

	SILABO DE ASIGNATURA Y/O CURSO		
	Código: MI-DO-FO13	Versión: 01	
	Aprobado: 18/01/2016	Página: 1 de 7	

IDENTIFICACIÓN DE LA ASIGNATURA

NOMBRE DE ASIGNATURA	DESARROLLO FORMAL DE SOFTWARE	CÓDIGO	IS0405
ÁREA DE FORMACIÓN	FORMACION ESPECIFICA	MODALIDAD	PRESENCIAL
CRÉDITOS	3	HABILITABLE	N/A
PROGRAMA (S)	INGENIERIA DE SISTEMAS	VALIDABLE	SI
SEMESTRE	06	PRERREQUISITOS	INGENIERÍA DE SOFTWARE
INTENSIDAD HORARIA (Horas Semanales)	PRESENCIAL	4	TRABAJO INDEPENDIENTE 5
UNIDAD ACADÉMICA	PROGRAMA DE INGENIERIA DE SISTEMAS		

DESCRIPCIÓN DE LA ASIGNATURA

ASPECTOS ACADÉMICOS DE LA ASIGNATURA	
PRESENTACIÓN	Desarrollo Formal de Software es una asignatura orientada a la construcción de software mediante prácticas de programación, especificación, verificación, pruebas y evaluación de la calidad. La asignatura articula los conocimientos adquiridos en Análisis de Sistemas, Diseño de Sistemas e Ingeniería de Software. A partir de una funcionalidad seleccionada, el estudiante construye software ejecutable, establece condiciones y propiedades que debe cumplir, incorpora mecanismos de especificación y contratos, aplica técnicas de verificación y pruebas, identifica defectos y propone mejoras de calidad. El curso busca que el estudiante evolucione de la comprobación basada exclusivamente en ejemplos hacia una práctica de desarrollo en la que pueda especificar qué debe cumplir un programa, construirlo y obtener evidencias de su correctitud y calidad.
JUSTIFICACIÓN	La ingeniería de software requiere desarrollar sistemas que no solamente respondan a los requerimientos establecidos, sino que presenten comportamientos previsibles, verificables y confiables. En este contexto, los métodos y técnicas formales permiten expresar de manera precisa propiedades y condiciones que debe cumplir un programa, proporcionando mecanismos para razonar sobre su comportamiento y detectar inconsistencias o defectos. De esta manera, el estudiante desarrolla la capacidad de pasar de un requerimiento y una especificación a una implementación que pueda ser analizada y verificada, fortaleciendo la confiabilidad, mantenibilidad y calidad del software.

COMPETENCIAS QUE DESARROLLA LA ASIGNATURA

PERSONALES	SER	Ciudadanos éticos, responsables, comprometidos con su comunidad, con el desarrollo sostenible y sustentable, con el reconocimiento de su identidad cultural y de los valores humanos
------------	-----	--

	SILABO DE ASIGNATURA Y/O CURSO		
	Código: MI-DO-FO13	Versión: 01	
	Aprobado: 18/01/2016	Página: 2 de 7	

	SABER	- Desempeñarse de manera efectiva en equipos de trabajo. - Identificar, formular y resolver problemas de ingeniería. - Concebir, diseñar y desarrollar proyectos de ingeniería. - Utilizar de manera efectiva las técnicas y herramientas de aplicación en la ingeniería. - Aprender en forma continua y autónoma.
	SABER HACER	- Planificar estrategias para el análisis y la resolución de situaciones problema, a partir de la identificación de los datos, la representación de los mismos y el establecimiento de relaciones, integrando los saberes de las ciencias, las matemáticas y las ciencias básicas de la ingeniería. - Implementar metodologías adecuadas para la gestión, planificación y desarrollo de proyectos de software.
GENERALES	Desarrollar software mediante técnicas de especificación, programación, verificación y pruebas que permitan establecer y comprobar el cumplimiento de propiedades y condiciones de correctitud y calidad.	
ESPECIFICAS	Comprender los fundamentos del desarrollo formal, la correctitud, la verificación, la validación, las pruebas y la calidad del software.	
	Formalizar propiedades y condiciones de comportamiento mediante precondiciones, postcondiciones, aserciones, invariantes y contratos.	
	Construir programas y componentes de software a partir de especificaciones previamente establecidas.	
	Aplicar técnicas de verificación de programas, incluyendo lógica de Hoare, invariantes, razonamiento sobre correctitud y terminación.	
	Utilizar herramientas de verificación automatizada para analizar propiedades de programas y reconocer contraejemplos o condiciones no satisfechas.	
	Aplicar pruebas unitarias, pruebas de casos límite y técnicas complementarias de análisis de calidad.	
	Integrar especificación, construcción, verificación, pruebas y calidad en una funcionalidad del proyecto de software.	
ALCANCES ESPERADOS	Al finalizar la asignatura, el estudiante estará en capacidad de construir una funcionalidad de software a partir de especificaciones previamente establecidas, expresar formalmente condiciones de comportamiento mediante contratos, precondiciones, postcondiciones e invariantes, aplicar técnicas de verificación y pruebas, utilizar herramientas de apoyo a la verificación y analizar la calidad del software desarrollado, documentando y sustentando las evidencias obtenidas.	

	SILABO DE ASIGNATURA Y/O CURSO		
	Código: MI-DO-FO13	Versión: 01	
	Aprobado: 18/01/2016	Página: 3 de 7	

CONTENIDOS MÍNIMOS

UNIDAD	CAPITULO	TEMAS
1. FUNDAMENTOS DEL DESARROLLO FORMAL Y CALIDAD DEL SOFTWARE	1. INTRODUCCIÓN AL DESARROLLO FORMAL	Concepto de desarrollo formal de software. Desarrollo de software y desarrollo formal. Correctitud del software. Error, defecto y fallo. Verificación y validación. Pruebas de software. Relación entre especificación, construcción, verificación y pruebas. Calidad del producto software.
	2. CALIDAD Y CONSTRUCCIÓN DE SOFTWARE	Características de calidad del software. Calidad funcional y calidad técnica. Código mantenible y legible. Revisión del código. Casos límite y condiciones de error. Introducción al análisis estático. Buenas prácticas para la construcción de software.
2. ESPECIFICACIÓN FORMAL Y DESARROLLO BASADO EN CONTRATOS	3. FUNDAMENTOS DE ESPECIFICACIÓN FORMAL	Concepto de especificación formal. Propiedades y condiciones de un programa. Lógica proposicional. Proposiciones y operadores lógicos. Tablas de verdad. Predicados. Cuantificadores básicos. Expresiones booleanas. Formalización de condiciones del software.
	4. CONTRATOS DE SOFTWARE	Precondiciones. Postcondiciones. Aserciones. Invariantes. Contratos de métodos y funciones. Contratos de clases. Diseño por contrato. Especificación de entradas y salidas. Condiciones de error y restricciones. Refinamiento de especificaciones.

	SILABO DE ASIGNATURA Y/O CURSO		
	Código: MI-DO-FO13	Versión: 01	
	Aprobado: 18/01/2016	Página: 4 de 7	

3. VERIFICACIÓN FORMAL DE PROGRAMAS	5. FUNDAMENTOS DE VERIFICACIÓN DE PROGRAMAS	Concepto de verificación formal. Verificación de programas frente a pruebas de software. Correctitud parcial. Correctitud total. Estados de un programa. Aserciones. Razón de ser de una prueba de correctitud. Tripletas de Hoare. Lógica de Hoare. Precondiciones y postcondiciones.
	6. VERIFICACIÓN DE ESTRUCTURAS DE PROGRAMACIÓN	Verificación de asignaciones. Verificación de secuencias. Verificación de estructuras condicionales. Verificación de ciclos. Invariantes de ciclos. Terminación de algoritmos. Verificación de funciones. Verificación de algoritmos. Verificación de programas recursivos. Contraejemplos y corrección de errores.
	7. VERIFICACIÓN DE PROGRAMAS ORIENTADOS A OBJETOS	Especificación de clases. Invariantes de clase. Contratos de métodos. Relaciones entre objetos. Herencia y contratos. Verificación modular. Verificación de componentes de software.
4. PRUEBAS, VERIFICACIÓN AUTOMATIZADA Y CALIDAD	8. PRUEBAS DE SOFTWARE	Pruebas unitarias. Pruebas funcionales. Pruebas de casos límite. Pruebas basadas en propiedades. Diseño de casos de prueba. Pruebas negativas. Pruebas de robustez. Relación entre pruebas y verificación formal.
	9. VERIFICACIÓN AUTOMATIZADA	Automatización de la verificación. Herramientas de verificación. Verificación estática. Solucionadores SAT y SMT. Contraejemplos. Verificación basada en contratos. Introducción al model checking. Propiedades de seguridad. Propiedades de vivacidad. Introducción a lógica temporal.

[illegible]

	SILABO DE ASIGNATURA Y/O CURSO		
	Código: MI-DO-FO13	Versión: 01	
	Aprobado: 18/01/2016	Página: 7 de 7	

ESTADO LEGAL INTERNO Y CONTROL DE SEGUIMIENTO/CAMBIOS DE LA ASIGNATURA

ELABORACIÓN			
ELABORARON	Yowanna Karina Caicedo Guerrero	REVISARON	
FECHA	Agosto 2026	FECHA	

REVISIONES/CAMBIOS		
	AUTOR	FECHA