

Nombre del Curso: Análisis y Diseño de Sistemas 1	
Código: 0283	Créditos 5
Escuela: Escuela de Ciencias y Sistemas	Área: Área de desarrollo de software
Pre requisito: 0774 - Sistemas de Bases de Datos 1	Post requisito: 0785 - Análisis y Diseño de Sistemas 2
Categoría: Obligatorio	Semestre: 2do Semestre 2023
Docente: William Samuel Guevara Orellana Edgar Francisco Rodas Robledo	Auxiliar: Bryan Alexander Portillo Alvarado Ludwing Gabriel Paz Hernandez
Edificio: MEET	Sección: A , B
Salon del curso: Salón 21 y Salon 31	Salon de laboratorio: Salon 10 y Salon 34
Horas por semana del curso 4 horas	Horas por semana del laboratorio: 2 horas
Días en que se imparte el curso: Martes y Jueves; Sabado	Días que se imparte el laboratorio: Sábado (Seccion A y B)
Horario del curso: 7:10 - 8:50(Sección A); 7:10 - 10:3 (Sección B)	Horario del laboratorio: 12:10 - 13:50 (Sección A y 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

3. Metodología

Se realizarán actividades 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.

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.
- Se requiere de un 90% de asistencia al laboratorio.
- Es obligatorio aprobar el laboratorio con una nota mínima de 61 puntos para tener derecho a examen final y aprobación del curso.

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. 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

2. Metodologías de Desarrollo

- 2.1. Introducción
- 2.2. Metodologías Robustas
 - 2.2.1. Cascada
 - 2.2.2. Prototipo
 - 2.2.3. Espiral
- 2.3. Metodologías Ágiles
 - 2.3.1. Kanban
 - 2.3.2. XP
 - 2.3.3. Scrum
 - 2.3.3.1. Estimaciones
 - 2.3.3.1.1. Herramientas de Estimación
 - 2.3.4. Agile Inception
- 2.4. Herramientas de Gestión de Proyectos
 - 2.4.1. Jira
 - 2.4.1.1. Introducción
 - 2.4.1.2. Flujos de Trabajo

3. Pruebas Unitarias

- 3.1. Introducción
- 3.2. Características
- 3.3. Ventajas y Desventajas
- 3.4. Conceptos de aserciones
- 3.5. Simular comportamientos y dependencias externas
- 3.6. Cobertura de código (code coverage) y métricas
- 3.7. Herramientas

4. **RUP**
 - 4.1. Conceptos básicos
 - 4.2. Fases del RUP relacionadas con la administración de requerimientos
 - 4.3. Roles y responsabilidades en el proceso
 - 4.4. Actividades y tareas involucradas
 - 4.5. Identificación y Documentación de Requerimientos
 - 4.5.1. Documento de Visión
 - 4.5.2. Casos de Uso
 - 4.5.3. Diagramas de Clases
 - 4.5.4. Diagramas de Secuencia
 - 4.5.5. Modelos de Datos
 - 4.5.6. Prototipos y Maquetas
 - 4.5.7. Diagramas de Despliegue

5. **Pruebas Funcionales y no Funcionales**
 - 5.1. Introducción
 - 5.2. Funcionales
 - 5.3. No Funcionales

6. **Pruebas E2E (End to End)**
 - 6.1. Introducción a las pruebas extremo a extremo
 - 6.2. Diseño de escenarios de prueba completos
 - 6.3. Herramientas para pruebas extremo a extremo

7. **Docker**
 - 7.1. Imágenes y contenedores.
 - 7.2. Uso del Docker Hub y repositorios públicos.
 - 7.3. Introducción a Docker Compose

8. **Integración Continua**
 - 8.1. Definición y conceptos fundamentales
 - 8.2. Beneficios y objetivos de la Integración Continua
 - 8.3. Herramientas
 - 8.4. Aplicación

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

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	10 pts
Proyecto 1	25 pts
Proyecto 2 Fase 1	25 pts
Proyecto 2 Fase 2	30 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