

# UNIVERSIDAD DEL NORTE

## 1. IDENTIFICACIÓN

<b>DIVISION ACADEMICA</b>	Ingenierías
<b>DEPARTAMENTO</b>	Sistemas
<b>PROGRAMA ACADEMICO</b>	Ingeniería de Sistemas
NOMBRE DEL CURSO	Diseño de software I
COMPONENTE CURRICULAR AL QUE PERTENECE EL CURSO (Básico, Básico Profesional, Profesional)	Profesional
MATERIA O ÁREA DEL CONOCIMIENTO	Ingeniería del Software
CODIGO DEL CURSO	IST 7121
Número de Registro del Curso (NRC)	3806
PRE-REQUISITO	
CO-REQUISITO	
NUMERO DE CREDITOS SEMESTRALES	3
TIPO DE CRÉDITO (Obligatorio, Parcialmente libres, libres)	Obligatorio
NÚMERO DE SEMANAS	16
INTENSIDAD HORARIA SEMANAL Número de horas teóricas semanales de trabajo con el profesor 3 Número de horas prácticas semanales de trabajo con el profesor Número de horas semanales de trabajo independiente 4	
NIVEL DEL CURSO	Pregrado
NOMBRE DEL PROFESOR	Wilson Nieto Bernal
UBICACIÓN DEL PROFESOR (Oficina, e-mail)	Bloque I 4 piso of. 35 <a href="mailto:wnieto@uninorte.edu.co">wnieto@uninorte.edu.co</a>

## 2. DESCRIPCION SINTETICA DE LA ASIGNATURA

En esta asignatura el estudiante conocerá y aplicará el paradigma del desarrollo de software orientado a objetos en el desarrollo de un sistema de información durante las fases de planeación, análisis y diseño del ciclo de vida de un sistema y bajo el enfoque del aseguramiento de la calidad.

## 3. DESCRIPCION AMPLIA DE LA ASIGNATURA

Los estudiantes de Ingeniería de sistemas desempeñan funciones de planeación, análisis, diseño y programación durante el desarrollo de un producto de software, en esta asignatura, Diseño de Software I, conocen y aplican diferentes estrategias propuestas por la Ingeniería del software con el fin de realizar la práctica de desarrollo de software de una manera disciplinada, organizada y estandarizada y que con lleve a construir productos de software de calidad. Para ello, se estudian los siguientes contenidos, Modelos para el proceso de desarrollo del software, ciclo de vida de un sistema de información y sus diferentes etapas, desarrollo de software orientado a objetos, uso apropiado de Estándares, herramientas y gestión de proyectos.

#### **4. JUSTIFICACION**

Es necesario que el estudiante de ingeniería de sistemas conozca las mejores prácticas para el desarrollo de software, utilizar las metodologías de acuerdo a la naturaleza de los proyectos (ágiles, robustas o básica). Usar apropiadamente las herramientas existentes y las tecnologías (EID, herramientas tipo case, herramientas para la gestión de proyectos, herramientas de gestión de código y herramientas de colaboración para trabajo en equipo) que faciliten y simplifiquen la práctica diaria de construcción y mantenimiento de software; pero que también contribuyan a la calidad del producto.

El cambio de paradigma en el desarrollo del software y los nuevos modelos de procesos para la construcción de software ofrecen excelentes oportunidades de mejorar las formas cómo se han venido construyendo el software y es mediante esta asignatura que se le prepara al estudiante en la formalidad que debe incluir en su labor de desarrollador de productos de software que le implica la aplicación y el conocimiento de nuevos métodos y nuevas tecnologías.

#### **5. OBJETIVOS**

##### **5.1. Objetivo General**

Comprender el proceso de desarrollo de software a partir de la conceptualización de los diferentes modelos de desarrollo (convencionales, estandarizados y ágiles) donde el estudiante aprenda a construir y mantener Sistemas de Información, basado en los principios que brinda la Ingeniería de Software, como marco de referencia en el proceso de construcción durante las fases de planeación, análisis, diseño, implementación, despliegue, pruebas y operación del software con altos estándares de calidad.

##### **5.2 Objetivos específicos**

El estudiante estará en capacidad de:

Estudiar las metodologías existentes para la construcción de software y que permiten el desarrollo de aplicaciones robustas, flexibles y reutilizables.

Integrar proceso de planeación para apoyar la gestión y control de proyectos de software.

Realizar el análisis, diseño, implementación de soluciones software, aplicando técnicas, metodologías, herramientas y estándares de desarrollo bajo enfoques como OO. OA y OE de Ingeniería del software.

#### **6. RESULTADOS DEL APRENDIZAJE**

- CO1. Conocer los conceptos de Ingeniería del software, los antecedentes, la crisis del software y las capas de la ingeniería del software
- CO2. Comprender los modelos de procesos para el desarrollo del software: Modelo en cascada, Modelo en espiral, modelos prototipo y modelos ágiles (Scrum, XP, Lean, Devops) para el desarrollo de software
- CO3. Apropiar el modelo de procesos y las fases del ciclo de vida a seguir durante el curso de Diseño de software I.

- CO4. Aprender a definir el plan de desarrollo de un software
- CO5. Aprender a modelar y diseñar soluciones software utilizando las técnicas orientadas a objetos y bajo especificaciones con herramientas tipo UML.
- CO6. Construir un software aplicando los principios y buenas practicas desde Ingeniería del software.
- CO7. Aprender a documentar un desarrollo de software

## **7. METODOLOGIA**

Se conformarán grupos de trabajo de dos estudiantes los cuales desarrollarán las actividades de clase como son el control de lecturas desde el texto guía, los talleres en el laboratorio.

El desarrollo del curso se efectuará por medio de explicaciones dadas por el profesor con respecto a los diferentes tópicos del contenido, a su vez los estudiantes consultarán temas complementarios que reforzarán su aprendizaje. Los temas de mayor actualidad serán tratados en discusiones de grupo y se realizarán ejercicios a manera de talleres.

Los estudiantes deben abordar el proceso de desarrollo de software, el cual debe definirse desde inicio del semestre e ir presentando informes de avance de las diferentes fases que requieren para construirlo. Estos avances deben cumplir con las especificaciones establecidos de acuerdo al método que se utilice y en la última semana de clases deben entregar el software con su documentación.

En el catálogo del curso permitirá interactuar con el estudiante a través de Chats, foros virtuales, trabajos en grupo, dirigidos. Etc. Los estudiantes conformaran grupos para realizar un trabajo de investigación sobre las tendencias de la Ingeniería del Software.

## **8. MEDIOS**

Además de los medios tradicionales para el desarrollo de las clases, se utilizarán los siguientes recursos:

- Diferentes textos de referencia UML, RUP y Scrum
- La Suite Software Enterprise Architect.
- Los recursos computacionales con que cuenta la Universidad del Norte y las redes locales, nacionales e internacionales con las que se encuentra conectada. Se pueden utilizar los recursos propios del estudiante si le es favorable.
- Los recursos audiovisuales disponibles en la Universidad del Norte para el ejercicio de la docencia.

## 9. CONTENIDO

No	TOPICOS	Horas.
1	INTRODUCCION A LA INGENIERIA DE SOFTWARE	3
1.1	El desarrollo software + laboratorio free (node + mongodb)	
1.2	Ingeniería de software y ética	
1.3	Ejemplos y casos. Lectura propuesta y taller	
2.	MODELOS DE PROCESO DEL SOFTWARE Y CICLOS DE VIDA	6
2.1	Modelo en cascada	
2.2	Desarrollo Incremental	
2.3	Modelo Espiral	
2.4	Actividades de procesos: El proceso de ingeniería de requisitos, Modelo general de diseño de software, Etapas del testing de software, Proceso de software en V	
2.5	Desarrollo Prototipos Lectura propuesta y taller	
3	PLANEACION DEL DESARROLLO DEL SOFTWARE	3
3.1	Conceptos de planificación de proyectos software (PP desde CMMI)	
3.2	Planeación de proyectos software	
3.3.	Ejercicios prácticos de planificación proyectos software Lectura y taller Jira; Microsoft Project, Wrike,	
4	MODELADO y ESPECIFICACIÓN DE SISTEMAS SOFTWARE	6
4.1	Requerimientos funcionales y no funcionales	
4.2	Identificación, Especificación, validación y gestión de requerimientos	
4.3	Modelo de casos de uso, modelo de los requerimientos Modelo de interfaces IU, modelos extendidos Lectura y taller con E.A. Parcial 1.-	
5	ENTORNO DE LA HERRAMIENTA CASE E.A. 15.X	3
5.1	Creando archivos de proyectos de desarrollo de software	
5.2	Conocimiento básicos del browser del proyecto	
5.3	Creando modelos, plantillas y edición del proyecto	
5.4	Publicación del proyecto de desarrollo Lectura y taller con E.A.	
6	MODELADO Y DISEÑO DEL SISTEMA SOFTWARE CON UML + HCASE	12
6.1	Modelo de contexto: Business (Process Model eriksson penker)	
6.2	Modelo de casos de uso extendido	
6.3	Modelo de requerimientos funcionales y no funcionales Taller con E.A.	
6.4	Modelo de análisis (diagramas de interacción de objetos)	
6.5	Modelo de datos, diseño de base de datos y tópicos base de datos (SQL-No SQL).- PARCIAL 2.- Taller con E.A.	
6.6	Modelo de dominio o clases	
6.7	Modelo de componentes Taller con E.A.	
6.8	Modelo de despliegue o redes	
6.9	Modelo de Casos de pruebas Taller con E.A.	
7.	LA ARQUITECTURA SOLUCION SOFTWARE	3
7.1	Requerimientos básicos para arquitectura	
7.2	Vistas Arquitectura: MVC, 4+1; N-LAYER, DISTRIBUIDAS, ELASTICAS Y VIRTUALES	



**Links**

[www.uml.org](http://www.uml.org)

[www.omg.org](http://www.omg.org)

[www.sparsystems.com](http://www.sparsystems.com)

[www.sei.cmu.edu](http://www.sei.cmu.edu)

**Unified Process**

<http://www.informit.com/articles/article.aspx?p=24671>>

"Use case diagrams are UML diagrams describing units of useful ..." 2010. 21 Jun. 2016 <<http://www.uml-diagrams.org/use-case-diagrams.html>>

*A basic explanation of UML Diagrams. Also includes example diagrams.*

"UML Class and Object Diagrams Overview - common types of class ..." 2011. 21 Jun. 2016

<<http://www.uml-diagrams.org/class-diagrams-overview.html>>

**Prototyping**

"SDLC - Software Prototype Model - Tutorialspoint." 2013. 7 Jul. 2016

<[http://www.tutorialspoint.com/sdlc/sdlc\\_software\\_prototyping.htm](http://www.tutorialspoint.com/sdlc/sdlc_software_prototyping.htm)>

**Enero 19 de 2024**