

UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD POLITÉCNICA
LICENCIATURA EN CIENCIAS INFORMÁTICAS
ÉNFASIS EN ANÁLISIS DE SISTEMAS INFORMÁTICOS
PLAN 2009
PROGRAMA DE ESTUDIOS
ANEXO 02

I. IDENTIFICACIÓN

- | | |
|--------------------|---|
| 1. Asignatura | : Electiva II - Programación Web <i>Front-End</i> |
| 2. Código | : 6.2.B |
| 3. Horas semanales | : 5 horas |
| 4. Total de horas | : 80 horas |

II. JUSTIFICACIÓN

En la actualidad, la globalización obliga a las empresas a tener un sitio web desde el cual ofrecer servicios o vender productos en forma electrónica. Por otra parte, incluso los sistemas informáticos de uso interno, de importancia crítica para las empresas, tienden a ser construidos como sistemas web, por su facilidad de administración y diversidad de medios de acceso en cuanto a tipos de dispositivos, sistemas operativos, navegadores, etc.

El énfasis del curso será el revisar las técnicas y herramientas para el diseño y desarrollo de las interfaces de usuario de los sistemas web (front-end), teniendo presente la heterogeneidad de dispositivos, navegadores, sistemas operativos y capacidades de conexión a la red, de forma independiente a la tecnología empleada del lado servidor (back-end).

III. OBJETIVOS GENERALES

1. Conocer los detalles que envuelven al diseño y programación de interfaces de usuario de tipo página web.
2. Conocer la semántica y la sintaxis de los diferentes lenguajes utilizados en el desarrollo de las interfaces de usuario en aplicaciones web.
3. Desarrollar la habilidad de diseñar e implementar interfaces de usuario en aplicaciones web.
4. Conocer los detalles de hardware, protocolos y estándares subyacentes a fin de tener argumentos suficientes para recomendar soluciones acordes a las necesidades.

IV. OBJETIVOS ESPECÍFICOS

A. Conocimientos

1. Herramientas de desarrollo de interfaces de sistemas web.
2. Criterios de diseño de interfaces de usuario de sistemas web, atendiendo a las buenas prácticas de usabilidad, diseño y estética.
3. Criterios de diseño que influyen en el rendimiento de las interfaces de usuarios de sistemas web.
4. Criterios de diseño que influyen en la seguridad de las interfaces de usuarios de sistemas web.
5. Protocolos y herramientas de comunicación entre la interfaz web y el back-end.

B. Habilidades

1. Evaluar y seleccionar las herramientas y plataformas más adecuadas según el tipo de aplicación de interfaces web.
2. Diseñar interfaces de usuario de sistemas web en base a los estándares y buenas prácticas de usabilidad, diseño y estética.
3. Diseñar interfaces de usuario de sistemas web con altos niveles de rendimiento atendiendo a los tipos de dispositivos en los que estas deban desplegarse.
4. Diseñar interfaces de usuario de sistemas web con altos niveles de seguridad atendiendo a los tipos de dispositivos en los que estas deban desplegarse.
5. Diagnosticar y corregir problemas de rendimiento en interfaces de usuario de sistemas web.
6. Detectar vulnerabilidades de seguridad en interfaces de usuario de sistemas web, y proponer acciones paliativas.

C. Competencias

1. Capacidad de aplicar los conocimientos en la práctica.
2. Disposición para el trabajo en equipo.
3. Capacidad de abstracción, análisis y síntesis y presentaciones orales.
4. Habilidades para buscar, procesar y analizar información procedente de fuentes diversas.
5. Capacidad para identificar, plantear y resolver problemas
6. Capacidad de comunicación oral y escrita.

V. PRE – REQUISITO

1. Electiva I.
2. Base de Datos II.



VI. CONTENIDO**6.1. Unidades programáticas**

1. Protocolos de comunicación.
2. Tecnologías de front-end exclusivamente client-side
3. Tecnologías de front-end server y client-side
4. Procesos de desarrollo de front-end
5. Performance en el front-end
6. Seguridad en el front-end

6.2. Desarrollo de las unidades programáticas

1. Protocolos de comunicación.
 - 1.1. HTTP, Web Socket, HTTP 2
 - 1.2. HTTPS
 - 1.3. Cookies
 - 1.4. Caché
 - 1.5. Proxies
 - 1.6. Balanceadores de carga
2. Tecnologías de front-end exclusivamente client-side
 - 2.1. HTML, Javascript y CSS
 - 2.2. AJAX
 - 2.3. Websocket
 - 2.4. Frameworks Javascript, como Angular o Backbone, 1 representativo por paradigma
 - 2.5. Frameworks y/o técnicas para animaciones exclusivamente con Javascript, HTML y CSS, implementación de efectos visuales
 - 2.6. Estrategias de comunicación con servicios REST de backend
 - 2.7. Construcción de widgets con capacidad de comunicación con el servidor.
 - 2.8. Reproducción multimedia y streaming con HTML5
 - 2.9. Upload y download de archivos con AJAX
3. Tecnologías de front-end server y client side
 - 3.1. Tecnologías estándares y representativas de los paradigmas vigentes que impliquen la programación de componentes client y server side como:
 - 3.1.1. Java Server Faces
 - 3.1.2. ASP .NET
 - 3.1.3. Ruby on Rails
 - 3.1.4. Django
4. Procesos de desarrollo de front-end
 - 4.1. Automatización de flujo de trabajo, por ejemplo con Grunt
 - 4.2. Test Driven Development, por ejemplo con Jasmin
 - 4.3. Minimización
 - 4.4. Ofuscación
 - 4.5. Versionado de scripts
 - 4.6. Análisis estático de código fuente y medidas de calidad
5. Performance en el front-end
 - 5.1. Performance en el uso de AJAX
 - 5.2. Creación de aplicaciones web responsive
 - 5.3. Identificar el Payload inicial
 - 5.4. Cargar scripts evitando bloqueos
 - 5.5. Acoplar scripts asíncronos
 - 5.6. Ubicación de scripts inline
 - 5.7. Escribir código Javascript eficiente
 - 5.8. Escalar con Comet
 - 5.9. Compresión de contenido
 - 5.10. Optimización de imágenes
 - 5.11. Herramientas y técnicas para evaluar el rendimiento del front-end de aplicaciones web.
6. Seguridad en el front-end
 - 6.1. Análisis de los tipos de vulnerabilidades de las aplicaciones web
 - 6.2. Análisis de los tipos de vulnerabilidades de los navegadores web
 - 6.3. Estudio de las mejores prácticas en el diseño arquitectónico del front-end en aplicaciones web considerando aspectos de seguridad

VII. ESTRATEGIAS METODOLÓGICAS

1. Exposición de fundamentos teóricos en clase por parte del profesor.
2. Trabajos de investigación y evaluación de herramientas y técnicas a desarrollarse fuera del horario de clases.
3. Trabajos de diseño y programación de interfaces de sistemas web a desarrollarse fuera del horario de clases.
4. Evaluación de trabajos en laboratorio, validaciones de caja negra y análisis de código fuente.

A. Medios Auxiliares

1. Pizarras acrílicas.



2. Marcadores.
3. Borrador de pizarra acrílica.
4. Computadoras.
5. Proyectores multimedia.
6. Parlantes para multimedia.
7. Plataforma virtual "EDUCA".
8. Sala de laboratorio equipada para las prácticas.
 - 8.1. Computadoras en red.
 - 8.2. Sistemas operativos Linux, Windows.
 - 8.3. Acceso a internet.

VIII. EVALUACIÓN

1. Para evaluar la asignatura se tienen en cuenta lo siguiente:
 1. Exámenes parciales de teoría con un % asignado.
 2. Examen final de teoría con un % asignado.
 3. Algunos trabajos prácticos tienen un % asignado.
2. Las calificaciones se basan en el reglamento de la Universidad.
3. Es imprescindible la entrega de todas las prácticas para poder calcular la nota de prácticas.

IX. BIBLIOGRAFÍA

A. Básica

- Steve Sounders. Even Faster Web Sites. O'Reilly (2014).
- Wade Alcorn, Christian Fritchot. The Browser Hacker's Handbook. Wiley (2014).
- Ilya Grigorik. High Performance Browser Networking. O'Reilly (2014).
- Subbu Allamaraju. Restful Web Services Cookbook. O'Reilly (2010).
- Kyle Simpson. Async & Performance JS. O'Reilly (2015).

B. Complementaria

- Kyle Simpson. this & Object Prototypes. O'Reilly (2015).
- Kyle Simpson. scope & closure. O'Reilly (2015).
- Nicholas C. Zakas. High Performance JavaScript. O'Reilly (2010).
- Leonard Richardson. RESTful Web APIs. O'Reilly (2015).
- Jim Webber. REST in Practice: Hypermedia and Systems Architecture. O'Reilly (2010).

C. Enlaces Web

- <http://www.w3.org/TR/html5/>
- <https://tools.ietf.org/html/rfc7540>
- <http://www.w3schools.com/>



