Prentice Hall, 2007. ISBN: 9788489660342

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

Sitio oficial de JAVA: <http://www.oracle.com/technetwork/java/javase/>

Head First Android Development: A Brain-Friendly Guide. Dawn Griffiths, David Griffiths. Editorial O'Reilly Media, 2015. ISBN: 9781449362188.

Android User Interface Design: Implementing Material Design for Developers, 2nd Edition. Ian G. Clifton. Editorial Addison-Wesley Professional, 2015. ISBN: 9780134191409.

Programming Android: Java Programming for the New Generation of Mobile Devices, 2nd Edition. Zigurd Mednieks, Laird Dornin, G. Blake Meike, Masumi Nakamura. O'Reilly Media, 2012. ISBN: 1449316646.

Sitio oficial de Android para desarrolladores: <https://developer.android.com>

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 de la Facultad de Informática (<http://catedras.info.unlp.edu.ar>) como 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, asimismo se dispone de acceso a Internet en las aulas, de esta manera es posible mostrar durante las clases ejemplos que pueden aplicarse a las clases prácticas. Estas explicaciones y guías se ponen a disposición de las y los estudiantes de la asignatura mediante su publicación en la plataforma virtual de la asignatura. Las clases prácticas se desarrollan en una de las salas de PC de la Facultad que cuentan con máquinas acceso a Internet y doble booteo permitiendo trabajar tanto en sistemas Microsoft Windows como GNU/Linux.

En las clases prácticas las y los estudiantes desarrollan los trabajos prácticos, las evaluaciones parciales y el proyecto final. Las y los estudiantes desarrollan el proyecto supervisados por los docentes, cada grupo tiene asignado un docente que acompaña el desarrollo del mismo.

La formulación del proyecto final se articula con la Secretaría de Extensión y da respuesta a necesidades de organizaciones de la sociedad civil y/o instituciones de gestión pública con las que se trabaja desde dicha Secretaría. Desde el año 2014 venimos trabajando junto a CEDICA (Centro de Equitación para Personas con Discapacidad y Carenciadas) en el desarrollo de aplicaciones móviles destinada a niñas, niños y jóvenes que concurren a la institución y que son utilizadas como complemento innovador a la terapia que allí se realiza.

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, tablets, ubicando a las y los estudiantes en un ambiente profesional actual. Esto permite consolidar la formación experimental del estudiante.

EVALUACIÓN

Para la aprobación de la asignatura, los/as estudiantes deberán desarrollar durante la cursada un proyecto cuyo producto final es una aplicación móvil destinada a dispositivos Android. La evaluación de la asignatura se organiza en 4 instancias parciales, que cubren todos los temas dados a lo largo de la cursada. Las dos primeras evaluaciones son individuales. El desarrollo del proyecto es grupal, los equipos se conforman por 2 estudiantes y comienza en la tercera evaluación parcial. La cuarta evaluación consiste en la entrega final del proyecto.

Las evaluaciones se realizan en la sala de PC, utilizando el equipamiento y el software provisto por la Facultad, 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 4 instancias de evaluación parcial y en cada una de las mismas deberán obtener al menos 4 puntos. La cuarta 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.

Adquisición de las competencias

CGS1- Desempeñarse de manera efectiva en equipos de trabajo, con capacidad para organizarlos y liderarlos.

En la asignatura se organizan actividades por equipos de trabajo, con un número variable de estudiantes entre 2 y 4, los cuales tienen roles específicos, en algunos casos que trabajarán como pares sin jerarquías en el grupo.

En la evaluación del trabajo en equipo se definen momentos para las presentaciones del equipo donde se responden aspectos generales de las tareas asignadas y explicaciones individuales que permiten calificar diferentes aptitudes de los miembros del equipo, entre ellas sus conocimientos, modo de expresarse y predisposición al trabajo colaborativo. Esto se refleja en planillas que mantiene el responsable de la práctica y son informadas al Profesor/Profesores de la asignatura para ser tenidas en cuenta en las evaluaciones de los estudiantes.

CGS2- Comunicarse con efectividad en forma oral y escrita.

En la cátedra se evalúa a los estudiantes en coloquios en los cuales expondrán la tarea realizada en forma individual y donde el docente evalúa sus conocimientos, la claridad de la presentación, su organización y la forma de expresión. Esto se refleja en planillas escritas que conforman documentación de evaluación del coloquio.

CGS3- Actuar con ética, responsabilidad profesional y compromiso social y ambiental, considerando el impacto económico, social y ambiental de su actividad en el contexto local, regional y global.

En la asignatura se trabaja esta competencia en base al planteo de casos, donde los estudiantes presentan sus ideas respecto de aspectos de responsabilidad profesional y ética personal y de la organización involucrada en el caso, en relación a los usos de las tecnologías digitales. Asimismo, se discute el impacto social y ambiental del producto informático que produce la organización.

En la evaluación de estos “análisis de caso” las producciones de los estudiantes se reflejan en una planilla detallada que son compartidas por los responsables de teoría y práctica para considerarlas en las evaluaciones de los estudiantes y como devolución que puede considerarse como “recomendación” o material de estudio complementario para ayudar a la formación en esta competencia.

CGS6- Capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras.

En la asignatura se plantean actividades planificadas para los estudiantes en las que se propone el estudio de tecnologías existentes y previstas para un tipo de problema y se los “desafía” a presentar la posible evolución de la solución para ese tipo de problema y en QUE podría mejorarse o no la solución.

Estas actividades conducen al estudiante a buscar bibliografía relacionada con el cambio tecnológico y formarse un criterio sobre las tendencias, entre ellas procesadores a utilizar, tecnologías de conectividad, extracción y análisis de información, servicios a incorporar, etc.

La cátedra acompaña el proceso del estudiante para contrastar las conclusiones del estudiante y validar su habilidad para esta competencia. La evaluación de esta competencia se refleja en una planilla detallada, donde se indican los resultados del estudio bibliográfico del estudiante y su capacidad para analizar críticamente el cambio tecnológico en el problema planteado.

CGT2- Concebir, diseñar y desarrollar proyectos de Informática.

En la cátedra se pone énfasis en el proceso de identificación de problemas reales, contextualizados, en la especificación de los mismos como problemas resolubles desde la informática, en la formulación de proyectos de software para su solución y en el análisis y diseño de las soluciones en el marco del proyecto formulado.

La evaluación de esta competencia forma parte de las evaluaciones de las etapas de entrega del proyecto final de la asignatura.

CGT3- Gestionar, planificar, ejecutar y controlar proyectos de Informática.

En la asignatura se hace un seguimiento de proyectos, trabajando con grupos de estudiantes con roles en el marco de un proyecto de software. El estudiante fortalece su formación en esta competencia en el desarrollo del curso, siendo evaluado por los docentes en el contexto del rol asignado en el proyecto.

La evaluación efectiva de esta competencia forma parte de las evaluaciones de entregas del proyecto final de la asignatura.

CGT8 Capacidad de interpretación y resolución de problemas multidisciplinarios, desde los conocimientos de la disciplina informática.

En la cátedra se tratan proyectos que involucran la participación de múltiples disciplinas, intentando un abordaje interdisciplinario. El docente acompaña al estudiante en la interpretación del rol del Informático en equipos de trabajo que involucran la participación de expertos extra-disciplinarios.

La cátedra acompaña el aprendizaje del estudiante sobre esta competencia con materiales bibliográficos de casos de interés que resulten informativos y motivadores para el estudiante.

La evaluación de esta competencia forma parte de las evaluaciones de las etapas del proyecto final de la asignatura.

LI- CE1 – Planificar, dirigir, realizar y/o evaluar proyectos de especificación, diseño, implementación, verificación, validación, puesta a punto, mantenimiento y actualización para arquitecturas de sistemas de procesamiento de datos, con capacidad de incorporar aspectos emergentes del cambio tecnológico.

LS-CE9- Analizar y evaluar proyectos de especificación, diseño, implementación, puesta a punto, mantenimiento y actualización de sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico.

En la cátedra se trabaja esta competencia mediante el seguimiento de proyectos de software concernientes al procesamiento de datos y aspectos innovadores en las tecnologías actuales, con especial atención en aplicaciones móviles. Los estudiantes trabajarán en equipos en el marco del proyecto, planificando etapas de entregas y de discusión y evaluación sobre las nuevas tendencias en tecnologías informáticas y las posibilidades de incorporación en sus proyectos. Los estudiantes fortalecen su formación en esta competencia durante el desarrollo del curso, siendo acompañados en la interpretación y aprendizaje de los nuevos emergentes tecnológicos.

La evaluación por los docentes se realizará mediante el seguimiento del proyecto final de la asignatura.

LI- CE4 – Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real, especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software/sistemas de información que se ejecuten sobre equipos de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

LS-CE1- Planificar, dirigir, realizar y/o evaluar proyectos de relevamiento de problemas del mundo real. Especificación formal, diseño, implementación, prueba, verificación, validación, mantenimiento y control de calidad de sistemas de software que se ejecuten sobre sistemas de procesamiento de datos, con capacidad de incorporación de tecnologías emergentes del cambio tecnológico. Capacidad de análisis, diseño y evaluación de interfases humano computador y computador-computador.

En la cátedra se trabaja esta competencia mediante la identificación de problemas reales, contextualizados; la especificación de los mismos como problemas abordables desde la informática, la

formulación de proyectos de software para su solución, análisis y diseño de las soluciones en el marco del procesamiento de datos y de aspectos innovadores en las tecnologías actuales. Se pondrá especial atención en aplicaciones móviles y, en aspectos de diseño de interfaces humano-computador y computador-computador. Los estudiantes fortalecen su formación en esta competencia durante el desarrollo del curso. Los docentes guían a los estudiantes en la interpretación, aprendizaje y evaluación, de todas las etapas del proyecto.

La evaluación de esta competencia forma parte de las evaluaciones parciales prácticas y el proyecto final de la asignatura y se refleja en las correcciones que recibe el estudiante.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos	Actividades
Clase 1	Miércoles 21/8	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 nombres 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 28/8	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 "Competencia de Bots" Objetivo del taller: desarrollar una estrategia ganadora.
Clase 3	Miércoles 4/9	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. Práctica 5: Clases anidadas y clase internas. Clases anónimas TALLER "Competencia de Bots continúa....."
Clase 4	Miércoles 11/9	Unidad 7: Construcción de Interfaces de Usuario en JAVA	Presentación de la API para construcción de interfaces de usuario gráficas. Manejo de eventos de interfaz de usuario. Práctica 3: Modelo de Delegación de Eventos.

			<p>Acceso a Recursos (imágenes). Aplicaciones de escritorio. Acceso a bases de datos mediante JDBC</p> <p>TALLER "Compositor de melodías musicales" (se comienza a trabajar)</p>
Clase 5	Miércoles 18/9	<p>Unidad 2: Tipos Enumerativos</p> <p>Unidad 7: Construcción de Interfaces de Usuario en JAVA (continuación)</p> <p>Unidad 10: Acceso a BD (JDBC). Patrón DAO.</p>	<p>Presentación de tipos Enumerativos. Presentación de acceso a BD en JAVA. Patrón DAO</p> <p>Se continúa con la Práctica 3. Práctica 4: Tipos Enumerativos.</p> <p>Explicación de la herramienta Java Web Start en la práctica (JWS).</p> <p>TALLER "Compositor de melodías musicales" (entrega al iniciar la práctica)</p> <p>TALLER "Singleton Charly García"</p>
Clase 6	Miércoles 25/9		Primera evaluación parcial.
Clase 7	Miércoles 2/10	<p>Unidad 6: "Genéricos" El framework de Colecciones</p>	<p>Presentación del framework de colecciones y de tipos genéricos.</p> <p>Práctica 6: Arreglos de primitivos y objetos, clases de la API para manejar colecciones de objetos y tipos Genéricos. Programación Lambda para procesamiento de colecciones.</p>
Clase 8	Miércoles 9/10	Unidad 12: Lambda y Streams	<p>Presentación de programación funcional en JAVA y su aplicación a procesamiento de colecciones. Práctica 6 (continuación): Arreglos de primitivos y objetos, clases de la API para manejar colecciones de objetos y tipos Genéricos. Programación Lambda para procesamiento de colecciones</p>
Clase 9	Miércoles 16/10	Unidad 8: Manejo de errores mediante excepciones	Presentación y desarrollo de manejo de errores en Java: Excepciones.

			Práctica 7: Conceptos y uso de excepciones. Las cláusulas try/catch/finally. Definir Excepciones.
Clase 10	Miércoles 23/10	Unidad 9: Anotaciones Explicación de la api de Reflection y su uso con anotaciones. Explicación sobre programación de sockets en JAVA. El paquete java.net.	Presentación y desarrollo de Anotaciones. Práctica 8: Conceptos y uso de anotaciones. Definir anotaciones.
Clase 11	Miércoles 30/10	Unidad 11: Concurrencia: multithreading en Java	Presentación y desarrollo de aplicaciones concurrentes en JAVA. Práctica 9: Threads. Ciclo de vida. Sincronización de threads. Ejecutores.
Clase 12	Miércoles 6/11		Segunda evaluación parcial.
Clase 13	Miércoles 13/11	Unidad 13: Aplicaciones móviles (primera parte)	Presentación del desarrollo de aplicaciones móviles con tecnologías Android. La arquitectura de Android. Las componentes de las aplicaciones. Explicación sobre Android Studio (entorno de desarrollo de Android) Práctica 10: Ambiente de desarrollo Android. Aplicación, Activities, Views, resources y Layouts. Diseño declarativo de interfaz gráfica.
FERIADO NACIONAL 20/11			
Día de la Soberanía Nacional			
Clase 14	Miércoles 27/11	Unidad 13: Aplicaciones móviles (segunda parte)	Explicación y publicación del enunciado del Trabajo Final. Se comienza a trabajar en el proyecto.
Clase 15	Miércoles 4/12		Consultas y guías para el desarrollo del trabajo final
Clase 16	Miércoles 11/12		Consultas y guías para el desarrollo del trabajo final. Evaluación Flotante para los

			estudiantes que adeudan entregas o no llegaron a la nota de la promoción.
Clase 17	Miércoles 18/12		Tercera evaluación parcial.
VACACIONES			
Febrero			Consultas y entrega final del Proyecto

Evaluaciones previstas	Fecha
<p>Primera evaluación parcial:</p> <p>Se evalúan las unidades: 2 (tipos enumerativos), 4 (herencia y polimorfismo en JAVA), 5 (clases anidadas y clases internas), y 7 (construcción de interfaces de usuario gráficas)</p>	25/9
<p>Segunda evaluación parcial:</p> <p>Se evalúan las unidades: 6 (colecciones y genericos), 8 (excepciones), 9 (anotaciones), 11 (threads) y 12 (Lambda & Streams).</p> <p>Evaluación Flotante para los/as estudiantes que adeudan evaluaciones parciales o no alcanzaron a la nota de la promoción directa</p>	6/11
<p>Tercera evaluación parcial:</p> <p>Prueba de concepto del trabajo final</p>	18/12
<p>Cuarta evaluación: Entrega Final del Proyecto</p> <p>La primera semana de febrero hay clase de consulta y la segunda se realiza la entrega final.</p>	Febrero de 2020

Contacto de la cátedra (mail, sitio WEB, 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 2019")



Firma del/los profesor/es