

PROGRAMA DE LA ASIGNATURA:
Ingeniería de Software I (IF031)

CÓDIGO: IF031
AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:
2 año
FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:
2022-11-10
CARRERA/S: Analista Universitario de Sistemas
050/2017, Licenciatura en Sistemas 049/2017,

CARÁCTER: CUATRIMESTRAL (2do)
TIPO: OBLIGATORIA
NIVEL: GRADO
MODALIDAD DEL DICTADO: PRESENCIAL
MODALIDAD PROMOCION DIRECTA: SI
CARGA HORARIA SEMANAL: 10 HS
CARGA HORARIA TOTAL: 150 HS

EQUIPO DOCENTE

Nombre y Apellido	Cargo	e-mail
Ezequiel Moyano	Profesor Adjunto	jmoyano@untdf.edu.ar
JORGELINA COLLI	Asistente de Primera	jcolli@untdf.edu.ar

1. FUNDAMENTACION

Esta asignatura resulta fundamental para el futuro ejercicio de la profesión de Licenciado en Sistemas, dado que constituye el primer contacto del alumno con los conceptos vinculados al software como producto, al proceso de desarrollo de software y a métodos y herramientas de la ingeniería de software. En la asignatura se realiza un especial énfasis en las etapas de requerimientos, análisis y diseño del mencionado proceso, teniendo en cuenta que el alumno ya tiene incorporados conceptos de sistemas de información y de programación.

- Con respecto a la ubicación de la materia dentro del Plan de estudios, cabe destacar que se trata de una asignatura del segundo año de la carrera, precedida por "Sistemas y Organizaciones" y Algorítmica y Programación I y II, donde ya han adquirido conceptos básicos de sistemas y programación fundamentales para comprender los procesos del desarrollo de software. Por ello es factible partir de dichos conocimientos para avanzar en conceptos de Ingeniería de Software

- Por otra parte, en el mismo cuatrimestre se dictan Bases de Datos I y Programación y diseño Orientado a Objetos, ambas con contenidos complementarios con esta asignatura, razón por la cual se debe coordinar el avance del dictado.

- Dado que la asignatura Ingeniería de Software II se ubica en el 3er año de la carrera, en esta asignatura se realiza una introducción a la Ingeniería de software sin profundizar en los conceptos vinculados a la administración de proyectos de software relacionados al siguiente curso, por cuanto el énfasis se pone en el proceso de desarrollo de software, desde la etapa de requerimientos hasta su verificación y validación.

- Se considera fundamental el trabajo en contacto permanente con los integrantes de otras cátedras relacionadas, tales como Programación y Diseño orientado a Objetos e Ingeniería de Software II a los fines de brindar la mayor coherencia posible en el tratamiento de los temas y evitar superposiciones. En el caso de materias de programación y Bases de Datos, la relación debe ser continua, con el fin de compatibilizar criterios, conocer la bibliografía utilizada por cada asignatura y ajustar planes de trabajo.

2. OBJETIVOS

a) OBJETIVOS GENERALES

Lograr que el alumno adquiera capacidad para la modelización y el desarrollo de software de calidad desde las primeras etapas de su concepción (con énfasis en las de requerimientos, análisis y diseño) aplicando métodos, herramientas y procedimientos adecuados a cada momento del proceso de desarrollo

b) OBJETIVOS ESPECIFICOS

Mediante el dictado de la materia se pretende que el alumno:

- Adquiera una base sólida en modelos de procesos de desarrollo de software, metodologías de análisis de requerimientos y de diseño de sistemas que le permita desarrollar software de calidad.
- Desarrolle habilidades para la modelización de sistemas
- Conozca las fases del proceso de desarrollo de software, valorando la importancia de llevar adelante adecuadamente las etapas de requerimientos, análisis y diseño en el marco de dicho proceso.
- Comprenda la importancia de la utilización de métodos, herramientas y procedimientos de la ingeniería de software en el desarrollo de sistemas informáticos.
- Sea capaz de realizar todos los pasos que comprende el análisis de un sistema y de diseñar los procesos y las estructuras de datos correspondientes.
- Esté preparado para realizar la especificación y la documentación de un sistema.
- Sea capaz de diseñar las interfaces del sistema.
- Comprenda la conveniencia de utilizar herramientas de automatización en la aplicación de las metodologías estudiadas para el desarrollo de sistemas informáticos.
- Valore la importancia de desarrollar apropiadas pruebas del software.

3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

Para la aprobación del cursado de la asignatura los alumnos trabajaran en pequeños grupos y deberá desarrollar un Trabajo Practico Integrador, sobre el cual se solicitarán entregas específicas en para determinadas etapas del desarrollo. La cátedra dispondrá una rúbrica de evaluación desde en comienzo de la cursada a los efectos que los alumnos conozcan los objetivos de cada etapa y los criterios de evaluación, el objetivo es promover el trabajo colaborativo y recrear una situación real de desarrollo.

Cada etapa tendrá una nota cualitativa de Aprobada o Desaprobado, y la posibilidad de una segunda entrega.

La regularidad se obtendrá con el 75% de la etapas aprobadas.

Para la evaluación se tomará en cuenta una Rúbrica diseñada a tal efecto.

Se contempla la promoción durante la cursada, el alumno deberá aprobar tener aprobadas las etapas desarrolladas al momento (mitad de cuatrimestre primer instancia y al final la segunda) La misma consistirá en la evaluación de los temas teórico vistos al momentos en cada etapa.

El alumno que se presente en condición de libre, deberá avisar y realizar con antelación el proceso de desarrollo de una situación particular entregada por la cátedra, aprobada la instancia se deberá presentar a una mesa para rendir la teoría correspondiente a toda las unidades de la materia

4. CONTENIDOS DE LA ASIGNATURA

MODULO I: INTRODUCCIÓN A LA INGENIERÍA DE SOFTWARE

Ingeniería de Software. Importancia del software. Evolución. Características. Ingeniería de software e Ingeniería de Sistemas. Problemas en el desarrollo de software. Necesidad de la Ingeniería de Software. El Producto Software. Características. Aplicaciones.

Visión general de la Ingeniería de software. Fases. El Proceso de desarrollo de software Métodos, herramientas y procedimientos. Paradigmas de la IS.

Concepto de ciclo de vida del Software. Modelos de ciclo de vida. Ciclos de desarrollo y ciclos de sistemas. Modelo lineal o secuencial, en cascada. Modelo DRA. Modelos de ciclo de vida de evolución. Modelo incremental y evolutivo. Modelo en espiral. Ensamblaje de componentes. Prototipos. Modelos formales: Modelo transformacional. Combinaciones de ciclos de vida. Proceso RUP. Metodologías ágiles.

MÓDULO II: INGENIERÍA DE REQUERIMIENTOS

Conceptos de requerimiento e Ingeniería de requerimientos. Tipos de requerimientos.

Requerimientos funcionales y no funcionales.

Procesos de la Ingeniería de requerimientos Productos entregables. Framework para los procesos de la IR. Procesos de: Elicitación, especificación, validación.

Elicitación de requerimientos. Conceptos. Técnicas de Elicitación: Partiendo del usuario (Entrevistas, cuestionarios, talleres, etc.), Escenarios, Análisis de objetivo y metas, Formularios, Reuso de requerimientos, Casos de uso, otras.

Especificación de requerimientos del software (ERS). Qué incluye. Criterios a cumplir por la ERS. Métodos formales de especificación. Documentación de requerimientos. Esquema del Documento de requerimientos. Estándar IEEE/ANSI 830/1998. Características. Secciones del Estándar. Validación de requerimientos.

MÓDULO III: MODELOS Y HERRAMIENTAS DEL ANÁLISIS

Concepto de Modelo. Ventajas y funciones de los modelos. Herramientas, Principios del análisis.

Modelado del análisis. Visión Esencial y de implementación. Dimensiones de los Sistemas.

Metodologías de análisis. Metodologías estructuradas orientadas a procesos. Análisis

Estructurado Moderno. Análisis Orientado a Objetos, DSBC (Desarrollo basado en componentes), MDD (Desarrollo dirigido por modelos).

Análisis Estructurado Moderno: Modelo esencial y de implementación. Fases. Elementos y Herramientas del Modelo del análisis. Concepto de Análisis Esencial. Componentes de la esencia del sistema. Actividades Esenciales. Memoria Esencial.

Análisis Orientado a Objetos. Abstracción y modelado de sistemas. Beneficios de OO. El Paradigma OO. Modelado con objetos. Objetos del dominio y objetos software.

Conceptos de AOO: Encapsulación, Ocultamiento de la Información. Clases e Instancias.

Identidad. Estado Comportamiento. Persistencia. Comunicación. Mensajes y Estímulos. Métodos.

Herramientas de abstracción. Concepto de Cliente- Servidor. Herencia. Superclases y subclasses.

Clases abstractas. Jerarquías de Herencia. Polimorfismo. Metodología CRC: Clases, Responsabilidades, Colaboraciones

MÓDULO IV: DISEÑO DE SISTEMAS

Concepto y Objetivos. Diseño de alto nivel y diseño detallado. Principios, claves y reglas del diseño. Actividades que comprende el diseño. Conceptos: Abstracción, refinamiento, modularidad división estructural. Mecanismos de abstracción. Modularización. Diseño modular efectivo.

Independencia funcional. Cohesión y Acoplamiento. Decisiones de diseño de alto nivel

(Arquitectura). Componentes, conectores. Vistas de la Arquitectura. Descomposición en

subsistemas, en capas, en particiones. Patrones arquitecturales. Frameworks. Relación ente arquitectura y diseño detallado. Transformación de modelos del análisis al diseño.

Diseño de interfaz. Modelos y procesos de diseño de interfaces. Definición de objetos y acciones.

Principios fundamentales. Usabilidad. Diseño centrado en el usuario.

MÓDULO V: UML – MODELOS, DIAGRAMAS HERRAMIENTAS.

Qué es UML. Historia de UML. Propósito de UML. Enfoques que aglutina. Aspectos

característicos e inconvenientes. Modelos y Diagramas de UML. Organización de los modelos en

UML. Paquetes en UML. Uso de una herramienta CASE para UML.

Modelando requerimientos: Diagramas de casos de Uso: Conceptos básicos. Actores.

Escenarios. Relaciones: Comunicación. Extensión. Inclusión. Herencia. Descripción textual de Casos de Uso. Descripción formal de Casos de uso. Guías para la construcción del Diagrama de Casos de Uso.

Modelando la estructura estática del sistema: Diagrama de Packages. Diagrama de componentes. Diagrama de clases: Clases y Objetos. Relaciones entre clases. Asociación. Tipos de asociaciones. Multiplicidad. Navegabilidad, Atributos y Operaciones. Visibilidad. Perspectivas del Diagrama de Clases. Generalización.

Modelado de la dinámica del sistema con UML: Mensaje y Estímulo. Comunicación. Concepto de Interacción. Análisis de Interacciones de Objetos. Diagramas de Interacciones. Diagrama de secuencia, Diagrama de comunicación.

Modelado del comportamiento del sistema con UML: Diagramas de Máquinas de Estado: Eventos, Estados y Transiciones. Acciones. Generalización de Estados. Diagramas de Actividad: Nodos de Actividad. Nodos de control y Objetos. Flechas de Actividad. Acciones. Flujo de control y datos en una acción. Tipos de acciones provistas por UML. Partición. Actividades sincronizadas.

MÓDULO VI: EL PROCESO DE DESARROLLO DE SOFTWARE CON UML. EL CONTEXTO Y LOS REQUERIMIENTOS DEL SISTEMA

El Proceso de Desarrollo de software con UML. RUP. Conceptos. Características. Fases y Dimensiones. RUP Agil. El proceso en una iteración:

Modelando el contexto del sistema: Modelo del Dominio y Modelo del Negocio. Diagramas utilizados para su construcción.

Modelando los Requerimientos del Sistema. Identificación de casos de uso. Descripción de casos de uso: textual y utilizando diagramas de actividad. Construcción de Diagramas de Casos de Uso partiendo del Modelo de Dominio o del Negocio. Partiendo del Sistema. Modelado de requerimientos no funcionales.

MÓDULO VII: EL PROCESO DE DESARROLLO DE SOFTWARE CON UML. ANÁLISIS Y DISEÑO DEL SISTEMA

Análisis estático del Sistema con UML: Construcción del Diagrama de Clases de nivel conceptual partiendo del modelo de Dominio o del Negocio. Enfoques para el Desarrollo. Subsistemas con diagramas de Packages.

Análisis dinámico del sistema: Realización de casos de uso. Enfoques para el Desarrollo.

Modelación de escenarios con diagramas de interacciones. Completando el Diagrama de clases. Diseño arquitectural. Definición de subsistemas y dependencias. Uso de patrones arquitecturales. Diseño de la estructura de un sistema en packages.

Diseño detallado. Mapeo del Diagrama conceptual al de diseño. Construcción del Diagrama de clases del Diseño.

MÓDULO VIII: GUIAS DE DISEÑO E IMPLEMENTACIÓN

Guías de diseño para Diagramas Flexibles. Aplicación de patrones de diseño. Cohesión.

Acoplamiento. Acoplamiento de identidad, representacional y de subclase. Uso de Interfaces.

Acoplamiento de Herencia. Generalización y Especialización. Generalización de Interfaces.

Especialización de clases. Diseño de una clase: Comportamiento de la clase. Nombres de los métodos. Construcción de Instancias. Destrucción de Instancias. Operaciones fundamentales.

Asociación de Clases del Modelo de Diseño con los Lenguajes de Programación. Asociaciones y Navegabilidad. Asociación con las estructuras de los Lenguajes de Programación. Visibilidad, de Atributo, de Parámetro, Local.

MÓDULO IX: VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE

El Proceso de verificación y validación. Diferencias. Técnicas de verificación y Validación.

Técnicas estáticas. Técnicas dinámicas. Depuración. Inspecciones del software. Prueba del software. Técnicas de prueba de software. Fundamentos. Objetivos, principios. Prueba vs.

Depuración. Normas de calidad asociadas. Trazabilidad de requerimientos. Metodologías de desarrollo y pruebas. Niveles de pruebas. Tipos de pruebas: estáticas y dinámicas. Enfoques de

diseño de pruebas. Pruebas de caja blanca. Pruebas de caja negra. Enfoque aleatorio. Diseño de casos de prueba. Prueba de unidad. Consideraciones y procedimientos. Prueba de integración. Tipos de pruebas de integración. Pruebas de sistema. Estrategias de prueba de software. Enfoque estratégico. Estructura del plan de pruebas. Herramientas Open Source. El proceso de depuración. Entrega del sistema. Los problemas del mantenimiento. Técnicas y herramientas para el mantenimiento.

5. RECURSOS NECESARIOS

- Proyector
- Pc
- Laboratorio Informatica

6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
1	I	Teoría: Ing. de Software Conceptos de Ing. de Software. El Producto. Proceso de desarrollo y Ciclos de vida	
2	II	Teoría: Ing. de requerimientos. Conceptos de Ing. de requerimientos. Elicitación de requerimientos. Técnicas.	
3	II	Teoría: Ing. de requerimientos. Especificación de requerimientos del software. Estándar IEEE-830	
4	III	Teoría: Modelos y Herramientas del Análisis Conceptos fundamentales del Análisis de Sistemas. Modelos Modelo de Análisis Estructurado	
5	III	Teoría: Modelo de Análisis OO Conceptos básicos del Análisis OO Método CRC. Ejemplo	
6	IV	Teoría: Diseño de sistemas Conceptos y Principios fundamentales del diseño. Diseño de alto nivel.	
7	V	Parcial práctico N° 1 Parcial teórico N° 1. Teoría: UML. Modelos y Diagramas	
8	V	Introducción a UML. Diagramas de casos de uso. Diagramas para Modelado Estático de un sistema Diagramas para Modelado Dinámico del Sistema	
9	VI	Teoría: El Proceso de desarrollo de software con UML El Proceso de Desarrollo de Software RUP Agil. Características y Modelos.	
10	VI	Teoría: El Proceso de desarrollo de software con UML Modelado del Contexto y los requerimientos del sistema Modelado de los requerimientos del Sistema.	
11	VII	Teoría: El Proceso de desarrollo de software con UML Modelado de la estructura del sistema. Modelado dinámico del sistema. Interacciones y Comportamiento.	
12	VII	Teoría: El Proceso de desarrollo de software con UML Diseño arquitectural. Diseño detallado.	
13	VIII	Teoría: Guías de diseño e implementación Guías de diseño estático y dinámico del sistema. Patrones Del diseño a la implementación.	
14	IX	Teoría: Verificación y validación del software Técnicas de verificación y validación. Prueba del software	

15	X	Parcial práctico N° 2. Parcial teórico N° 2	
16	X	Parcial recuperatorio N° 2. Presentación TP Integrador	

7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
Roger S. Pressman	2005	Ingeniería de Software - Un enfoque práctico. 6a. Ed.	1-2-3-4		Mc Graw Hill
Carlo Ghezzi, et al	2000	Fundamentals of Software Engineering	1-2		Prentice Hall
Sommerville I.	2005	Ingeniería del software	1-2-3-4-5-6		Addison Wesley
Lawrence S.	2002	Ingeniería del software. Teoría y práctica	1-2-3		Prentice Hall
Schach S.	2006	INGeniería de software clásica y orientada a objetos. 6a. ed.	1-2		Mc Graw Hill
Braude	2007	Ingeniería del software. Una perspectiva OO	1-2-3-5		Alfaomega
Sommerville I, Sawyer	1999	Requirements Engineering	1-2-3-4-5		Wiley & Sons
Booch, Rumbaugh, Jacobson	2007	Object oriented analysis and design with applications: 3a ed			Pearson ed.
Kendall & Kendall	2005	Análisis y diseño de sistemas. 6a ed.			Pearson ed.
Fowler M., Scott K.	2004	UML Gota a gota			Prentice Hall
Larman C.	2003	UML y Patrones. 2a ed.	1-2-3-4-5-6-7-8-9-10		Prentice Hall
Jacobson, Booch, Rumbaugh	2000	El Proceso unificado de desarrollo de software			Addison Wesley
Booch, Rumbaugh, Jacobson	2006	El Lenguaje UML. 2a ed.			Pearson ed.
Lano K.	2005	Advanced Systems Design with Java, UML and MDA			Elsevier Ltd.
Weilkiens T.	2008	Systems Engineering with SysML/UML			The MK/OMG Press

Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	

Este programa de estudio tiene una validez de hasta tres años o hasta que otro programa lo reemplace en ese periodo