

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

Información de la asignatura

Programa	Ingeniería de sistemas
Periodo Académico	2026-2
Grupo	5
NRC	
Nombre de la asignatura	Seminario de Ingeniería de Software
Código de la asignatura	09825
Intensidad horaria¹	144 horas
Intensidad semanal¹	4 horas/semana.
Créditos¹	3 créditos
Docente(s)	Gonzalo Llano Ramírez/Jose Luis Jurado

Introducción o presentación general del curso

El curso **Seminario de Ingeniería de Software** ofrece un espacio de actualización, exploración e integración de tecnologías, metodologías y prácticas emergentes que están transformando el ejercicio profesional de la Ingeniería de Software. A través de una secuencia de aprendizaje que combina **investigación, experimentación y desarrollo**, los estudiantes fortalecen su capacidad para identificar tendencias tecnológicas, evaluar evidencia científica y aplicar herramientas contemporáneas en la construcción de soluciones de software.

El curso se estructura en cuatro ejes articulados. En primer lugar, desarrolla competencias para realizar revisiones **sistemáticas de literatura y utilizar herramientas de inteligencia artificial como apoyo en la búsqueda, análisis y síntesis de información literaria con rigor científico**. Posteriormente, introduce el uso de **asistentes inteligentes para el desarrollo de software**, abordando estrategias para gestionar el contexto de trabajo, extender las capacidades de los asistentes mediante protocolos y herramientas especializadas, y mejorar la productividad en proyectos de desarrollo.

Como complemento, el curso presenta los fundamentos de la computación en la nube, permitiendo analizar y diseñar arquitecturas basadas en servicios cloud para responder a diferentes necesidades de ingeniería. Finalmente, los estudiantes aplican los conocimientos adquiridos en el desarrollo de una aplicación frontend multiplataforma asistida por inteligencia artificial, integrada con servicios backend en la nube y construida bajo principios modernos de ingeniería de software y desarrollo guiado por especificaciones.

A lo largo del semestre, el aprendizaje se desarrolla mediante actividades de investigación, laboratorios, talleres y un proyecto integrador que promueve el pensamiento crítico, el aprendizaje autónomo, el trabajo colaborativo y el uso ético, crítico y estratégico de la inteligencia artificial como herramienta para

¹ La intensidad horaria, intensidad semanal y créditos, deben estar 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.

la ingeniería. De esta manera, el curso contribuye a formar profesionales capaces de adaptarse a un entorno tecnológico en constante evolución y de incorporar de manera fundamentada nuevas tecnologías en el diseño y desarrollo de soluciones de software

Formación en competencias

Este curso contribuye al desarrollo de las siguientes competencias:

- **SO-1.** Identificar, formular y resolver problemas complejos susceptibles de ser resueltos con ingeniería de sistemas de software, utilizando pensamiento crítico y sistémico, y aplicando principios y métodos de las ciencias, las matemáticas, la ingeniería y, en particular, de las Ciencias de la Computación y la Ingeniería de Software.
- **SO-2.** Diseñar soluciones basadas en desarrollo de software que satisfagan necesidades específicas, considerando la salud pública, la seguridad y el bienestar de las personas, así como factores globales, culturales, sociales, ambientales y económicos.
- **SO-3** Comunicación efectiva: Comunicarse efectivamente de forma oral y escrita, considerando la audiencia, tanto en español como en inglés.
- **SO-7.** Adaptarse a entornos cambiantes y aprender de forma autónoma buscando continuamente oportunidades de éxito personal y profesional

Objetivo general de aprendizaje

Al terminar este curso, los estudiantes habrán adquirido las competencias necesarias para mantenerse actualizados en el campo de la Ingeniería de Software, mediante la aplicación de varias estrategias, a saber: **(i)** la identificación de necesidades de información y cómo buscar, recuperar, analizar y aplicar; **(ii)** la exploración y aplicación de tecnologías emergentes y en evolución, como el desarrollo en la nube y de aplicaciones multiplataforma, asistidas por herramientas de inteligencia artificial.

Resultados de aprendizajes de la asignatura (RAA)

Resultado de aprendizaje del curso o asignatura (RAA)	Competencia (s) en formación	Resultado de aprendizaje de la competencia de egreso del programa al que se contribuye (RAP)
RA1. Aplicar métodos de revisión sistemática de literatura, estrategias avanzadas de búsqueda y herramientas de inteligencia artificial para identificar, evaluar y sintetizar evidencia científica que le permita sustentar de forma oral el análisis de tecnologías y tendencias en Ingeniería de Software.	SO-7 SO-3	SO7-PI1 (T) SO7-PI2 (T) SO3 -PI2 (T)
RA2. Diseñar una solución de software basada en servicios de computación en la nube, seleccionando modelos, servicios y componentes adecuados de acuerdo con los requerimientos funcionales y no funcionales del problema	SO-1 SO-2	SO1-PI2(T) SO2-PI1(T)

Resultado de aprendizaje del curso o asignatura (RAA)	Competencia (s) en formación	Resultado de aprendizaje de la competencia de egreso del programa al que se contribuye (RAP)
RA3. Configurar asistentes inteligentes y protocolos de extensión, para desarrollar soluciones de software.	SO-1	SO1-PI2(T)
RA4. Desarrollar una aplicación frontend multiplataforma asistida por inteligencia artificial e integrada con servicios backend en la nube, aplicando patrones de diseño para construir soluciones funcionales garantizando atributos de calidad.	SO-1 SO-2 SO-7	SO1-PI2(T) SO2-PI2(A) SO7-PI3 (T)

Evaluación de aprendizajes

Código de evaluación	Mecanismo o actividad evaluativa	Relación con resultados de aprendizaje de la Asignatura (RAA)	Nivel de uso de IAG (descripción corta)	Porcentaje de la nota final ²
	Poster con presentación oral de la revisión sistémica de un tema seleccionado	RA1	IAG - Completa	25 %
	Taller de desarrollo de la solución en nube	RA2, RA3	Colaboración con IAG	25 %
	Diseño e Implementación de una solución con aplicaciones multiplataforma	RA4	Exploración con IAG	50 %

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, como: textos, imágenes, música o código, a partir de modelos entrenados con grandes volúmenes de datos.

En coherencia con el Lineamiento 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 especifique el profesor.

² Organización por cortes parciales y reporte de calificaciones

Las actividades evaluativas deben organizarse por cortes parciales, entendidos como momentos institucionales de reporte. En consecuencia:

- Las actividades correspondientes al **corte parcial 1 (semana 8)** deberán permitir que las calificaciones acumuladas sumen, como mínimo, el **30 % de la nota final del curso**.
- Para el **corte parcial 2 (semana 12)**, deberá encontrarse **reportado en el sistema** el **60 % de la nota final del curso**.
- El **40 % restante** corresponderá a las actividades del **corte final**.

Se debe tener en cuenta que la clasificación de cada mecanismo o actividad 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 establecida, 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.

Nivel de uso de IAG	Descripción	Mecanismo o actividad formativa o evaluativa (descripción corta)
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.	No aplica
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.	Laboratorios o retos
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.	Laboratorios o retos
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.	Laboratorios o retos
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.	No aplica

Metodologías de aprendizajes

El curso se desarrolla mediante estrategias de aprendizaje activo centradas en la investigación aplicada, el aprendizaje basado en proyectos, la resolución de problemas y la experimentación con tecnologías

emergentes. Las actividades se organizan en tres momentos que favorecen el desarrollo progresivo de los resultados de aprendizaje del curso.

Antes de la sesión (Preparación)

Los estudiantes preparan las actividades que servirán de base para el trabajo en clase mediante procesos de aprendizaje autónomo.

- Estudiar los conceptos, principios y tecnologías definidos para cada unidad.
- Analizar artículos científicos, documentación técnica y casos de estudio relacionados con el tema.
- Explorar tecnologías emergentes y tendencias en Ingeniería de Software.
- Resolver ejercicios preparatorios.
- Aplicar estrategias de búsqueda y revisión de literatura para identificar información relevante.
- Explorar herramientas y asistentes de inteligencia artificial para comprender sus capacidades y limitaciones.
- Preparar avances del proyecto integrador y de los talleres programados.
- Gestionar responsablemente el aprendizaje autónomo.
- Preparar preguntas y reflexiones para enriquecer la discusión en clase.
- Utilizar la inteligencia artificial con criterios de honestidad académica, pensamiento crítico y transparencia sobre su uso.

La inteligencia artificial se emplea como apoyo para planificar actividades, explorar conceptos, organizar información y preparar insumos para las sesiones. En las actividades de revisión sistemática de literatura se permite un uso de IAG Completa, orientado a fortalecer la búsqueda, clasificación y síntesis de información científica, manteniendo siempre la validación crítica por parte del estudiante.

Durante la sesión (Construcción del aprendizaje)

Las sesiones combinan discusión académica, resolución de problemas, laboratorios, talleres y desarrollo del proyecto integrador.

- Analizar conceptos y discutir fundamentos teóricos.
- Comparar enfoques, arquitecturas y tecnologías para resolver un problema de ingeniería.
- Argumentar decisiones técnicas con base en evidencia científica.
- Resolver problemas de ingeniería mediante talleres prácticos.
- Diseñar arquitecturas de soluciones en la nube.
- Configurar y utilizar asistentes inteligentes para apoyar el desarrollo de software.
- Implementar componentes de la solución y validar su funcionamiento mediante actividades prácticas.
- Participar activamente en discusiones académicas.
- Trabajar colaborativamente en la resolución de problemas.
- Justificar decisiones técnicas con argumentos y evidencia.
- Asumir una actitud crítica frente a las recomendaciones generadas por la inteligencia artificial.

Durante los talleres de desarrollo de soluciones en la nube (RA2 y RA3) se promueve un nivel de Colaboración con IAG, donde los asistentes inteligentes apoyan la generación de código, documentación técnica, configuración de servicios y resolución de problemas, mientras que los estudiantes validan, adaptan y justifican las soluciones obtenidas.

Después de la sesión (Consolidación y reflexión)

Los estudiantes fortalecen los aprendizajes mediante actividades de profundización y mejora continua de los productos desarrollados.

- Relacionar los conceptos estudiados con otros cursos y con el ejercicio profesional.
- Profundizar en tecnologías y buenas prácticas identificadas durante las sesiones.
- Refinar los productos desarrollados en talleres y en el proyecto integrador.
- Documentar decisiones técnicas y lecciones aprendidas.
- Incorporar la retroalimentación recibida para mejorar las soluciones implementadas.
- Reflexionar sobre el proceso de aprendizaje y las decisiones adoptadas.
- Evaluar el impacto técnico, ético y profesional del uso de inteligencia artificial en el desarrollo de software.
- Fortalecer el aprendizaje autónomo y la mejora continua.

En el desarrollo del proyecto integrador (RA4) se promueve un nivel de Exploración con IAG, permitiendo experimentar con asistentes inteligentes, nuevos modelos y herramientas emergentes para construir soluciones innovadoras, evaluar su impacto y proponer mejoras fundamentadas, manteniendo siempre la responsabilidad sobre el producto final.

Unidades de aprendizaje**Unidad de aprendizaje #1****Estrategia de búsqueda y análisis de información (8 sesiones)**

Esta unidad desarrolla las competencias necesarias para identificar, recuperar, evaluar y sintetizar evidencia científica relacionada con la Ingeniería de Software mediante métodos de revisión sistemática de literatura y el uso crítico de herramientas de inteligencia artificial. Su énfasis está en fundamentar decisiones tecnológicas a partir de información confiable y actualizada, fortaleciendo el pensamiento crítico y el aprendizaje autónomo. En esta unidad se desarrolla el RA1, relacionado con la aplicación de métodos de revisión sistemática de literatura y herramientas de IA para el análisis de evidencia científica.

Saberes, problemas o ejes de trabajo de la unidad

1. Systematic Literature Review (SLR)
2. Bases de Datos Bibliográficas: Exploración y uso eficaz.
3. Métodos Avanzados de Búsqueda de Información en Línea: Estrategias y técnicas.
4. Utilización de Herramientas de Inteligencia Artificial para la Búsqueda y Clasificación de Información: Aplicaciones prácticas.
5. Evaluación Crítica de Fuentes de Información: Criterios y análisis.
6. Aplicación de técnicas SLS en el análisis de impacto y proyección profesional de las trayectorias de formación del programa.

Unidad de aprendizaje #2**Fundamentos de computación en la nube (8 sesiones)**

La unidad proporciona una visión integral de los fundamentos de la computación en la nube y de los principales servicios ofrecidos por las plataformas cloud para el desarrollo de software. Su énfasis está en el análisis de requerimientos, la selección de arquitecturas y el diseño de soluciones soportadas en servicios de nube que respondan a necesidades específicas de ingeniería. En esta unidad se desarrolla el RA2, relacionado con el diseño de arquitecturas de soluciones basadas en computación en la nube.

Saberes, problemas o ejes de trabajo de la unidad

- Fundamentos de la computación en la nube: principios, características, beneficios y aplicaciones del Cloud Computing en el desarrollo de soluciones de Ingeniería de Software.
- Modelos de prestación de servicios en la nube: análisis de los modelos IaaS, PaaS, SaaS, FaaS, sus características, ventajas, limitaciones y criterios de selección según las necesidades del problema.
- Arquitecturas para soluciones en la nube: principios de diseño arquitectónico, patrones básicos, escalabilidad, alta disponibilidad, tolerancia a fallos y optimización del uso de recursos en entornos cloud.
- Servicios de computación en la nube: exploración y aplicación de servicios de almacenamiento, autenticación y otros servicios gestionados para la construcción e integración de soluciones de software.

Unidad de aprendizaje #3**Fundamentos de Desarrollo de soluciones asistido por IA (4 sesiones)**

Esta unidad introduce el uso de asistentes inteligentes como herramientas de apoyo al desarrollo de software. Los estudiantes experimentan con la configuración de asistentes, la gestión del contexto de trabajo, los mecanismos de extensión y las estrategias de interacción para incrementar la productividad y mejorar la calidad del desarrollo de software. En esta unidad se desarrolla el **RA3**, orientado a la configuración y utilización de asistentes inteligentes para el desarrollo y documentación de soluciones de software.

Saberes, problemas o ejes de trabajo de la unidad

1. Asistente IA en consola. Salto respecto a una IA de chat web hacia un asistente IA de consola con tool system propio. Experimentar tareas de análisis y documentación sobre repositorios de código. Lectura de archivos, búsqueda en el código, ejecución de comandos en shell.
2. Modos de operación. Confirmación manual, niveles de autonomía y modo de planeación previa a la ejecución.
3. Configuración de proyectos asistido por IA. Creación del archivo de contexto del proyecto, donde se documentan convenciones, arquitectura y el flujo agéntico esperado, aportando el contexto en cada prompt.
4. **Model Context Protocol**. Qué es MCP y qué problema resuelve. Explicar sus piezas básicas (servidor, cliente, herramientas expuestas). Conectar el asistente con MCP usuales, (GitHub, google, etc).
5. **Skills**. Usar y crear skills para el asistente, distinguiendo la estructura de carpetas para ubicarlo a nivel global o solo dentro del proyecto actual.

6. **Contexto:** visualización de la ventana de contexto, factores que consumen el contexto disponible durante una sesión y limitaciones derivadas de su saturación. Estrategias para optimizar su uso, como el resumen de información, la delegación de tareas en subagentes y modular instrucciones.

Unidad de aprendizaje #4

Fundamentos de desarrollo Frontend asistido por Inteligencia Artificial (12 sesiones)

Esta unidad integra los conocimientos desarrollados durante el curso mediante la construcción de una aplicación frontend multiplataforma asistida por inteligencia artificial e integrada con servicios backend en la nube. El énfasis se centra en la aplicación de principios modernos de desarrollo frontend, el uso de especificaciones técnicas como guía del desarrollo y la integración de asistentes inteligentes para apoyar la implementación de soluciones de software. En esta unidad se desarrolla el RAA4, orientado al desarrollo de aplicaciones frontend asistidas por IA e integradas con servicios de computación en la nube.

Saberes, problemas o ejes de trabajo de la unidad

- Panorama del desarrollo frontend: responsabilidades, capas y relación con el backend.
- Paradigma declarativo vs. imperativo en la construcción de interfaces.
- Componentes: estructura, ciclo de vida, props y composición.
- Estado: local, compartido y global; reactividad y efectos secundarios.
- Navegación declarativa y gestión del estado en aplicaciones multipantalla.
- Patrones de diseño aplicados al frontend.
- Spec Driven Development: anatomía de una especificación técnica de frontend
- Integración cliente-backend mediante SDK: autenticación, operaciones de datos y almacenamiento de archivos

Medios Educativos

Bibliografía básica

- Brindle, M., & Cox, N. (2024). *Engineering Software as a Service: An Agile Approach Using Cloud Computing* (3rd ed.). Kendall Hunt Publishing.
- Martin, R. C. (2021). *Clean Craftsmanship: Disciplines, Standards, and Ethics*. Pearson.
- Pressman, R. S., & Maxim, B. R. (2024). *Software Engineering: A Practitioner's Approach* (10th ed.). McGraw-Hill.
- Sommerville, I. (2021). *Software Engineering* (11th ed.). Pearson.
- Van Vliet, H., & Brinkkemper, S. (2024). *Software Architecture* (4th ed.). Springer.

Bibliografía complementaria

Erl, T., Puttini, R., & Mahmood, Z. (2013). *Cloud Computing: Concepts, Technology & Architecture*. Pearson.
Freeman, A., Robson, E., Sierra, K., & Bates, B. (2020). *Head First Design Patterns* (2nd ed.). O'Reilly Media.
Richter, W. (2016). *Modern Software Engineering: Doing What Works to Build Better Software Faster*. Apress.

Documentación técnica y contenidos digitales

- Anthropic. (2025). *Claude Code Documentation*. <https://docs.anthropic.com>

- Google. (2025). *Gemini for Developers Documentation*. <https://ai.google.dev>
- Microsoft. (2025). *Azure Architecture Center*. <https://learn.microsoft.com/azure/architecture/>
- Microsoft. (2025). *Visual Studio Code Documentation*. <https://code.visualstudio.com/docs>
- Model Context Protocol. (2025). *Official Documentation*. <https://modelcontextprotocol.io>
- OpenAI. (2025). *OpenAI Platform Documentation*. <https://platform.openai.com/docs>
- React. (2025). *React Documentation*. <https://react.dev>
- Supabase. (2025). *Supabase Documentation*. <https://supabase.com/docs>
- Vercel. (2025). *Next.js Documentation*. <https://nextjs.org/docs>

Recursos para revisión sistemática de literatura

Kitchenham, B., & Charters, S. (2007). *Guidelines for Performing Systematic Literature Reviews in Software Engineering*. Keele University & Durham University.