

**JAVA Y APLICACIONES SOBRE
INTERNET****Carrera/ Plan:***Ingeniería en Computación Plan 2008 / Plan 2011 / Plan 2024***Año 2026****Año:** 4to/5to**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Optativa**Correlativas:** Taller de Lenguajes II - Ingeniería de Software**Profesor/es:** Claudia Queiruga y Jorge Rosso**Hs. semanales teoría:** 2 hs.**Hs. semanales teoría:** 4 hs.**FUNDAMENTACIÓN**

Java y Aplicaciones Avanzadas sobre Internet es una materia optativa de cuarto y quinto año de las carreras Licenciatura en Informática y Licenciatura en Sistemas.

Las y los estudiantes que cursan **Java y Aplicaciones Avanzadas sobre Internet** manejan el lenguaje JAVA y están entrenados en el uso de algún IDE (Integrated Development Environment) relacionado a tecnologías JAVA y en herramientas de desarrollo.

Los desarrolladores de hoy reconocen cada vez más la necesidad de aplicaciones distribuidas, transaccionales y portables que aprovechan la velocidad, seguridad y fiabilidad de la tecnología del lado del servidor. **Java y Aplicaciones Avanzadas sobre Internet** le provee al estudiante de los fundamentos para la construcción de aplicaciones de gran envergadura, server-side, en tecnologías JAVA, caracterizadas por ser distribuidas, multiusuario, de alta disponibilidad, de acceso seguro, portables, escalables e interoperables.

Asimismo la tecnología elegida en esta asignatura le proporciona al estudiante de habilidades para manejar un potente conjunto de APIs, que a la vez le permiten acortar el tiempo de desarrollo, reducir la complejidad de la aplicación y mejorar el rendimiento de las mismas.

Las habilidades que el estudiante adquiere en esta asignatura le permitirán programar y desplegar aplicaciones web server-side programadas en JAVA, usando las componentes estándares que provee la especificación JEE (Java Enterprise Edition) en las diferentes capas físicas del modelo multi-capas y también usando frameworks de nivel aplicación, que facilitan y mejoran la calidad de las aplicaciones desarrolladas y desplegadas. Pondrán en práctica plataformas de despliegue y ejecución de aplicaciones en contenedores, al estilo Docker.

OBJETIVOS GENERALES

Construir aplicaciones empresariales server-side distribuidas, multiusuario, de alta disponibilidad, de acceso seguro, escalable e interoperable usando tecnologías JAVA server-side. Consolidar la formación experimental del estudiante.

COMPETENCIAS

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-CE5- Planificar, dirigir, realizar y/o evaluar proyectos de sistemas de software de base: Sistemas Operativos, Sistemas Operativos Distribuidos, Sistemas Operativos Dedicados. Especificación, diseño, implementación, prueba, verificación, validación, mantenimiento y control de eficiencia de los sistemas de administración de recursos que se implanten como software de base de datos sobre sistemas de procesamiento de datos, incluyendo la virtualización de recursos físicos y lógicos.

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-CE8- Planificar, dirigir, realizar y/o evaluar proyectos de sistemas de administración de recursos. Especificación formal de los mismos, diseño, implementación, prueba, verificación, validación, mantenimiento y control de eficiencia/calidad de los sistemas de administración de recursos que se implanten como software sobre sistemas de procesamiento de datos.

CONTENIDOS MÍNIMOS

La asignatura tiene un enfoque de una temática experimental sobre tecnologías de desarrollo para aplicaciones de gran envergadura.

PROGRAMA ANALÍTICO

Unidad 1: Introducción a JEE

Objetivos de aprendizaje:

- Aplicar el modelo de componentes multi-tier (múltiples capas físicas) promovido por JEE en el diseño de aplicaciones.
- Identificar las funcionalidades que provee cada componente y cómo interactúan entre ellas.

Contenido:

Las componentes del modelo JEE: componentes clientes, componentes web y componentes empresariales.

La componente fundacional del módulo web de JEE: Servlets.

Las componentes empresariales: EJB (Enterprise Java Beans) de Sesión, de Entidad y Orientados a Mensajes.

Interacción entre las diferentes componentes JEE.

Los contenedores JEE. Servicios provistos implícitamente a las componentes que gerencia.

Los archivos descriptores de la aplicación JEE: ejb-jar.xml, web.xml, application-client.xml.

El modelo de despliegue de una aplicación JEE.

Unidad 2: Servlets HTTP

Objetivos de aprendizaje:

- Programar servlets HTTP.
- Relacionar los métodos del protocolo HTTP con las peticiones a un servlet.
- Analizar el funcionamiento de un servlet, su interacción con el cliente web y el manteniendo de sesiones con diferentes clientes web.
- Desplegar servlets en servidores de testeo y analizar el archivo descriptor de la aplicación web, web.xml.

Contenido:

El ciclo de vida de un Servlet: inicialización, pedido de servicio y destrucción.

La API (Application Programming Interface) de Servlets: la interface javax.servlet.Servlet. La anatomía de un Servlet.

La clase GenericServlet y la clase HttpServlet.

La interface HttpServletRequest y la interface HttpServletResponse.

Soporte de Sesiones con Servlets: la interface HttpSession.

Unidad 3: Servlet Listener y Servlet Filtros

Objetivos de aprendizaje:

- Programar listener de contexto y de sesión.
- Analizar los beneficios que aportan las componentes listeners respecto del código producido.
- Programar requerimientos no funcionales con servlets filtros: seguridad, auditoría, logging, compresión de la respuesta, etc.

Contenido:

El contexto de servlets: la interface ServletContext.

Componentes web que atienden los eventos del ciclo de vida de la aplicación, de las sesiones y de las peticiones: ServletListeners. Las interfaces listeners.

Relación entre el ServletContext y los listeners del ciclo de vida de la aplicación.

Componentes web que permiten realizar pre-procesamiento y post-procesamiento de peticiones HTTP: Servlets Filtros. Cómo usar servlets filtros para implementar requerimientos no-funcionales.

Unidad 4: Conexión a Bases de Datos

Objetivos de aprendizaje:

- Programar componentes que accedan a una base de datos.
- Analizar los beneficios de establecer conexiones usando DataSource.

Contenido:

La API JDBC (Java DataBase Connectivity). Tipos de Drivers JDBC

Establecimiento de una Conexión: la clase DriverManager y la interface DataSource. Ejecución de Sentencias SQL. Las clases Statement, PreparedStatement y CallableStatement. Recuperación de resultados: la interface ResultSet.

Soporte de Transacciones. Manejo de excepciones.

JPA (Java Persistente API). Introducción. Beans de Identidad. Anotaciones. Anotaciones del ciclo de vida.

Unidad 5: Arquitectura de Aplicaciones Web: REST y SPA

Objetivos de aprendizaje:

- Introducir los conceptos del desarrollo de aplicaciones basadas en servicios REST y clientes web SPA (Single Page Application).
- Desarrollar aplicaciones bajo los lineamientos de REST y SPA.

Contenido:

Conceptos básicos de la Arquitectura REST. Web Services del tipo RESTful.

Desarrollar APIs utilizando la API de Java, JAX-RS (JSR 311 - Java API para RESTful Web Services).

Técnica de Inyección de Dependencias (JSR 330 - Dependency Injection for Java) implementada por HK2

El framework SPA: Angular.

Tecnologías de Seguridad de aplicaciones Angular-REST: CORS y JWT.

Unidad 6: Integración y Despliegue continuo de Aplicaciones

Objetivos de aprendizaje:

- Introducir el concepto de contenedores Docker y su utilización para el despliegue de aplicaciones.
- Introducir los conceptos de CI/CD: integración continua y entrega continua.
- Desarrollar aplicaciones utilizando imágenes Docker y CI/CD

Conceptos claves de Docker: contenedores, imágenes, dockerfile y registro de docker.

Volúmenes y redes de Docker. Docker compose.

Cómo construir, probar y desplegar aplicaciones de manera rápida y eficiente utilizando Gitlab y Docker.

BIBLIOGRAFÍA

Servlets and JavaServer Pages: The J2EE Technology Web Tier. Jayson Falkner and Kevin Jones.

Addison-Wesley Professional, 2003. Disponible para su descarga.

Professional JSP, 2nd Edition. Simon Brown, Robert Burdick, Jayson Falkner, Ben Galbraith, Rod Johnson, Larry Kim, Casey Kochmer, Thor Kristmundsson, Sing Li. Editorial Apress, 2001. ISBN: 1861004958

Java Platform, Enterprise Edition: The Java EE Tutorial. Release 8. Disponible en <https://javaee.github.io/tutorial/>

Expert one-on-one J2EE design and development. Johnson, Rod. Editorial Wrox Press, 2003. ISBN: 0764543857.

Java Persistence with Hibernate. Christian Bauer, Gavin King. Editorial Manning Publications, 2007. ISBN: 1932394885

RESTful Java with JAX-RS 2.0, 2nd Edition. Designing and Developing Distributed Web Services. Editorial O'Reilly Media, 2013. ISBN: 978-1-44936-134-1.

RESTful web APIs, 1st Edition. Leonard Richardson, Sam Ruby, Mike Amundsen. Editorial O'Reilly Media, 2013. ISBN: 9781449358068.

Microservicios: un enfoque integrado. Ra-Ma, 2018. Roldán Martínez, David; Valderas Aranda, Pedro J y Torres Bosch, Victoria.

Sitio oficial de Angular: <https://angular.io>

JSON Web Token (JWT) - RCF 7519: <https://tools.ietf.org/html/rfc7519>.

JWT.IO: <https://jwt.io>

Jersey RESTful Web Services: <https://eclipse-ee4j.github.io/jersey/>

METODOLOGÍA DE ENSEÑANZA

La asignatura consolida la formación experimental y profesional del estudiante ubicándolo en un entorno de trabajo similar al real y cotidiano. Los contenidos de la asignatura se encuentran articulados y se organizan en instancias prácticas y teóricas semanales.

El desarrollo conceptual consiste en explicaciones orales que luego son retomadas para la resolución de los trabajos prácticos y del trabajo final integrador.

Las actividades prácticas se trabajan en grupos de estudiantes, se utilizan herramientas específicas de soporte y desarrollo típicas en la comunidad de software libre, entre ellas, servidor web, motor de base de datos, máquinas virtuales, IDEs para desarrollo, repositorios de software, tecnologías Docker, de licencias libres. Se permite el uso de herramientas de inteligencia artificial como apoyo al proceso de aprendizaje para la generación de código, el diseño y el maquetado, entre otras actividades. El equipo docente lleva a cabo instancias de revisión de código y coloquios en las que las y los estudiantes deben fundamentar las decisiones adoptadas y acreditar la comprensión del código desarrollado.

Se prevén instancias de resolución de trabajos prácticos en clase, de carácter individual y grupal, permitiendo el seguimiento del proceso formativo. Durante la cursada las y los estudiantes desarrollan un **proyecto final integrador**, organizado en diferentes entregables.

En síntesis, **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,
- muestras y coloquios de las producciones para el seguimiento del proceso formativo,
- el desarrollo de un proyecto de software final integrador, organizado en entregables.

Las y los estudiantes desarrollan el proyecto final en equipos de 2 estudiantes, en algunos casos se admiten grupos de 3. A cada grupo se le asigna 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 integrador se articula con la Secretaría de Extensión de la Facultad y tiene por objetivo dar respuesta a problemáticas de organizaciones de la sociedad civil y/o instituciones de gestión pública con las que se trabaja desde dicha Secretaría y se intentará articular con proyectos o equipos de extensión.

Se utiliza el EVEA (Entorno Virtual de Enseñanza y Aprendizaje) del LINTI <http://catedras.linti.unlp.edu.ar> como apoyo adicional a las actividades presenciales y como herramienta fundamental para la modalidad de cursada semipresencial. Los materiales con los que se trabaja son puestos a disposición a través de dicha plataforma y se utiliza la facilidad de tareas programadas para completar las actividades entregables y las evaluaciones.

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

Para el desarrollo de las clases teóricas, en esta modalidad, se utiliza una PC y un cañón óptico dispuestos 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. Estas explicaciones y guías se ponen a disposición de las y los estudiantes mediante su publicación en el EVEA.

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

EVALUACIÓN

Para la aprobación de la asignatura las y los estudiantes deben desarrollar un proyecto final integrador a lo largo de la cursada, organizado en 6 etapas entregables. Las mismas son de seguimiento y evaluación

con calificación. El sexto entregable consiste en la entrega del proyecto final integrador completo con toda la funcionalidad implementada y en funcionamiento en el servidor de prueba provisto por la cátedra.

La aprobación de la asignatura es mediante promoción directa: se deberá obtener al menos 6 puntos de promedio entre las 6 entregas y no menos de 4 puntos en cada entrega individual. Durante estas instancias de entregables se realiza un coloquio sobre la entrega desarrollada, con la intención que las y los estudiantes puedan explicar la solución planteada y las decisiones tomadas. La sexta entrega consiste en integrar los diferentes módulos que componen el proyecto final integrador, que fueron desarrollados durante los entregables, su puesta en funcionamiento en un servidor de prueba provisto por la cátedra y la presentación del mismo en modalidad exposición. La evaluación de cada entrega se realiza mediante rúbricas de desempeño previamente definidas y comunicadas a los estudiantes, que explicitan criterios y niveles de logro esperados. El uso de rúbricas tiene un propósito formativo: orientar el trabajo de los estudiantes, hacer explícitas las expectativas, favorecer la autoevaluación y brindar una retroalimentación cualitativa para la mejora progresiva de las producciones. Las devoluciones de cada instancia de evaluación además de fundamentar la calificación obtenida, también señalan fortalezas, aspectos a revisar y sugerencias de mejora, promoviendo la reflexión sobre el propio proceso de aprendizaje y de desarrollo del proyecto.

Para exposición del proyecto final integrador (sexta entrega) se guía a los grupos en la elaboración de una presentación en soporte digital que los ayude en la presentación del trabajo realizado en el proyecto. Participan en dicha presentación, además de los profesores, referentes de las organizaciones y/o instituciones adoptantes del trabajo final integrador con el objetivo de trabajar desde el espacio de cátedra con problemas auténticos de organizaciones o instituciones con las que se articula desde la Secretaría de Extensión.

Los estudiantes que no alcancen los requisitos de la promoción directa y hayan obtenido al menos 4 puntos en cada una de las primeras 5 entregas, aprueban la cursada. En estos casos para aprobar la asignatura deben rendir un examen final escrito en alguna de las mesas de examen contempladas en el calendario académico.

Las evaluaciones se realizan en la sala de PC en la que se desarrollan habitualmente los trabajos prácticos semanales, 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.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos	Actividades
Clase 1	Miércoles 11/3	Arquitectura JEE Desarrollo de Servlets HTTP	Presentación de la materia.
			Presentación de la arquitectura JEE y sus componentes. La componente fundacional: Servlets.
			Práctica 1: Servlets http.
			Presentación de las herramientas de desarrollo (IDE, github, servidor Tomcat, servidor de BD)
Clase 2	Miércoles 18/3	Soporte de Sesiones HTTP	Presentación de Sesiones HTTP en la API de Servlets.
			Práctica 2: Sesiones HTTP en servlets.
Clase 3	Miércoles 25/3	Servlet Listeners	Presentación de objetos listeners y su aplicación en desarrollo de aplicaciones web. Problemas de concurrencia en aplicaciones web java. Delegación de peticiones HTTP y redirecciones respuestas HTTP.
			Práctica 3: Desarrollo de servlets listeners, alcances y redireccionamiento HTTP.
Clase 4	Miércoles 1/4	Persistencia en Aplicaciones JAVA server-side. Persistencia de objetos mediante un ORM: JPA & Hibernate Integración con Maven	Presentación de acceso a bases de datos a través de JDBC (Java DataBase Connectivity) y DataSource. Patrón DAOs (Data Access Objects). Presentación de JPA (Java Persistence API). Hibernate.
			Presentación del proyecto final integrador. Etapas del proyecto final integrador: publicación de un documento modelo que sirve de guía para describir el análisis y diseño (historias de usuario y maquetado).
			Práctica 4: Persistencia usando JPA.
Clase 5	Miércoles 8/4	Servlet Filter	Presentación de objetos filtros y su aplicación a requerimientos no funcionales.

			Práctica 5: Servlets Filtros, JDBC, DataSource y Patrón DAO.
			Consultas etapa 1 del proyecto. Publicación de la rúbrica de evaluación.
Clase 6	Miércoles 15/4	Primera instancia de evaluación	
		Consultas y entrega del prototipo del proyecto final	
		Entrega y presentación de la etapa 1 del proyecto final: “Análisis, diseño y maquetado del sistema”	
Clase 7	Miércoles 22/4	Publicación del alcance del proyecto final. Se solicitará a cada grupo que adecúe el documento de análisis y diseño a lo establecido y que lo re-entregue como parte de la segunda instancia de evaluación.	
		Publicación del alcance de la etapa 2 del trabajo final: “Definición de objetos del modelo” y de la rúbrica de evaluación.	
Clase 8	Miércoles 29/4	Segunda instancia de evaluación	
		Webservices tipo RESTful	Consultas sobre la etapa 2 del proyecto final.
			Entrega y presentación de la etapa 2 del trabajo final “Definición de los objetos del modelo” y de las adecuaciones del documento de análisis y diseño.
			Presentación de Arquitectura REST, Web Services de tipo RESTful.
Clase 9	Miércoles 6/5	Integración y despliegue continuo aplicaciones: explicación sobre despliegue de aplicaciones usando Docker.	
		Explicación y publicación de los alcances de la etapa 3 del trabajo final: “Desarrollo de la capa de persistencia”. Publicación de la rúbrica de evaluación.	
Clase 10	Miércoles 13/5	Se trabaja en la etapa 3 del proyecto final “Desarrollo de la capa de persistencia”.	
Clase 11	Miércoles 20/5	Tercera instancia de evaluación	
		Entrega y presentación de la etapa 3 del trabajo final: “Desarrollo de la capa de persistencia”	
		Práctica 6: Desarrollo de APIs utilizando la API de Java JAX-RS (JSR 311 - Java API para RESTful Web Services).	

Clase 12	Miércoles 27/5	Aplicaciones SPA: Angular - Parte 1	Presentación del framework Angular. Elementos principales de un proyecto Angular. Binding, formularios y ruteo.
			Explicación sobre los alcances de la etapa 4 del trabajo final: “Desarrollo de una API REST”. Publicación de la rúbrica de evaluación.
			Consultas de la etapa 4 del trabajo final “Desarrollo de una API REST”
		Inyección de dependencias	Se presenta la Inyección de dependencias y su implementación con HK2
Clase 13	Miércoles 3/6	Aplicaciones SPA: Angular - Parte 2	Presentación de ruteo avanzado de Angular, servicios e inyección de dependencias y comunicaciones con servidor remoto utilizando HTTP.
			Consultas de la etapa 4 del trabajo final “Desarrollo de una API REST”
			Explicación de Swagger
Clase 14	Miércoles 10/6	Aplicaciones SPA: Angular - Parte 3	Cuarta instancia de evaluación
			Entrega y presentación de la etapa 4 del trabajo final: “Desarrollo de una API REST”
			Presentación de la comunicación entre componentes Angular. Ruteo avanzado. Configuración de ambientes para conectar la aplicación Angular con la API REST. Uso de interceptores.
			Prácticas 7 y 8: Angular
			Explicación y publicación de los alcances de la etapa 5 del trabajo final: “Desarrollo de la vista del proyecto usando Angular e integración con la capa de servicios”. Publicación de la rúbrica de evaluación.
			Consultas de la etapa 5 del trabajo final: “Desarrollo de la vista del proyecto usando Angular e integración con la capa de servicios”
Clase 15	Miércoles 17/6	Consultas de la etapa 5 del trabajo final: “Desarrollo de la vista del proyecto usando Angular e integración con la capa de servicios”	

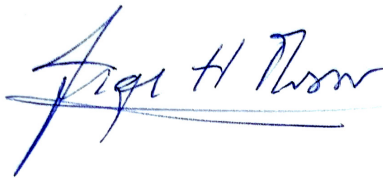
Clase 16	Miércoles 24/6	Quinta instancia de evaluación
		Explicación de la modalidad de presentación del proyecto final.
		Entrega etapa 5, con capa de servicios y conexión a servicios REST.
Clase 17	Miércoles 1/7	Consultas
Clase 18	Miércoles 8/7	Sexta instancia de evaluación (primera fecha) En caso de ser necesario en esta fecha se podrá hacer la re-entrega Etapa 5 (en caso de ser necesario)
Clase 19	Miércoles 15/7	Sexta instancia de evaluación (segunda fecha)

Cronograma de evaluaciones

Evaluaciones previstas	Fecha
Entrega etapa 1 del proyecto final	Miércoles 15/4
Entrega etapa 2 del proyecto final En esta entrega también se re-entrega el diseño con adecuaciones	Miércoles 29/4
Re-entrega de la etapa 1 del proyecto final Para las y los estudiantes que no alcanzaron los objetivos de la entrega 1	
Entrega etapa 3 del proyecto final	Miércoles 20/5
Re-entrega de la etapa 2 del proyecto Para las y los estudiantes que no alcanzaron los objetivos de esta etapa	
Entrega etapa 4 del proyecto final	Miércoles 10/6
Re-entrega de la etapa 3 del proyecto Para las y los estudiantes que no alcanzaron los objetivos de esta etapa	
Entrega etapa 5 del proyecto final Con la aprobación de esta entrega y las anteriores se aprueba la cursada	Miércoles 24/6
Re-entrega etapa 4 del proyecto final	
Entrega etapa 6: proyecto final completo (primera fecha) Con la aprobación de esta etapa y las anteriores, con promedio al menos 6 y al menos 4 en cada entrega, se promociona la materia	Miércoles 8/7
Re-entrega etapa 5 del proyecto final Con la aprobación de esta entrega y las anteriores se aprueba la cursada	
Entrega etapa 6: proyecto final completo (segunda fecha) Con la aprobación de esta etapa y las anteriores, con promedio al menos 6 y al menos 4 en cada entrega, se promociona la materia	Miércoles 15/7

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):Profesora Titular: Claudia Queiruga: claudiaq@info.unlp.edu.arProfesor Adjunto: Jorge Rosso: jrosso@info.unlp.edu.arJefe de Trabajos Prácticos: Diego Bellante: diegobellante@gmail.comPlataforma virtual: <https://catedras.linti.unlp.edu.ar/> (categoría "Cursos 2026")


Claudia Queiruga



Jorge Rosso