

FACULTAD/ESCUELA DE BARBERI DE INGENIERÍA, DISEÑO Y CIENCIAS APLICADAS

DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS INTELIGENTES

Información de la asignatura

Nombre de la asignatura	Computación en Internet II
Código de la asignatura	09769 - TIC
Periodo Académico	202510
Nrc	12346
Grupo	005
Programas/Semestres	SIS 06
Intensidad horaria	2
Intensidad Semanal	4
Créditos	3
Docente(s)	Domiciano Rincón Niño

Introducción o presentación general del curso

La Computación en Internet abarca todo un conjunto de áreas relacionadas con el contexto y problemas que ha introducido la Internet en el diseño y desarrollo de software, tales como: comunicación entre computadores, conceptos y protocolos de red de alto nivel, estándares y tecnologías Web, siendo en general la base para el entendimiento y el diseño de sistemas complejos de software distribuido.

En este curso se continúa sobre los fundamentos vistos en Computación en Internet I, centrándose en el desarrollo de aplicaciones web, entendiendo principios de arquitectura, implementando patrones de diseño e integrando tecnologías frontend y backend para la creación de un producto funcional. Para ello se combinará tanto teoría como práctica con la realización de talleres. Estos talleres retoman buenas prácticas de la industria, reforzando la comprensión del estudiante de los conceptos y su aplicación a problemas del mundo real, incluyendo modelado, evaluación de alternativas de diseño basado en tecnologías representativas y existentes en el mercado, y análisis y desarrollo de software usando diversos protocolos de comunicación de datos.

Formación en competencias

- **SO-1. Solución de problemas:** Identificar, formular y resolver problemas complejos de ingeniería aplicando pensamiento crítico y principios de las ciencias, las matemáticas, la ingeniería y, en particular, de las Ciencias de la Computación y de la Ingeniería de Software.
- **SO-2. Diseño de ingeniería:** Diseñar soluciones y procesos basados en software que satisfagan necesidades específicas y generen valor a sus usuarios, considerando la salud pública, la seguridad y el bienestar de las personas, así como factores globales, culturales, sociales, ambientales y económicos.

Objetivo general de aprendizaje

Al finalizar el curso, en un entorno productivo empresarial, industrial, propio, o como empleado, los estudiantes estarán en capacidad de diseñar, implementar, integrar y probar las funcionalidades básicas que se requieren para construir aplicaciones Web interactivas y

transaccionales sencillas, soportando despliegue independiente entre el Frontend y el Backend y asegurando el uso de principios para la calidad de software.

(RA1) Explicar las arquitecturas usadas para el desarrollo de aplicaciones Web que se soportan en los protocolos de comunicaciones de alto nivel.

(RA2) Diseñar y construir funcionalidades back-end que requieren las aplicaciones Web interactivas y transaccionales sencillas, siguiendo los lineamientos arquitectónicos del framework de desarrollo.

(RA3) Diseñar, construir e integrar funcionalidades front-end que requieren las aplicaciones Web interactivas y transaccionales sencillas, con base en los conceptos del protocolo HTTP y sus extensiones.

(RA4) Utilizar frameworks, librerías y herramientas para el desarrollo, pruebas y preparación para el despliegue que dan solución o soporte a funcionalidades comunes y repetitivas en los proyectos Web.

Objetivos terminales - Resultados de aprendizajes

Resultado de aprendizaje del curso o asignatura	Competencia en formación	Resultado de aprendizaje de la competencia de egreso al que se contribuye
(RA1) Explicar las arquitecturas usadas para el desarrollo de aplicaciones Web que se soportan en los protocolos de comunicaciones de alto nivel.	SO2-Diseño	SO2-PI2-T Diseño de alto nivel
(RA2) Diseñar y construir funcionalidades back-end que requieren las aplicaciones Web interactivas y transaccionales sencillas, siguiendo los lineamientos arquitectónicos del framework de desarrollo.	SO1-Solución de problemas; SO2-Diseño	SO1-PI2-T Solución de problemas (ingeniería y ciencias); SO2-PI1-A Diseño detallado
(RA3) Diseñar, construir e integrar funcionalidades front-end que requieren las aplicaciones Web interactivas y transaccionales sencillas, con base en los conceptos del protocolo HTTP y sus extensiones.	SO1-Solución de problemas; SO2-Diseño	SO1-PI2-T Solución de problemas (ingeniería y ciencias); SO2-PI1-A Diseño detallado
(RA4) Utilizar frameworks, librerías y herramientas para el desarrollo, pruebas y preparación para el despliegue que dan solución o soporte a funcionalidades comunes y repetitivas en los proyectos Web.	SO1-Solución de problemas	SO1-PI2-T Solución de problemas (ingeniería y ciencias)

Unidades de aprendizaje

Unidad 1: Fundamentos Generales

- Identificar los protocolos y servicios de alto nivel en las distintas funcionalidades de una aplicación web: http, smtp. Complementariamente, scp, ssh
- Explicar y aplicar los principales estilos arquitectónicos de software, incluyendo cliente/servidor, multicapa y microservicios, según los requisitos de un proyecto determinado.

Unidad 2: Introducción al Spring Framework

- Identificar los conceptos generales de Spring
- Explicar los tipos de aplicaciones empresariales para los que el framework es aplicable
- Explicar y aplicar el concepto de inyección de dependencias e inversión de control
- Aplicar la autoconfiguración en el framework
- Diferenciar los conceptos de service, controller, repository

Unidad 3: Capa de integración a Base de datos

- Explicar y usar los conceptos asociados a ORM (Object-Relational Mapping) y cómo se aplican en el desarrollo de aplicaciones de backend.
- Aplicar JPA (Java Persistence API) y explicar las ventajas que presenta sobre JDBC (Java Database Connectivity), como estrategias para acceder a bases de datos relacionales en aplicaciones de Java.
- Explicar los conceptos fundamentales de Spring Data y cómo se utilizan en el desarrollo de aplicaciones de backend.
- Explicar las diferencias y similitudes entre DAO y DTO, incluyendo su propósito, función y responsabilidades en la arquitectura de una aplicación.

Unidad 4: Patrón y Arquitectura Modelo Vista Control

- Explicar los tres conceptos principales del patrón MVC y su papel en la separación de responsabilidades y la organización de la lógica de la aplicación.
- Aplicar conceptos de desarrollo en el framework Spring Boot.
- Explicar los conceptos fundamentales de la programación en el lado del servidor.
- Utilizar Postman de manera efectiva para probar, documentar y colaborar en el desarrollo de APIs en la capa backend.
- Explicar y usar los conceptos fundamentales de la programación en el lado del cliente (ReactJS).

Unidad 5: Diseño y desarrollo de Backend

- Explicar los conceptos fundamentales de la capa backend, incluyendo la gestión de datos, la lógica de negocio y la integración con la capa frontend.
- Implementar pruebas unitarias y de integración para garantizar la calidad del código y la funcionalidad de las aplicaciones backend.
- Documentación de API REST usando Swagger.
- Identificar el concepto y las características de la interfaz REST.
- Describir, comprender e implementar el patrón de diseño Business Delegate.
- Desarrollar, diseñar y utilizar APIs RESTful para proporcionar una interfaz de programación de aplicaciones segura, eficiente y escalable a las aplicaciones frontend y móviles.

Unidad 6: Introducción React

- Diseñar e implementar componentes de React reutilizables utilizando JSX y la sintaxis de JavaScript moderna.
- Utilizar el estado y las propiedades de React para controlar el flujo de datos en sus aplicaciones, permitiendo actualizaciones en tiempo real de la interfaz de usuario.
- Utilizar React Router para crear aplicaciones de una sola página (Single-Page Applications), proporcionando una experiencia de usuario fluida y sin interrupciones.
- Explicar y usar y la sintaxis de los Hooks más comunes de React, como useState, useEffect, useContext, useMemo y useCallback, así como su aplicación en el desarrollo de componentes.
- Realizar la integración con backend consumiendo las apis creadas previamente.

Metodologías de aprendizajes

Antes de la clase, el estudiante deberá:

- Preparar los temas que asigne el profesor. Bajo el esquema de trabajo de este curso, preparar un tema significa ESTUDIARLO, es decir hacer una lectura crítica (análisis y síntesis) del tema asignado, indagar sobre los aspectos desconocidos, resolver las preguntas y los ejercicios planteados.
- Preparar los informes de las actividades realizadas durante la semana, estas actividades pueden ser de investigación sobre nuevas tecnologías o sobre el desarrollo del proyecto.

Durante la clase, el estudiante deberá:

- Participar en las discusiones del tema y en la formulación de preguntas orientadas a resolver las dudas que hayan surgido al realizar las actividades mencionadas y las propuestas por el profesor; para esto, el estudiante deberá llevar a clase en formato digital, las soluciones planteadas a los ejercicios propuestos como preparación del tema con el fin de discutir estas propuestas con todo el grupo.
- Trabajar en la solución de los problemas de aplicación que se propongan.

Después de la clase:

- Realizar los trabajos asignados por el profesor para poner en práctica y afianzar conceptos.
- Establecer las relaciones entre los temas tratados en la clase y el conocimiento previamente adquirido en el curso y otros cursos, así como con su futuro desempeño profesional como Ingeniero de Sistemas.

Evaluación de aprendizajes

Código evaluación	Mecanismo o actividad evaluativa	Porcentaje de la nota final	Relación con objetivos terminales - resultado de aprendizaje del curso	Relación con el resultado de aprendizaje de la competencia de egreso
T.Backend	T.Backend	20	RA2	SO1-PI2, SO2-PI2
T.Frontend	T.Frontend	20	RA3	SO1-PI2, SO2-PI2
T.MVC	T.MVC	15	RA1	SO2-PI2
T.Persist	T.Persist	15	RA4	SO1-PI2
T.Spring	T.Spring	10	RA4	SO1-PI2
Tareas	Tareas	20	RA1	SO2-PI2

Recursos de apoyo



No es necesario comprar ningún libro para el curso. Todo el material necesario para el curso será suministrado a través de Intu, como los materiales bibliográficos, guías de clases, videos, blogs, cursos en línea y otros.

Algunos libros, enlaces y herramientas relevantes:

- The Scrum Guide <http://www.scrumguides.org/>
- James F. Kurose y Keith W. Ross. Computer Networking: A Top-Down Approach. Pearson, 2020.
- Doug Lowe. Networking all-in-one for Dummies, séptima edición. Wiley, 2018.
- Gordon Davies. Networking Fundamentals. Packt Publishing, 2019.