

**FORMATO DE SYLLABUS**

Macroproceso: Direccionamiento Estratégico

Código: AA-FR-003

Versión: 01

Proceso: Autoevaluación y Acreditación

Fecha de Aprobación:
27/07/2023**FACULTAD:****FACULTAD DE INGENIERÍA****PROYECTO CURRICULAR:**

Maestría en Ciencias de la Información y las Comunicaciones

CÓDIGO PLAN DE ESTUDIOS:**I. IDENTIFICACIÓN DEL ESPACIO ACADÉMICO****NOMBRE DEL ESPACIO ACADÉMICO: INGENIERÍA DE SOFTWARE II**

Código del espacio académico:	79502003	Número de créditos académicos:			4	
Distribución horas de trabajo:	HTD	48	HTC	16	HTA	128
Tipo de espacio académico:	Asignatura	X	Cátedra			

NATURALEZA DEL ESPACIO ACADÉMICO:

Obligatorio Básico	X	Obligatorio Complementario		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

Desarrollo de software, metodologías de desarrollo de software.

III. JUSTIFICACIÓN DEL ESPACIO ACADÉMICO

El Ingeniero de Software se enfrenta normalmente al modelado y construcción de máquinas lógicas para dar solución a problemas de automatización de procesos o mejora en los procesos de las organizaciones desde el punto de vista del procesamiento de información. Este espacio académico es indispensable para consolidar en el futuro magíster de la Universidad Distrital la disciplina metodológica para desarrollar software de alta calidad.

Este espacio académico da mayor énfasis al área de conocimiento de diseño de software aplicado a la ingeniería directa para el desarrollo de soluciones basadas en software. La formación sólida en ciencias de la computación y programación de computadores es un requisito necesario para abordar este curso.

Debido a que esta asignatura requiere formar en el estudiante disciplina metodológica y ejercitarse el uso de un metamodelo para generar modelos de software, la formación se orienta al ejercicio teórico-práctico basado en un proyecto de aula a través del cual se enfrenta al estudiante al escenario de elaboración de los modelos funcionales basados en casos de uso, estructurales y dinámicos de una aplicación basada en software. A lo largo del semestre se profundizará en la comprensión de los conceptos fundamentales del metamodelo, en este caso, UML (Unified Modeling Language), con la aplicación simultánea de los conceptos al problema real de análisis y diseño de software del proyecto de curso.

De otra parte, se abordará la revisión de temas tecnológicos que complementen el conocimiento del estado del arte de las tecnologías que apoyan el desarrollo de software. El ejercicio de la lectura comprensiva será parte del desarrollo del curso. Se abordarán lecturas técnicas relacionadas con las áreas de conocimiento de semánticas de lenguajes de programación, principios y atributos de calidad del software y modelos de valoración de calidad de productos basados en software.

IV. OBJETIVOS DEL ESPACIO ACADÉMICO (GENERAL Y ESPECÍFICOS)

OBJETIVO GENERAL: Revisar de manera conciencia con los estudiantes los elementos teóricos básicos de gestión, análisis, diseño, implementación y pruebas de proyectos de desarrollo de software con un ejercicio de estudio de caso práctico que refuerce la adquisición de conocimientos. Así mismo, enfrentar al estudiante con literatura técnica que lo ubique en el estado del arte y ejercite en él la disciplina de la actualización técnica a través de la lectura continua

OBJETIVOS ESPECÍFICOS

- Contextualizar al estudiante en lo que comprende la disciplina de Ingeniería de Software y comprender los niveles semánticos de especificación de lenguajes de programación
- Brindar elementos conceptuales que permitan controlar el desarrollo de un proyecto de desarrollo de software. Revisar y aplicar de manera práctica algunas de las métricas de software más conocidas.
- Aplicar el Lenguaje de Modelado Unificado y un Modelo de Proceso Unificado simplificado a un proyecto real de desarrollo de software durante el semestre.
- Revisar de manera general los atributos de calidad del software e introducir en el diseño conducido por atributos de calidad.

V. PROPÓSITOS DE FORMACIÓN Y DE APRENDIZAJE (PFA) DEL ESPACIO ACADÉMICO

Competencias	Dominio-Nivel	RA	Resultados de Aprendizaje
Básicas	Diferenciar los niveles semánticos para especificar un lenguaje de programación en concordancia con la teoría de computación.		
	Diferenciar los niveles de metamodelo, meta-modelo y modelo en el aprendizaje y desarrollo de fundamentos teóricos en esta disciplina.		
	Diferenciar técnicas de medición de técnicas de estimación en el desarrollo de software.		
	Reconocer y diferenciar atributos de calidad del software.		
Laborales	Aplicar las unidades lingüísticas de UML 2 en el diseño detallado de software acorde con las recomendaciones de los estándares internacionales.		
	Reconocer de manera diferenciada los enfoques prevalecientes en la actualidad para el desarrollo de software.		
Ciudadanas	Trabajar en equipo y acordar la distribución de tareas de manera equilibrada.		
	Argumentar de manera sólida sobre temas relacionados con las tecnologías que soportan la Ingeniería de Software.		
	Cumplir los acuerdos y reglas establecidas para la formación en una asignatura de acuerdo a la metodología propuesta por el profesor.		

VI. CONTENIDOS TEMÁTICOS

Unidad 1. Introducción

- 1.1. Conceptos de la Ingeniería de Software
- 1.2 Áreas de conocimiento de la Ingeniería de Software.
- 1.3. Niveles de abstracción en ingeniería. Diferenciación conceptual de modelo, metamodelo y meta-metamodelo.

Unidad 2. Métricas De Proceso, Proyecto y Producto

- 2.1. Análisis de los componentes claves en el desarrollo de un proyecto: El proceso, el recurso humano y el proyecto.
- 2.2 Revisión de las métricas más comunes de producto, proyecto y proceso.
- 2.3 Estimación vs. medición: Técnica de estimación por puntos de función.

Unidad 3. Análisis y Diseño Orientado a Objetos con UML

- 3.1. Contexto histórico de la evolución de UML.
- 3.2 Alcance de la recomendación de superestructura de UML.
- 3.3. Niveles de conformidad de las herramientas de modelado respecto a UML.
- 3.4 Áreas semánticas de UML.
- 3.5. Modelado funcional basado en casos de uso: Diagramas y enfoques de especificación.
- 3.6. Modelado estático o estructural: Diagramas de clases, asociaciones y colaboración estructural, diagramas de componentes, diagramas de despliegue.
- 3.7. Mapeo o traducción de modelos de dominio o lógica de negocio a modelos de persistencia.
- 3.8. Modelado dinámico

Unidad 4. Atributos de Calidad de Software

- 4.1. Revisión a los atributos de calidad relevantes en el desarrollo de software: Desempeño, Facilidad de prueba, Mantenibilidad, Facilidad de comprensión, Seguridad, Disponibilidad, Confiabilidad, Usabilidad, Modificabilidad, Facilidad de reuso, Integridad conceptual, Latencia, Conurrencia.
- 4.2. Revisión sucinta al diseño conducido por atributos de calidad.

VII. ESTRATEGIAS DE ENSEÑANZA QUE FAVORECEN EL APRENDIZAJE

Tradicional		Basado en Proyectos		Basado en Tecnología	
Basado en Problemas		Colaborativo		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					
RA02	X					
RA03					X	
RA04			X			
RA05						
RA06						
RA07						
RA08						
RA09						
Tipo de evaluación**						
Porcentaje de evaluación (%)	20%		30%		30%	
Trabajo Individual (I) o Grupal (G)						
Tipo de nota	0-5	0-5	0-5	0-5	0-5	0-5

IX. MEDIOS Y RECURSOS EDUCATIVOS**Medios y ayudas:**

- Sala o salón de conferencias
- Videobeam y computador personal.
- Herramienta para modelado de software usando UML 2
- Herramientas de desarrollo de software

Aulas virtuales**Correo Institucional****Portal Web Institucional (PWI)****X. PRÁCTICAS ACADÉMICAS - SALIDAS DE CAMPO**

XI. BIBLIOGRAFÍA

Básicas:

- Pfleeger, Shari Lawrence. "Ingeniería de software. Teoría y práctica". Prentice Hall. Primera Edición. 2002
- Pressman, Roger S. "Ingeniería del Software. Un Enfoque Práctico". Séptima Edición. Mc. Graw Hill. 2010
- Sommerville, Ian. "Software Engineering". Séptima Edición. Addison Wesley. 2004 LEN BASS P. C., AND KAZMAN, R. "Software Architecture in Practice". Addison-Wesley, 1998.
- Taylor, Richard N.; Medvidovic, Nenad; Dashofy, Eric M. "Software Architecture. Foundations, Theory and Practice". John Wiley & Sons Inc. 2010.
- Weitzenfeld, Alfredo. "Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet". Thomson. 2005
- Sommerville, Ian. "Software Process Models". Computer Science Handbook. Second Edition.A63 Editor-in – Chief Allen B. Tucker. Chapman&Hall/CRC in cooperation with ACM. 2004.
- Ghezzi, Carlo; Mehdi, Jazaheri and Mandrioli, Dino. "Software Qualities and Principles". Computer Science Handbook. Second Edition. Editor-in –Chief Allen B. Tucker. Chapman&Hall/CRC in cooperation with ACM. 2004.
- Kapfhammer, Gregory. "Software Testing". Computer Science Handbook. Second Edition. Editor-in – Chief Allen B. Tucker. Chapman&Hall/CRC in cooperation with ACM. 2004.
- Martin, Robert C. "Object-Oriented C++ Applications Using The Booch Method". Prentice Hall.1.995.
- Meyer, Bertrand. "Object Oriented Software Construction". Prentice Hall, Englewood Cliffs, N.J. 1.988.
- Coad, Peter y Yourdon, Edward. "Object Oriented Analysis". Prentice Hall Cliffs, N.J. 1.990
- Coad, Peter y Yourdon, Edward. "Object Oriented Modeling and Design". Prentice Hall Cliffs, N.J. 1.991
- Tanenbaum, Andrew S. "Sistemas Operativos Modernos".Prentice Hall.1.992
- Newman, Alexander et al. "Using JAVA". Que Corporation. 1996.
- W. , Richard Stevens. "UNIX Network Programming". Prentice Hall. 1990
- Collins, Dave. "Designing Object Oriented Interfaces". Benjamin /Cummings Publishing Company.Inc. 1.995.
- Rumbaugh, James; Blaha, Michael; Premerlani, William; Frederick, Eddy y Lorenzen, William. "Modelado y Diseño Orientado Objetos. Metodología OMT". Prentice Hall. 1996
- Booch, Grady. "Análisis y Diseño Orientado a Objetos". Addison-Wesley/Díaz de Santos. 1996
- Baker, Seán."CORBA Distributed Objects". Addison-Wesley.1997 Booch, Grady; Rumbaugh, James; Jacobson, Ivar; "El Lenguaje Unificado de Modelado". Addison Wesley y ACM Press Books. 1999
- Booch, Grady; Rumbaugh, James; Jacobson, Ivar; "El Proceso Unificado de Desarrollo de Software". Addison Wesley y ACM Press Books. 1999
- Larman, Craig;"UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos". Prentice Hall. 1999 Fowler, Kendall; Scott, Kendall; "UML Gota a Gota". Addison Wesley Longman., Pearson y Prentice Hall. 1999

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 desempeño	ED
----------------------------	----