

## FACULTAD/ESCUELA DE BARBERI DE INGENIERÍA, DISEÑO Y CIENCIAS APLICADAS DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS INTELIGENTES

### *Información de la asignatura*

<b>Nombre de la asignatura</b>	Computación en Internet II
<b>Código de la asignatura</b>	09769 - TIC
<b>Periodo Académico</b>	202610
<b>Nrc</b>	11151
<b>Grupo</b>	003
<b>Programas/Semestres</b>	SIS 06
<b>Intensidad horaria</b>	2
<b>Intensidad Semanal</b>	4
<b>Créditos</b>	3
<b>Docente(s)</b>	Kevin David Rodriguez Belalcazar

### *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***

<b>Resultado de aprendizaje del curso o asignatura</b>	<b>Competencia en formación</b>	<b>Resultado de aprendizaje de la competencia de egreso al que se contribuye</b>
(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.

## **Composición de los talleres**

Las calificaciones del curso se componen de cinco (5) talleres que contribuyen progresivamente al desarrollo de un proyecto final. Cada taller tiene dos componentes: uno individual (60%) y otro grupal (40%).

El componente grupal corresponde a la implementación técnica que aportará directamente al proyecto final del curso y será desarrollado en equipo.

El componente individual consistirá en un examen diseñado para evaluar las competencias específicas de cada unidad del curso, y será realizado de forma individual.

## **Acuerdos sobre el uso de Inteligencia Artificial Generativa en la asignatura**

La Inteligencia Artificial Generativa (IAG) es un tipo de Inteligencia Artificial (IA) capaz de crear contenido nuevo, como: textos, imágenes, música o código, a partir de modelos entrenados con grandes volúmenes de datos. Se centra en la creación de nuevo contenido, a partir de datos existentes. A diferencia de otras formas de IA que analizan o clasifican información, la IAG produce materiales originales en respuesta a indicaciones específicas o prompts (Lim et al., 2023).

En coherencia con el Manifiesto sobre IA y educación de la Universidad Icesi, este curso promueve un uso ético, crítico y transparente de las tecnologías de IAG, permitiendo su aplicación en las actividades formativas y evaluativas de acuerdo con los niveles de uso que se especifican en la siguiente tabla. Se debe tener en cuenta que la ubicación de cada mecanismo o actividad en la tabla se refiere al máximo nivel posible de uso de IAG que se espera de los estudiantes en dicho mecanismo o actividad, no obstante, se pueden emplear también los niveles anteriores según se especifique en las guías de aprendizaje, consignas o enunciados.

Durante el desarrollo de la asignatura, cuando se le indique, el estudiante deberá aplicar y declarar el nivel de uso de la IAG según la escala declarada, verificando la calidad, veracidad y pertinencia del contenido generado. Este acuerdo tiene como propósito fortalecer una relación formativa, reflexiva y dialógica con la IAG, orientada al desarrollo de capacidades creativas, investigativas y colaborativas.

\*\*\*

### **Niveles de IA y su uso en el curso**

#### **1. No IAG**

*La actividad se desarrolla en entornos controlados donde no se permite ningún uso de IAG. Los estudiantes pueden emplear solamente sus propios conocimientos y habilidades.*

El componente individual de cada taller será sin IA

**Actividades en esta categoría: E. Backend, E.Frontend, E.MVC, E.Persist, E.Spring**

#### **2. Planificación con IAG**

*Los estudiantes pueden usar IAG para apoyar la fase de planificación o generación de ideas, sin embargo, el producto final debe ser de su propia autoría y no puede contener contenido creado con IAG.*

**Actividades en esta categoría: T. Persist, en la creación del modelo de datos de la aplicación**

#### **3. Colaboración con IAG**

*Los estudiantes pueden apoyarse en la IAG para completar tareas o desarrollar entregables asociados a la actividad, aprovechando las capacidades de estas herramientas para mejorar los productos. Asimismo, se espera que los estudiantes lleven un registro de sus interacciones con la IAG y estén en la capacidad de modificar los resultados generados, demostrando comprensión y dominio conceptual. Este registro debe contener tanto el contenido de autoría propia proporcionado a la IAG como los prompts empleados.*

**Actividades en esta categoría: T. MVC y T. Spring, Tareas**

#### **4. IAG completa**

*Los estudiantes pueden usar estratégicamente la IAG para resolver tareas complejas en cualquier etapa de la actividad. Los estudiantes deben hacer un uso crítico y reflexivo de estas herramientas justificando su aplicación para alcanzar los resultados de aprendizaje. Cualquier contenido creado con IAG debe ser citado. No aplica*

**Actividades en esta categoría: T. Backend y T. Frontend**

#### **5. Exploración con IAG**

*Los estudiantes tienen libertad para experimentar e integrar tecnologías IAG emergentes en el diseño de propuestas para solucionar problemas complejos. Los estudiantes asumen un rol investigador o innovador colaborando activamente con docentes y expertos y están en la capacidad de evaluar el impacto ético, técnico y académico de sus creaciones*

*(Sin presencia en el curso)*

### **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
E.Backend	E.Backend	12	RA2	SO1-PI2, SO2-PI2
E.Frontend	E.Frontend	12	RA3	SO1-PI2, SO2-PI2
E.MVC	E.MVC	12	RA1	SO2-PI2
E.Persist	E.Persist	12	RA4	SO1-PI2
E.Spring	E.Spring	6	RA4	SO1-PI2
T.Backend	T.Backend	8	RA2	SO1-PI2, SO2-PI2
T.Frontend	T.Frontend	8	RA3	SO1-PI2, SO2-PI2
T.MVC	T.MVC	8	RA1	SO2-PI2
T.Persist	T.Persist	8	RA4	SO1-PI2
T.Spring	T.Spring	4	RA4	SO1-PI2
Tareas	Tareas	10	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.