



**Universidad Ricardo Palma**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA**  
**DEPARTAMENTO ACADÉMICO DE INGENIERÍA**

**PLAN DE ESTUDIOS 2006-II**

**SÍLABO**

**I. DATOS ADMINISTRATIVOS**

1.1. Nombre del curso	: <b>Arquitectura de Software</b>
1.2. Código	: IF 0804
1.3. Tipo del curso	: Teórico – Laboratorio
1.4. Área Académica	: INGENIERÍA DE SOFTWARE
1.5. Condición	: Obligatorio
1.6. Nivel	: VIII Ciclo
1.7. Créditos	: 03
1.8. Horas semanales	: Teoría = 2, Laboratorio = 2
1.9. Requisito	: IF 0703 (Pruebas de Software)

**II. SUMILLA**

El curso de Arquitectura de Softwares, corresponde al VIII ciclo de formación de la Escuela Académico Profesional de Ingeniería Informática. El curso es de naturaleza teórico-práctica y brinda a los participantes los conocimientos y habilidades para la resolución de sistemas complejos mediante el diseño de la arquitectura del software utilizando eficazmente los diferentes estilos y patrones de la arquitectura así como de los métodos de diseño de ésta. Las unidades de aprendizaje son: Introducción a la arquitectura de software, vistas y estilos de la arquitectura, patrones de arquitectura, diseño de la arquitectura y arquitectura orientada a servicios

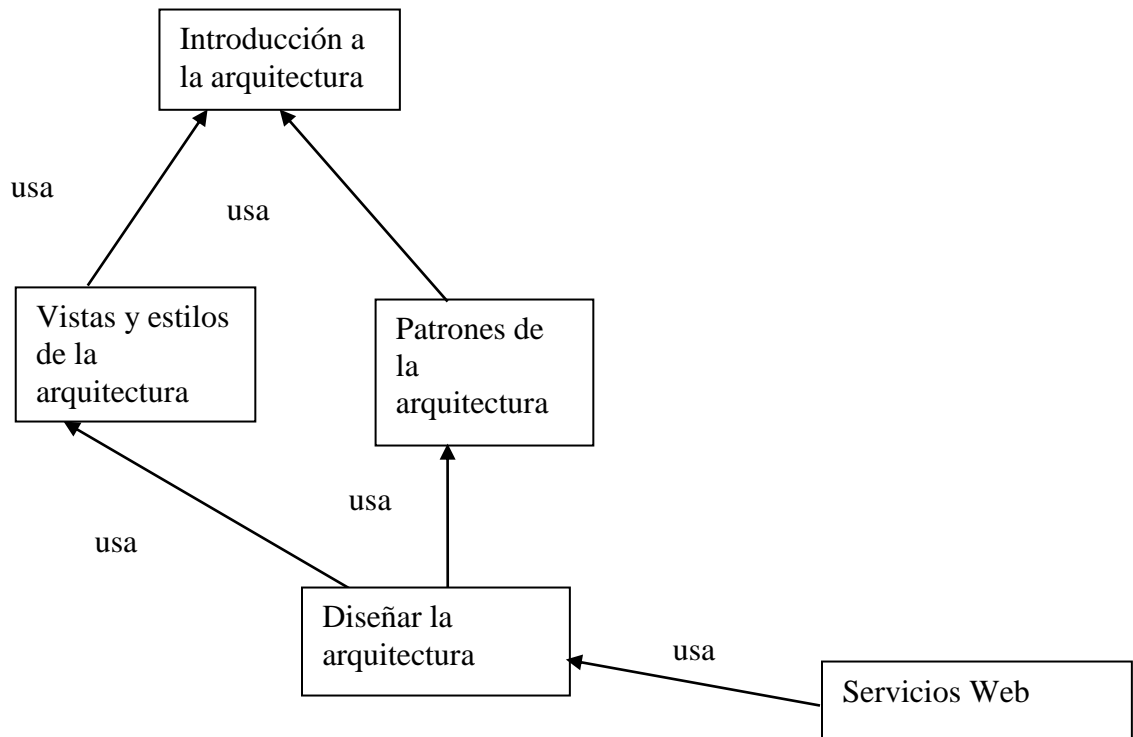
**III. COMPETENCIAS DE LA CARRERA**

Desarrolla y mantiene sistemas de software confiable y eficiente y que sea económico desarrollarlos y mantenerlos y que satisfagan los requisitos definidos por los clientes.

**IV. COMPETENCIA DEL CURSO**

1. **Aplica el estado del arte en el campo de la arquitectura del software**
2. **Utiliza las vistas y los estilos de la arquitectura para la solución de problemas.**
3. **Utiliza patrones de arquitectura para la solución de problemas**
4. **Diseña e implementa la arquitectura para las soluciones informáticas**
5. **Diseña e implementa la arquitectura basada u orientada a servicios**

## V. RED DE APRENDIZAJE



## VI. UNIDADES DE APRENDIZAJE

### Unidad de aprendizaje 1: Introducción a la arquitectura

**Competencia: Aplica el estado del arte en el campo de la arquitectura del software**

Sesión / Temas	Actividades	Fecha / Semana
Definición. Qué es la arquitectura. Por qué diseñar la arquitectura Rol y el equipo. Aspectos históricos	Exposición del profesor. Ejercicios: Dar varias variantes de arquitecturas y analizar si responde a la definición de la arquitectura	1
Acoplamiento y cohesión	Analizar si es o no una buena arquitectura utilizando los conceptos de cohesión y acoplamiento en la clase de taller	2

### Unidad de aprendizaje 2: Vistas y estilos de arquitectura Competencia: Utiliza las vistas y los estilos de la arquitectura para la solución de problemas

Sesión / Temas	Actividades	Fecha / Semana
Tipos de vista de módulos ➤ Estilos de descomposición, uso, generalización y capas Tipo de vista componente-conector ➤ Estilos: Pipes-filters, publicar-suscribir y Cliente-servidor	En clase de taller hacer el análisis de arquitecturas propuestas que usan los estilos mencionados. Hacer sus implementaciones en .net o Java	3
Tipo de vista componente-conector ➤ Estilos: Peer to peer, communicating process Tipo de vista de Asignación ➤ Estilos: Despliegue, implementación y asignación de tareas	En clase de taller hacer el análisis de arquitecturas propuestas que usan los estilos mencionados. Hacer sus implementaciones en .net o Java	4

### Unidad de aprendizaje 3: Patrones de la arquitectura

Competencia: Utiliza patrones de arquitectura para la solución de problemas

Sesión / Temas	Actividades	Fecha / Sem.
Patrones de arquitectura <ul style="list-style-type: none"><li>➤ Remote Facade</li><li>➤ Composite</li><li>➤ Observer o publicar-Suscribir</li><li>➤ Arquitectura de capas</li><li>➤ Model View Controller</li><li>➤ Front Controller</li><li>➤ Page Controller</li></ul>	Hace en taller la Implementación de los patrones estudiados en los ambientes de programación disponibles	5
Patrones de la arquitectura de aplicaciones Web <ul style="list-style-type: none"><li>➤ Cliente delgado</li><li>➤ Cliente grueso</li><li>➤ Cliente distribuido</li></ul>	Práctica calificada 1 Hace en taller la Implementación de los patrones estudiados en los ambientes de programación disponibles	6

### Unidad de aprendizaje 4: Diseño de la arquitectura

Competencia: Diseña e implementa la arquitectura para las soluciones informáticas

Sesión / Temas	Actividades	Fecha / Sem.
Método de diseño y atributos de calidad de la arquitectura del software: <ul style="list-style-type: none"><li>➤ Desempeño</li><li>➤ Seguridad</li><li>➤ Protección</li><li>➤ Disponibilidad</li><li>➤ Facilidad de mantenimiento</li></ul>	El tema iniciado en teoría se corresponde con el primer paso del método de diseño ADD. En la próxima clase practica se inicia un caso para aplicar todos los pasos	7
Prueba Parcial		8
Solución de la prueba parcial	Ejercitación de diseño de la arquitectura y de sus atributos de calidad Aplicar el paso de ADD, definir los atributos de calidad a un caso	9

Tácticas: <ul style="list-style-type: none"> <li>➤ Disponibilidad</li> <li>➤ Detección de fallas</li> <li>➤ Recuperación de fallas</li> <li>➤ Prevención</li> <li>➤ Facilidad de modificación</li> <li>➤ Localizar modificaciones</li> <li>➤ Efecto Ripley</li> <li>➤ Demanda de recursos</li> <li>➤ Administración de recursos</li> <li>➤ Arbitraje de recursos</li> <li>➤ Seguridad</li> <li>➤ Facilidad de prueba</li> </ul>	Continuar con el caso iniciado en la clase anterior y para el realizar la definición de las tácticas estudiadas	<b>10</b>
Método de diseño de la arquitectura <b>Método ADD (Attribute-Driven-Design)</b>	Aplicación del método al caso iniciado antes	<b>11</b>
Documentación de la arquitectura	Aplicación del método al caso iniciado antes	<b>12</b>

## Unidad de aprendizaje 5: Arquitectura basada en servicios

**Competencia: Diseña e implementa la arquitectura basada u orientada a servicios**

Sesión / Temas	Actividades	Fecha / Sem.
Introducción a arquitectura basada en servicios	Análisis de casos que utilizan una arquitectura basada en servicios o modelar uno en particular	<b>13</b>
Criterios de diseño	Práctica Calificada 2	<b>14</b>
Implementación en ambientes de desarrollo. Se dará en que arquitectura se basa la Service Software Factory de Microsoft	Utilizar la Service Software Factory de Microsoft	<b>15</b>
<b>EXAMEN FINAL</b>		<b>16</b>
<b>EXAMEN SUSTITUTORIO</b>		<b>17</b>

## VII. METODOLOGÍA

La metodología del curso está estrictamente orientada a la adquisición de conocimientos básicos y, sobre todo, de habilidades, por lo cual los temas teóricos son reforzados en gran medida por estudio de casos en la misma clase y en el laboratorio. El profesor expondrá inicialmente los temas teóricos y proseguirá con el desarrollo de aplicaciones, los cuales irán progresivamente incrementando su dificultad.

## VIII. EVALUACIÓN

- Los criterios que se usarán para la evaluación de los alumnos:
  - Habilidades adquiridas y demostradas en el laboratorio para la aplicación de los temas del curso.
  - Nivel de aprendizaje en el laboratorio.
- La nota final será la resultante de la siguiente fórmula:

$$PF = 0.30EP + 0.40EF + ((PL1+PL2/2)*0.30)$$

EP = examen parcial

EF = examen final

PL<sub>i</sub> = i-ésimo laboratorio calificado

## IX. Bibliografía.

	Introducción	Vistas y estilos	Patrones	Diseño de la arquitectura	Servicios
Bass, L., Clements, P., Kazman, R. (2003) <u>Software architecture in practice</u> . USA: Addison-Wesley- SEI series in software engineering.	X			X	
Cerami E. (2002) <u>Web services Essentials</u> USA: O'Reilly Media, Inc.					X
Clements, P., Bass, L.(2003) <u>Documenting software architectures : views and beyond</u> . USA:Addison-Wesley, SEI.		X			
Conallen, J. (2003). <u>Building Web Application with UML</u> (2a. ed.). USA: Addison-Wesley Object Technology Series.			X		
Fowler M., Rice D., Foemmel M., Hieatt E., Mee R., Stafford R. (2002). <u>Patterns Of Enterprise Application Architecture</u> . USA : Addison-Wesley.	X		X		X
Sommerville, I. (2005). <u>Ingeniería de Software (7a. ed.)</u> . España: Addison Wesley.	X				