

- a. Aplicar los conceptos de la orientación a objetos en la construcción de aplicaciones de software.
- b. Desarrollar aplicaciones cuya implementación y estructura esté separada en capas lógicas.
- c. Formular e implementar modelos de clases basados en el lenguaje de modelado UML.
- d. Construir aplicaciones Web básicas.

4. Contenidos

1. Modelado de sistemas orientados a objetos con UML
 - 1.1. El lenguaje de modelado unificado (UML), importancia y utilización.
 - 1.2. Diagramas de Clase: Identificación de Clases (atributos, operaciones), herencia e interfaces.
 - 1.3. Relaciones de clases: dependencia, generalización, asociación (nombre, rol, multiplicidad, navegación, agregación y composición).
2. Manejo dinámico de memoria
 - 2.1. Concepto y ejemplos de estructuras de datos dinámicas y punteros.
 - 2.2. Concepto y utilización de las colecciones.
 - 2.2.1. Concepto y aplicación de Collection y Collections, ArrayList, Iterator
 - 2.2.2. Utilización de los métodos para manipular colecciones
3. Patrones de diseño
 - 3.1. Principios de diseño: definición, importancia y utilización.
 - 3.1.1. Principio abierto/cerrado
 - 3.1.2. Principio de inversión de dependencias
 - 3.1.3. Principio de segregación de interfaces
 - 3.1.4. Principio de responsabilidad única
 - 3.1.5. Principio de sustitución de Liskov
 - 3.2. Patrones de diseño: definición, características, importancia y utilización.
 - 3.3. Patrones de creación: concepto y utilización
 - 3.3.1. Abstract Factory (Fabrica abstracta)
 - 3.3.2. Builder (Constructor)
 - 3.3.3. Prototype (Prototipo)
 - 3.3.4. Singleton (Único)
 - 3.4. Patrones de comportamiento: concepto y utilización
 - 3.4.1. Observer (Observador)
 - 3.4.2. Memento (Recuerdo)
 - 3.4.3. Strategy (Estrategia)
 - 3.5. Patrones estructurales: concepto y utilización
 - 3.5.1. Composite (Compuesto)
 - 3.5.2. Decorator (Decorador)
4. Pruebas Unitarias
 - 4.1. Definición prueba de software, tipos de pruebas, pruebas unitarias
 - 4.2. Realización de Pruebas Unitarias utilizando JUnit
5. Hilos
 - 5.1. Concepto y características de hilos
 - 5.2. Creación, administración y destrucción de hilos
 - 5.3. Múltiples hilos
 - 5.4. Los hilos en las aplicaciones Cliente-Servidor
 - 5.5. Sincronización

6. Sockets
 - 6.1. Concepto y características de sockets
 - 6.2. Conexión de aplicaciones mediante sockets
 - 6.3. Sockets en aplicaciones Cliente - Servidor

7. Archivos especializados (XML)
 - 7.1. Definición de archivo XML, características y utilización
 - 7.2. Creación, lectura y escritura
 - 7.3. Parseo de sintaxis con archivo XML

8. Introducción al desarrollo de aplicaciones web
 - 8.1. Introducción a las aplicaciones web (definición, características)
 - 8.2. Historia de la Web
 - 8.3. Arquitectura básica web (cliente-servidor)
 - 8.4. Doctype, html, head, body
 - 8.5. Imágenes, listas, tablas, formularios, otros
 - 8.6. Estructura de documentos XHTML
 - 8.7. Estándares y Validación W3C
 - 8.8. Lenguajes del lado del servidor

5. Metodología

El curso presenta un eje de desarrollo teórico-práctico. El profesor desarrolla clases magistrales y realiza asignaciones (quices, tareas cortas, entre otros) para que los estudiantes apliquen los conceptos discutidos. Se realizan laboratorios durante el curso. Los estudiantes desarrollan proyectos programados donde ponen en práctica y amplían los conocimientos adquiridos en el curso. Además, se asignarán temas de investigación a los estudiantes.

6. Evaluación

Descripción	Porcentaje
Examen Parcial I	20%
Examen Parcial II	20%
Examen Parcial III	10%
Quices, Tareas Cortas , otros	10%
Laboratorios	10%
Tarea Programada 1	10%
Tarea Programada 2	10%
Trabajo de investigación	10%
Total:	100%

Consideraciones sobre la evaluación

- ☐ Los Quices se harán sin previo aviso en cualquier momento de la lección y no se harán reposiciones (excepto por las disposiciones establecidas en el reglamento de Régimen Académico Estudiantil).
- ☐ Las fechas de las evaluaciones del cronograma están sujetas a cambio dependiendo del avance de los temas vistos.
- ☐ Los rubros de calificación de cada una de las evaluaciones serán entregadas junto con la especificación de la misma.
- ☐ Las tareas programadas se realizarán en grupos establecidos por el profesor(a).
- ☐ Para las tareas programadas se realizará una comprobación individual para determinar la participación de los integrantes, de la cual dependerá el total de la nota.
- ☐ Las tareas programadas deben ser entregadas a la hora y fecha indicadas en el enunciado del proyecto. En caso de que se atrase un día(desde un minuto hasta 24 horas exactas), perderán el 40% del valor del trabajo.
- ☐ Para los demás aspectos de evaluación no se aceptarán entregas después de la fecha y hora solicitada (excepto por las disposiciones establecidas en el reglamento de Régimen Académico Estudiantil).
- ☐ Para cualquier ejercicio programado(independientemente del rubro de evaluación al que pertenezca), es responsabilidad del estudiante entregar una versión de la aplicación que compile sin errores, si el resultado entregado no cumple con esta característica el ejercicio será calificado con nota 0.
- ☐ Los temas de investigación también serán evaluados dentro de rubro los exámenes.
- ☐ Mantener celulares en modo silencioso o apagado durante las lecciones.
- ☐ La aparición de un celular durante un examen o quiz anulará automáticamente el mismo.
- ☐ La comprobación de que alguna tarea, laboratorio, proyecto, examen u otra evaluación es una copia implicará un 0 en la nota de los involucrados y hará que se apliquen las sanciones que contempla la reglamentación universitaria.
- ☐ Para comunicación con el profesor(a) se utilizará únicamente el correo de la carrera (@ucrsi.info) por lo cual el estudiante debe tenerlo activo y revisarlo continuamente, ya que es el medio que se utilizará para avisarle sobre actualizaciones en el sitio digital del curso, avisos, asignación de tareas, etc.
- ☐ Se utilizará la plataforma de Moodle de Informática Empresarial como apoyo del curso, entrega de material del curso por parte del profesor(a) y para recibo y calificación de algunas de las evaluaciones.



7. Cronograma.

	SEMANA	TEMA
1	10 – 16 Marzo	Presentación curso y entrega carta al estudiante Tema 1: Modelado de sistemas orientados a objetos con UML Tema 2: Manejo dinámico de memoria
2	17– 23 Marzo	Tema 2: Manejo dinámico de memoria Tema 3: Patrones de diseño
3	24 – 30 Marzo	Tema 3: Patrones de diseño
4	31 – 06 Abril	Tema 3: Patrones de diseño I Examen
5	07 – 13 Abril	Tema 4: Pruebas Unitarias
6	14 – 20 Abril	Semana Santa
7	21 – 27 Abril	Semana Universitaria Tema 4: Hilos
8	28 – 04 Mayo	I Examen Tema 4: Hilos
9	05 – 11 Mayo	Tema 4: Hilos Entrega I Tarea Programada
10	12 – 18 Mayo	Tema 5: Sockets Comprobación I Tarea Programada
11	19 – 25 Mayo	Tema 5: Sockets
12	26 – 01 Junio	Tema 6: Archivos especializados (XML)
13	02 – 08 Junio	II Examen Tema 7: Archivos especializados (XML)
14	09 – 15 Junio	Tema 8: Introducción al desarrollo de aplicaciones web
15	16 – 22 Junio	Tema 8 Introducción al desarrollo de aplicaciones web
16	23 – 29 Junio	Tema 8: Introducción al desarrollo de aplicaciones web
17	30 – 06 Julio	III Examen
18	07 – 13 Julio	Entrega II Tarea Programada / Comprobación II Tarea Programada Entrega de promedios
19	14 – 20 Julio	Ampliación

