

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA  
FACULTAD DE INGENIERIA  
ESCUELA DE CIENCIAS y SISTEMAS



**NOMBRE DEL CURSO**  
ANÁLISIS Y DISEÑO DE SISTEMAS 1

<b>CODIGO:</b>	<b>0283</b>	<b>CREDITOS:</b>	<b>5</b>
<b>ESCUELA:</b>	Escuela de Ciencias y Sistemas	<b>ÁREA A LA QUE PERTENECE:</b>	Área de desarrollo de software
<b>PRE-REQUISITO:</b>	0774 - Sistemas de Bases de Datos 1	<b>POST REQUISITO:</b>	0785 - Análisis y Diseño de Sistemas 2
<b>CATEGORIA:</b>	Obligatorio	<b>SEMESTRE:</b>	Primer Semestre 2023
<b>CATEDRÁTICO:</b>	Ing. Edgar Rodas	<b>Tutor:</b>	Ludwing Paz
<b>EDIFICIO:</b>	MEET	<b>SECCIÓN:</b>	B
<b>SALON DEL CURSO:</b>	SALON 34	<b>SALON DEL LABORATORIO:</b>	SALON 22
<b>HORAS POR SEMANA DEL CURSO:</b>	5	<b>HORAS POR SEMANA DEL LABORATORIO:</b>	2
<b>DÍAS QUE SE IMPARTE EL CURSO:</b>	Sábado	<b>DÍAS QUE SE IMPARTE EL LABORATORIO:</b>	Sábado
<b>HORARIO DEL CURSO</b>	07:10 - 12:10	<b>HORARIO DEL LABORATORIO</b>	12:10 - 13:50

**DESCRIPCIÓN DEL CURSO:**

En el laboratorio de Análisis y diseño de sistemas 1 se complementa el contenido dado en la clase magistral, se imparten temas para que los estudiantes puedan implementar sus proyectos de forma mucho más eficiente, con la utilización de nuevas herramientas y técnicas que serán de mucha ayuda para los estudiantes.

**OBJETIVO GENERAL:**

Lograr que el estudiante adquiera los conocimientos necesarios para poder analizar y diseñar un sistema conforme a las nuevas tecnologías y herramientas que actualmente están disponibles, adoptando para ellos buenas prácticas de análisis y de programación de sistemas

**OBJETIVOS ESPECÍFICOS:**

- Que el estudiante utilice de manera correcta las herramientas de control de versiones
- Que el estudiante pueda utilizar técnicas de branqueo para facilitar la administración de un repositorio que se trabaja en equipo.
- Que el estudiante pueda identificar de forma precisa los requerimientos de un sistema por medio de historias de usuario.
- Que el estudiante pueda hacer uso de herramientas para la administración de requerimientos
- Que el estudiante pueda implementar de forma correcta las pruebas unitarias y conocer la importancia que estas tienen en el desarrollo de una aplicación.
- Que el estudiante pueda aplicar de forma correcta las pruebas unitarias y de integración para la parte de testing de una aplicación.
- Que el estudiante pueda utilizar de forma correcta las herramientas de CI/CD en un Proyecto.

**HABILIDADES:**

- Que el estudiante sea capaz de investigar, comprender y aplicar los conocimientos adquiridos durante los cursos anteriores para analizar sistemas de forma óptima.
- Que el estudiante sea capaz de analizar un enunciado para la toma de requerimientos de software para la implementación de un sistema cumpliendo las etapas del ciclo de vida del software.
- Extraer y representar de forma escrita los requerimientos para construir una solución para un proyecto de software cumpliendo con los requisitos necesarios de un problema propuesto.

**COMPETENCIAS:**

- Que el estudiante pueda aplicar técnicas de ramificación de forma correcta.
- Que el estudiante sea capaz de analizar y determinar los requerimientos de software.
- Resolver y reconocer los problemas comunes que se presentan en la implementación de una solución.
- Capacidad de adaptarse a los cambios que el usuario pueda hacer de los requerimientos iniciales del software.

**METODOLOGIA:**

- Se realizarán exámenes cortos como parte de la evaluación del contenido del laboratorio.
- Se realizarán actividades como prácticas y proyectos con la intención que el estudiante conozca y aplique los conocimientos y herramientas impartidas en el laboratorio.
- Se dejarán tareas de investigaciones para que el estudiante pueda complementar y comprender de mejor manera los temas tratados en el laboratorio.
- Se harán preguntas a los integrantes de los grupos durante las calificaciones con el objetivo de verificar que todos tengan los conocimientos que se desarrollaron en dicha actividad.

**EVALUACIÓN DEL RENDIMIENTO ACADEMICO:**

De acuerdo con el Normativo de Evaluación y Promoción del estudiante de Pregrado de la Facultad de Ingeniería, se procederá así:

**EVALUACIÓN DEL RENDIMIENTO ACADÉMICO:** El laboratorio se evalúa sobre una nota de 100 puntos, la nota mínima de promoción es de 61 puntos, y se debe de cumplir con las normas establecidas para poder aprobar el curso.

<b>ACTIVIDAD</b>	<b>PONDERACIÓN</b>
Tareas	10 pts
Cortos	10 pts
Practica 1	15 pts
Practica 2	15 pts
Proyecto FASE 1	15 pts
Proyecto FASE 2	25 pts
Examen Final	10 pts
<b>TOTAL</b>	<b>100 pts</b>

**REQUISITOS PARA APROBAR EL LABORATORIO:**

- El laboratorio se gana con 61% de la nota.
- Las tareas, prácticas y proyectos se entregan en la fecha indicada. Es de hacer notar que la PUNTUALIDAD es importante en la entrega de cualquiera de las actividades, de no cumplirse la nota de la actividad se verá afectada, así como también queda a discreción del auxiliar el recibir alguna fuera del tiempo estipulado.
- Cualquier copia de prácticas, tareas o fases de proyecto, será sancionada con la pérdida automática del laboratorio y la sanción correspondiente por parte de la Escuela de Sistemas
- El grupo que no entregue una fase del proyecto no tendrá derecho a entregar la siguiente.

Si alguna persona no puede asistir a la calificación de un proyecto o práctica, debe informar tanto al auxiliar como al catedrático el motivo de su falta, luego se procederá a determinar si se puede o no realizar la calificación del proyecto o práctica y bajo qué términos se efectuará si se realiza. Solamente el catedrático puede autorizar la calificación de algún proyecto o práctica después de la fecha establecida, acompañando la solicitud con una prueba que demuestre el motivo por el cual no pudo asistir a la misma.

## **CONTENIDO**

### **INTRODUCCIÓN**

Ingeniería de software  
Atributos del buen software  
Actividades de la ingeniería del software

### **UNIDAD 1: Metodologías de Desarrollo**

- 1.1 Introducción
- 1.2 Metodologías Robustas
  - 1.2.1 Cascada
  - 1.2.1 Prototipo
  - 1.2.1 Espiral
- 1.3 Metodologías Ágiles
  - 1.3.1 Scrum
  - 1.3.1 Kanban
  - 1.3.1 XP
- 1.4 Requerimientos
  - 1.4.1 Introducción
  - 1.4.2 Historias de usuario
  - 1.4.3 Casos de uso
  - 1.4.4 Herramientas para la Administración de requerimientos
- 1.5 Jira
  - 1.5.1 Introducción
  - 1.5.2 Flujos de Trabajo
  - 1.5.3 métricas ágiles
  - 1.5.4 Confluence

### **UNIDAD 2: Control de versiones**

- 2.1 Introducción
- 2.2 Sistemas de control de versiones
  - 2.2.1 Local
  - 2.2.2 Centralizados
  - 2.2.3 Distribuidos
- 2.3 Repositorios
  - 2.3.1 Públicos
  - 2.3.1 Privados
- 2.4 Git
  - 2.4.1 Introducción
  - 2.4.2 Comandos

## 2.5 GitFlow

2.5.1 Flujo de trabajo

2.5.2 Características

2.5.3 Ramas

2.5.4 Colaboradores

## **UNIDAD 3: Pruebas unitarias**

3.1 Introducción

3.2 Características

3.3 Ventajas y Desventajas

3.4 Herramientas

## **UNIDAD 4: Pruebas funcionales y no funcionales**

4.1 Introducción

4.2 Funcionales

4.3 No funcionales

## **UNIDAD 5: Integración continua**

5.1 Introducción

5.2 Beneficios

5.3 Entrega y despliegue continuo

5.4 Herramientas

5.5 Aplicación

## **BIBLIOGRAFÍA:**

- “Ingeniería de software”, Novena Edición, PEARSON EDUCACIÓN, México, 2011
- “Version Control with Subversion”, Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michal Pilato.
- “Continuous Delivery”, Jez Humble, David Farley