

**Nombre del Curso: Análisis y Diseño de Sistemas 1**

<b>Código:</b> 0283	<b>Créditos:</b> 5
<b>Escuela:</b> Escuela de Ciencias y Sistemas	<b>Área:</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>Categoría:</b> Obligatorio	<b>Semestre:</b> 1er Semestre 2025
<b>Docente:</b> Edgar Francisco Rodas Robledo	<b>Auxiliar:</b> Ronald Geovany Ordoñez Xiloj
<b>Edificio:</b> Meet	<b>Sección:</b> B
<b>Salón del curso:</b> Virtual	<b>Salón de laboratorio:</b> Virtual
<b>Horas por semana del curso:</b> 4 horas	<b>Horas por semana del laboratorio:</b> 2 horas
<b>Días en que se imparte el curso:</b> <b>Sábados</b>	<b>Días que se imparte el laboratorio:</b> Miércoles
<b>Horario del curso:</b> <b>7:10- 10:30</b>	<b>Horario del laboratorio:</b> <b>10:40 - 12:20</b>

**1. Descripción del laboratorio**

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

**2. Objetivos**

**Generales:**

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

**Específicos**

Obtener habilidades para identificar, documentar y gestionar los requerimientos de un proyecto de manera estructurada y eficiente; apoyarse mediante una herramienta especializada que facilite la planificación, seguimiento y colaboración durante todo el ciclo de vida de un proyecto.

Adquirir habilidades prácticas para aplicar una metodología de desarrollo ágil en la ejecución de proyectos.

Asegurar la calidad del software y validar su funcionamiento en diferentes niveles, garantizando una cobertura integral de pruebas que permita detectar errores tempranamente y mejorar la confiabilidad del sistema en su conjunto

2

0

2

5

### 3. Metodología

Se realizarán exámenes cortos como parte de la evaluación del contenido del laboratorio.

Se realizarán actividades como práctica y proyectos con la intención de que el estudiante conozca y aplique los conocimientos y herramientas impartidas en el laboratorio.

Se dejarán tareas 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.

### 4. Competencias Terminales

Al finalizar el laboratorio el estudiante desarrolla las siguientes competencias:

Utilizar de manera correcta las herramientas de control de versiones

Manejar técnicas de "branching" para facilitar la administración de un repositorio que se trabaja en equipo.

Identificar de forma precisa los requerimientos de un sistema e introducir el uso de herramientas para la administración de requerimientos

Implementar de forma correcta las pruebas unitarias y conocer la importancia que estas tienen en el desarrollo de una aplicación.

Introducir al manejo de las herramientas de CI/CD en un Proyecto.

### 5. Observaciones

- 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.
- 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.
- El estudiante que no entregue un proyecto o una práctica 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.
- Es obligatorio aprobar el laboratorio con una nota mínima de 61 puntos para tener derecho a examen final y aprobación del curso.
- Es necesario tener un mínimo del 80% de asistencia al laboratorio para aprobar el mismo.

## 6. Contenido Temático del Laboratorio

1. Control de Versiones
  - 1.1. Introducción
  - 1.2. Sistemas de Control de Versiones
    - 1.2.1. Local
    - 1.2.2. Centralizados
    - 1.2.3. Distribuidos
  - 1.3. Repositorios
    - 1.3.1. Públicos
    - 1.3.2. Privados
  - 1.4. Git
    - 1.4.1. Introducción
    - 1.4.2. Buenas Prácticas en Git
    - 1.4.3. Estructuras de un commit
    - 1.4.4. Comandos
  - 1.5. GitFlow
    - 1.5.1. Flujo de trabajo
    - 1.5.2. Características
    - 1.5.3. Ramas
    - 1.5.4. Colaboradores
  - 1.6. Versionamiento Semántico
  - 1.7. Releases
2. Metodologías de Desarrollo
  - 2.1. Introducción
  - 2.2. Factores de selección de una metodología
  - 2.3. Tendencias
  - 2.4. Metodologías Robustas
    - 2.4.1. Introducción
    - 2.4.2. Ciclo de vida de desarrollo de software
    - 2.4.3. Cascada
    - 2.4.4. Prototipo
    - 2.4.5. Espiral
    - 2.4.6. Aplicaciones
  - 2.5. Metodologías Ágiles
    - 2.5.1. Introducción y principios el manifiesto ágil
    - 2.5.2. Ágil Inception
    - 2.5.3. Iteraciones y entregas incrementales
    - 2.5.4. Scrum
    - 2.5.5. Kanban
    - 2.5.6. XP
    - 2.5.7. Scrum
      - 2.5.7.1. Introducción
      - 2.5.7.2. Roles
      - 2.5.7.3. Artefactos
      - 2.5.7.4. Eventos
      - 2.5.7.5. Ciclo de vida
      - 2.5.7.6. Técnicas de estimación
  - 2.6. Herramientas de Gestión de Proyectos
    - 2.6.1. Introducción
    - 2.6.2. Jira
    - 2.6.3. Trello
    - 2.6.4. Asana
    - 2.6.5. Comparación y selección de herramientas

3. Diseño y Desarrollo de Sistemas
  - 3.1. Análisis de Requerimientos y Prototipado
    - 3.1.1. Introducción
    - 3.1.2. Ingeniería de requerimientos
    - 3.1.3. Historias de usuarios
    - 3.1.4. Casos de uso expandidos
    - 3.1.5. Requerimientos funcionales y no funcionales
    - 3.1.6. Prototipado rápido
    - 3.1.7. Iteración y mejora del prototipo
  - 3.2. Modelo de Sistemas
    - 3.2.1. Introducción
    - 3.2.2. Diagrama de caso de uso
    - 3.2.3. Diagrama de clases
    - 3.2.4. Diagrama de secuencias
    - 3.2.5. Patrones de diseño
    - 3.2.6. Arquitectura
    - 3.2.7. Documentación del diseño del sistema
  - 3.3. Arquitectura de Software
    - 3.3.1. Introducción
    - 3.3.2. Modelos de arquitectura
    - 3.3.3. Modelo vista controlador
    - 3.3.4. Microservicios
    - 3.3.5. Arquitectura basada en eventos
    - 3.3.6. Diseño para la escalabilidad y rendimiento
    - 3.3.7. Documentación y comunicación de la arquitectura
4. Herramientas, Pruebas e Integración Continua (CI/CD)
  - 4.1. Pruebas Unitarias
    - 4.1.1. Introducción
    - 4.1.2. Características
    - 4.1.3. Ventajas y Desventajas
    - 4.1.4. Conceptos de aserciones
    - 4.1.5. Simular comportamientos y dependencias externas
    - 4.1.6. Cobertura de código (code coverage) y métricas
    - 4.1.7. Herramientas
  - 4.2. Pruebas Funcionales y no Funcionales
    - 4.2.1. Introducción
    - 4.2.2. Tipo de Pruebas funcionales
    - 4.2.3. Pruebas E2E (End to End)
    - 4.2.4. Pirámide de pruebas
    - 4.2.5. Herramientas de pruebas funcionales
    - 4.2.6. Tipos de pruebas no funcionales
    - 4.2.7. Beneficios de realizar pruebas
  - 4.3. Docker
    - 4.3.1. Imágenes y contenedores.
    - 4.3.2. Uso del Docker Hub y repositorios públicos.
    - 4.3.3. Introducción a Docker Compose
  - 4.4. Integración Continua
    - 4.4.1. Definición y conceptos fundamentales
    - 4.4.2. Beneficios y objetivos de la Integración Continua
    - 4.4.3. Herramientas
    - 4.4.4. Aplicación

\*Nota: Además se abordarán temas complementarios que ayuden a los estudiantes a complementar sus habilidades en la elaboración y desarrollo del Software

2

0

2

5

## 7. Evaluación del Rendimiento Académico

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.

Actividad	Ponderación
Tareas	12 pts
Cortos	10 pts
Práctica	18 pts
Proyecto Fase 1	23 pts
Proyecto Fase 2	27 pts
Examen Final	10 pts

## 8. 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

2

0

2

5