



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

## JAVA Y APLICACIONES AVANZADAS EN INTERNET

Año 2012

### Carrera/ Plan :

Licenciatura en Sistemas Plan 2003/07  
Licenciatura en Informática Plan 2003/07  
Licenciatura en Informática Plan 90  
Licenciatura en Informática Plan 2012  
Licenciatura en Sistemas Plan 2012

**Área:** Arquitectura, Sistemas Operativos  
y Redes

**Año:** 4º o 5º año

**Régimen de Cursada:** *Semestral*

**Carácter:** Optativa

**Correlativas:**

Laboratorio de Software

**Profesor:** *Claudia. Queiruga*

**Hs Semanales:**

---

### 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 alumno que cursa **Java y Aplicaciones Avanzadas en Internet** maneja fluidamente el lenguaje JAVA y está entrenado en el uso de IDEs (Integrated Development Environment) relacionados a tecnologías JAVA.

Los desarrolladores de hoy reconocen cada vez más la necesidad de aplicaciones distribuidas, transaccionales y portables que se aprovechan de la velocidad, seguridad y fiabilidad de la tecnología del lado del servidor. **Java y Aplicaciones Avanzadas en Internet** le provee al alumno 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 aplicaciones.

Las habilidades que el alumno 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.
- Desarrollar custom tags controlados por tag handlers y por tag files y distribuirlos en un archivo JAR.
- 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>`); acciones customizadas y el lenguaje de expresiones JSP EL.

JSP y Java Beans.

Desarrollar tags customizados en JSP usando el modelo clásico y el "nuevo y simple" de JSP 2.0. Ventajas de trabajar con librerías de tags customizados.

El archivo descriptor de la librería de custom tags, TLD.

Objetos que controlan los tags customizados: tag handlers y tag files.

Empaquetado de librerías de custom tags en un archivo JAR.



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.

### **Unidad 6: El lenguaje AspectJ**

#### Objetivos:

- Programar aspectos que permitan securitizar y auditar las aplicaciones web

Conceptos básicos sobre los elementos que conforman el lenguaje AspectJ: Aspect, Join points, Pointcut, Advice y Weaving

### **Unidad 7: El framework JSF (JavaServer Faces)**

#### Objetivos:

- Construir una aplicación prototipo usando JSF.
- Comparar JSF con Swing.
- Analizar los eventos generados en el cliente web y consumidos en el servidor.
- Experimentar con las facilidades provistas por los IDEs para programar vistas (páginas) JSF.

El primer framework estándar JSF (JavaServer Faces) incorporado en la especificación JEE: JSF. Construcción de aplicaciones web programando componentes de interfaz de usuario y manejadores de eventos.

Componentes fundamentales de JSF: la vista, componentes de IU, eventos, manejadores de eventos, convertidores, validadores, mensajes, renders, el archivo faces-config.xml. Fases del procesamiento de una petición HTTP.

### **METODOLOGÍA DE ENSEÑANZA**

La asignatura consolida la formación experimental y profesional del alumno 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 conceptuales 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 opensource Moodle (<http://moodle.org>) para publicación de apuntes y guías elaborados por la cátedra y para comunicación entre docentes y alumnos.

### **EVALUACIÓN**

Para la aprobación de los trabajos prácticos el alumno debe realizar el desarrollo de un proyecto que tendrá pautadas 6 entregas. En cada una de estas instancias, el alumno deberá presentar una producción de su proyecto. Estas entregas son de seguimiento y de evaluación con calificación.

La aprobación final de la asignatura consiste en una evaluación del trabajo desarrollado durante la cursada y una defensa oral del mismo en la que se indaga sobre los conceptos teóricos aplicados en el desarrollo, o en una prueba escrita sobre los contenidos desarrollados en la teoría y la práctica a lo largo de la asignatura.

### **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

**J2EE open source toolkit. Building an enterprise platform with open source tools.** Bell, John T.; Ng, Stanford; Lambros, James T. Editorial Wiley, 2003

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

**Core J2EE patterns. Best practices and design strategies** Alur, Deepak; Crupi, John; Malks, Dan- Prentice Hall/Sun Microsystems Press, 2003

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



### CRONOGRAMA DE CLASES Y EVALUACIONES

<b>Clase</b>	<b>Actividades</b>
Presentación de la materia Arquitectura JEE Servlets	Presentación del proyecto: <i>"Gestión de proyectos comunitarios"</i>  Práctica de servlets Herramientas de desarrollo
Soporte de Sesiones	Práctica de sesiones Consignas de la primera entrega
Etapas 1 del proyecto	Primera entrega y evaluación
Servlet Listeners	Presentación de diferentes tipos de objetos listeners y su aplicación.  Desarrollo de Servlets listeners, de contexto, de sesión, de requerimiento.  Consignas de la 2° entrega
Etapas 2 del proyecto	Segunda entrega y evaluación
Persistencia	Presentación de JDBC  Presentación de Hibernate con JPA  Ejemplos de DAOs.  Consultas sobre el proyecto.  Consignas de la 3° entrega
Servlet Filtro	Presentación de diferentes tipos de objetos filtros.  Desarrollo de casos que aplican filtros en seguridad de aplicaciones web, auditoría y modificación de la respuesta del servidor.  Consultas sobre el proyecto.
Etapas 3 del proyecto	Tercera entrega y evaluación



Programación Orientada a Aspectos usando AspectJ	Presentación del lenguaje orientado a aspectos, AspectJ y su aplicación para el desarrollo de requerimientos no funcionales.  Desarrollo de Aspectos relacionados al proyecto  Consignas de la 4° entrega
JSP	Tags estándares provistos por JSP.  Desarrollo de páginas usando tag estándares.  Consultas sobre el proyecto
Etapa 4 del proyecto	Cuarta entrega y evaluación
JSP	Técnicas de construcción de custom tags.  Desarrollo de custom tags personalizados.  Consignas de la 5° entrega
Etapa 5 del proyecto	Quinta entrega y evaluación
JSF	Presentación del framework JSF  Desarrollo de la vista del proyecto usando JSF  Consignas de la 6° entrega
	Consultas sobre el proyecto.
Etapa 6 del proyecto	Sexta entrega y evaluación



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

**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)**

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

Plataforma virtual: <https://catedras.info.unlp.edu.ar/> (categoría “Cursos 2012”)

Firmas del/los profesores responsables: