
	FORMATO DE SYLLABUS	Código: AA-FR-003	
	Macroproceso: Direccionamiento Estratégico	Versión: 01	
	Proceso: Autoevaluación y Acreditación	Fecha de Aprobación: 27/07/2023	

FACULTAD:	FACULTAD DE INGENIERÍA
PROYECTO CURRICULAR:	Ingeniería de Sistemas CÓDIGO PLAN DE ESTUDIOS:

I. IDENTIFICACIÓN DEL ESPACIO ACADÉMICO

NOMBRE DEL ESPACIO ACADÉMICO: Construcción de software

Código del espacio académico:		Número de créditos académicos:			3	
Distribución horas de trabajo:	HTD	2	HTC	2	HTA	5
Tipo de espacio académico:	Asignatura	X	Cátedra			

NATURALEZA DEL ESPACIO ACADÉMICO:

Obligatorio Básico		Obligatorio Complementario	X	Electivo Intrínseco		Electivo Extrínseco	
--------------------	--	----------------------------	---	---------------------	--	---------------------	--

CARÁCTER DEL ESPACIO ACADÉMICO:

Teórico		Práctico		Teórico-Práctico	X	Otros:		Cuál: _____
---------	--	----------	--	------------------	---	--------	--	-------------

MODALIDAD DE OFERTA DEL ESPACIO ACADÉMICO:

Presencial	X	Presencial con incorporación de TIC		Virtual		Otros:		Cuál: _____
------------	---	-------------------------------------	--	---------	--	--------	--	-------------

II. SUGERENCIAS DE SABERES Y CONOCIMIENTOS PREVIOS

Para tener éxito en el curso de construcción de software, es recomendable que los estudiantes tengan los siguientes saberes y conocimientos previos:

Conocimientos básicos de informática:

- * Comprensión de los conceptos básicos de la informática, como el hardware, el software, las redes y la seguridad informática.
- * Familiaridad con el uso de ordenadores personales y sistemas operativos comunes.

Habilidades de programación:

- * Experiencia en al menos un lenguaje de programación de propósito general, como Java, Python o C++.
- * Comprensión de los conceptos básicos de programación, como variables, tipos de datos, operadores, estructuras de control y funciones.
- * Habilidad para escribir código legible y mantenible.

Pensamiento lógico y resolución de problemas:

- * Habilidad para analizar problemas y descomponerlos en tareas más pequeñas y manejables.
- * Capacidad para pensar de forma creativa y encontrar soluciones innovadoras a los problemas.
- * Habilidad para depurar código y solucionar problemas.

Conocimientos matemáticos básicos:

- * Comprensión de conceptos matemáticos básicos, como álgebra, trigonometría y cálculo.
- * Habilidad para realizar operaciones matemáticas básicas con facilidad.

Habilidades de comunicación e interpersonales:

- * Habilidad para comunicarse de manera efectiva por escrito y oralmente.
- * Habilidad para trabajar en equipo y colaborar con otros.
- * Capacidad para aprender de forma autónoma y buscar información por sí mismo.

Además de estos saberes y conocimientos previos, es importante que los estudiantes tengan una actitud positiva hacia el aprendizaje y estén dispuestos a trabajar de forma exigente. La construcción de software es una disciplina desafiante, pero también es muy gratificante. Con el trabajo duro y la dedicación, los estudiantes pueden adquirir las habilidades y los conocimientos necesarios para tener éxito en este campo.

III. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

En la era digital actual, el software se ha convertido en una herramienta indispensable para el funcionamiento de las empresas y organizaciones de todo tipo. La demanda de profesionales capacitados en la construcción de software es alta y sigue creciendo, lo que hace que este curso sea una oportunidad valiosa para aquellos que buscan ingresar a este campo prometedor.

Este curso proporcionará a los estudiantes los conocimientos y habilidades necesarios para diseñar, desarrollar, implementar y mantener software de alta calidad. Aprenderán sobre los principios fundamentales de la ingeniería de software, así como las metodologías y herramientas más utilizadas en la industria.

Al finalizar el curso, los estudiantes estarán preparados para trabajar como desarrolladores de software junior en una variedad de entornos. También tendrán la base para continuar sus estudios y especializarse en áreas específicas del desarrollo de software, como el desarrollo web, el desarrollo de aplicaciones móviles o la ciencia de datos.

En resumen, este curso es una excelente manera de prepararse para una carrera gratificante y desafiante en el campo de la construcción de software.

IV. OBJETIVOS DEL ESPACIO ACADÉMICO (GENERAL Y ESPECÍFICOS)

Objetivo General:
Dotar a los participantes de los conocimientos, habilidades y aptitudes necesarias para llevar a cabo de manera efectiva la implementación de soluciones de software en entornos reales. Esto implica una amplia gama de actividades, que abarcan desde la planificación y preparación del entorno de implementación hasta la prueba, el despliegue y el soporte posterior a la implementación.

- Objetivos específicos:**
- * Brindar una comprensión profunda de los principios y las mejores prácticas de la implementación de software.
 - * Desarrollar habilidades prácticas para la implementación de software en diferentes entornos.
 - * Utilizar herramientas y metodologías para implementar software en entornos locales, en la nube y en entornos híbridos.
 - * Fomentar la capacidad de trabajar en equipo y colaborar con diferentes partes interesadas.
 - * Preparar a los participantes para obtener certificaciones profesionales en construcción de software.

V. PROPÓSITOS DE FORMACIÓN Y DE APRENDIZAJE (PFA) DEL ESPACIO ACADÉMICO			
Competencias	Dominio-Nivel	RA	Resultados de Aprendizaje
Desarrolla implementaciones haciendo uso patrones, estilos, arquitecturas y frameworks de software; que permiten integrar modelos diseñados con la planeación y seguimiento de un proyecto informático.	Cognitivo - comprender	1	Comprende los principios y las mejores prácticas de la implementación de software.
	Cognitivo - aplicar	2	Desarrolla soluciones basadas en tecnologías haciendo uso de diferentes frameworks de software
	Cognitivo - aplicar	3	Utiliza metodologías y herramientas para implementar software en diferentes entornos.
	Afectivo	4	Participa activamente en el trabajo en equipo y colabora en diferentes partes del desarrollo del proyecto
	Cognitivo - evaluar	5	Obtiene el puntaje para aprobar certificaciones profesionales en construcción de software.

VI. CONTENIDOS TEMÁTICOS					
<p>1. Prácticas de Código Limpio</p> <ul style="list-style-type: none">* Convenciones de código y estilo* Refactorización de código* Pruebas unitarias y de integración* Integración continua y entrega continua (CI/CD) <p>2. Herramientas y Frameworks</p> <ul style="list-style-type: none">* Entornos de desarrollo integrado (IDEs)* Sistemas de control de versiones (Git)* Frameworks web y de aplicaciones* Herramientas de automatización de pruebas <p>3. Seguridad en el Desarrollo de Software</p> <ul style="list-style-type: none">* Vulnerabilidades comunes de software* Prácticas de codificación segura* Pruebas de seguridad* Seguridad en la nube <p>4. DevOps</p> <ul style="list-style-type: none">* Generalidades de DevOps* Despliegue de aplicaciones mediante Docker* Jenkins* Kubernetes <p>5. IA Generativa</p> <ul style="list-style-type: none">* Generalidades IA Generativa* Herramientas IA Generativa para el desarrollo de software* Prompt Engineering					
VII. ESTRATEGIAS DE ENSEÑANZA QUE FAVORECEN EL APRENDIZAJE					
Tradicional	X	Basado en Proyectos	X	Basado en Tecnología	
Basado en Problemas		Colaborativo	X	Experimental	
Aprendizaje Activo		Autodirigido		Centrado en el estudiante	

VIII. EVALUACIÓN						
Resultados de aprendizaje (RA) a ser evaluados:	Resultados de aprendizaje asociados a las evaluaciones					
	Actividades Entregables	Talleres	Parciales	Informes de proyecto final	Proyecto final	Exposiciones
RA01	X	X	X	X	X	X
RA02	X	X	X	X	X	
RA03	X	X	X	X	X	
RA04		X			X	X
RA05	X	X		X	X	X
Tipo de evaluación**	EHP	EHP	EE	EBP	EBP	EOP
Porcentaje de evaluación (%)	20	15	30	10	20	5
Trabajo Individual (I) o Grupal (G)	I,G	I,G	I	G	G	G
Tipo de nota	0-5	0-5	0-5	0-5	0-5	0-5
IX. MEDIOS Y RECURSOS EDUCATIVOS						
<ul style="list-style-type: none"> Salón normal con pizarrón para sesiones de cátedra y para sesiones de discusión. Acceso a Videobeam. Página web para publicar material didáctico, guías de trabajo, talleres, etc. Videos didácticos alrededor de los temas de la asignatura. Ttalleres investigativos y prácticos. Acceso al material bibliográfico recomendado 						
X. PRÁCTICAS ACADÉMICAS - SALIDAS DE CAMPO						
No aplica						
XI. BIBLIOGRAFÍA						
<p>Básicas:</p> <p>[1] Robert C. Martin. Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education, 2008.</p> <p>[2] Uncle Bob (Robert C. Martin). Clean Code: A Tutorial and Reference. Pearson Education, 2011.</p> <p>[3] M. Howard and D. LeBlanc, Writing Secure Code, 2nd ed. Redmond, WA, USA: Microsoft Press, 2020.</p> <p>[4] M. Hüttermann, DevOps for Developers. Apress, 2017.</p> <p>[5] David Foster, Generative Deep Learning: A Practical Guide to Autoencoders, Variational Autoencoders, Generative Adversarial Networks, and Beyond, 2021.</p> <p>Complementarias:</p> <p>[A] Joshua Bloch. Effective Java (Second Edition). Addison-Wesley, 2008.</p> <p>[B] Jon Skeet. Code That's Easy to Test: And Other Testable Code Patterns. O'Reilly Media, 2011.</p> <p>[C] Eric Lippert. Code Your Life: Programming for a Brighter Future. Addison-Wesley, 2016.</p> <p>[D] Martin Fowler. Patterns of Design. Addison-Wesley, 1996.</p> <p>[E] Dmitri Yabsley. Patterns for Effective Error Handling in Java. Apress, 2008.</p> <p>[F] R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, 3rd ed. Indianapolis, IN, USA: Wiley, 2020.</p> <p>[G] L. Bass, I. Weber, y L. Zhu, DevOps: A Software Architect's Perspective. Addison-Wesley Professional, 2015.</p> <p>[H] G. Kim, P. Debois, J. Willis, J. Humble, y P. Oppenheimer, The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations. IT Revolution Press, 2016.</p> <p>Páginas web</p> <ul style="list-style-type: none"> DevOps.com. Disponible: https://devops.com DZone DevOps Zone. Disponible: https://dzone.com/devops The DevOps Handbook (Website). Disponible: https://itrevolution.com/the-devops-handbook/ DevOps Institute. Disponible: https://devopsinstitute.com Docker Blog. Disponible: https://www.docker.com/blog CloudBees Blog. Disponible: https://www.cloudbees.com/blog AWS DevOps Blog. Disponible: https://aws.amazon.com/blogs/devops/ 						
XII. SEGUIMIENTO Y ACTUALIZACIÓN DEL SYLLABUS						
Fecha revisión por Consejo Curricular:						
Fecha aprobación por Consejo Curricular:					Número de acta:	

**Tipo de Evaluación	Abreviatura
1. Evaluación de habilidad	EHP
2. Evaluación basada en p	EBP
3. Evaluación oral o prese	EOP
4. Evaluación escrita	EE
5. Evaluación formativa	EF
6. Evaluación de desempe	ED