

Syllabus Asignatura: Ingeniería de Software II

Semestre: 2024-1

Unidad Académica Responsable: Departamento de Informática y Ciencias de la Computación

CARRERA a las que se imparte: Ingeniería Civil Informática

I.- IDENTIFICACION

Nombre: Ingeniería de Software II		
Código: 503355	Créditos: 4	Créditos SCT: 6
Prerrequisitos: 503354 Ingeniería de Software I		
Modalidad: remota/presencial	Calidad: obligatoria	Duración: semestral
Semestre en el plan de estudios: 9	Ingeniería Civil Informática	
Trabajo Académico 10		
Horas Teóricas: 3 Horas Prácticas: 2 Horas Laboratorio: 0		
Horas de otras actividades: 5		
Docente Responsable	Gonzalo Rojas Durán	
Duración (semanas)	16 semanas	
Fecha: 06 de marzo de 2023		Aprobado por:

II.- DESCRIPCION

En esta asignatura se complementarán y aplicarán conceptos fundamentales de Análisis, Diseño e Implementación de Software, mediante la revisión de las características y pertinencias de distintos métodos y procesos de desarrollo.

Esta asignatura contribuye a las siguientes competencias del perfil de egreso:

- Aplicar principios de matemáticas, ciencias de la ingeniería y ciencias de la computación, a problemas de ingeniería informática.
- Identificar necesidades de usuarios, a partir de las cuales formula, diseña e implementa soluciones informáticas que satisfagan las especificaciones.
- Analizar e interpretar grandes volúmenes de datos que describen tanto sistemas como sus componentes.
- Diseñar y conducir experimentos para evaluar, caracterizar y modelar sistemas informáticos complejos.
- Trabajar en equipo para solucionar problemas de manera colaborativa e incentivando el trabajo multidisciplinario.
- Comprender, desenvolverse y comunicarse efectivamente en los ambientes laborales delimitados por la responsabilidad profesional y ética.
- Involucrarse en procesos de auto-aprendizaje continuo que le permita adaptarse a la evolución de la teoría y tecnología.

III.- RESULTADOS DE APRENDIZAJE ESPERADOS

Al finalizar la asignatura, el/la estudiante debe ser capaz de:

1. Reconocer los conceptos básicos de Ingeniería de Software
2. Describir modelos de desarrollo tradicionales para sistemas software
3. Aplicar métodos de desarrollo actuales en entornos de desarrollo de mediana escala
4. Identificar los principales conceptos de Testing de Software en el diseño e implementación de sistemas software

IV.- CONTENIDOS

1. Introducción
2. Principios de Ingeniería de Software
3. Procesos de Desarrollo
 - a. Modelos de Desarrollo Tradicionales
 - b. Proceso de Desarrollo RUP
 - c. Métodos Ágiles
4. Testing de Software

V.- METODOLOGÍA

La asignatura se realizará en modalidad presencial, con el apoyo de plataformas en línea (Canvas). En ella, semanalmente, se publicará material que permitirá revisar los contenidos de la asignatura. Se habilitarán foros de discusión, donde los estudiantes podrán formular comentarios y preguntas sobre los contenidos semanales.

Se realizarán clases teóricas, donde se revisarán los contenidos de la asignatura, se presentarán ejemplos y se resolverán dudas. En las clases prácticas, se revisará la aplicación de estos conceptos orientados a casos prácticos y al proyecto semestral.

Para organizar mejor su tiempo, se sugiere a los y las estudiantes el destinar las horas de clase establecidas para revisar el material y formular preguntas y comentarios. En las horas de práctica, se insta a realizar coordinaciones intensivas con sus compañeros y compañeras de equipo.

La asignatura contempla el desarrollo de un proyecto en equipo. Se utilizarán plataformas electrónicas de coordinación, que ya se han utilizado exitosamente en semestres anteriores, y que permiten una visualización del grado de avance grupal e individual.

VI.- EVALUACIÓN

La Nota de Presentación (NP) será calculada a partir de las siguientes evaluaciones, con sus correspondientes ponderaciones.

- Proyecto semestral (en equipo): 75%. Se realizará un proyecto de desarrollo semestral, basado en un conjunto de requerimientos que serán desarrollados siguiendo un proceso Scrum. Se realizarán tres iteraciones con sus correspondientes entregas, con ponderaciones incrementales de 20%, 25% y 30%, respectivamente
- Investigación y Discusión: 25%. Grupalmente, se asignarán temas relevantes de Ingeniería de Software a ser investigados, expuestos y discutidos por los alumnos.

El cálculo de la nota final (NF) se realiza como sigue:

1. Si $NP \geq 3,95$, $NF = NP$
2. Si $NP < 3,95$, el estudiante reprueba la asignatura. Para aprobarla, debe rendir una evaluación de recuperación (ER): - $NF = NP * 0,60 + ER * 0,40$

Entrega de evaluaciones

Las evaluaciones cuentan con un plazo de entrega anunciado en este syllabus, el cual, de no mediar razones de fuerza mayor, no es postergable. Razones justificadas podrán otorgar un plazo mayor, con una penalización de 1 décima por día de atraso.

La no entrega injustificada de evaluaciones y controles programados significarán una evaluación final de NCR. En casos debidamente justificados, las entregas y controles faltantes podrán ser reemplazados por una evaluación adicional a concertar con el profesor o con la rendición de la evaluación de recuperación, que reemplazará la nota faltante.

Participación

La participación de los y las estudiantes en la asignatura y, principalmente, en el trabajo de su equipo es parte fundamental para el logro de los resultados de aprendizaje de esta. Por ello, el incumplimiento de este requisito es causal de reprobación de la asignatura. El/la o los/las integrantes de un equipo de trabajo que, sin una justificación atendible a su condición de estudiantes de últimos años de la carrera, no participen activamente en las actividades de proyecto semestral e investigación en equipo, serán evaluados con la calificación NCR, sin derecho a evaluaciones de recuperación o sustitutivas.

Plagio

Se considerará plagio cualquier intento de copia de trabajos de estudiantes, actuales o de otras ediciones de esta u otras asignaturas, como también la transcripción o traducción textual de trabajos, parcial o total. El uso de material bibliográfico debe ser debidamente referenciado. La copia de artefactos software, de cualquier nivel de abstracción, entre equipos distintos de trabajo, también se considerará plagio. Quienes efectúen la copia y quienes lo permitan, serán calificados con el concepto NCR en la asignatura.

Puntualidad

Se solicita puntualidad en clases y prácticas, y en la entrega de las evaluaciones consideradas

VII.- BIBLIOGRAFÍA Y MATERIAL DE APOYO

Bibliografía Básica:

- Kruchten, Philippe. The rational unified process: an introduction. Addison-Wesley Professional, 2004.
- Sommerville, Ian. Software Engineering. Addison Wesley, 2006.

Bibliografía Sugerida:

- Schwaber, Ken. Agile project management with Scrum. Microsoft Press, 2004.

Material de Apoyo:

- Presentaciones con contenidos de la asignatura
- Material complementario publicado en plataforma de la asignatura

VIII.- PLANIFICACIÓN

Semana	Actividad	Responsable	TA	RA
01 04-mar	Presentación del Curso Ingeniería de Software, conceptos fundamentales.	Docente	5	Todos
	Revisión de Proyecto Semestral y guías generales para el semestre, conformación de grupos de trabajo	Estudiante	5	Todos
02 11-mar	Procesos de Desarrollo	Docente	5	1,2
	Asignación de roles, planificación de actividades, organización de entregables, inducción a plataforma	Estudiante	5	1-3
03 18-mar	Rational Unified Process Descripción <ul style="list-style-type: none">• Roles, actividades, artefactos• Revisión de disciplinas principales y complementarias• Diseño de Arquitectura• Gestión del Riesgo• Detalle de Fases (inicio, elaboración, implementación y transición)	Docente	5	3-4

	Definición de arquitectura del sistema Especificación de riesgos	Estudiante	5	Todos
04 25-mar	Métodos ágiles <ul style="list-style-type: none"> • Manifiesto • Historias de usuario • Test Driven Development 	Docente	5	1,3,4
	Definición de Historias de Usuario Condiciones de aceptación	Estudiante	5	1,3,4
05 01-abr	Métodos ágiles <ul style="list-style-type: none"> • Agilidad Remota • Ciclos iterativos • Inducción a herramienta 	Docente	5	3,4
	Sprint Planning	Estudiante	5	1,3,4
06 08-abr	Scrum: <ul style="list-style-type: none"> • Antecedentes • Principios • Conceptos clave 	Docente	5	3
	Comienzo Sprint1: Product y sprint backlog	Estudiante	5	1,3,4
07 15-abr	Scrum: Artefactos	Docente	5	3
	Sprint 1	Estudiante	5	Todos
08 22-abr	Scrum: Roles	Docente	5	3
	Entrega Sprint 1	Estudiante	5	3,4
09 29-abr	Scrum: Ceremonias; Sprint 1: Review y Retrospective	Docente	5	3
		Estudiante	5	1,3,4
10 06-may	Scrum: Artefactos	Docente	5	3
	Comienzo Sprint 2	Estudiante	5	1,3,4
11 20-may	EVALUACIÓN: Investigación y Discusión	Docente	5	4
	Sprint 2	Estudiante	5	1,3,4
12 27-may	Calidad en Scrum;	Docente	5	4
	Entrega Sprint 2	Estudiante	5	3
13 03-jun	Gestión de Calidad Sprint 2 Review y Retrospective	Docente	5	4
	Comienzo Sprint 3	Estudiante	5	1,3,4
14	Testing de Software	Docente	5	4

10-jun	Sprint 3	Estudiante	5	1,3,4
15 17-jun	Testing de Software	Docente	5	1,3, 4
	Entrega Sprint 3	Estudiante	5	1,3,4
16 24-jun	Sprint 3: Review y Retrospective	Docente y Estudiante	5	1,3, 4

IX.- OTROS

Plataforma del Curso

Como plataforma de comunicación de la asignatura, se utilizará el sistema Canvas, donde se podrán responder consultas, realizar discusiones y compartir material del curso.

Horario de clases:

Lunes de 10:15 a 11:45 hrs (TM3-4); miércoles de 9:15 a 10:00, IS 2-3

Prácticas: Viernes, de 15:15 a 17:00 hrs, IS 2-2

Horario de consultas:

Lunes 12:15 a 13:00 hrs, Of. 303. En cualquier momento, se podrán formular consultas a través la plataforma Canvas en cualquier momento. Se insta a que las consultas sean públicas, para compartirlas con el resto de la clase. Serán respondidas principalmente en el horario de clases y prácticas.