

### TS4A4 ENTORNOS DE DESARROLLO DE SOFTWARE

Nombre del programa académico	Tecnología en Desarrollo de Software
Nombre completo de la asignatura	Entornos de Desarrollo de Software
Área académica o categoría	Tecnólogos
Semestre y año de actualización	Semestre 2 - 2023
Semestre y año en que se imparte	Semestre 1- Año 1
Tipo de asignatura	[ x ] Obligatoria [ ] Electiva
Número de créditos ECTS	4
Director o contacto del programa	Guillermo Roberto Solarte Martínez
Coordinador o contacto de la asignatura	

#### Descripción y contenidos

<p>1. Breve descripción</p> <p>La asignatura de Entornos de Desarrollo de Software es de naturaleza teórica, el propósito que tiene es el estudio de las áreas del conocimiento de la ingeniería del software, más concretamente los requerimientos del software, diseño mediante modelamiento UML y pruebas del software.</p>
<p>2. Objetivos</p> <p><b>Correspondencia con los objetivos del programa:</b></p> <p>1. El estudiante conocerá cuales son las KA (áreas de conocimiento) de la ingeniería del software especificadas en el SWEBOK.</p> <p>2. El estudiante conocerá las fases de un proceso de desarrollo de software. 3. El estudiante adquirirá la capacidad para realizar la fase de requerimientos de un proyecto de desarrollo de software.</p> <p>4. El estudiante adquirirá la capacidad de realizar la fase de pruebas del software. 5. El estudiante maneja los conceptos y el lenguaje específico de la asignatura, podrá entender los textos de la materia y escribir sobre el tema.</p>

### 3. Resultados de aprendizaje

#### **Competencias específicas:**

Reconocer las áreas del conocimiento del software descritas en el libro swebok. Reconocer los tipos de requerimientos de software, funcional y no funcional. Realizar una elicitación de requerimientos a un cliente de producto del software. Realizar una especificación y validación de requerimientos.

Diseñar artefactos UML de software.

Reconocer las pruebas unitarias, de integración, punto a punto y pruebas de carga. Diseñar e implementar pruebas unitarias del software en cualquier lenguaje de programación. **Otras**

#### **competencias por formar:**

Diseñar y modelar procesos en la notación BPMN.

Reconocer los patrones de diseño más comunes.

Principios del diseño e implementación del software.

Reconocer metodologías de desarrollo y ciclos de vida.

### 4. Contenido

- a. T1: Contextualización a la teoría general de sistemas y desarrollo de software. (4h)
- b. T2: Áreas del conocimiento del software. (12 h).
- c. T3: Requerimientos del software (18 h).

- d. T4: Modelamiento de procesos (4 h).
- e. T5: Modelamiento de artefactos UML (18 h).
- f. T6: Metodologías de desarrollo (6 h).
- g. T7: Patrones de diseño (12 h).
- h. T8: Teoría y conceptos de pruebas del software (6 h).
- i. T9: Diseño e implementación de pruebas unitarias (12 h).

### 5. Requisitos

- a. Asignatura: Programación Orientada a Objetos.
- b. Competencias: Capacidad de reconocer los procesos empresariales y cómo funcionan. Entendimiento de la teoría general de sistemas y los componentes principales del mismo. Capacidad de abstracción de procesos lógicos matemáticos.

## 6. Recursos

### a. Libros de texto:

- i. SWEBOK, Software Engineering Body of Knowledge.
- ii. Alexander Shvets patrones de diseño.

### b. Herramientas informáticas.

- i. Draw.io
- ii. Lenguajes de programación de preferencia. Ej: Java, Python, C++,

## etc. 7. Herramientas técnicas de soporte para la enseñanza

- a. Trabajos escritos de presentación de requerimientos.
- b. Talleres en clase, quizzes y evaluaciones.
- c. Exposición presencial o digital.
- d. Examen parcial de conocimientos generales.
- e. Trabajo final de programación de pruebas del software.

## 8. Trabajos en laboratorio y proyectos

Esta asignatura está estructurada para que el estudiante haga un trabajo tipo proyecto final sobre las pruebas unitarias del software, a partir de los requerimientos del software presentados en una entrega anterior. También el estudiante deberá programar ejemplos de aplicación de patrones de diseño de su elección.

## 9. Métodos de aprendizaje

- a. Clases magistrales.
- b. Talleres en clase.
- c. Videos y presentaciones.
- d. Planteamiento de requerimientos reales.

## 10. Métodos de evaluación

- a. Trabajo 1:
  - i. Primera entrega: Elicitación de requerimientos.
  - ii. Segunda entrega: Especificación de requerimientos.
- b. Actividades y talleres: El estudiante deberá realizar las actividades y/o talleres propuestos en clase.
- c. Exposición o presentación: Presentación de algún patrón de diseño sugerido por el docente y elegido por el estudiante.
- d. Trabajo Final: Pruebas del software a partir de TDD (Test Driven Development).