1. Identificación del curso

Laboratorio abierto: Construcción							
Programa educativo				**Departamento de adscripción**			
Ingeniería en Computación				Departamento de Ingenierías			
Área de formación				**Tipo de Unidad de Aprendizaje**			
Básica particular obligatoria				Taller			
Carga horaria						**Créditos**	**Clave**
Teoría	0	**Práctica**	80	**Total**	80	5	IL373
Modalidad de Enseñanza - Aprendizaje				**Prerrequisito**			
Presencial				Laboratorio Abierto: Construcción			
Academia				**Profesor responsable**			
Ciencias Computacionales				Héctor González Sánchez			
Elaboró / Modificó				**Fecha de elaboración / modificación**			
Héctor González Sánchez/ María Obdulia González Fernández				16 de junio de 2025			

2. Competencias que abonan al perfil de egreso

Transversal	Disciplinar	Profesional
Posee habilidades de trabajo en equipo que le permita desarrollarse como líder de proyectos en su campo profesional o integrarse a un grupo ya establecido. Reconoce sus responsabilidades éticas y profesionales para actuar con rigor en su desarrollo como ingeniero.	Posee capacidad de razonamiento crítico, lógico y matemático para resolver problemas dentro de su área de estudio a través de modelos abstractos que reflejan situaciones reales. Posee saberes, conceptos, principios y teorías relacionadas con las ciencias computacionales y a sus disciplinas afines.	Se formará con ética y responsabilidad, en búsqueda de la calidad y la innovación tecnológica en las organizaciones. Podrá trabajar en equipo, con liderazgo y una visión emprendedora para aportar soluciones de ingeniería en los contextos global, económico, sustentable y social.

3. Saberes previos

De manera directa, los estudiantes tienen como prerrequisito la materia de Laboratorio abierto: Diseño. Pero también requieren contar con conocimientos sobre programación, desarrollo web, arquitectura de computadoras, bases de datos e ingeniería del software.

4. Presentación de la unidad de aprendizaje

Unidad de aprendizaje orientada a fortalecer el perfil de egreso del ingeniero en computación a través de la aplicación de los conocimientos integrales de la carrera como programación, ingeniería de software, arquitectura de computadores y redes, con la intención de desarrollar un prototipo funcional que forma parte del proyecto aplicativo integrador (PAI). Cabe mencionar que dicha asignatura cuenta como antecedente el Laboratorio abierto: Diseño, donde los estudiantes proyectaron y planificaron los diseños de una problemática a ser resuelta por medio de un sistema computacional. Finalmente, mencionar que los estudiantes cursarán de manera simultánea la asignatura de Seminario integrador: Desarrollo, donde complementarán sus conocimientos.

5. Objetivo de aprendizaje

Aplicar los principios de programación y desarrollo de proyectos tecnológicos para la construcción de un prototipo funcional, como parte del producto de los denominados Proyectos integradores aplicativos.

6. Competencia general de la unidad de aprendizaje

C.23 Capacidad de emprender, completar y presentar un proyecto integrador. (AIS/ACM/IEEE C.23)

7. Habilidades, valores y actitudes

Liderazgo, respeto a las opiniones de los demás, colaboración y trabajo en equipo. Muestra interés en el aprendizaje continuo. Valora la retroalimentación grupal.

8. Elementos de competencia

Bloque No. I: Fundamentos de la Fase de Programación		
Sub-competencia	Identificar los conceptos básicos, el rol de la programación en el desarrollo de software y cómo se relaciona con otras fases	
Cognitivos (Contenido)		
• ¿Qué es la fase de programación? • Relación entre diseño y codificación • Introducción a lenguajes de programación utilizados en ingeniería de software • Herramientas básicas: editores de código, compiladores y sistemas de control de versiones		
Procedimentales		
Investiga sobre los diferentes enfoques de la Inteligencia Artificial, así como las técnicas actuales de la industria		
Estrategias didácticas		
Exposición por parte del profesor y de los alumnos Estrategias para motivar el trabajo en equipo		
Criterios de desempeño	**Producto esperado**	**Sesiones estimadas**
Orden, limpieza, puntualidad en la entrega Presentar trabajos de investigación con información clara y concisa	Genera mediante la fase de diseño con modelos abstractos a la fase de codificación con un código real escrito en un lenguaje de programación que cumpla los requerimientos planteados	20
Área de conocimiento	6.3 Sistemas de software	

Bloque No. II: Implementación de Código Limpio y Eficiente	
Sub-competencia	implementa conocimientos y habilidades para escribir con patrones de diseño, código de calidad, entendible y fácil de mantener.
Cognitivos (Contenido)	
• Principios de código limpio (Clean Code). • Refactorización y eliminación de "deuda técnica". • Patrones básicos de diseño y su implementación. • Testing básico durante la codificación (pruebas unitarias)	
Procedimentales	
Investiga información acerca de definiciones y elementos de los sistemas basados en conocimiento, así como la construcción de agentes mediante programación lógica/funcional para resolver problemas cotidianos.	
Estrategias didácticas	
Exposición por parte del profesor Estrategias para motivar el trabajo en equipo	

Criterios de desempeño	Producto esperado	Sesiones estimadas
Orden, limpieza, puntualidad en la entrega. Presentar trabajos de investigación con información clara y concisa. Programación de ejercicios.	Implementa en tu proyecto de software la aplicación práctica de código limpio y eficiente	28
Área de conocimiento	6.3 Sistemas de software	

Bloque No. III: Optimización y Herramientas de Soporte a la Programación		
Sub-competencia	aplica las técnicas de optimización y el uso de herramientas avanzadas para el desarrollo organización y mantenimiento de software	
Cognitivos (Contenido)		
• Herramientas de análisis estático y dinámico de código. • Integración y despliegue continuo (CI/CD). • Técnicas de optimización de rendimiento. • Introducción a metodologías ágiles aplicadas a la programación (Scrum, Kanban).		
Procedimentales		
Investiga y resuelve problemas basados en búsquedas mediante algoritmos adecuados y óptimos que ofrezcan un resultado confiable/óptimo según su contexto.		
Estrategias didácticas		
Exposición por parte del profesor. Resolución de problemas/ejercicios. Trabajo colaborativo.		
Criterios de desempeño	**Producto esperado**	**Sesiones estimadas**
Orden, limpieza, puntualidad en la entrega. Presentar trabajos de investigación con información clara y concisa.	genera un proyecto de software optimizado que no solo funcione correctamente, sino que también permita apreciar las mejoras de rendimiento gracias a la aplicación de herramientas de análisis y técnicas de optimización.	32
Área de conocimiento	7.3.3 Desarrollo e implantación.	

Nota: 1 sesión = 1 hora;

9. Recursos requeridos

Videoproyector, computadora, paquetería de Ofimática, programas.

10. Evaluación y acreditación de la unidad de aprendizaje

- Actividades de Investigación 15%
- Reportes de avances 60%
- Proyecto 20%
- Participación 5%

11. Referencias (APA)

Básica
Hussain, K., & Hussain, F. (2023). Clean code: An agile guide to software craft. Sonar Publishing. - Martin, R. C. (2017). Clean architecture: A craftsman's guide to software structure and design. Prentice Hall
Kroll, P., Kruchten, P. y Booch P. (2003). The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP. USA: Addison-Wesley.
Scott, K., (2001). The Unified Process Explained. USA: Addison-Wesley.
Complementaria
Pressman, R. S., (2006), Ingeniería del Software: un Enfoque Práctico. México: McGraw Hill.
Sitios web
IONOS. (s.f.). Clean Code: Qué es el código limpio. https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/clean-code-que-es-el-codigo-limpio/
GoDaddy LATAM. (s.f.). Clean Code: qué es y cómo escribir código limpio. https://www.godaddy.com/resources/latam/desarrollo/clean-code-que-es

12. Campo de aplicación profesional

El estudiante es capaz de comprender los diferentes entornos de aplicación de la IA en la industria, así como, los beneficios que conlleva mediante su implementación en problemas con necesidades específicas.

13. Perfil docente

El docente de esta materia deberá ser un profesionista con formación en las áreas de las ciencias computacionales con enfoque en programación; capaz de motivar a la investigación y creación de conocimiento, con habilidades para transmitir sus conocimientos y enseñar de forma interactiva propiciando en los alumnos el autoaprendizaje.

CENTRO UNIVERSITARIO DE LOS ALTOS
DIVISIÓN DE CIENCIAS AGROPECUARIAS E INGENIERÍAS
DEPARTAMENTO DE INGENIERÍAS

Dr. Cesar Eduardo Aceves Aldrete	**Mtro. Héctor González Sánchez**
Jefe de departamento de ingenierías	Presidente de la academia