

CARTA AL ESTUDIANTE

CODIGO: EIF-209
NOMBRE: Programación IV
REQUISITOS: Programación III
NATURALEZA: Teórico/práctico
AREA DISCIPLINARIA: Ingeniería de software
NIVEL: III nivel
CICLO LECTIVO: I ciclo lectivo 2017
PROFESORES: Georges Alfaro (I:13), Prof. Cristhian Garita (I:17), José Sánchez (M: 15, V:15)
(Horas de consulta)
COORDINADOR: Prof. José Sánchez

Créditos	Horas Semanales	Horas Presenciales		Horas estudio independiente
4	11	Prácticas	Teóricas	7
		2	2	

DESCRIPCIÓN

Este curso trata varios temas relacionados con el desarrollo de aplicaciones *web* que operan en Internet. Se construyen aplicaciones de gestión estructuradas por capas y con interfaz *web*. En la parte del cliente, se integran los conceptos del lenguaje de marcas HTML con páginas dinámicas utilizando lenguajes de *scripting* y hojas de estilo. En la parte servidor se integra un lenguaje orientado a objetos, un modelo de aplicación, conexión a bases de datos y plataformas de trabajo basadas en componentes (*frameworks*). La práctica consiste en proyectos que los estudiantes irán desarrollando a lo largo del curso.

OBJETIVO GENERAL

Que el estudiante sea capaz de desarrollar aplicaciones *web* que operen en Internet, integrando diversas técnicas y tecnologías en la implementación tanto de la parte cliente como de la parte del servidor y aplicando principios adecuados de ingeniería de software.

OBJETIVOS ESPECÍFICOS: Que el estudiante sea capaz de:

1. Comprender el modo de operación de internet y de la *web*
2. Describir la estructura de una página utilizando HTML y HTML 5.
3. Trabajar el modelo de objetos (DOM) de JavaScript y las API de HTML5.
4. Utilizar XML, JSON para la descripción y serialización de datos.
5. Comprender y utilizar el modelo de páginas dinámicas por medio de *servlets* y JSP
6. Desarrollar aplicaciones *web* por capas con conexión a bases de datos relacionales
7. Utilizar AJAX para la construcción de páginas de contenido rico (*Rich Content Pages*)
8. Usar plataformas basadas en componentes para el desarrollo de aplicaciones *web* en Java

HABILIDADES Y COMPETENCIAS

Al terminar el curso, el estudiante deberá tener las habilidades y competencias necesarias para:

1. Diseñar y construir aplicaciones *web*.
2. Organizar adecuadamente aplicaciones *web* según prácticas comunes de diseño.
3. Comprender y utilizar las diferentes técnicas y herramientas disponibles para la construcción de sitios *web* funcionales y seguros.
4. Trabajar adecuadamente en equipos de trabajo, asumiendo sus roles y responsabilidades y resolviendo adecuadamente los conflictos.
5. Evaluar su proceso de aprendizaje y el de sus compañeros de grupo.

CONTENIDOS:

1. Conceptos de programación en Internet y *web* (Objetivo 1)

- 1.1. Internet y *web*
- 1.2. Protocolos TCP/IP y HTTP
- 1.3. Servidor *web*, sitio *web* y página *web*
- 1.4. El servidor *web*
- 1.5. Las aplicaciones *web*

2. Introducción al lenguaje de marcas HTML (Objetivo 2)

- 2.1. Principales marcas
- 2.2. Enlaces
- 2.3. Formularios
- 2.4. Hojas de estilo en cascada (CSS)
- 2.5. Fundamentos de HTML5

3. Introducción a la programación con *scripts* (Objetivo 2)

- 3.1. El lenguaje JavaScript
- 3.2. Estructuras de datos y de control
- 3.3. Funciones definidas por el usuario y de biblioteca
- 3.4. El modelo DOM y el procesamiento de eventos
- 3.5. Expresiones regulares
- 3.6. Errores comunes
- 3.7. La API de HTML5
 - 3.7.1. Herramientas gráficas y capacidades multimediales
 - 3.7.2. Verificación de formularios y bases de datos locales
 - 3.7.3. Geolocalización

4. El modelo de capas (Objetivo 4)

- 4.1. Arquitectura del modelo de capas
- 4.2. Capa de presentación
- 4.3. Capa de la lógica de aplicación
- 4.4. Capa de acceso a la base de datos
- 4.5. Modelo conceptual y modelo de clases en UML

5. Capa de presentación con páginas dinámicas (Objetivo 3)

- 5.1. Páginas dinámicas
- 5.2. Procesamiento de peticiones (*request*)
- 5.3. Generación de respuestas (*response*)
- 5.4. *Servlets* y *Java Server Pages* JSP
- 5.5. Conceptos básicos de AJAX

6. **Capa lógica de la aplicación (Objetivo 4)**
 - 6.1. Objetos, clases e interfaces
 - 6.2. Asociaciones y herencias
 - 6.3. Gestores
7. **Capa de acceso a la base de datos (Objetivo 4)**
 - 7.1. Modelo de objetos versus modelo de datos
 - 7.2. Conexión a bases de datos
 - 7.3. Objetos de datos: conexiones, sentencias o comandos, resultados
8. **Validación en el cliente y en el servidor (Objetivo 3)**
 - 8.1. Validación mediante *scripts* en el cliente
 - 8.2. Validación en el servidor
9. **Administración de sesiones (Objetivo 3)**
 - 9.1. Definición y seguimiento de sesiones
 - 9.2. Justificación del uso de sesiones
10. **Uso de *frameworks* (Objetivo 5)**
 - 10.1. El patrón de arquitectura MVC (*Model-View-Controller*)
 - 10.2. Las fases del ciclo de procesamiento
 - 10.3. Ligado de componentes (*backing beans binding*)
 - 10.4. Ligado de propiedades (*value binding*)
 - 10.5. Bibliotecas de componentes
11. **Autenticación y seguridad (Objetivo 3)**
 - 11.1. Protección de datos en la *web*
 - 11.2. Autenticación en el protocolo http
 - 11.3. Autenticación usando la base de datos
12. **Diseño de aplicaciones *web* (Objetivo 4)**
 - 12.1. Servicios *web*

METODOLOGÍA:

Las aplicaciones *web*, al contrario de las aplicaciones locales, se componen al menos de dos partes, que operan en distintas computadoras interconectadas por la red Internet. La implementación tanto de la parte cliente como de la parte servidor de la aplicación requiere no solo conocer las tecnologías que se usan en cada parte sino los mecanismos para su adecuada integración.

La estrategia seguida en este curso divide su temática en cuatro grandes secciones. Primero se ubica al estudiante en un marco general de cómo operan las aplicaciones *web*. Luego se estudian las tecnologías y lenguajes requeridos en la parte cliente de la aplicación. Posteriormente se hace lo propio con la parte servidor, para tener así una aplicación completa. En la última parte del curso se estudia y aplica alguna plataforma de componentes (*framework*) que permita una mayor productividad en el desarrollo de este tipo de aplicaciones.

El curso consta de clases magistrales, impartidas por el profesor, seguidas por sesiones prácticas o pequeños laboratorios para que el estudiante aplique los conceptos expuestos.

También habrá proyectos extra clase más complejos donde el estudiante deberá integrar los distintos conceptos al crear aplicaciones de software para *web*, por supuesto con el asesoramiento del profesor.

También se invitará a expertos en algunos temas para complementar y motivar el aprendizaje.

El curso propicia el desarrollo de destrezas de diseño de software, enfrentando a los estudiantes a proyectos de mediana complejidad que ellos deberán completar de principio a fin. También se refuerza la habilidad de la reutilización de software, pues tanto en las sesiones de laboratorio como en los exámenes y proyectos los estudiantes tendrán a disposición todos los ejemplos y prácticas anteriores, de manera que rápidamente puedan adaptar una solución existente a una nueva necesidad más o menos semejante.

EVALUACION:

Descripción	Porcentaje
Examen parcial Cubre los objetivos del curso hasta los temas de la semana 6.	25%
Examen Final Cubre todos los objetivos del curso.	25%
Proyectos Los proyectos o tareas grupales servirán para evaluar aspectos prácticos concretos de los temas estudiados en el curso. Pueden realizarse en grupos más dos personas, según lo indique el profesor en el momento que se entregue el enunciado de este.	30%
Evaluaciones cortas	20%

NOTA: Por ser este un curso de naturaleza teórico/práctica no tiene examen extraordinario.

CRONOGRAMA:

	<i>Contenidos (temas)</i>
Semana 1	1
Semanas 2-3	2
Semanas 4-5	3
Semanas 6-9	4,5
Semana 7	Examen parcial (Domingo 2 Abril 1pm)
Semana 10	Proyecto #1
Semanas 10-13	6,7
Semanas 14-16	8-11
Semana 16	Proyecto final (entrega final)
Semana 17	Examen final (Domingo 18 Junio 1pm)

BIBLIOGRAFIA Y MATERIAL DE APOYO:

- Castro, Elizabeth. *XML Guía de Aprendizaje*. Prentice Hall, 2001.
- Ceballos, Francisco. *Java 2. Interfaces gráficas y aplicaciones para Internet*. Alfaomega Grupo Editor, 2005.
- Fields, Duane K; Kolb, Mark A. *Web Development with Java Sever Pages*. 2nd edition, Manning Publications Company, 2001.
- Firtman, Maximiliano R. *Ajax. Web 2.0 para profesionales*. 1era edición. Alfaomega Grupo Editor S.A. de C.V. México, 2008
- Flanagan, David. *JavaScript: The Definitive Guide*. 5th edition, O'Reilly, 2006.
- Hall, Marty; Brown, Larry; Chaikin, Yaakov. *Core Servlets and Java Server Pages. Volume 2: Advanced Technologies*. 2nd edition, Prentice Hall, 2007.
- Hall, Marty; Brown, Larry. *Core Servlets and Java Server Pages*. 2nd edition, Prentice Hall, 2004
- Hunter Jason; Crawford, William. *Java Servlet Programming*. 2da Edición. O'Reilly. 2003.
- Lemay, Laura; Cadenhead, Roger. *Sams Teach Yourself Java 6 in 21 Days*. 5th edition, Prentice-Hall, 2007.
- Lubbers, Peter; Albers, Brian and Salim, Frank. *Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development*. Apress, Springer Science+Business Media, LLC. New York, USA, 2010.
- Negrino, Tom; Smith, Dori. *JavaScript Guía de Aprendizaje*. 3era edición, Prentice Hall, 2000.
- Stefanov, Stoyan; Chetan Sharma, Kumar. *Object-Oriented JavaScript*. 2nd edition, Packt Publishing. Birmingham B3 2PB, UK. 2013