



Fecha: diciembre de 2021

**PROGRAMA ACADÉMICO:** Ingeniería De Sistemas Y Computación

**SEMESTRE:** Séptimo

**ASIGNATURA:** Ingeniería de software II

**CÓDIGO:**

**NÚMERO DE CRÉDITOS:** 4

### PRESENTACIÓN

Es un curso diseñado, junto a Ingeniería de Software I, para proveer una amplia introducción en los fundamentos de la Ingeniería de Software.

El curso de Ingeniería de software II, aborda los temas subsiguientes, luego de conocer los procesos de: requisitos, paradigmas de desarrollo, prácticas de análisis y diseño, construcción y pruebas.

Se abordarán temas como: arquitectura de software, integración, mantenimiento, así como contenidos transversales, por ejemplo: la calidad, la administración de la configuración, la administración de proyectos, finalmente, se realiza un paseo por temas de actualidad que se aconseja sean profundizados más adelante por los participantes del curso.

### JUSTIFICACIÓN

Los ingenieros de sistemas y computación de la Uptc, deben ser capaces de aplicar sus conocimientos a la realización de tareas y resolución de problemas en diferentes áreas relacionadas con las Tecnologías de la Información y las Comunicaciones, por ello, es imprescindible formar personas competentes en el área de programación e ingeniería del software, así como en manejo de tecnologías de actualidad, además, son profesionales que van a tener que funcionar eficazmente en esta nueva sociedad del conocimiento, donde la necesidad del aprendizaje continuo a lo largo de toda su vida profesional es ya una realidad.

### COMPETENCIAS

La asignatura está ligada a fortalecer competencias referentes al proceso de ingeniería de software, enfatizando las mejores prácticas, al final del curso se espera que el estudiante:

- Interprete los procesos que se presentan en el desarrollo de un software, comparando paradigmas y metodologías de desarrollo, para identificar las mejores prácticas en determinada situación.
- Deducir los efectos de seguir métodos de aseguramiento de la calidad, de medición y de integración de equipos humanos involucrados en el desarrollo de aplicaciones, a partir de conocer técnicas de automatización de tareas, para lograr un desarrollo ágil.
- Construir comentarios argumentados sobre tecnologías que le permiten mejorar la toma de decisiones sobre la infraestructura de desarrollo adecuada a determinado ambiente de trabajo.



### RESULTADOS DE APRENDIZAJE

- Comparo paradigmas y metodologías de desarrollo, mediante la Interpretación de los procesos que se presentan en el desarrollo de un software, para identificar las mejores prácticas a aplicar en una determinada situación.
- Desarrollo aplicaciones de calidad de manera ágil, siguiendo métodos de aseguramiento de la calidad, de medición y de integración de equipos humanos involucrados en el desarrollo de aplicaciones, a partir de conocer técnicas de automatización de tareas.
- Construyo comentarios argumentados sobre tecnologías que permiten mejorar la toma de decisiones sobre la infraestructura de desarrollo adecuada a determinado ambiente de trabajo.

### METODOLOGÍA

Se realizará exposición magistral de los temas del contenido temático, los cuales deberán ser entendidos desde las mejores prácticas y teorías del conocimiento.

Se darán los parámetros indispensables para que los estudiantes realicen prácticas con herramientas necesarias para la implementación de los conceptos explicados, estos conocimientos deben ser plasmados en un proyecto colectivo, que se organiza de la siguiente manera:

Semana	Actividad
1	Conformación de equipos de trabajo y presentación de idea del proyecto a desarrollar
2-3-4	Iteración 1
5-6-7	Iteración 2
8-9-10	Iteración 3
11-12-13	Iteración 4
14-15-16	Iteración 5

Se proponen temas para que el estudiante avance en su proyecto y presente los resultados de la misma mediante:

- socializaciones,
- foros,
- debates,
- y otras actividades.

Lo anterior permite afianzar el espíritu crítico del grupo, de igual manera, el uso y conocimiento de herramientas relacionadas con proceso de software y calidad, complementa en gran parte los contenidos teóricos y el proceso de aprendizaje.

### INVESTIGACIÓN

Los estudiantes tendrán espacios propios para la lectura y análisis de artículos y documentos entregados por el docente, así como tendrán la oportunidad de realizar consultas sobre temas específicos (nuevas tecnologías, conceptos, otros). En los temas abordados deberán conocer el estado del arte y elaborar escritos que plasmen sus experiencias y opiniones.

### MEDIOS AUDIOVISUALES



Para el mejor desarrollo de las actividades propuestas, se hará uso de: herramientas de comunicación síncrona como videoconferencias y asíncronas como foros; se acudirá al uso de la plataforma institucional Moodle con el fin de socializar material de estudio y lectura, efectuar la entrega de trabajos y tareas, además para la realización de algunas evaluaciones.

### EVALUACIÓN

#### EVALUACIÓN COLECTIVA

Se incentivará el trabajo colaborativo, para que los equipos de trabajo en sinergia puedan contar con los recursos tecnológicos necesarios para el desarrollo de la asignatura.

Para la evaluación se propone actividades que pueden ser realizadas de manera colectiva, se encuentran las siguientes:

Ítem	Porcentaje
Talleres	15%
Seguimiento al desarrollo de proyectos asignados en grupo	35%

#### EVALUACIÓN INDIVIDUAL

Se acudirá a técnicas de gestión de proyectos, en donde, se realizará monitoreo continuo.

Ítem	Porcentaje
Evaluaciones	35%
Participación en actividades	15%

### CONTENIDOS TEMÁTICOS CENTRALES

Contenidos temáticos
Presentación del Curso
Fundamentos del diseño de Software
Elementos clave del Diseño de Software
Estructura y Arquitectura de Software
Análisis de Calidad y Evaluación
Notaciones de Diseño de Software
Estrategias y métodos de Diseño
Fundamentos de construcción de Software
Gestión de la Construcción y Consideraciones prácticas
Fundamentos de Pruebas de Software
Niveles de Pruebas
Técnicas de Pruebas
Medidas de Pruebas
Proceso de Pruebas



	Gestión del proceso SCM	
	Identificación de la configuración	
	Control de Configuración de Software	
	Registro del estado de la configuración	
	Auditoria de Configuración de Software	
	Gestión de lanzamiento y entrega	

#### LECTURAS MÍNIMAS

- Documentos y artículos relacionados con la asignatura, proporcionados por el docente.
- clean architecture

#### BIBLIOGRAFÍA

- ALISTAIR, COCKBURN. Writing Effective Use Cases (Agile Software Development Series). 2000.
- BRAUDE, Eric J. Ingeniería de Software: una perspectiva orientada a objetos, Editorial Alfaomega, ISBN 9701508513, 2003
- BOOCH, G. Análisis y Diseño Orientado a Objetos con aplicaciones, 2ª edición, Addison-Wesley/Diaz de Santos, 1996.
- BOOCH, G. et al. Unified Modeling Language UML Notación guide and UML. Semantics, version, 1.3, mayo 1999, <http://www.rational.com>
- BRUEGGE, B. Ingeniería de Software Orientada a Objetos, Prentice Hall, 2002.
- COHN, MIKE. User Stories Applied: For Agile Software Development (Addison Wesley Signature Series). 2004.