

**LABORATORIO DE SOFTWARE****Carrera/ Plan:**

*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 2021**

**Correlativas:** *Proyecto de Software*

**Profesores:** *Claudia Queiruga y Pablo Iuliano*

**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.

**RESULTADOS DE APRENDIZAJE**

1.3. Describir los avances informáticos actuales e históricos y demostrar cierta visión sobre tendencias y avances futuros (Básico).

1.4. Aplicar e integrar conocimientos de otras disciplinas informáticas como apoyo al estudio de la propia área de especialidad (o áreas de especialidad) (Básico).

- 1.5. Demostrar sensibilización ante la necesidad de contar con amplios conocimientos a la hora de crear aplicaciones informáticas en otras áreas temáticas (Básico).
- 2.5. Analizar la medida en la que un determinado sistema informático cumple con los criterios definidos para su uso actual y desarrollo futuro (Básico).
- 3.1. Definir y diseñar hardware/software informático/de red que cumpla con los requisitos establecidos (Básico).
- 3.2. Describir las fases implicadas en distintos modelos de ciclo de vida con respecto a la definición, construcción, análisis y puesta en marcha de nuevos sistemas y el mantenimiento de sistemas existentes (Básico).
- 3.3. Elegir y utilizar modelos de proceso adecuados, entornos de programación y técnicas de gestión de datos con respecto a proyectos que impliquen aplicaciones tradicionales así como aplicaciones emergentes (Adecuado).
- 3.4. Describir y explicar el diseño de sistemas e interfaces para interacción persona-ordenador y ordenador-ordenador (Básico).
- 3.5. Aplicar las correspondientes competencias prácticas y de programación en la creación de programas informáticos y/u otros dispositivos informáticos (Básico).
- 4.3. Identificar los requisitos jurídicos que rigen las actividades informáticas, incluyendo la protección de datos, derechos de propiedad intelectual, contratos, cuestiones de seguridad del producto y responsabilidad, cuestiones personales y riesgos laborales (Básico).
- 5.1. Demostrar conocimientos sobre los códigos y estándares de cumplimiento del sector (Básico).
- 5.5. Diseñar y llevar a cabo investigaciones prácticas (por ejemplo, de rendimientos de sistemas) para interpretar datos y extraer conclusiones (Adecuado).

## **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.

## **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 disponible? 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: 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 8: 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 9: 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 10: 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 11: 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 12: 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. 3rd ed. Joshua Bloch. Addison-Wesley Professional. 2018.

The Java module system. Parlog, N. Manning. 2019.

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**

La modalidad de dictado de la asignatura es presencial de acuerdo a lo establecido en el plan de estudio de las carreras a las que pertenece, sin embargo la emergencia de la pandemia COVID-19, las disposiciones nacionales tanto de aislamiento y distanciamiento, social preventivo y obligatorio, como las disposiciones de la UNLP, han requerido suspender las clases presenciales en 2020. En 2021 la UNLP prevé continuar con la modalidad virtual al menos durante el 1er semestre para el dictado de las clases. Es por ello que en este apartado se describen ambas modalidades.

En términos generales se ha intentado conservar la naturaleza de la metodología de enseñanza de asignatura, sobre todo el trabajo en talleres, a pesar del cambio a la modalidad virtual.

Las clases están organizadas en actividades teóricas y prácticas, en las clases teóricas se trabajan contenidos conceptuales que son vistos en forma aplicada durante las prácticas. Las clases se desarrollan bajo una metodología de trabajo en talleres en donde la teoría y práctica se encuentran estrechamente vinculadas.



Las estrategias empleadas para el dictado de la asignatura combinan:

- la exposición oral para el desarrollo de los contenidos teóricos,
- explicaciones de herramientas y/o tecnologías necesarias para el desarrollo de determinadas actividades,
- la resolución de trabajos prácticos, con entregas pautadas,
- la resolución de problemas con entregas pautadas,
- muestras y coloquios de las producciones para el seguimiento del proceso formativo,
- el desarrollo de un **proyecto de software final integrador**.

En las clases prácticas, las y los estudiantes desarrollan los trabajos prácticos, participan de talleres con actividades entregables, realizan las muestras y coloquios y, desarrollan el proyecto final.

Las y los estudiantes desarrollan el proyecto final en equipos de 2 estudiantes, en algunos casos se admiten grupos de 3 y son supervisados por los docentes. Cada grupo tiene asignado un docente que acompaña la evolución de los aprendizajes y el desarrollo del proyecto final. Esta asignación se realiza al comenzar la cursada y es el mismo docente el que acompaña todas las actividades desarrolladas tanto individuales como grupales favoreciendo el seguimiento del proceso formativo.

La formulación del proyecto final se articula con la Secretaría de Extensión de la Facultad 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.

Para la implementación de los trabajos prácticos se utilizan herramientas de desarrollo y soporte típicas en la comunidad de software libre y en ambientes de desarrollo profesional, ejemplo de ellos son la utilización de sistemas de versionado de código, IDEs de desarrollo, virtualización, emuladores de dispositivos móviles, servidores web, motor de BD, dispositivos móviles (tablets), ubicando a las y los estudiantes en un ambiente profesional actual. Esto favorece la consolidación de la formación experimental del estudiante.

### **Modalidad Virtual**

Las clases están organizadas en actividades teóricas y prácticas en modalidad virtual, para ello se utiliza el EVEA (Entorno Virtual de Enseñanza y Aprendizaje) de la Facultad de Informática <http://catedras.info.unlp.edu.ar>.

Las clases se desarrollan de manera sincrónica utilizando la herramienta BBB (Big Blue Button) incorporada en el EVEA. Se dispone de una sala para la clase teórica, y de múltiples salas asignadas a los docentes en la que cada uno trabaja con sus grupos asignados. De esta manera las salas no son numerosas y la atención puede hacerse en forma personalizada. Las clases teóricas se graban facilitando el acceso a las mismas en forma diferida atendiendo a los posibles cortes de energía y/o Internet o a la imposibilidad de conectarse en el horario establecido. Para las muestras y coloquios se asigna a las y los estudiantes el perfil presentador, facilitando de esta manera el diálogo.

El material generado para las clases teóricas, explicaciones y guías se publican en el EVEA.

Se utiliza la facilidad de tareas programadas del EVEA para completar las actividades entregables y las evaluaciones.

Se habilitan 2 foros, uno de avisos usado frecuentemente para comunicar novedades y en varios casos también para consultas técnicas a lo largo de la cursada y otro destinado al proyecto final. Asimismo la mensajería privada del EVEA está habilitada para la comunicación personal entre los docentes y estudiantes.

### **Modalidad Presencial**

Las clases están organizadas en actividades teóricas y prácticas en modalidad presencial y se utiliza el EVEA (Entorno Virtual de Enseñanza y Aprendizaje) de la Facultad de Informática <http://catedras.info.unlp.edu.ar> como apoyo adicional a las actividades presenciales.

En esta modalidad para la exposición de las clases se utiliza una PC y un cañón óptico dispuesto en el aula, asimismo se dispone de acceso a Internet, de esta manera es posible mostrar on-line, durante las clases, ejemplos que pueden aplicarse a las clases prácticas. Estas explicaciones y guías se ponen a disposición de las y los estudiantes mediante su publicación en el EVEA. Se utiliza la facilidad de tareas programadas del EVEA para completar las actividades entregables y las evaluaciones.

Las clases prácticas y las evaluaciones se desarrollan en una de las salas de PC de la Facultad, se dispone de computadoras con acceso a Internet y doble booteo, permitiendo trabajar tanto en sistemas Microsoft Windows como GNU/Linux.

## **EVALUACIÓN**

La evaluación se organiza en 5 instancias de evaluación parcial, que cubren todos los temas dados a lo largo de la cursada: las 2 primeras evaluaciones son individuales y las 3 restantes grupales y relativas al proyecto integrador, en formato de “entregables”. La última evaluación consiste en la entrega final del proyecto.

En la modalidad presencial 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.

En la modalidad virtual las evaluaciones se desarrollan en salas de BBB y los estudiantes tienen asignada la sala de su docente guía. En el caso de las evaluaciones individuales, al iniciar la evaluación se explican las consignas del trabajo y qué se espera que hagan las y los estudiantes, en una sala general; se atienden consultas y luego cada estudiante continúa en su sala asignada. Al finalizar se realiza un breve coloquio, con la intención que las y los estudiantes puedan explicar la solución planteada y las decisiones tomadas. En el caso de las evaluaciones grupales, referidas al proyecto final, las evaluaciones son “entregables”; también se realizan coloquios en la sala de BBB.

**El régimen de aprobación de la materia es por promoción directa:** los estudiantes deberán obtener al menos 6 puntos de promedio entre las 5 instancias de evaluación parcial y en cada una de ellas deberán obtener al menos 4 puntos. La quinta entrega consiste en integrar los diferentes módulos que componen el proyecto y su puesta en funcionamiento.

Las y los estudiantes que no alcancen los requisitos de la promoción directa y hayan aprobado al menos con 4 puntos las primeras 4 evaluaciones, 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 3, los cuales tienen roles específicos, en algunos casos 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 prácti-



ca 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 seguridad, responsabilidad profesional, riesgo laboral 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 qué 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 multidisciplinares, 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-disciplinares.

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 docen-

tes 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 25/8	<p>Unidad 1: La Plataforma JAVA</p> <p>Unidad 2: Clases, Espacios de Nombres, Especificadores de Acceso</p>	<p>Presentación de la materia, de los y las docentes y de la metodología de trabajo.</p> <p>Elección de grupos.</p> <p>Presentación de la plataforma JAVA y de la iniciativa Android.</p> <p>Presentación del tema espacios de nombres y especificadores de acceso en JAVA y su relación con el ocultamiento de información y la herencia.</p> <p><b>Práctica 1:</b> especificadores de acceso, constructores y clases abstractas.</p>
Clase 2	Miércoles 1/9	<p>Unidad 2: Interfaces y Clases Abstractas</p> <p>Unidad 4: Herencia y Polimorfismo en JAVA</p>	<p>Presentación y desarrollo de interfaces JAVA y su relación con la herencia.</p> <p>Presentación de herencia y polimorfismo en JAVA.</p> <p><b>Práctica 2:</b> Interfaces y Polimorfismo.</p> <p><b>TALLER 1: “Competencia de Bots” - Parte 1:</b> Se comienza a trabajar y se entrega el 21/9 a través de una tarea.</p> <p>Objetivo del taller: desarrollar una estrategia ganadora.</p>
Clase 3	Miércoles 8/9	Unidad 5: Clases Anidadas y Clases Internas	Presentación y desarrollo de clases internas, anidadas, su relación con el ocultamiento de

			<p>información y con la herencia.</p> <p>Competencia de robots en VIVO.</p> <p><b>TALLER 1: “Competencia de Bots continúa.....”- Parte 2.</b></p> <p><b>Coloquio sobre Taller 1</b></p>
Clase 4	Miércoles 15/9	Unidad 6: “Genéricos” El framework de Colecciones	<p>Presentación del framework de colecciones y de tipos genéricos</p> <p><b>Práctica 3:</b> Clases anidadas y clases internas. Clases anónimas..</p> <p><b>Práctica 4:</b> Colecciones y tipos genéricos</p> <p><b>Coloquio Taller 1-Parte 2</b></p>
<b>Clase 5</b>	<b>Miércoles 22/9</b>	<b>Primera evaluación parcial (individual)</b>	
Clase 6	Miércoles 29/9	Unidad 2: Tipos Enumerativos Unidad 11: Expresiones Lambda	<p>Presentación de tipos Enumerativos.</p> <p>Presentación de programación funcional en JAVA.</p> <p><b>TALLER 2: “Luis Alberto Spinetta”</b></p> <p><b>Práctica 5:</b> Tipos Enumerativos y expresiones Lambda.</p>
Clase 7	Miércoles 6/10	Unidad 8: Anotaciones	<p>Presentación y desarrollo de manejo de errores en Java: Excepciones.</p> <p>Presentación y desarrollo de Anotaciones &amp; Reflection.</p> <p>Presentación de acceso a BD en JAVA. Patrón DAO</p> <p><b>TALLER 3: Mapeador de BD</b></p> <p><b>Práctica 7:</b> Conceptos y uso de anotaciones. Definir anotaciones.</p>
Clase 8	Miércoles 13/10	Unidad 7: Manejo de errores mediante excepciones	<p><b>Entrega del TALLER 3, antes de la práctica.</b></p> <p><b>Coloquio del TALLER 3.</b></p> <p>Se convoca a los usuarios o interesados en el proyecto que serán los adoptantes del trabajo final.</p>

			<p>A partir de esta charla se solicitará un documento preliminar de análisis del problema planteado junto con historias de usuario que describen los requerimientos del sistema (miércoles 17/11)</p> <p><b>Práctica 6:</b> Conceptos y uso de Excepciones.</p>
Clase 9	Miércoles 20/10	Unidad 10: Concurrencia: multithreading en Java	<p>Presentación y desarrollo de aplicaciones concurrentes en JAVA.</p> <p>Explicación sobre programación de sockets en JAVA.</p> <p>El paquete java.net.</p> <p>Explicación de gitlab</p> <p><b>Práctica 8:</b> Threads. Ciclo de vida. Sincronización de threads. Ejecutores.</p>
<b>Clase 10</b>	<b>Miércoles 27/10</b>	<b>Segunda evaluación parcial (individual)</b>	
Clase 11	Miércoles 3/11	Unidad 12: Aplicaciones móviles en Android	<p>Presentación del desarrollo de aplicaciones móviles con tecnologías Android. La arquitectura de Android. Las componentes de las aplicaciones: activities e intents. El ciclo de vida de una aplicación Android.</p> <p>Explicación sobre Android Studio (entorno de desarrollo de Android)</p> <p><b>Práctica 9:</b> Ambiente de desarrollo Android. Aplicación, Activities, Views, resources y Layouts. Diseño declarativo de interfaz gráfica.</p>
Clase 12	Miércoles 10/11	Unidad 12: Aplicaciones móviles (segunda parte)	<p>Presentación de construcción de interfaces de usuario en Android.</p> <p><b>Puesta en común del trabajo final y publicación del enunciado.</b></p> <p>Se comienza a trabajar en el proyecto.</p>
Clase 13	Miércoles 17/11	<b>Tercera evaluación parcial (grupal)</b>	
		Unidad 12: Aplicaciones móviles (tercera parte)	<b>Tercera evaluación parcial:</b> Entrega del documento de análisis preliminar y de las historias de



			<p>usuario del proyecto final.</p> <p>Presentación de mapas en Android: la librería OpenStreetMap</p> <p>Explicación de Retrofit</p> <p>Consultas y guías para el desarrollo del trabajo final</p>
<b>Clase 14</b>	<b>Miércoles 24/11</b>	<b>Evaluación Flotante de la 1ra Evaluación Parcial Individual</b>	
			<p><b>Evaluación Flotante de la 1ra Evaluación Parcial:</b> Para los estudiantes que aún no aprobaron la primera evaluación parcial o que no alcanzaron la nota de promoción o deseen subir nota.</p> <p>Consultas y guías para el desarrollo de la prueba de concepto del trabajo final.</p>
<b>Clase 15</b>	<b>Miércoles 1/12</b>	<b>Evaluación Flotante de la 2da Evaluación Parcial Individual</b>	
			<p><b>Evaluación Flotante de la 2da Evaluación Parcial:</b> Para los estudiantes que aún no aprobaron la segunda evaluación parcial o que no alcanzaron la nota de promoción o deseen subir nota.</p> <p>Consultas y guías para el desarrollo de la prueba de concepto del trabajo final.</p>
<b>Miércoles 8/12 - Feriado Nacional: Día de la Inmaculada Concepción de María</b>			
<b>Clase 16</b>	<b>Miércoles 15/12</b>	<b>Cuarta evaluación parcial (grupal)</b>	
			<p><b>Cuarta evaluación parcial:</b> prueba de conceptos del proyecto final.</p> <p>Clases de consultas.</p>
<b>VACACIONES</b>			
<b>Clase 17</b>	<b>Miércoles 9/2</b>		<p>Clases de consultas.</p> <p>Explicación sobre la elaboración del informe.</p>
<b>Clase 18</b>	<b>Miércoles 16/2</b>	<b>Recuperatorio Flotante</b>	
		<b>Cuarta evaluación parcial (grupal)</b>	
			<p><b>Recuperatorio Flotante:</b> para lxs estudiantes que adeudan la aprobación de la 1ra y/o 2da</p>

			instancia de evaluación parcial.  <b>Cuarta evaluación parcial:</b> prueba de conceptos del proyecto final (grupos que no hayan entregado en diciembre).
<b>Clase 19</b>	<b>Miércoles 23/2</b>	<b>Quinta evaluación parcial (grupal)</b>	
			<b>Quinta evaluación parcial:</b> <b>Entrega Final del Proyecto</b>
<b>Clase 20</b>	<b>Miércoles 2/3</b>	<b>Quinta evaluación parcial (grupal)</b>	
			<b>Quinta evaluación parcial:</b> <b>Entrega Final del Proyecto</b>

Evaluaciones previstas	Fecha
<b>Primera evaluación parcial:</b>  Se evalúan las unidades: 2 (tipos enumerativos), 4 (herencia y polimorfismo en JAVA), 5 (clases anidadas y clases internas)	22/9
<b>Segunda evaluación parcial:</b>  Se evalúan las unidades: 6 (colecciones y genéricos), 7 (excepciones), 8 (anotaciones), 10 (threads) y 11 (Lambda & Streams).	27/10
<b>Recuperatorio - 1ra Evaluación Parcial:</b>  Para lxs estudiantes que aún no aprobaron la primera evaluación parcial o que no alcanzaron la nota de promoción o desean subir nota.	24/11
<b>Recuperatorio- 2da Evaluación Parcial:</b>  Para lxs estudiantes que aún no aprobaron la primera evaluación parcial o que no alcanzaron la nota de promoción o desean subir nota.	1/12
<b>Tercera evaluación parcial:</b>  Entrega del documento de análisis preliminar y de las historias de usuario del proyecto final.	17/11

<b>Cuarta evaluación parcial:</b>	15/12/2021
Diseño, PoC del trabajo final. Se desdobra en 2 fechas: 15/12 y 16/2/2022	16/2/2022
<b>Evaluación Flotante:</b>	16/2/2022
Para lxs estudiantes que adeudan la aprobación de la 1ra y/o 2da instancia de evaluación parcial.	
<b>Quinta evaluación: Entrega Final del Proyecto</b>	23/2/2022
Se desdobra en 2 fechas: 23/2 y 2/3, atendiendo la situación de lxs estudiantes que rindieron en la fecha flotante.	2/3/2022

**Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):**

Profesores: Claudia Queiruga ([claudiaq@info.unlp.edu.ar](mailto:claudiaq@info.unlp.edu.ar)) y Pablo Iuliano ([piuliano@info.unlp.edu.ar](mailto:piuliano@info.unlp.edu.ar))

Jefes de Trabajos Prácticos: Diego Bellante ([diegobellante@gmail.com](mailto:diegobellante@gmail.com)) e Isabel Kimura ([ikimura@linti.unlp.edu.ar](mailto:ikimura@linti.unlp.edu.ar))

Plataforma virtual: <https://catedras.info.unlp.edu.ar/> (categoría "Cursos 2021")

Firma del/los profesor/es



**Claudia Queiruga**