

TS4C4 Técnicas de Modelado de Software

Nombre del programa académico	Tecnología en Desarrollo de Software
Nombre completo de la asignatura	Técnicas de Modelado de Software
Área académica o categoría	
Semestre y año de actualización	Semestre I - 2023
Semestre y año en que se imparte	Semestre IV – Año 2
Tipo de asignatura	<input checked="" type="checkbox"/> Obligatoria <input type="checkbox"/> Electiva
Número de créditos ECTS	4
Director o contacto del programa	Guillermo Roberto Solarte Martínez
Coordinador o contacto de la asignatura	Daniel Andrés Duque Ramirez

Descripción y contenidos

1. Breve descripción

Esta asignatura estudia principalmente técnicas de modelado de software; así como el lenguaje de modelado más utilizado en la ingeniería informática: UML (Unified Modeling Language). Antes de profundizar en estos conceptos, se hará una introducción a los procesos de desarrollo de software, requerimientos y conceptos básicos de la ingeniería de software.

2. Objetivos

- Al finalizar el curso, el estudiante está en la capacidad de describir, comprender y diseñar software utilizando los diferentes recursos como diagramas UML, Diagramas de Flujo, Diagramas Entidad Relación.
- Trabajar en equipos de desarrollo.
- Seleccionar una metodología de trabajo según el tipo de software a implementar.

Correspondencia con los objetivos del programa:

Obj1. Desarrollar habilidades para el diseño, implementación, y socialización de proyectos informáticos pertinentes y adaptables al entorno empresarial, soportados en la ingeniería de software; teniendo en cuenta aspectos económicos, sociales y ambientales.

Obj2. Fomentar comportami

entos éticos con integridad y responsabilidad, promoviendo la equidad y la diversidad, en ambientes individuales y grupales.

Obj3. Producir software de calidad con estándares internacionales mediante estrategias computacionales basadas en la lógica de la programación en diferentes entornos de desarrollo.

3. Resultados de aprendizaje

Competencias específicas:

RA1: Identifica el diseño, la implementación y la socialización de proyectos de desarrollo de software como una respuesta a problemáticas relacionados con la productividad en el manejo de la información actual y futura de la empresa, y de la sociedad en condiciones de responsabilidad, eticidad y equidad (COMPRENDE)

RA2: Implementa estándares internacionales de calidad en la producción de software en cada una de sus etapas. (HACE)

4. Contenido

T1. Introducción a la Ingeniería de software
T2. Ciclos de vida de SW
T3. Ingeniería de requerimientos
T4. Introducción a la POO
T5. Diagramas UML + ER + Diagramas de Flujo
T6. Desarrollo web base, HTML, CSS, JS, JSON.

5. Requisitos

---- N/A ----

6. Recursos

- Libro: "Ingeniería de Software: Un Enfoque Práctico" de Roger S. Pressman.
- Artículo: "The Mythical Man-Month" de Frederick P. Brooks Jr.
- Libro: "Software Engineering: A Practitioner's Approach" de Roger S. Pressman.
- Artículo: "A Spiral Model of Software Development and Enhancement" de Barry Boehm.
- Libro: "Requirements Engineering: From System Goals to UML Models to Software Specifications" de Axel van Lamsweerde.
- Artículo: "The Role of the User in Software Development" de E. S. Raymond.
- Libro: "UML Distilled: A Brief Guide to the Standard Object Modeling Language" de Martin Fowler.
- Artículo: "The Unified Modeling Language User Guide" de Grady Booch, James Rumbaugh, Ivar Jacobson.
- Libro: "Agile Estimating and Planning" de Mike Cohn.
- Artículo: "Scrum: A Pattern Language for Hyperproductive Software Development" de Jeff Sutherland y Ken Schwaber.
- Libro: Bourque, Pierre. Fairley, Richard. SWEBOK. Versión 3.0, 2014. IEEE Computer society.
- Libro: Eloquent JavaScript 3rd edition (2018), Marijn Haverbeke.

7. Herramientas técnicas de soporte para la enseñanza

- e-Learning Skillsforall.com
- Zoom
- VS Code
- w3schools.com
- learngitbranching.js.org

8. Trabajos en laboratorio y proyectos

Los estudiantes desarrollan un proyecto sencillo guiado a lo largo de la asignatura en donde se trabaja la concepción del problema, diseño y diagramación, el desarrollo y despliegue.

Paralelamente, los estudiantes deben trabajar las mismas bases utilizando una metodología ágil (SCRUM) para el desarrollo de las entregas de un proyecto/problema propio que deben plantear.

9. Métodos de aprendizaje

- Clases magistrales
- Clases prácticas virtuales
- Plataforma de e-Learning
- Lecturas en portales especializados.
- Videos guía para profundización en temas puntuales o uso de herramientas
- Tutorías

10. Métodos de evaluación

- Primer examen parcial: Conceptos base de ingeniería de software y ciclos de vida junto con entrega práctica sitio web. 20%
- Entrega I: definición del problema + tablero. 15%
- Entrega II: Modelos UML + Diagrama de Flujo. 15%
- Entrega III: Modelado de datos ER. 15%
- Entrega Final: Presentación y entrega del proyecto. 15%
- Trabajos en clase. 20%