

**JAVA Y APLICACIONES AVANZADAS  
SOBRE INTERNET****Año 2017****Carrera/ Plan :***Licenciatura en Informática*

Plan2003-07/ Plan 2012/Plan 2015

*Licenciatura en Sistemas*

Plan2003-07/ Plan 2012/Plan 2015

**Año:** 4to-5to**Régimen de cursada:** *Semestral***Carácter:** *Optativa***Correlativas:** Proyecto de Software**Profesores:** *Claudia Queiruga y Jorge Rosso***Hs Semanales:** *6 hs***FUNDAMENTACIÓN**

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

El/la estudiante que cursa **Java y Aplicaciones Avanzadas en Internet** maneja el lenguaje JAVA y está entrenado 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 de la velocidad, seguridad y fiabilidad de la tecnología del lado del servidor. **Java y Aplicaciones Avanzadas en 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 de 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-capa y también usando frameworks de nivel aplicación, que facilitan y mejoran la calidad de las aplicaciones desarrolladas y desplegadas.

**OBJETIVOS GENERALES**

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

**CONTENIDOS MINIMOS**

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:

- Aplicar el modelo de componentes multi-tier (múltiples capas físicas) promovido por JEE en el diseño de aplicaciones.
- Establecer qué funcionalidad debe proveer cada componente y cómo interactúan entre ellas.

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

La componente fundacional del módulo web de JEE: Servlets.

Páginas dinámicas escritas en JAVA: JSP (Java ServerPages).

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:

- Programar servlets HTTP como una extensión de un servidor 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.

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:

- 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, auditoria, logging, compresión de la respuesta, etc.

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.

Componente 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: Java ServerPages**

##### Objetivos:

- Programar JSPs usando los diferentes elementos promovidos por la distribución de referencia de JEE.
- Programar descriptivamente los manejadores que atenderán las excepciones y errores.

El ciclo de vida de una JSP. El soporte de servlets provisto para JSP.

Diferentes elementos para programar JSP: scripting (scriptlets, expresiones, declaraciones); directivas (@page, @include y @taglib); acciones estándares (<jsp:useBean>, <jsp:forward>, <jsp:include>, <jsp:param>, <jsp:plugin>).

Manejo de Errores en JSP: tipos de Errores y excepciones.

#### **Unidad 5: Conexión a Bases de Datos**

##### Objetivos:

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

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 6: Arquitectura de Aplicaciones Web: REST y SPA**

##### Objetivos:

- 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

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).

El framework SPA: Angular.

#### **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 que luego son retomadas para la resolución de las prácticas de laboratorio.

Las actividades prácticas son grupales, se desarrollan en laboratorios de PCs, se utilizan herramientas específicas de soporte y desarrollo típicas en la comunidad de software libre, entre ellas SVN, servidor web, motor de base de datos, máquinas virtuales, IDE Eclipse y plugins. Se estipulan instancias de resolución de trabajos prácticos en clase y de carácter individual, permitiendo el seguimiento del proceso formativo.

Se utilizan PCs y presentaciones proyectables. Se utiliza la plataforma virtual de la Facultad de Informática, <http://catedras.info.unlp.edu.ar>, basada en la herramienta *opensource* Moodle (<http://moodle.org>) para publicación de apuntes y guías elaborados por la cátedra y para comunicación entre docentes y estudiantes.

A partir de 2013, se incorpora en esta asignatura la modalidad de cursada semi-presencial. Los/as estudiantes que cursen en esta modalidad, utilizarán la plataforma virtual para mantenerse comunicados con la asignatura, se abrirá un foro de consulta especial para los/as estudiantes de esta modalidad. En esta modalidad se contemplan actividades e instancias de evaluación presencial. Asimismo se recomienda que los/as estudiantes que opten por la modalidad semi-presencial, cuenten con buen manejo de Java Standard Edition.

### EVALUACIÓN

Para la aprobación de la asignatura, el alumno deberá desarrollar un proyecto a lo largo de la cursada que tendrá pautadas 6 entregas, las mismas son de seguimiento y de evaluación con calificación. La sexta entrega consiste en la entrega del proyecto con la funcionalidad completa.

Los alumnos que cursan en modalidad semi-presencial realizarán 4 entregas de las 6 en forma presencial. A su vez, esos mismos días serán evaluados acerca de los temas de las entregas (en forma de coloquio o escrito). Las restantes 2 entregas se realizarán en forma virtual.

La aprobación de la asignatura será por 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.

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, aprobarán la cursada. Para aprobar la asignatura deberán rendir un examen final escrito en alguna de las mesas de examen contemplada en el calendario académico.

### BIBLIOGRAFÍA

**Servlets and JavaServer Pages: The J2EE Technology Web Tier.** Jayson Falkner and Kevin Jones. Addison-Wesley Professional

**The Java EE 6 Tutorial. Basic Concepts - Fourth Edition.** Eric Jendrock, Ian Evans, Devika Gollapudi, Kim Haase, Chinmayee Srivathsa. Addison Wesley

**Core JavaServer Faces (3rd Edition).** David Geary, Cay S. Horstmann. Prentice Hall.

**JavaServer Faces in action.** Mann, Kito D.; Burns, Alan. Editorial Manning, 2005

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

**AspectJ In Action. Second Edition.** Laddad, R. Manning.

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

**CRONOGRAMA DE CLASES Y EVALUACIONES**

Clase	Contenidos Teóricos	Actividades	Evaluaciones previstas
<b>Clase 1: Miércoles 15/3</b>	<b>Arquitectura JEE Desarrollo de Servlets HTTP</b>	Presentación de la materia.	
		Presentación de la arquitectura JEE y sus componentes. La componente fundacional: Servlets.	
		Presentación del proyecto final	
		Práctica de servlets http.	
		Presentación de las herramientas de desarrollo (IDE, SVN, servidor Tomcat, servidor de BD)	
		Elección de la modalidad de cursada (presencial, semipresencial) mediante la administración de encuesta on-line.	
<b>Clase 2: Miércoles 22/3</b>	<b>Soporte de Sesiones HTTP</b>	Presentación de Sesiones HTTP en la API de Servlets.	
		Práctica de sesiones http.	
<b>Clase 3: Miércoles 29/3</b>	<b>Servlet Listeners</b>	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 sobre desarrollo de servlets listeners, alcances y redireccionamiento HTTP.	
		Encuesta relevamiento grupos de trabajo	
		<b>Explicación y publicación de los alcances de la primera entrega del trabajo final: "Construcción del prototipo"</b>	
<b>Clase 4: Miércoles 5/4</b>	<b>Persistencia en Aplicaciones JAVA server-side.</b>	Presentación de acceso a bases de datos a través de JDBC (Java DataBase Connectivity y DataSource.	
		Ejemplos de DAOs (Data Access Objects)	
		<b>Se trabaja en la etapa 1 del proyecto final.</b> Consultas y guías para el desarrollo de la primera entrega del proyecto final.	

<b>Clase 5: Miércoles 12/4</b>		Consultas y entrega del prototipo del trabajo final.	<b>Evaluación de la primera entrega (alumnos modalidad presencial).</b>  <b>Alumnos de modalidad semi-presencial: evaluación parcial presencial.</b>
<b>Clase 6: Miércoles 19/4</b>	<b>Servlet Filter</b>	Presentación de objetos filtros y su aplicación a requerimientos no funcionales.  Práctica sobre desarrollo de servlets filtros, uso de conexiones JDBC y DATASOURCE para conexiones a bases de datos y patrón DAO.	
<b>Clase 7: Miércoles 26/4</b>		<b>Explicación y publicación de los alcances de la segunda entrega del trabajo final: "Definición de objetos del modelo"</b>	Evaluación de la re-entrega de la etapa 1 del proyecto final (para alumnos que no alcanzaron los objetivos de la entrega 1).
<b>Clase 8: Miércoles 3/5</b>	<b>JavaServerPage (primera parte)</b>	Presentación de JSP. Ciclo de Vida. Programar JSPs.  Práctica de JSP – Primera Parte.	
<b>Clase 9: Miércoles 10/5</b>		<b>Se trabaja en la etapa 2 del proyecto final.</b>  Consultas y entrega de la segunda etapa del trabajo final.	<b>Evaluación de la segunda entrega del proyecto final (alumnos modalidad presencial).</b>
<b>Clase 10: Miércoles 17/5</b>	<b>JavaServer Pages (segunda parte)</b>	Presentación y desarrollo de tags estándares y Java Beans en JSP.  Práctica de JSP – Segunda Parte.	
<b>Clase 11: Miércoles 24/5</b>	<b>Persistencia de objetos mediante un ORM: JPA &amp; Hibernate</b>	Presentación de JPA (Java Persistence API). Hibernate  Práctica sobre persistencia usando JPA.  <b>Explicación y publicación de los alcances de la tercera entrega del trabajo final: "Desarrollo de la capa de persistencia"</b>  Consultas y guías para el desarrollo de la tercera etapa del trabajo final.	Evaluación de la re-entrega de la etapa 2 del proyecto (para estudiantes que no alcanzaron los objetivos de esta etapa)
<b>Clase 12: Miércoles 31/5</b>		<b>Se trabaja en la etapa 3 del proyecto final.</b>  Consultas y guías para el desarrollo de la tercera etapa del trabajo final.	

<b>Clase 13: Miércoles 7/6</b>		Consultas y entrega de la tercera etapa del trabajo final.	<b>Evaluación de la tercera entrega (alumnos modalidad presencial).</b>  <b>Alumnos de modalidad semi-presencial: evaluación parcial presencial.</b>
<b>Clase 14: Miércoles 14/6</b>	<b>Arquitectura de Aplicaciones Web: REST y SPA</b>  <b>Primera parte: webservices tipo RESTful</b>	Presentación de Arquitectura REST, Web Services de tipo RESTful. Desarrollo de APIs utilizando la API de Java JAX-RS (JSR 311 - Java API para RESTful Web Services) <b>Explicación sobre los alcances de la cuarta entrega del trabajo final: "Desarrollo de una API Rest"</b>  Se comienza a trabajar en la entrega 4	
<b>Clase 15: Miércoles 21/6</b>	<b>Arquitectura de Aplicaciones Web: REST y SPA</b>  <b>Segunda parte: Angular</b>	<b>Presentación del framework Angular.</b>  <b>Se trabaja sobre la entrega 4</b>	
<b>Clase 16: Miércoles 28/6</b>		Consultas y entrega de la cuarta etapa del trabajo final.	<b>Evaluación de la etapa 4 del proyecto final.</b>  <b>Evaluación de la re-entrega de la etapa 3 del proyecto (para estudiantes que no alcanzaron los objetivos de esta etapa)</b>
<b>Clase 17: Miércoles 5/7</b>		<b>Práctica de Angular</b>  <b>Explicación y publicación de los alcances de la quinta entrega del trabajo final: "Desarrollo de la vista (parcial) del proyecto usando Angular"</b>	
<b>Clase 18: Miércoles</b>		Consultas y entrega de la quinta etapa del trabajo final.	<b>Evaluación de la etapa 5 del proyecto final.</b>



<b>12/7</b>			<b>Alumnos de modalidad semipresencial: evaluación parcial presencial</b>  <b>Evaluación de la reentrega de etapa 4 del proyecto final.</b>
<b>Vacaciones de invierno del 17/07 al 29/07</b>			
<b>Clase 19: Miércoles 3/8</b>		Consultas	<b>Re-entrega de las etapa 5 del proyecto final o entrega final</b>

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

**Prof. Claudia Queiruga: [claudiaq@info.unlp.edu.ar](mailto:claudiaq@info.unlp.edu.ar)**

**Prof. Jorge Rosso: [jrosso@info.unlp.edu.ar](mailto:jrosso@info.unlp.edu.ar)**

**JTP Diego Bellante: [diegobellante@gmail.com](mailto:diegobellante@gmail.com)**

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

Firmas del/los profesores responsables: