



UNIVERSIDAD DE LIMA  
FACULTAD DE INGENIERÍA Y ARQUITECTURA  
CARRERA DE INGENIERÍA DE SISTEMAS

## Plan de Trabajo

**CURSO: INGENIERÍA DE SOFTWARE 2**

**Semestre: 2024-1**

### Resultados del Estudiante

**2.2** Aplicar el diseño de ingeniería tomando en cuenta la salud pública, seguridad y bienestar, así como factores globales, culturales, sociales ambientales y económicos.

**4.1** Reconocer las responsabilidades éticas y profesionales en el ámbito de ingeniería, analizando el impacto de las soluciones de ingeniería en los contextos globales, económicos, ambientales y sociales.

**5.1** Demuestra habilidades efectivas de trabajo en equipo contribuyendo al logro de los objetivos del grupo.

Semana	Avance de la asignatura (syllabus)	Detalle	Controles/Asignaciones y Evaluaciones	Criterio ABET (colocar letra y número)
1	DISEÑO ORIENTADO A OBJETOS USANDO UML Métodos de Ingeniería de Software. Métodos formales. Principios de diseño. Proceso del diseño de software. Modelado y programación orientada a objetos. Diseño detallado. Modelos de diseño. Aspectos de implementación del software.	Actividades: <ul style="list-style-type: none"><li>- Presentar el curso: Sílabo, Evaluaciones, Plan de Trabajo</li><li>- Revisar los proyectos y grupos de trabajo y especificar sus actividades y entregables.</li><li>- Revisar las principales herramientas del curso</li><li>- Guía de Github como repositorio de código fuente</li><li>- Guía de herramienta de modelado Visual Paradigm Community Edition</li><li>- Guía de Herramienta para el monitoreo del desarrollo de aplicaciones de Software (Jira, Plantillas Excel, Trello,</li></ul>		

	Medidas de calidad del diseño OO.	<p>Scrum, Kanban y XP)</p> <ul style="list-style-type: none"> <li>- Analizar la aplicación de conceptos de POO (Herencia, Polimorfismo, Abstracción, Interfaces, Relaciones entre clases) en el diseño del Software.</li> <li>- Implementación de ejercicios de POO</li> <li>- Cada equipo debe elaborar el diseño simple para el proyecto de software</li> <li>- Cada equipo debe elaborar un modelo de datos para el desarrollo del proyecto de software</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Presentación sobre diseño OO con UML</li> </ul>		
2	<p>PRINCIPIOS DEL DISEÑO ORIENTADO A OBJETOS</p> <p>Principios DRY (Don't Repeat Yourself). Principio Law of Demeter, Principios SOLID.</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Analizar la aplicar principios SOLID en el diseño del Software</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Presentación Sobre Principios de Demeter y DRY (Don't Repeat Yourself). Principios SOLID</li> </ul>		
3	<p>DISEÑO DE SOFTWARE CON PATRONES</p> <p>Patrón de Diseño en Ingeniería del Software. Categoría de patrones. Patrones creacionales: Factory Method, Abstract Factory, Singleton.</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Analizar la aplicación de Patrones Adapter, Factory, Singleton y Fachada en de diseño de Software</li> <li>- Implementar patrones de diseño Adapter, Factory, Singleton y Fachada para casos de estudio</li> </ul> <p>Recursos</p> <ul style="list-style-type: none"> <li>- Guía de Programación sobre patrones de diseño de Software: Adapter, Factory, Singleton y Fachada.</li> </ul>		

4	PATRONES ESTRUCTURALES Y DE COMPORTAMIENTO Patrones Estructurales: Adapter, Facade. Patrones de comportamiento: Strategy, Command.	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Analizar e implementar patrones estructurales y de comportamiento en de Diseño de Software</li> <li>- Seguimiento a los equipos de trabajo sobre el avance del proyecto</li> </ul> <p>Recursos</p> <ul style="list-style-type: none"> <li>- Guía de Programación sobre patrones estructurales y de comportamiento en de Diseño de Software.</li> <li>- Guía de diagrama de estados.</li> </ul>		
5	DISEÑO DEL SISTEMA Y ARQUITECTURA DEL SOFTWARE Métodos de Ingeniería de Software. Métodos formales. Principios de diseño. Proceso del diseño de software. Proceso de diseño de la arquitectura: Requerimientos arquitecturales, Atributos de calidad, Especificación de requisitos de atributos de calidad. Estilos de arquitectura. Software de gestión de versiones.	<p>Actividades:</p> <ul style="list-style-type: none"> <li>- Analizar y reflexionar sobre el diseño la arquitectura del Software.</li> <li>- Cada equipo debe diseñar la arquitectura del Software para su proyecto</li> <li>- Detallar las especificaciones del trabajo recopilatorio sobre Ingeniería del Software y Sostenibilidad</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Presentación de diapositivas sobre Diseño de Arquitectura de Software</li> </ul>	EVC: Evaluación continua Nota 1: Evaluación en clase / Individual	
6	DOCUMENTACIÓN DE LA ARQUITECTURA Documentación de la arquitectura: Documentación de la arquitectura modelo 4+1 vistas. Documentación de la arquitectura modelo C4.	<p>Actividades:</p> <ul style="list-style-type: none"> <li>- Documentar la arquitectura de un software basado vistas y puntos de vistas</li> <li>- Seguimiento a los equipos de trabajo sobre el avance del proyecto</li> <li>- Cada grupo debe plantear la documentación de su Arquitectura de Software</li> </ul> <p>Recursos:</p>		

		<ul style="list-style-type: none"> <li>- Presentación sobre documentación de arquitectura de Software</li> <li>- Plantilla para documentar la arquitectura de Software</li> <li>- Guía sobre Diagramas de despliegue y de subsistemas.</li> </ul>		
7	<b>PRUEBAS UNITARIAS</b> Concepto de pruebas unitarias. Pruebas unitarias estáticas y dinámicas. Herramientas para pruebas. Pruebas unitarias en eXtreme Programming.	Actividades <ul style="list-style-type: none"> <li>- Analizar y reflexionar sobre la ejecución de pruebas de software</li> <li>- Cada equipo debe presentar la propuesta de implementación de pruebas unitarias</li> </ul> Recursos <ul style="list-style-type: none"> <li>- Presentación pruebas unitarias del software</li> </ul>		
8	<b>DISEÑO ÁGIL</b> Conceptos de diseño ágil. principios de diseño de paquetes. Arquitectura en proyectos ágiles.	Actividades <ul style="list-style-type: none"> <li>- Analizar el diseño ágil frente al diseño con enfoques no ágiles</li> </ul> Recursos <ul style="list-style-type: none"> <li>- Presentación sobre Diseño ágil</li> </ul>		
9	<b>REFACTORIZACIÓN</b> Conceptos de Refactorización. Principios en Refactorización. Bad Smells in Code. Anti-patronos.	Actividades <ul style="list-style-type: none"> <li>- Analizar la aplicación de Refactoring y anti-patronos en de Diseño de Software</li> <li>- Analizar Principales smells en el desarrollo de Software.</li> <li>- Aplicar las técnicas de reconocimiento de smells en el desarrollo de Software</li> </ul> Recursos <ul style="list-style-type: none"> <li>- Presentación sobre Refactoring y Anti-patronos</li> </ul>	EC: Evaluación continua Nota 2: Entrega de trabajo / Individual y grupal	

10	<p>ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE</p> <p>Validación y verificación (V&amp;V). Conceptos básicos de pruebas. Tipos de pruebas: Pruebas de funcionalidad, Pruebas de usuario, Pruebas de performance y carga, Pruebas de seguridad. Pruebas de caja blanca y Negra. Diseño de casos de prueba. Diseño de Plan de pruebas.</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Analizar y reflexionar sobre la validación y verificación del Software</li> <li>- Analizar y reflexionar sobre los diferentes tipos de pruebas de software</li> <li>- Cada equipo debe formular las pruebas de usuario y pruebas funcionales</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Presentación Sobre pruebas de software</li> <li>- Guía de plan de pruebas de software</li> </ul>		
11	<p>PRUEBAS RENDIMIENTO</p> <p>Introducción a las Pruebas de Rendimiento. Tipos de Pruebas de Rendimiento Performance, Stress y Carga. Concurrencia, Latencia y throughput</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Analizar y reflexionar sobre los tipos de prueba de rendimiento.</li> <li>- Cada equipo debe formular las pruebas de Rendimiento</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Presentación Sobre pruebas de rendimiento de software</li> <li>- Guía de plan de pruebas de software</li> </ul>		
12	<p>FRAMEWORK PARA PRUEBAS UNITARIAS</p> <p>Elementos del framework para pruebas unitarias. Automatización de pruebas unitarias.</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Implementar pruebas unitarias con Junit</li> <li>- Seguimiento a los equipos de trabajo sobre el avance del proyecto</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Guía de pruebas unitarias con Junit 4</li> </ul>	EVC: Evaluación continua Nota 3: Evaluación en clase / Individual	
13	<p>PRUEBAS DE MUTACIÓN Y PRUEBAS DE INTEGRACIÓN</p> <p>Fundamentos de pruebas de mutación. Tipos de pruebas de integración. Framework para mocks. Desarrollo basado en pruebas.</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Implementar pruebas unitarias aplicando mocks.</li> <li>- Analizar e implementar pruebas de integración de software</li> </ul> <p>Recursos:</p> <ul style="list-style-type: none"> <li>- Presentación Sobre pruebas de mutación</li> </ul>		

		<ul style="list-style-type: none"> <li>- Guía de uso de mocks</li> </ul>		
14	<p>FUNDAMENTOS DE INTEGRACIÓN CONTINUA</p> <p>El problema de la entrega de software. Implementación de integración continua. Implementación de una estrategia de prueba</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Reflexionar y analizar conceptos de Integración continua</li> <li>- Reflexionar y analizar conceptos de mejora continua</li> <li>- Lectura sobre integración continua</li> <li>- Seguimiento a los equipos de trabajo sobre el avance del proyecto</li> </ul> <p>Recursos</p> <ul style="list-style-type: none"> <li>- Presentación de diapositivas sobre integración continua y mejora continua</li> <li>- Texto: Systematic Literature Review on DevOps Capabilities and Areas</li> </ul>		
15	<p>PROYECTO</p> <p>Proyecto integrador del curso</p>	<p>Actividades</p> <ul style="list-style-type: none"> <li>- Sustentación de Proyectos por equipos.</li> </ul> <p>Recursos</p> <ul style="list-style-type: none"> <li>- Equipo Multimedia</li> </ul>	<p>EVC: Evaluación continua</p> <p>Nota 4: Entrega de trabajo / Individual y grupal</p>	ABET 2.1
16	Retroalimentación del aprendizaje.			