Facultad de Ingeniería - Universidad Nacional de Cuyo P1- PROGRAMA DE ASIGNATURA			
Asignatura:	**Ingeniería de Software II**		
Profesor Titular:			
Carrera:	**Licenciatura en Ciencias de la computación**		
Año: 2019	**Semestre: 6to**	**Horas Semestre: 80**	**Horas Semana: 5**

OBJETIVOS

- Aplicar los procesos para el diseño de aplicaciones informáticas que se ajusten a las necesidades de las organizaciones.
- Especificar los componentes y funciones de un sistema con suficiente detalle para la construcción del software.
- Aplicar técnicas y metodologías de diseño con el propósito de definir un dispositivo, proceso o sistema con el detalle que viabilice la posterior fase de codificación de aplicaciones.
- Aplicar la metodología del paradigma de Orientado a Objetos y el uso de patrones en el diseño de sistemas de software.
- Reconocer las principales herramientas de verificación y validación de software y su utilidad en las diferentes fases del desarrollo de sistemas
- Organizar el diseño de pruebas que verificarán el posterior correcto funcionamiento de los programas, ajustado a los requisitos de análisis y diseño.
- Reconocer los problemas que implican los sistemas de tiempo real.
- Reconocer los diferentes procesos de mantenimiento en vistas a la ejecución de técnicas de reingeniería de software.

CONTENIDOS

Unidad 1: Introducción al diseño de software

1.A. Definición y alcances

Introducción al Diseño. Definición de Diseño y Arquitectura de Software. Fases del diseño. Diseño Lógico y Diseño Físico. Propósito del Diseño de Sistemas. Diferencias entre Arquitectura y Diseño.

1.B Metodologías de diseño

Diseño orientado a Objeto, basado en ocultamiento de la información, basado en tipos abstractos, orientado a aspectos y centrado en el usuario.

1.C Patrones de Software

Concepto de patrón. Propósito. Ventajas y desventajas de su uso. Categorías de patrones: arquitecturales, de diseño, idiomáticos (Idioms).

Unidad 2: Arquitecturas de Software.

2.A Problemática de las arquitecturas de Software

Introducción a la problemática de arquitecturas. Estilos arquitectónicos y arquitecturas de referencia. Principales estilos arquitectónicos: Flujo de datos (Data-Flow), Cliente-Servidor, Llamada y Retorno (Call and Return), Repositorio, Guiados por eventos (Event-Driven).

Especificación de Atributos de Calidad. Quality Attribute Workshops (QAW) y Architecture Tradeoff Analysis Method (ATAM).

2.B Principales patrones arquitectónicos

Sistemas Multi-capas. Modelo-Vista-Controlador. Inyección de Dependencias. Tuberías y filtros (Pipe-and-filter). Pizarra (Blackboard). Orientada a Servicios (SOA). Entre Pares (Peer-2-Peer).

Unidad 3: Diseño de Software orientado a objetos

3.A Clasificación de patrones de Diseño

Patrones creacionales. Patrones estructurales. Patrones de comportamiento. Relaciones entre los diferentes clases de patrones. Refactorización mediante patrones.

3.B Principales patrones de diseño

Singleton. Fábrica. Composite. Estrategia. Observador. Adaptador. Comando. Decorador. Fachada. Puente. Plantilla. Iterador.

3.B Gestión de Persistencia

Definición y Concepto de Persistir. Mapeo Objeto - Relacional (ORM). Arquitectura de un framework de persistencia. Patrones para la gestión de la persistencia.

Unidad 4: Verificación y Validación del Software

4.A Técnicas de revisión

Planificación de la verificación y validación. Efecto de los defectos del software en el costo. Inspección de Software. Análisis Estático Automatizado. Revisiones de pares.

4.B Pruebas de Software

El proceso de pruebas. Planificación de las pruebas. Estrategias de pruebas. Pruebas de caja-negra. Pruebas Estructurales. Pruebas de Interfaces. Pruebas Unitaria, Integración y Sistema. Diseño de pruebas unitarias y de integración con mock objects. Pruebas de Regresión y Automatización de Tests en el Diseño. Pruebas sobre Requerimientos No Funcionales. Performance, Carga, Stress.

Unidad 5: Sistemas de tiempo Real (STR)

5.A Introducción a los STR

Definición. Características. Ejemplos.

5.B STR: problemática de diseño e implementación

Diseño de STR. Lenguajes para STR y sus características necesarias. Fiabilidad y tolerancia a fallos. Planificadores y procesos. Comunicación y sincronización. Concurrencia.

Unidad 6: Mantenimiento y reingeniería del software

6.A Mantenimiento

Concepto de mantenimiento del software. Tipos de mantenimientos.

6.B. Reingeniería

El modelo de proceso de reingeniería de software. Actividades de reingeniería de software. Ingeniería inversa: ingeniería inversa para comprender datos, Ingeniería inversa para entender el procesamiento, Ingeniería inversa de interfaces de usuario. Reestructuración. Ingeniería hacia adelante para arquitecturas cliente-servidor y para arquitecturas orientadas a objetos

METODOLOGÍA DE ENSEÑANZA

La materia se organiza en clases teóricas, clases prácticas y actividades de laboratorio.

En las clases teóricas se brindan los contenidos fundamentales de la asignatura. Se desarrolla actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura. Se propicia el uso adecuado de conceptos, y de terminología científico-tecnológica.

Las estrategias pedagógicas permitirán al alumno:

- observar y analizar fenómenos y problemáticas propias del campo ocupacional,
- relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante y proponer situaciones que permitan al estudiante la integración de contenidos.

En las clases prácticas se ejercita sobre los temas cubiertos en la teoría, con especial énfasis en actividades de ingeniería del software aplicadas a casos reales. Llevar a cabo actividades prácticas que promuevan el desarrollo de habilidades para la experimentación, el planteamiento de hipótesis, y trabajo en equipo. Se utilizarán recursos pedagógicos basados en tecnologías de la información y la comunicación para permitir una interacción y comunicación docente-alumno continua y eficiente.

La resolución de problemas debe conducir al desarrollo de las competencias necesarias para la identificación y solución de problemas abiertos de ingeniería, entendiendo como tal

aquellas situaciones reales o hipotéticas cuya solución no es única y requiere la aplicación de los conocimientos de las ciencias básicas y de las tecnologías.

Se seleccionarán estrategias de enseñanza que fomenten actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.

Actividad	**Carga horaria por semestre**
Teoría y resolución de ejercicios simples	32
Formación práctica	
Formación Experimental – Laboratorio	18
Formación Experimental - Trabajo de campo	0
Resolución de problemas de ingeniería	30
Proyecto y diseño	0
Total	**80**

BIBLIOGRAFÍA

Bibliografía básica

Autor	Título	Editorial	Año	Ejemplares en biblioteca
Pressman, Roger	Ingeniería de Software. Un enfoque práctico	McGraw-Hill	2010	
Sommerville, Ian	Ingeniería de Software	Person Education	2011	
Gamma, Erich. Helm, Richard. Johnson, Ralph. Vlissides, John.	Patrones de diseño	Person Education	2003	

Bibliografía complementaria

Autor	Título	Editorial	Año	Ejemplares en biblioteca
Larman, Craig	UML y Patrones	Person Education	2004	
Burns, Alan. Wellings, Andy	Sistemas de tiempo real y lenguajes de programación	Addison Wesley	2003	
Rumbaugh, J. Jacobson, I. Booch, G.	El Lenguaje Unificado De Modelado. Manual De Referencia	Addison Wesley	2007	

EVALUACIONES (S/ Ord. 108-10_CS)

Se realizará una evaluación integral y continua durante el cursado.

Para obtener la regularidad de la materia el alumno deberá:

- Cumplir con un 75% de asistencia a clase.
- Aprobar el 80% de los trabajos prácticos, los que no sean entregados en tiempo y forma se considerarán desaprobados.
- Aprobar las actividades presentadas en la plataforma virtual educativa que fortalecerán los aprendizajes desarrollados en las clases teóricas.
- Durante el curso de la materia se ejecutará un proyecto de ingeniería real y se tomarán dos (2) exámenes parciales sobre el proyecto, ambos con posibilidad de recuperatorio. El alumno deberá aprobar los dos exámenes parciales o respectivos recuperatorios con un mínimo de sesenta por ciento (60%).

Evaluación sumativa de resultado: el examen final será oral y consistirá en una parte teórica (40%) y en una parte práctica (60%). El alumno deberá aprobar el examen final con un mínimo de seis (6) puntos para acceder a la aprobación de la materia.

FECHA, FIRMA Y ACLARACIÓN TITULAR DE CÁTEDRA