LABORATORIO DE SOFTWARE

Año 2023

Carrera/ Plan:

Licenciatura en Informática Planes 2021/2015/2012
Licenciatura en Sistemas Planes 2021/2015/2012

Año: *4°(Lic en Informática)*
4º y 5º (Lic en Sistemas)
Régimen de Cursada: *Semestral*
Carácter:
Obligatoria (Lic en Informática)
Optativa Área Ing. de Software y Base de Datos (Lic en Sistemas)
Correlativas: *Proyecto de Software*
Profesor/es: *Claudia Queiruga y Pablo Iuliano*
Hs. semanales teoría: 2 hs.
Hs. semanales teoría: 4 hs.

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 desarrollo de software.

Laboratorio de Software provee a los estudiantes de 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 y KOTLIN. 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, base de datos y redes. Mediante este trabajo el estudiante se enfrenta a problemas reales y a la utilización de tecnologías de desarrollo de software actuales. Pondrán en práctica plataformas de despliegue y ejecución de aplicaciones en contenedores, al estilo Docker.

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 desarrollo de software como Java y Aplicaciones Avanzadas sobre Internet, Diseño de Experiencia de Usuario. 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 interfaces 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 perfomance 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: Las Plataformas JAVA y KOTLIN

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.
- Vincular los principios multiplataforma de JAVA y KOTLIN.

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.

La multiplataforma KOTLIN: características y su interoperabilidad con JAVA.

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

Objetivos de aprendizaje:

- Relacionar conceptos de orientación a objetos, su implementación en JAVA y KOTLIN.
- Definir tipos de datos usando interfaces y tipos enumerativos.
- Usar el formato de empaquetado y compresión de archivos disponible para distribución de aplicaciones.
- Usar la documentación en línea de la API JAVA y KOTLIN.

Definición de clases. Miembros de una clase: atributos y métodos. Miembros inmutables y mutables. Creación e inicialización de objetos. Constructores en JAVA y KOTLIN. Las palabras claves this y this(). Especialización de clases de acuerdo a los enfoques de JAVA y KOTLIN Especificadores de acceso disponibles en JAVA y KOTLIN. 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 y Kotlin: ¿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.

Unidad 4: Herencia y Polimorfismo

Objetivos de aprendizaje:

- Profundizar en el concepto de herencia simple, su implementación en JAVA y KOTLIN.
- Definir interfaces 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.
- 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 de JAVA y la clase Any de KOTLIN: 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 en JAVA. Los atributos open y de override de KOTLIN

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 de JAVA y KOTLIN.
- Usar colecciones de tipos genéricas e inferencias de tipos.

Tipos Genéricos y Parametrizados. Comodines. Métodos genéricos. Arquitectura del framework de colecciones en JAVA y KOTLIN. Interfaces core en JAVA: Collections, Set, List, SortedSet, Map, SortedMap. Interfaces inmutables y mutables en KOTLIN.

Interfaces para ordenación de objetos: Comparable y Comparator.

Interfaces para iterar: Iterator, ListIterator.

Implementaciones en JAVA: 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 enfoque del tratamiento de excepciones de JAVA y KOTLIN.
- 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 en los programas 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 y KOTLIN en forma independiente del motor de base de datos utilizado.
- Escribir aplicaciones 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: Expresiones Lambda

Objetivos de aprendizaje:

- Introducirse en la programación funcional en JAVA y KOTLIN: expresiones Lambda.
- Identificar cuándo usar Lambda y cuándo no.

De las clases anónimas en JAVA 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 Kotlin como tecnología de desarrollo en 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. Componente ViewModel y RecyclerView.

Uso de Corrutinas.

Persistencia de datos de la aplicación: tecnología Room.

Preferencias de usuario.

Presentación de mapas en Android: tecnología OpenStreetMap.

Peticiones HTTP en contexto de interacciones con servicios RESTful en Android: librería Retrofit.

Nociones de desarrollo seguro enfocado en el desarrollo de aplicaciones móviles.

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

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.

Atomic Kotlin, Mindview LLC, 2021. Eckel, Bruce y Isakova, Svetlana.

Kotlin Language Documentation 1.8.0, disponible en https://kotlinlang.org/docs/kotlin-pdf.html. Último acceso: 15/2/2023

Sitio oficial de Android para desarrolladores Kotlin: https://developer.android.com/kotlin

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

Sitio oficial de Kotlin: https://kotlinlang.org/

Sitio oficial de OpenJDK: https://openjdk.java.net/

METODOLOGÍA DE ENSEÑANZA

Los contenidos de la asignatura se encuentran articulados y están organizados en actividades teóricas y prácticas, semanales. 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**.

Para el desarrollo de las clases teóricas se utiliza una PC y un cañón óptico dispuesto en el aula, asimismo se dispone de acceso a Internet, posibilitando mostrar on-line, durante las clases, ejemplos que pueden aplicarse a las clases prácticas.

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. Estas actividades se desarrollan en una de las salas de PC de la Facultad, donde se dispone de computadoras con acceso a Internet y doble booteo.

Se utilizará 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. Los materiales con los que se trabaja serán puestos a disposición a través de dicha plataforma y se utilizará la facilidad de tareas programadas para completar las actividades entregables y las evaluaciones.

Sobre los materiales: 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, tecnologías Docker, ubicando a las y los estudiantes en un ambiente profesional actual, intentando favorecer la consolidación de la formación experimental del estudiante. Estos materiales son puestos a disposición por la asignatura y con el apoyo del área de soporte técnico de la facultad.

Sobre el proyecto final integrador: 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 por las y los estudiantes, tanto las individuales como las 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, intentando articular con proyectos y actividades de Extensión.

Se **utilizará en repositorio de software GitLab** disponible en el LINTI para los entregables del proyecto final (https://gitlab.linti.unlp.edu.ar).

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 y la presentación del mismo.

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 el caso de las evaluaciones individuales, al iniciar la evaluación se explican las consignas del trabajo y qué se espera que desarrollen las y los estudiantes. 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” y durante las mismas, después de la entrega, se realizará un breve coloquio sobre la entrega desarrollada, con la intención que las y los estudiantes puedan explicar la solución planteada y las decisiones tomadas.

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, que fueron desarrollados durante los entregables, su puesta en funcionamiento y la presentación del mismo. Para ello se guiará a los grupos en la elaboración de una presentación en soporte digital que los ayude en la presentación del mismo.

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.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos	Actividades

Clase 1	Miércoles 23/8	Unidad 1: Las Plataformas JAVA y KOTLIN Unidad 2: Clases, Espacios de Nombres, Especificadores de Acceso	Presentación de la materia, de los y las docentes y de la metodología de trabajo. Elección de grupos. Presentación de las plataformas JAVA, KOTLIN y de la iniciativa Android. Presentación del tema espacios de nombres y especificadores de acceso y su relación con el ocultamiento de información y la herencia. **Práctica 1:** especificadores de acceso, constructores y clases abstractas.
Clase 2	Miércoles 30/8	Unidad 2: Interfaces y Clases Abstractas Unidad 4: Herencia y Polimorfismo	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 1: "Competencia de Bots"** - **Parte 1:** Se comienza a trabajar y se entrega el 4/9 a través de una tarea. Objetivo del taller: desarrollar una estrategia ganadora.
Clase 3	Miércoles 6/9	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. Competencia de robots en VIVO. **TALLER 1: "Competencia de Bots continúa….."- Parte 2.** **Coloquio sobre Taller 1**

Clase 4	Miércoles 13/9	Unidad 2: Tipos Enumerativos Unidad 6: “Genéricos” Se retoman las unidades 2, 3 y 4 y su implementación en KOTLIN.	Presentación del framework de colecciones y de tipos genéricos **Práctica 3:** Clases anidadas y clases internas. Clases anónimas. Tipos enumerativos **Taller de KOTLIN**, que retome los temas de java dados hasta el momento: constructores, tipos anulables y no-anulables, clases y herencia, clases abstractas e interfaces y especificadores de acceso. **Coloquio Taller 1-Parte 2 -**
Clase 5	**Miércoles 20/9**	**Primera evaluación parcial (individual)**	
Clase 6	Miércoles 27/9	Unidad 6: “Genéricos” (continuación) El framework de Colecciones Unidad 11: Expresiones Lambda	Presentación de tipos Enumerativos y expresiones Lambda. **Práctica 4:** Colecciones, tipos genéricos y expresiones Lambda.
Clase 7	Miércoles 4/10	Unidad 8: Anotaciones	Presentación y desarrollo de Anotaciones & Reflection. Charla sobre desarrollo seguro en aplicaciones móviles. **TALLER 3: Integración Kotlin-Java** **Práctica 5:** Conceptos y uso de anotaciones. Definir anotaciones. Publicación de material sobre java.net. Publicación de material de gitlab
Clase 8	Miércoles 11/10	Unidad 7: Manejo de errores mediante excepciones	Presentación y desarrollo de manejo de errores en Java: Excepciones. **Coloquio del TALLER 3.** Visita de la organización social o institución adoptante del **trabajo final integrador** con el objetivo de explicar la situación-problema que se desarrollará. Se entrega un documento guía de análisis y diseño.

			Práctica 6: Conceptos y uso de Excepciones.
Clase 9	Miércoles 18/10	Unidad 10: Concurrencia: multithreading en Java Se retoman las unidades 6, 7 y 11 y su implementación en KOTLIN.	Presentación y desarrollo de aplicaciones concurrentes en JAVA. **Práctica 7:** Threads. Ciclo de vida. Sincronización de threads. Ejecutores. Consultas sobre la segunda evaluación. Consultas sobre el diseño y análisis del proyecto final. **Taller de KOTLIN,** que retome los temas de java dados hasta el momento: funciones, colecciones, expresiones lambdas y excepciones.
Clase 10	Miércoles 25/10	Unidad 12: Aplicaciones móviles en Android	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. Explicación sobre Android Studio (entorno de desarrollo de Android) Consultas sobre la segunda evaluación.
Clase 11	**Miércoles 1/11**	**Segunda evaluación parcial (individual)**	
Clase 12	**Miércoles 8/11**	**Tercera evaluación parcial (grupal)** Unidad 12: Aplicaciones móviles (segunda parte)	Presentación de construcción de interfaces de usuario en Android. Explicación de Retrofit Entrega del documento de análisis preliminar y de las historias de usuario del proyecto final. **Práctica 8:** Ambiente de desarrollo Android. Aplicación, Activities, Views, resources y Layouts. Diseño declarativo de interfaz gráfica.
Clase 13	Miércoles 15/11	Unidad 12: Aplicaciones móviles (tercera parte)	Presentación de mapas en Android: la librería OpenStreetMap Explicación de Docker: ambiente de desarrollo. Se comienza a trabajar en el proyecto.

			Consultas y guías para el desarrollo del trabajo final
Clase 14	**Miércoles 22/11**	**Evaluación Flotante de la 1ra Evaluación Parcial Individual**	**Evaluación Flotante de la 1ra Evaluación Parcial:** Para los estudiantes que aún no aprobaron la primera evaluación parcial o que no alcanzaron la nota de promoción o deseen subir nota.
			Consultas y guías para el desarrollo de la prueba de concepto del trabajo final.
Clase 15	**Miércoles 29/11**	**Evaluación Flotante de la 2da Evaluación Parcial Individual**	**Evaluación Flotante de la 2da Evaluación Parcial:** Para los estudiantes que aún no aprobaron la segunda evaluación parcial o que no alcanzaron la nota de promoción o deseen subir nota.
			Consultas y guías para el desarrollo de la prueba de concepto del trabajo final.
Clase 16	Miércoles 6/12		Consultas para el desarrollo de la prueba de concepto del trabajo final.
Clase 17	**Miércoles 13/12**	**Cuarta evaluación parcial (grupal)**	
			Cuarta evaluación parcial: prueba de conceptos del proyecto final.
			Clases de consultas.
VACACIONES			
Clase 18	Miércoles 7/2		Clases de consultas. Explicación sobre la elaboración del informe.
Clase 19	**Miércoles 14/2**	**Recuperatorio Flotante**	
		Cuarta evaluación parcial (grupal)	
			Recuperatorio Flotante: para lxs estudiantes que adeudan la aprobación de la 1ra y/o 2da instancia de evaluación parcial. **Cuarta evaluación parcial:** prueba de conceptos del proyecto final (grupos que no hayan entregado en diciembre).
Clase 20	**Miércoles 21/2**	**Quinta evaluación parcial (grupal)**	
			Quinta evaluación parcial: Entrega Final del Proyecto

Clase 21	**Miércoles 28/2**	**Quinta evaluación parcial (grupal)**	
			Quinta evaluación parcial: Entrega Final del Proyecto

Evaluaciones previstas	Fecha
Primera evaluación parcial - individual Se evalúan las unidades: 2 (tipos enumerativos), 4 (herencia y polimorfismo en JAVA), 5 (clases anidadas y clases internas)	20/9
Segunda evaluación parcial - individual Se evalúan las unidades: 6 (colecciones y genéricos), 7 (excepciones), 8 (anotaciones), 10 (threads) y 11 (Lambda & Streams).	1/11
Tercera evaluación parcial - grupal Entrega del documento de análisis preliminar y de las historias de usuario del proyecto final.	8/11
Recuperatorio - 1ra Evaluación Parcial Para lxs estudiantes que aún no aprobaron la primera evaluación parcial o que no alcanzaron la nota de promoción o desean subir nota.	22/11
Recuperatorio- 2da Evaluación Parcial Para lxs estudiantes que aún no aprobaron la primera evaluación parcial o que no alcanzaron la nota de promoción o deseen subir nota.	29/11
Cuarta evaluación parcial - grupal Diseño, PoC del trabajo final. Se desdobla en 2 fechas: 13/12 y 14/2	13/12 14/2/2024
Recuperatorio Flotante Para lxs estudiantes que adeudan la aprobación de la 1ra y/o 2da instancia de evaluación parcial.	14/2/2024
Quinta evaluación: Entrega Final del Proyecto Se desdobla en 2 fechas: 21/2 y 28/2, atendiendo la situación de lxs estudiantes que rindieron en la fecha flotante.	21/2/2024 28/2/2024

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

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

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

Plataforma virtual: https://catedras.info.unlp.edu.ar/ (categoría “Cursos 2023”)

Firma del/los profesor/es

Claudia Queiruga

Pablo Iuliano