



## UNIVERSIDAD LIBRE SECCIONAL PEREIRA

### FACULTAD DE INGENIERIAS

### PROGRAMA ACADÉMICO DE INGENIERIA DE SISTEMAS

#### PLAN DE ASIGNATURA

NOMBRE DE LA ASIGNATURA:	<b>ARQUITECTURA DE SOFTWARE</b>	
CODIGO DE LA ASIGNATURA:	02607	
CICLO DE FORMACIÓN:	BÁSICO	
	PROFESIONAL	X
COMPONENTE DE FORMACIÓN	BÁSICO	
	HUMANÍSTICO	
	PROFESIONAL	X
	ELECTIVO	
	OPTATIVO	
UBICACIÓN ASIGNATURA: (Semestre/año)	SEPTIMO SEMESTRE	
NIVEL DE FORMACIÓN:	POSGRADO	
	PREGRADO	X
	TECNOLÓGICO	
	TÉCNICO	
INTENSIDAD HORARIA SEMANAL:	<b>No. HORAS</b>	
	Presenciales	4
	Independientes	5
	<b>TOTAL HORAS</b>	9
CREDITOS ACADÉMICOS:	3	
PRERREQUISITOS:		
CORREQUISITOS:		
MODALIDAD:	PRESENCIAL	X
	A DISTANCIA	
	TUTORIADA	
	VIRTUAL	
CARÁCTER DE LA ASIGNATURA:	OBLIGATORIA	X
	ELECTIVA	
TIPO ASIGNATURA:	TEORICA	
	PRÁCTICA	
	TEÓRICO-PRÁCTICA	X

#### JUSTIFICACIÓN

El software que requieren las empresas deben ser robustos y de fácil mantenimiento para ellos, los proyectos de sistemas de información se desarrollan en las etapas de análisis, diseño, desarrollo e implementación utilizando metodologías de trabajo modernas. Siguiendo este proceso de desarrollo de los sistemas, corresponde a esta asignatura, el trabajo de la etapa de



## UNIVERSIDAD LIBRE SECCIONAL PEREIRA

diseño del sistema, basado en la etapa de análisis. El analista-diseñador debe retomar la información del análisis de sistemas y canalizarla a un diseño de tipo lógico, relacionando de una forma coherente, las características de las entradas, las salidas, los controles, los datos, los procesos y procedimientos.

### OBJETIVO GENERAL

Aunque la asignatura trata del diseño del sistema, se parte con un recorrido por las actividades del análisis del sistema, terminando con un diagnóstico de los problemas y necesidades detectados en esta etapa. En la etapa de diseño se debe plantear la solución. Para tal fin, se debe preparar al profesional en sistemas, competente en el uso de las herramientas de diseño para la realización de soluciones creativas que les permita a los usuarios, apoyar la toma de decisiones. El curso brinda al estudiante las herramientas, técnicas y métodos para la comprensión y diseño de sistemas complejos de software. Al final del curso el estudiante debe estar en capacidad de comprender los distintos estilos, patrones y vistas de una arquitectura de software y elegir entre ellas para modelar adecuadamente un sistema dado.

### OBJETIVOS ESPECÍFICOS

El estudiante deberá ser capaz de abstraer la arquitectura de un sistema en su nivel de tecnología, información y aplicación como una forma de alinear la arquitectura de tecnología de información y la del sistema de información.

Adicionalmente, deberá poder modelar formalmente los aspectos de integración del sistema con otros sistemas, protocolos, aplicaciones y plataformas.

Finalmente, el estudiante podrá llevar a cabo todo lo anterior haciendo uso de los últimos estándares y metodologías disponibles (UML, XML, IEEE 1471, TOGAF y otros). Apropiar de manera rápida casos de estudio de las principales arquitecturas multicapa existentes, en este caso J2EE y .NET.

### PRINCIPIOS DE FORMACION Y METAS DE APRENDIZAJE EN TÉRMINOS DE COMPETENCIAS

#### A. COMPETENCIA DE APRENDIZAJE Y DOMINIOS GENERALES:

#### B. COMPETENCIAS ESPECÍFICAS

- **COMPETENCIA INTERPRETATIVA:** El estudiante de análisis y diseño está en capacidad de interpretar la información de un sistema ya sea manual a automatizado, seguir una metodología y plantear soluciones.

- **COMPETENCIA ARGUMENTATIVA:** Una vez interpretado el sistema el estudiante desarrolla las etapas de análisis, diseño desarrollo e implementación.

- **COMPETENCIA PROPOSITIVA:** El estudiante está en capacidad de proponer solución a problemas, analizarlos, diseñarlos y plantear un sistema como solución a esos problemas.

### METODOLOGÍA GENERAL

1. Aprendizaje directivo: el profesor realizará exposición de los temas para explicitar complementar y clarificar los temas de la asignatura.
2. Auto-aprendizaje: Los estudiantes deberán preparar previamente la clase mediante la lectura crítica y activa de los temas del día (deben llegar con la lectura, conclusiones y preguntas).
3. Componente práctico: Los contenidos del curso se irán trabajando grupal mente fuera del salón de clase en equipos de estudiantes que resolverán en entregas (correspondientes al avance temático del curso) un proyecto cuyo enunciado es dado por el profesor. Para



## UNIVERSIDAD LIBRE SECCIONAL PEREIRA

la construcción de modelos y la generación de código, los estudiantes cuentan con los laboratorios de la Carrera que podrán usar fuera de las horas presenciales.

4. En promedio se realizará un taller cada dos semanas de los temas vistos en clase, donde, con la asistencia del profesor, se estudiará o resolverá un caso en equipos y para desarrollar en la hora de clase.

### ESTRATEGIAS PEDAGÓGICAS Y DIDACTICAS

Clase Magistral	X	Talleres de refuerzo	X	Lecturas previas	X
Valoración y motivación de aptitudes e intereses	X	Trabajos en grupo	X	Exposiciones	X
Presentación de contenidos mediante síntesis, cuadros, mapas conceptuales	X	Ejemplificación del contenido	X	Preguntas en clase	X
Realización de ejercicios y problemas por parte del profesor	X	Evaluación grupal	X	Diagnóstico de conocimientos previos	X
Verificación y síntesis de contenidos previos	X	Implementación de recursos didácticos	X	Seguimiento de actividad en la clase	X

### ESTRATEGIAS Y PORCENTAJES DE EVALUACIÓN

FORMAS DE EVALUACIÓN				PORCENTAJE DE EVALUACIÓN	
Evaluación escrita	X	Exposición	X	Primer Parcial	30
Quicez	X	Laboratorio	X	Segundo Parcial	30
Talleres Individuales	X	Trabajo de campo	X	Examen Final	40
Talleres Grupales	X	Participación en Clase	X		

### CONTENIDO PROGRAMATICO

UNIDAD TEMATICA	TEMA O SUBTEMA	BIBLIOGRAFÍA
1	1.1 Introducción a la arquitectura de software. 1.2 Contenido del curso -Motivación -Objetivos - Características -Conceptos generales. 1.3 Diseño preliminar o de alto nivel -Arquitectura de sistemas integrada con arquitectura de software.	Ver bibliografía y webgrafía asociada al curso.
2	2.1 Vistas de la arquitectura -Vista conceptual - Vista Lógica -Vista Física -Vista de implementación	Ver bibliografía y webgrafía asociada al curso.
3	3.1 Lenguaje de notación UML Introducción. 3.2 Conceptos básicos Diagramas UML. 3.3 Diagramas de casos de uso. 3.4 Diagrama de clases. 3.5 Diagramas de interacción. 3.6 Diagramas de objetos 3.7 Diagramas de actividad. 3.8 Diagramas de estados. 3.9 Diagramas de paquetes. 3.10 Diagramas de puesta en marcha. 3.11 Estereotipos Evolución de UML	Ver bibliografía y webgrafía asociada al curso.



## UNIVERSIDAD LIBRE SECCIONAL PEREIRA

4	<p>4.1 Metodología para la elicitación de requisitos del Sistemas de Software I.</p> <p>4.2 Objetivos de la metodología.</p> <p>4.3 Tareas recomendadas</p>	Ver bibliografía y webgrafía asociada al curso.
5	5.1 Metodología para la elicitación de requisitos del Sistemas de Software II.(Técnicas).	Ver bibliografía y webgrafía asociada al curso.
6	6.1 Metodología para la elicitación de requisitos del Sistemas de Software III.(Vista Conceptual)	Ver bibliografía y webgrafía asociada al curso.
7	<p>7.1 Metodología para la elicitación de requisitos del Sistemas de Software IV.(Vista Lógica).</p> <p>7.2 Catalogo de requisitos del sistema.</p> <p>7.3 Requisitos de información.</p> <p>7.4 Requisitos funcionales.</p> <p>7.5 Diagramas de casos de uso.</p> <p>7.6 Definición de actores.</p> <p>7.7 Casos de uso del sistema.</p>	Ver bibliografía y webgrafía asociada al curso.
8	8.1 Herramientas para la construcción de diagramas.	Ver bibliografía y webgrafía asociada al curso.
9	<p>9.1 Metodología para la elicitación de requisitos del Sistemas de Software V.(Vista Física).</p> <p>9.2 Requisitos No funcionales.</p> <p>9.3 Matriz de rastreabilidad objetivos/requisitos.</p>	Ver bibliografía y webgrafía asociada al curso.
10	<p>10.1 Metodología para la elicitación de requisitos del Sistemas de Software VI.(DRS).</p> <p>10.2 Documento de Requisitos del sistema.</p>	Ver bibliografía y webgrafía asociada al curso.
11	11.1 Herramientas para la construcción de patrones	
12	<p>12.1 Calidad y arquitectura de software.</p> <p>12.2 Selección de arquitectura.</p> <p>12.3 Diseño con calidad</p>	Ver bibliografía y webgrafía asociada al curso.
13	<p>13.1 Diseño de Interfaces de Usuario Principios, Prototipos y Heurísticas para Evaluación.</p> <p>13.2 Conceptos Generales.</p> <p>13.3 Principios para el Diseño de Interfaces de Usuario.</p> <p>13.4 Heurísticas para la Evaluación de IU.</p> <p>13.5 Caso Práctico -Conclusiones</p>	Ver bibliografía y webgrafía asociada al curso.
14	<p>14.1 Modelo Cliente / Servidor Database.</p> <p>14.2 Management System Modelo E-R Modelo RBS</p> <p>14.3 Gestores DBMS.</p> <p>14.4 Herramientas de modelado E-R Front End en DBMS.</p> <p>14.5 Laboratorio</p>	Ver bibliografía y webgrafía asociada al curso.
15	<p>15.1 Modelo Cliente / Servidor.</p> <p>15.1.1 Capa de Acceso a Datos.</p> <p>15.2 Laboratorio</p>	Ver bibliografía y webgrafía asociada al curso.
16	<p>16.1 Modelo Cliente / Servidor.</p> <p>16.1.1 Capa Lógica.</p> <p>16.1.2 Capa presentación.</p> <p>16.2 Laboratorio</p>	Ver bibliografía y webgrafía asociada al curso.
17	<p>17.1 ARQUITECTURA J2EE - Descripción de la arquitectura J2EE.</p> <p>17.2 Aplicaciones Distribuidas, RMI, RMI-IIOP,</p>	Ver bibliografía y webgrafía asociada al curso.



## UNIVERSIDAD LIBRE SECCIONAL PEREIRA

	JNDI. 17.3 Servicios de lógica no funcional, Contenedor.	
18	18.1 ARQUITECTURA .NET Arquitectura de Aplicaciones 18.2 Conceptos Patrones empresariales. 18.3 Diseño de capas de presentación ASP .NET Web Forms ASP .NET Web Services	Ver bibliografía y webgrafía asociada al curso.
19	19.1 Diseño de Capas de Datos ADO NET. 19.2 Patrones de Acceso a Datos	Ver bibliografía y webgrafía asociada al curso.

### BIBLIOGRAFÍA Y WEBGRAFÍA

1. Amescua, A. (2015). Análisis y diseño estructurado y orientado a objetos de sistemas informáticos. Madrid, España: McGraw-Hill.
2. Whitten, J. (2003). Análisis y diseño de Sistemas de Información. México: McGraw-Hill.
3. Bruegge, B. (2002). Ingeniería de Software Orientado a objetos. México: Pearson Educación.
4. Stevens, P. (2007). Utilización de UML en Ingeniería de Software con objetos y componentes. Madrid: Pearson Educación.
5. Weitzenfeld, A. (2005). Ingeniería de Software Orientada a Objetos con UML, Java e Internet. Mexico: Thomson.
6. Jeya, D. (2013). Object Oriented Analysis and Design using UML. Recuperado de: [https://books.google.es/books?id=IRB\\_AqAAQBAJ&pg=PT114&dq=uml+diagrams&hl=es&sa=X&ved=0ahUKEwjg4-brz7XNAhVLLB4KHdljDBI4ChDoAQhSMAQ#v=onepage&q=uml%20diagrams&f=false](https://books.google.es/books?id=IRB_AqAAQBAJ&pg=PT114&dq=uml+diagrams&hl=es&sa=X&ved=0ahUKEwjg4-brz7XNAhVLLB4KHdljDBI4ChDoAQhSMAQ#v=onepage&q=uml%20diagrams&f=false)
7. Fernandez, D. (2016). Definición de una arquitectura de software para el diseño de aplicaciones web basados en tecnología Java-J2EE. Recuperado de: [http://s3.amazonaws.com/academia.edu.documents/34702113/ArquitecturaJ2EE.PDF?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1466394978&Signature=6xEJUp87fultkpo05E3inpPMkoU%3D&response-content-disposition=inline%3B%20filename%3DArquitectura\\_Java\\_J2EE.pdf](http://s3.amazonaws.com/academia.edu.documents/34702113/ArquitecturaJ2EE.PDF?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1466394978&Signature=6xEJUp87fultkpo05E3inpPMkoU%3D&response-content-disposition=inline%3B%20filename%3DArquitectura_Java_J2EE.pdf)
8. Conessa, J. (2010). Introduccion a .NET. Recuperado de: [https://books.google.es/books?hl=es&lr=&id=9\\_9Uxdmb\\_QUC&oi=fnd&pg=PA13&dq=plataforma+.net&ots=xn1u\\_ltST3&sig=Odutgyrl05FNat8nygO50G-5Y2o#v=onepage&q=plataforma%20.net&f=false](https://books.google.es/books?hl=es&lr=&id=9_9Uxdmb_QUC&oi=fnd&pg=PA13&dq=plataforma+.net&ots=xn1u_ltST3&sig=Odutgyrl05FNat8nygO50G-5Y2o#v=onepage&q=plataforma%20.net&f=false)