

4122 - DESARROLLO DE SOFTWARE PARA LA WEB

I - Datos de identificación de la asignatura

Carrera:	Licenciatura en Análisis de Sistemas		
Código:	4122	Plan:	2024
Denominación:	Desarrollo de software para la web		
Área:	Énfasis propio de la carrera		
Año:	Segundo		
Horas con acompañamiento docente (HTD), semanal			4
Horas de Trabajo Independiente del estudiante (HTI), semanal			6
Horas semanales (HS)			10
Cantidad de sesiones			32
Total Horas de Trabajo con el docente (THTD)			128
THD teóricas	0	THD prácticas	128
Total de Horas de Trabajo Independiente del estudiante (THTI)			192
Total Horas Académicas (THA)			320
Crédito académico (CA)			12,8
Pre-requisito:	Fundamentos de algorítmica y programación		

II - Fundamentación

La asignatura tiene como objetivo principal brindar a los estudiantes los conocimientos y habilidades necesarias para diseñar, desarrollar e implementar aplicaciones web de alta calidad.

En un entorno donde la popularidad de Internet ha llevado a un aumento significativo en el uso de aplicaciones web, es fundamental comprender las demandas y los desafíos asociados con el desarrollo de estas aplicaciones. La exposición a una amplia variedad de usuarios de diferentes culturas y la necesidad de adaptarse constantemente a nuevos requerimientos son factores clave que deben abordarse de manera efectiva.

En esta asignatura, se presentan y se aplican procesos de desarrollo específicos para promover el desarrollo de aplicaciones web con características particulares. Se enfatiza la utilización de metodologías de ingeniería web para obtener aplicaciones de mayor calidad, reduciendo tanto los costos como el esfuerzo necesario para su desarrollo y mantenimiento en comparación con enfoques ad-hoc.

Los estudiantes adquirirán un sólido entendimiento de los conceptos de diseño y programación de aplicaciones web, aprendiendo a utilizar las herramientas y tecnologías utilizadas tanto en el lado del cliente como en el lado del servidor. A través de la práctica, diseñarán e implementarán un sitio web utilizando los estándares actuales y las mejores prácticas de la industria.

La asignatura combina una base teórica sólida con actividades prácticas, lo que permite a los estudiantes aplicar los conceptos aprendidos en situaciones reales. Se fomenta el desarrollo de habilidades de resolución de problemas, trabajo en equipo y creatividad, elementos clave para el éxito en el desarrollo de aplicaciones web.

III - Competencias a desarrollar

Competencias genéricas

1. Analizar, delinear, gestionar, desarrollar, implementar y evaluar proyectos con innovación y creatividad.
2. Conocer y saber aplicar técnicas y herramientas actualizadas en sus áreas de competencia.
3. Diseñar, programar, ejecutar, analizar e interpretar resultados de pruebas realizadas en sus áreas de competencia.
4. Integrar equipos multidisciplinarios y realizar trabajos colaborativos.

Competencias específicas

1. Comprender y explicar con claridad los protocolos fundamentales de Internet (TCP/IP, HTTP, HTTPS, DNS), para interpretar su funcionamiento y su uso en la comunicación y transferencia de datos a través de la web.
2. Desarrollar con habilidad aplicaciones web del lado del cliente, utilizando HTML para estructura, CSS para estilo, y JavaScript junto con bibliotecas modernas, para crear interfaces interactivas, dinámicas y funcionales.
3. Implementar de manera efectiva aplicaciones web del lado del servidor, utilizando un lenguaje de back-end y su framework asociado, para gestionar la lógica de negocio, la conexión con bases de datos y la interacción con el cliente.
4. Integrar de forma coherente las tecnologías de front-end y back-end, para desarrollar aplicaciones web completas con persistencia de datos, empleando herramientas actuales que optimicen la eficiencia y calidad del desarrollo.
5. Identificar y aplicar críticamente patrones arquitectónicos web (MVC, capas, microservicios), para seleccionar el enfoque más adecuado según los requerimientos y contexto del proyecto.
6. Aplicar con responsabilidad prácticas de seguridad en el desarrollo web, protegiendo contra vulnerabilidades comunes (SQLi, XSS, CSRF), garantizando autenticación, autorización y manejo seguro de datos sensibles.
7. Detectar y corregir con precisión errores sintácticos y lógicos en aplicaciones web, utilizando herramientas de depuración y técnicas de análisis, para mejorar el rendimiento, la calidad y la estabilidad del software.
8. Aplicar de forma organizada las mejores prácticas de despliegue web en entornos de producción, incluyendo configuración de servidores, optimización de rendimiento y monitoreo, para asegurar la disponibilidad, escalabilidad y eficiencia del sistema.

IV - Cuerpo de conocimientos

Unidad 1: Comprensión operativa de los protocolos que permiten la Internet.

Contenidos:

- Finalidad del cliente (front-end)
- Propósito del servidor (backend)
- Interacción de un servidor de Internet con un cliente

Unidad 2: Tecnologías de desarrollo de aplicaciones web del lado del cliente (Front-end), incluidas HTML, CSS, JavaScript y bibliotecas CSS y JavaScript.

Contenidos:

- Lado del cliente (Front-end)
- Introducción a la programación del lado del cliente
- HTML (Fundamentos del lenguaje de marcado HTML, Estructura básica de un documento HTML, Sintaxis de etiquetas y etiquetas básicas, Etiquetas para texto, listas, imágenes, enlaces, tablas y formularios, Etiquetas semánticas, Creación de páginas estáticas en HTML).
- CSS (Fundamentos del lenguaje de estilos CSS, Conceptos básicos sobre diseños de interfaces de usuarios, Sintaxis de CSS, Selectores y propiedades básicas y avanzadas, Patrones de layouts en css, Flexbox y CSS Grid, Pseudo clases y pseudo elementos, Estilización de páginas web estáticas utilizando CSS, Librerías de css).
- JavaScript (Fundamentos del lenguaje de programación JavaScript, Tipos de datos, variables, funciones y objetos, Notación de objetos JavaScript (JSON), Librerías de JavaScript, Modelo de objeto de documento (DOM) y su manipulación con JavaScript, Consumo de APIs REST)
- Desarrollo colaborativo de software (Conceptos básicos sobre intérpretes de comandos e interfaces de líneas de comandos, Sistemas de control de versiones y sus fundamentos, Plataformas para el desarrollo de software colaborativo)

Unidad 3: Tecnologías de desarrollo de aplicaciones web del lado del servidor (back-end) utilizando un lenguaje de programación de back-end (ejemplo, Java/Spring, Node/Express, u otro).

Contenidos:

- Lado del servidor (back-end)
- Lenguaje del lado del servidor (es decir, Java/Spring, Node/Express, Python/Django, u otro)

Unidad 4: Creación de una aplicación web funcional.

Contenidos:

- Desarrollo del servidor en conjunto con el cliente.
- Habilidades para crear una aplicación web usando front-end y back-end, incorporando la funcionalidad de la base de datos (CRUD)
- Conocer motores de renderización en el desarrollo web.
- Capa de Modelo y Persistencia de Datos
- Implementación de persistencia de datos.
- Manejo de Requisiciones Web
- Componentes de la capa de Control.
- Mapear atributos de objetos a relaciones de cardinalidad en una base de datos relacional.

Unidad 5: Patrones de arquitectura de aplicaciones web

Contenidos:

- Patrones de diseño (MVC, MVVM, u otro)
- Ventajas de las desventajas de los diferentes patrones de diseño.
- Arquitectura RESTful, y herramientas de prueba.

Unidad 6: Seguridad de un sitio web.

Contenidos:

- Autorización y Autenticación
- Cuando usar la seguridad y por qué

Unidad 7: Errores sintácticos y lógicos.

Contenidos:

- Proceso de depuración de una aplicación web
- Diferencia entre error lógico y de sintaxis.
- Depuración de sitios web
- Pruebas unitarias, pruebas END-2-END y TDD (Test driven development)

Unidad 8: Implementación de un sitio web en un hosting.

Contenidos:

- Servidor de hospedaje archivos
- Despliegue de sitio web y archivos.
- Configuración del hosting
- Paquete de aplicaciones para la web: CMS, comercio electrónico, blogs, sitio web institucional, landing page, etc.
- Tendencias tecnológicas para la web.

V- Estrategias didácticas a ser implementadas en el proceso de enseñanza aprendizaje. (abarcando actividades de formación e investigación)

La asignatura consolida la formación experimental y profesionalizante de los y las estudiantes ubicándolos/as en un entorno de trabajo similar al real y cotidiano.

Al comenzar la cursada, se realiza una evaluación diagnóstica sobre los conocimientos iniciales de los y las estudiantes, a través de una encuesta en línea. En esta encuesta, además se releva otra información de interés como ser su situación laboral, aspectos de conectividad y su situación respecto de las restantes asignaturas.

La teoría y la práctica, dictadas semanalmente, se encuentran estrechamente vinculadas. La teoría trabaja lineamientos conceptuales que se aplican en los desarrollos prácticos. Se organizan actividades planificadas para los estudiantes, en las que se proponen “desafíos” que deben convertirse en “ideas de proyecto” y posteriormente en potenciales desarrollos. Se trata que el estudiante logre abstraer una serie de pasos que respondan a una metodología clásica de investigación y lo ayuden a aprender en forma continua y autónoma, con capacidad de planificar este aprendizaje:

- búsqueda de bibliografía actualizada sobre el tema;
- discusión de alternativas tecnológicas para resolver el tema propuesto;
- abstracción del desafío y/o problema como una “idea de proyecto a resolver”;
- expresión sintética de la especificación del proyecto, con recursos humanos requeridos y plan de tareas;
- implementación y defensa, oral y escrita, de la solución al desafío.

En las instancias prácticas los estudiantes resuelven las actividades propuestas utilizando herramientas de soporte y desarrollo típicas en la comunidad de software libre, entre las cuales se menciona: un servidor web, un motor de base de datos, un sistema de versionado de código, ambientes integrados de desarrollo (IDE - Integrated Development Environment), herramientas de debugging y validación.

Se discuten casos en los que el producto informático que se genera tiene un impacto económico/social o ambiental. En la evaluación de estos “análisis de casos” las respuestas del estudiante se reflejan en una planilla detallada compartida por los responsables de teoría y práctica para considerarlas en las evaluaciones de los estudiantes o darles una devolución específica que puede ser una “recomendación” o material de estudio complementario.

Se trabaja en la capacidad para interpretar la evolución de la Informática con una visión de las tendencias tecnológicas futuras, analizando tendencias sobre distintas tecnologías utilizadas y/o a utilizar y se “desafía” a los estudiantes a presentar posible evolución de la solución para ese tipo de problema y en qué podría mejorarse la solución/soluciones actuales. Los estudiantes deben acompañar su trabajo con un informe final. Para ello, se realiza un taller en donde se indican las pautas para su correcta redacción. En este taller se trabajan aspectos de redacción, reglas de estilo y formas de

incluir citas bibliográficas. También se dictan los lineamientos para la presentación oral de su trabajo.

Se trabaja con los siguientes recursos:

- guías, diapositivas, videos, libros, tutoriales y especificaciones de estándares a utilizar;
- demostraciones de usos de herramientas con ejemplos en vivo;
- herramientas de versionado de código, servidor web, IDE de desarrollo, base de datos, API y frameworks de desarrollo;

VI - Estrategias de evaluación.

A modo de ejercitación y evaluación se plantean, a lo largo de la cursada, entregas de ejercicios que los estudiantes deben desarrollar y entregar en las prácticas y teorías.

Se entrega un trabajo en distintas etapas. Se presenta una producción que los estudiantes defienden en forma de coloquio oral. En esa instancia, además, se indaga sobre los conceptos teóricos vistos en esta etapa del desarrollo. Esto es un requisito para la aprobación de la cursada. Estas entregas son de seguimiento y de evaluación con calificación.

Toda evaluación realizada a los estudiantes queda plasmada en una planilla muy bien detallada, donde se indican los resultados de las diferentes evaluaciones realizada a los mismos: capacidad del estudiante para desarrollar su aprendizaje, claridad de las presentaciones realizadas, forma de organización y expresión en las diferentes instancias de evaluación oral, formulación de la solución de los diferentes desafíos en forma autónoma, entre otros.

Para la obtención de calificaciones parciales y finales se tendrá en cuenta el Reglamento Académico de la universidad.

VII - Actividades de extensión y de responsabilidad social universitaria.

Rige de acuerdo al reglamento de la Universidad y el reglamento interno de la facultad.

VIII - Fuentes bibliográficas

Básica

- Pro Git. Scott Chacon & Ben Straub. Disponible en: <https://git-scm.com/book/en/v2>.
- CSS practica/ CSS Instant result: Cascading Style Sheets For Web. York, Richard.
- Introducción a Javascript. Javier Eguiluz. Disponible en <http://librosweb.es/libro/javascript/>.
- Introducción a CSS. Javier Eguiluz. Disponible en <http://librosweb.es/libro/css/>.
- La seguridad en aplicaciones web – Top Ten https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.
- RESTful Web Services. Leonard Richardson & Sam Rub, disponible en <http://restfulwebapis.org/rws.html>.
- Guía oficial Vue.js: <https://es.vuejs.org/v2/guide>.
- Perry, B.W. (2004). Java Servlet & JSP. Cookbook. O'Reilly Media: Estados Unidos de América.
- Bryant, J. (2012). Java EE7 for Absolut Beginners. Apress
- Linwood, J. (2ª Ed.).Beginners Hibernate. Apress.

- Fernando, B. (2020) Spring Boot – Acelere la creación de microservicios.
- Martin Fowler. "Patterns of Enterprise Application Architecture" Addison-Wesley, 2002.
- Fernando, B. (2020) Spring Boot – Acelere la creación de microservicios.

Complementaria

- Dino Esposito, Andrea Saltarello. "Microsoft .NET: Architecting Applications for the Enterprise". 2ª Edición. Microsoft Press, 2014.
- Leonard Richardson, Mike Amundsen. "RESTful Web APIs". O'Reilly. 2013.
- Christian Bauer, Gavin King, Gary Gregory. "Java Persistence with Hibernate". 2ª Edición. 2014.
- Emmit Scott. "SPA Design and Architecture". Manning Publications. 2015.
- Willie Wheeler, Joshua White. "Spring in Practice". Manning Publications. 2013
- Jennifer Niederst Robbins. "HTML5 Pocket Reference". 5ª Edición. O'Reilly, 2013.
- Eric A. Meyer. "CSS Pocket Reference". O'Reilly, 2011.
- Sai Srinivas Sriparasa. "JavaScript and JSON Essentials". Packt Publishing. 2013.
- Eric Evans. "Domain-Driven Design". Addison-Wesley, 2003.