

**FACULTAD DE INGENIERÍA, DISEÑO Y CIENCIAS APLICADAS /  
ESCUELA DE TECNOLOGÍA, DISEÑO E INNOVACIÓN  
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS INTELIGENTES**

***Información de la asignatura***

<b>Nombre de la asignatura</b>	Computación en Internet III
<b>Código de la asignatura</b>	09770
<b>Período académico</b>	2024-I
<b>Intensidad horaria<sup>1</sup></b>	4 horas / 3 créditos
<b>Intensidad semanal<sup>1</sup></b>	4 horas /Semana
<b>Créditos<sup>1</sup></b>	3
<b>Apoyo a Currículo Central</b>	N/A
<b>Profesor(es)</b>	Leonardo Bustamante

***Introducción o presentación general del curso***

El propósito de este curso es consolidar y fortalecer en el estudiante el desarrollo de habilidades en programación full stack. Esto se logrará mediante la creación y perfeccionamiento de diversas funcionalidades, abarcando tanto el front-end como el back-end de aplicaciones web. Se emplearán herramientas y frameworks comúnmente utilizados en la industria global del software para proporcionar una experiencia práctica y alineada con las demandas actuales del sector.

***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.

---

<sup>1</sup> La intensidad horaria, intensidad semanal y créditos están alineados con la “Política de definición de créditos académicos de la Universidad Icesi” – Resolución N°. 80 de Junta Directiva del 22 de septiembre de 2019.

- **SO7 - Aprendizaje autónomo para la vida:** Aprender de forma autónoma y aplicar nuevos conocimientos según sea necesario, utilizando estrategias de aprendizaje adecuadas.
- **SO9 - Despliegue y operación:** Desplegar y poner en funcionamiento soluciones de software desarrolladas.

### ***Objetivo general (meta 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 avanzadas que se requieren para construir aplicaciones Web interactivas y transaccionales, soportando despliegue independiente entre el Frontend y el Backend, asegurando el uso de principios para la calidad de software, a través de uso de herramientas y frameworks que se encuentran a la vanguardia de la tecnología.

### ***Objetivos terminales (resultados de aprendizaje de la asignatura)***

<b>Objetivos Terminales</b>	<b>Competencia Egreso*</b>	<b>RA <math>\equiv</math> PI*</b>
(OT1) Seleccionar arquitectura, herramientas y frameworks a utilizar para el desarrollo y pruebas de una aplicación web con requerimientos específicos, de entre los comúnmente utilizados en la industria.	SO2-Diseño  SO7- Aprendizaje autónomo para la vida	SO2-PI2 Diseño de alto nivel  SO7-P1 Aprendizaje permanente SO7-PI2 Analizar información relevante SO7-PI3 Adquirir nuevos conocimientos
(OT2) Diseñar, implementar, compilar, probar y desplegar interfaces de usuario web (Frontend), considerando principios de usabilidad y experiencia de usuario.	SO2-Diseño  SO9-Despliegue	SO2-PI1 Diseño detallado  SO9-PI1 Despliegue de software
(OT3) Diseñar, implementar, probar, compilar y desplegar servicios Backend, usando el conjunto de herramientas y frameworks seleccionados, incluyendo mecanismos de persistencia según las necesidades del proyecto (SQL, NoSQL)	SO1-Solución de problemas  SO9-Despliegue	SO1-PI2 Solución de problemas (ingeniería y ciencias)  SO9-PI1 Despliegue de software
(OT4) Integrar pruebas automatizadas para asegurar la calidad tanto del Frontend como del Backend.	SO9-Despliegue	SO9-Despliegue

\*NOTA: La descripción completa de los SOs y de los PIs puede consultarse haciendo click en los siguientes enlaces: [Student Outcomes](#) y [Performance Indicators](#)

## ***Unidades de aprendizaje***

### **Unidad 1: Introducción al curso**

- Identificar las capas para el desarrollo de las aplicaciones web.
- Diferenciar entre un desarrollador del lado del cliente (frontend) y desarrollador del lado del servidor (backend)
- Explicar el concepto de entrega continua, integración continua.
- Explicar el concepto automatización de la infraestructura y pruebas.
- Listar los lenguajes de programación, librerías y frameworks más utilizados en la programación web.
- Identificar los componentes en los stacks de desarrollo web más utilizados.

### **Unidad 2: Programación del lado del servidor – NodeJS**

- Explicar los conceptos de NodeJS y cuál es la diferencia con servidores http tradicionales
- Identificar los tipos de aplicaciones para el uso de NodeJS
- Administrar módulos en NodeJS usando el manejador de paquetes NPM
- Implementar aplicaciones utilizando el framework Express
- Definir los requerimientos necesarios para el desarrollo de aplicaciones con acceso a bases de datos NoSQL
- Implementar servicios basados en la arquitectura REST
- Implementar aplicaciones que requieran soporte de disponibilidad para una gran cantidad de usuarios
- Entender los fundamentos de TypeScript, incluyendo el sistema de tipos estáticos.

- Aplicar tipos a variables y funciones para mejorar la seguridad y claridad del código.
- Implementar middleware para realizar tareas específicas antes o después del manejo de las solicitudes.
- Desplegar aplicación en la nube

**Unidad 3: Programación del lado del servidor – Introducción a framework NestJS**

- Aplicar conceptos de desarrollo en el framework NestJS.
- Crear controladores para gestionar las solicitudes HTTP y responder a ellas.
- Implementar servicios para encapsular la lógica de negocio en una aplicación NestJS.
- Comprender y aplicar la inyección de dependencias en el contexto de NestJS
- Dominar el sistema de enrutamiento en NestJS y crear rutas para distintos recursos.

**Unidad 4: Programación del lado del servidor – Framework NestJS**

- Aplicar técnicas de autenticación y autorización utilizando middleware en NestJS.
- Integrar TypeORM en un proyecto NestJS para la interacción con bases de datos.
- Implementar pruebas unitarias y de integración utilizando el marco de pruebas Jest.
- Gestionar errores y excepciones de manera eficiente en una aplicación NestJS.
- Configurar el entorno de producción y desplegar la aplicación en servicios de hosting.

**Unidad 5: Programación del lado del cliente – Introducción a framework NextJS**

- Conocer las ventajas de la pre-renderización y la generación de páginas estáticas.
- Desarrollar páginas estáticas y dinámicas en Next.js.
- Manejar parámetros en las rutas y utilizar el enrutamiento dinámico.
- Crear y reutilizar componentes en una aplicación Next.js.
- Integrar Redux en una aplicación Next.js para gestionar el estado de manera centralizada.
- Implementar pruebas unitarias y de integración utilizando el marco de pruebas Jest.
- Configurar el entorno de producción y desplegar la aplicación en servicios de hosting.

**Unidad 6: Aspectos Avanzados de la Computación en Internet**

- Estudiar y aplicar estrategias avanzadas de comunicación, seguridad, en el desarrollo web.
- Promover la innovación a través de proyectos colaborativos de investigación y desarrollo, alentando la creatividad y la aplicación práctica de soluciones avanzadas.
- Considerar la sostenibilidad en el diseño y desarrollo de aplicaciones web.

### ***Metodologías de aprendizaje***

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***

<b>Mecanismo o actividad evaluativa</b>	<b>Porcentaje de la nota final</b>	<b>Relación con objetivo terminales - resultado de aprendizaje del curso</b>	<b>Relación con el resultado de aprendizaje de la competencia de egreso = relación con el Indicador de desempeño del student outcome</b>
Taller nodejs	15 %	OT3	SO2-PI1 SO1-PI2
Taller nestjs	20 %	OT3, OT4	SO9-PI1 SO9-PI2
Taller nextjs	15%	OT2	SO9-PI1 SO9-PI2
Taller Aspectos Avanzados	10%	OT1	SO2-PI1 SO2-PI2
Tareas y quices	20%	OT1	SO1-PI2
Exposición	20%	OT1	SO7-PI1 SO7-PI2 SO7-PI3

### ***Medios Educativos***

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:*