



UNIVERSIDAD NACIONAL DE CATAMARCA				
FACULTAD DE TECNOLOGÍA Y CIENCIAS APLICADAS				
INGENIERIA EN INFORMATICA				
<b>PROGRAMA DE:</b>			<b>Código:</b> 7045	
<b>PROGRAMACION III</b>			<b>Área:</b> Tecnologías Básicas	
<b>Carga horaria Total: 105</b>			<b>Plan:</b> 2011	
<b>Horas</b>			<b>Régimen:</b> Cuatrimestral	
<b>Cuerpo Docente</b>				
<b>Teórico – Práctico</b>	<b>Actividad Práctica</b>			Profesor Adjunto a cargo: Esp. Gallardo Cecilia
	FE <sup>1</sup>	RPI <sup>2</sup>	ADyP <sup>3</sup>	Profesor Adjunto: Lic. Oscar Quinteros
105	70	35	--	Ay. Diplomado: Esp. Miranda Marta
<b>Correlativas</b>				
Base de Datos				
<b>OBJETIVOS DE ENSEÑANZA:</b>				
Se pretende que el estudiante obtenga los conocimientos y capacidades necesarias para desarrollar una Aplicación Web, tomando como base diversas tecnologías y estándares Web del Consorcio W3C (HTML, CSS, JavaScript, XML.) y el lenguaje de programación del lado del servidor web en particular, utilizando buenas prácticas de desarrollo y aplicando pautas de seguridad de la información.				
<b>CONTENIDOS MINIMOS:</b>				
La Web. Navegadores Web. Servidores Web. Programación del Lado del Servidor. Protocolo. Lenguajes de hipertexto. Hojas de Estilo en Cascada (CSS). Lenguajes de Marcas Extensible. Lenguajes de Programación para el Desarrollo de Aplicaciones Web. Acceso a Bases de Datos. Seguridad. Tipos de seguridad. Autorización, autenticación y acceso. Servicios Web.				
<b>PROGRAMA ANALITICO:</b>				
<b>UNIDAD N° 1: CONCEPTOS BÁSICOS DE LA WEB</b>				
<ul style="list-style-type: none"><li>▪ <b>La Web:</b> Internet y la World Wide Web. Estándares del consorcio W3C. Componentes básicos de la Web.</li><li>▪ <b>Protocolos Web:</b> Definición. Tipos. Protocolo de transporte de hipertexto (HTTP). Métodos de Petición de HTTP. Códigos de respuesta HTTP. Cabeceras HTTP. Protocolo HTTPS.</li><li>▪ <b>Navegadores Web:</b> Definición. Flujos de procesamiento. Ejemplos de navegadores web.</li><li>▪ <b>Servidores Web:</b> Definición. Funcionamiento básico. Servidores web más utilizados.</li><li>▪ <b>Aplicaciones Web:</b> Definición. Beneficios. Arquitecturas</li><li>▪ Herramientas y recursos para el desarrollo de Aplicaciones Web.</li></ul>				

<sup>1</sup> FE: Horas dedicadas a la Formación Experimental

<sup>2</sup> RPI: Horas dedicadas a la Resolución de Problemas de Ingeniería

<sup>3</sup> ADyP: Horas dedicadas a las Actividades de Diseño y Proyecto



#### **UNIDAD N° 2: LENGUAJES DE MARCAS Y ESTILOS**

- **Lenguaje de marcas de hipertexto:** Definición. Evolución. Lenguaje HTML: definición, versiones de HTML, estructura y etiquetas principales, atributos, elementos, codificación de caracteres, enlaces, tipos de enlaces, listas, imágenes, tablas, formularios, estructura y layout, metainformación, metadatos, validación.
- **Hojas de Estilo en Cascada (CSS):** Definición, versiones de CSS. Inclusión de CSS en Documentos HTML. Sintaxis básica. Medios de salida. Agrupación de propiedades CSS. Diferentes tipos de Selectores. Particularidades de estilos: herencia, colisiones. Unidades de medida. Modelo de Cajas. Propiedades de posicionamiento. Propiedades de visualización. Diseño web responsivo. Framework front-end basado en HTML, CSS y Javascript.

#### **UNIDAD 3: PROGRAMACIÓN DEL LADO DEL CLIENTE**

- **Lenguajes de Programación del lado del Cliente:** Definición. Scripts. Ejemplos.
- **Lenguaje de scripting JavaScript:** Definición. Inclusión en páginas HTML. Soporte en navegadores. Sintaxis básica. Variables. Tipos de variables. Operadores. Estructuras de control. Funciones. Objetos
- **DOM (Modelo de objetos del documento):** Tipos de nodos. Acceso directo a nodos. Creación, eliminación y reemplazo de nodos. Acceso directo a atributos. Modelo de eventos. Manejadores de eventos. Manejo de formularios. Validación de formularios. Manejo de errores. Ventanas emergentes.

#### **UNIDAD 4: PROGRAMACIÓN DEL LADO DEL SERVIDOR**

- **Lenguajes de Programación del lado del Servidor:** Definición. Evolución tecnológica. Lenguajes disponibles.
- **Tecnología Servlet-Java-JSP (Java Server Pages):** Funcionamiento básico y elementos de sintaxis de Servlet y JSP para interactuar con una petición HTTP. Relación entre Servlet y JSP
- **Patrón de Diseño Modelo-Vista-Controlador:** Definición. Utilización. Ventajas.
- **Frameworks para el desarrollo de Aplicaciones Web:** Definición. Ventajas y desventajas.
- **Framework de desarrollo Grails:** Definición. Arquitectura. Componentes principales. Interacción y flujo de componentes.
- **Lenguaje Dinámico Groovy:** Definición. Estructura del lenguaje. Sintaxis básica. Diferencias entre Groovy y Java. Listas y Mapas. Closures.
- **Patrón MVC: Modelo en Grails.** Implementación de mapeo relacional de objetos de Grails (GORM). Clases de Dominio. Especificación de restricciones y relaciones entre clases de dominio. Interacción con base de datos (consultas).
- **Patrón MVC: Controladores en Grails:** Definición de clases controladores. Acciones. Redirecciones. Enlace de datos (data binding) entre peticiones HTTP. Interacción de controladores con Capa de Servicios. Sesiones: definición, formas de implementación y manejo.
- **Patrón MVC: Vistas en Grails:** Tecnología GSP (Groovy Servers Pages). Estructura. Principales etiquetas. Templates.
- **Seguridad en Aplicaciones Web:** Definiciones. Clasificación de amenazas a la Seguridad Web. OWASP Top 10. Formas de prevención de los riesgos de seguridad más conocidos.



#### **UNIDAD 5: SERVICIOS WEB**

- **Lenguaje de Marcas Extensible (XML):** Definición. Diferencias entre XML y HTML. Estructura. Componentes básicos. XML Namespaces. Documentos XML bien formados y válidos. Validadores de XML: Definición de Tipo de Documento (DTD) y Definición de Esquema XML (XSD).
- **Notación de Objetos JavaScript (JSON):** Sintaxis. Tipos de datos. Diferencias entre JSON y XML.
- **Servicios Web (WS):** Definición. Objetivos. Arquitectura de Aplicaciones con WS. Escenarios de uso. Formas de implementación: SOAP y REST. Implementación de WS en Grails.

#### **ACTIVIDADES PRÁCTICAS:**

- Consignas de resolución práctica de temas abordados en las clases teóricas organizadas en base a la explicación didáctica.
- Clases prácticas de resolución de ejercicios.
- Trabajos prácticos que refuerzan y complementan los conocimientos teóricos desarrollados.
- Trabajo integrador de laboratorio, que consiste en la construcción de una Aplicación Web, incrementando el desarrollo a medida que se abordan los conocimientos.

<b>Modalidad de la Actividad Práctica</b>	<b>Horas Totales</b>
Formación Experimental (simulación, otros)	40
Resolución de Problemas de Ingeniería	25
Actividades Proyecto y Diseño	--

#### **BIBLIOGRAFÍA:**

<b>Título</b>	<b>Autores</b>	<b>Editorial</b>	<b>Año Edic.</b>
Beginning JSP, JSF and Tomcat	Giulio Zambon	Springer Science + Business Media New York	2012
Programming Groovy 2. Dynamic Productivity for the Java Developer	Venkat Subramaniam	The Pragmatic Bookshelf	2013
The Definitive Guide to Grails. Second Edition	Graeme Rocher & Jeff Brown	Apress	2009
Grails 3 - Step By Step	Cristian Olaru	Ebook – Plataforma Leanpub	2017



### **SITIOS WEB DE REFERENCIA:**

- W3Schools Online Web Tutorial – URL: <https://www.w3schools.com/>
- World Wide Web Consortium (W3C) – URL: <https://www.w3.org/>
- Bootstrap – URL: <https://getbootstrap.com/>
- Lenguaje de Programación Groovy – URL: <http://groovy-lang.org/>
- Framework Grails – URL: <https://grails.org/>
- Libro Eloquent JavaScript (3ra edición): <http://eloquentjavascript.net/>

### **CONDICIONES PARA REGULARIZAR LA ASIGNATURA**

- Tener como mínimo 70% de asistencia a clases teóricas
- Tener como mínimo 80% de asistencia a clases prácticas
- Tener como mínimo 80% de asistencia a clases de laboratorio
- Entregar resueltos todos los Trabajos Prácticos propuestos
- Aprobar 2 (dos) exámenes parciales teórico-prácticos con nota mayor o igual a 5 (Escala 0-10). Se pueden recuperar todos los parciales en caso de ser desaprobados. Los recuperatorios serán posteriores al segundo parcial.
- Presentar y aprobar Trabajo Final de Laboratorio

### **EXAMEN FINAL REGULAR**

El alumno debe extraer al azar dos bolillas correspondientes a las unidades temáticas del programa actualizado de la materia, pudiendo escoger un tema de alguna de las unidades sorteadas para comenzar la exposición oral. Posteriormente el tribunal podrá realizar preguntas sobre cualquier otro ítem del programa. Por cada tema se deben exponer los conceptos teóricos abordados en la cátedra, ejemplificando según el caso, con algún breve ejemplo práctico.

### **EXAMEN FINAL LIBRE**

- El Alumno deberá presentar, con 10 (diez) días hábiles de anticipación a la fecha del examen final, todos los trabajos prácticos y de laboratorio correspondientes al año en curso, o en su defecto, al año inmediato anterior.
- Una vez verificados y aprobados los trabajos prácticos y de laboratorio, el alumno deberá desarrollar en computadora, una Aplicación Web conforme a los requisitos de un caso de estudio provisto por el tribunal, correspondiente a los temas abordados en la cátedra. Para realizar esto, el alumno tendrá un tiempo disponible de 2 (dos) horas.
- Una vez aprobadas las dos instancias anteriores de evaluación, el alumno estará en condiciones de rendir el examen oral bajo las mismas condiciones del “Examen Final Regular”.



*Universidad Nacional de Catamarca  
Facultad de Tecnología y Ciencias Aplicadas  
Departamento Informática  
Año académico: 2019*

- Solamente se podrá rendir en condición de Libre, en los turnos de exámenes ordinarios (Febrero-Marzo, Julio-Agosto y Noviembre-Diciembre).

A handwritten signature in black ink, appearing to read 'Cecilia E. Gallardo'.

Lic. Cecilia E. Gallardo  
Prof. Adj. A/C Programación III