



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

**PLAN DE ESTUDIOS 2006-II**

**SÍLABO**

**1. DATOS ADMINISTRATIVOS**

Nombre del curso	:	Taller de Proyectos II
Código del curso	:	IF0606
Créditos	:	4
Naturaleza del curso	:	Taller
Número de horas de teoría	:	0
Número de horas de práctica	:	2
Número de horas de taller	:	6
Horas Totales	:	8
Requisitos	:	IF0507 Taller de Proyectos I
Profesores	:	Francisco Aguilar, Javier Añaños, Humberto Linárez, Luis Palacios, Jorge Rodríguez, Yolanda Yopla.

**2. SUMILLA**

El curso de Taller de Proyectos II corresponde al sexto ciclo de formación de la Escuela Académico Profesional de Ingeniería Informática. El curso es de naturaleza de taller y brinda a los participantes habilidades que permitan la integración de las áreas de conocimiento que componen la Ingeniería de software a través del documento SWEBOOK (Software Engineering Body of Knowledge), la forma de desarrollar enmarcada en el Proceso unificado de Desarrollo de Software, con sus tres principios: guiado por Casos de Uso, basado en la arquitectura y que sea iterativo e incremental.

Así como experiencia del trabajo en equipo, evaluando, aprendiendo y usando nuevas herramientas para resolver problemas del mundo real con sponsors reales.

El 60% de su tiempo lo dedica a la programación (escribe, prueba y construye software) y el 40% en otras tareas como la transformación de los requisitos en las especificaciones del análisis del sistema, administración de la configuración, obtener la documentación técnica.

**3. COMPETENCIAS DE LA CARRERA**

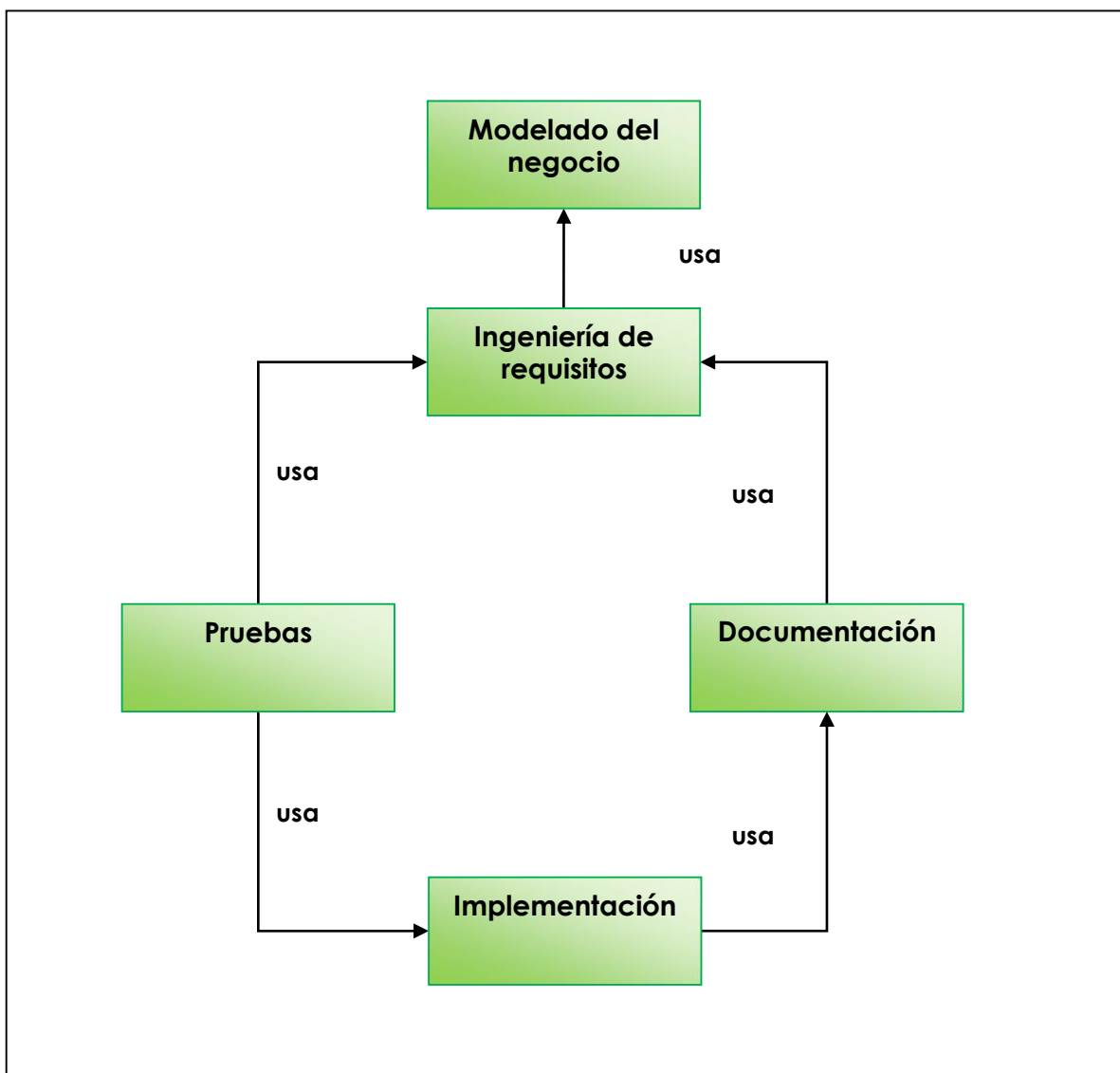
Desarrolla y mantiene sistemas de software confiable y eficiente, que sean económicos desarrollarlos y mantenerlos, y que satisfagan los requisitos definidos por los clientes. Versatilidad y adaptabilidad para trabajar e interactuar en los diferentes niveles de un proyecto de ingeniería y con una clara orientación al logro y alcance de objetivos. Profesionalismo para conducirse correctamente, respetando los estándares

de la profesión. Comunicación tanto oral como escrita, participa y se integra en forma efectiva en equipos multidisciplinarios de trabajo. Experimentación y pruebas analizando e interpretando sus resultados.

#### 4. COMPETENCIAS DEL CURSO

- Aplica las mejores prácticas para el modelado de negocio.
- Aplica las mejores prácticas para la ingeniería de requisitos.
- Aplica las mejores prácticas para la realización de pruebas.
- Aplica las mejores prácticas para la obtención de la documentación.
- Aplica las mejores prácticas para la implementación.
- Aplica las mejores prácticas para asegurar las pruebas de calidad del software desarrollado.

#### 5. RED DE APRENDIZAJE



## 6. DISTRIBUCIÓN SEMANAL

Sesión / Temas	Fecha / Semana	Feridos
Organización de equipos y proyecto a desarrollar	1	
Modela lo esencial del negocio; usar modelo para identificar los CU del sistema; diagramar los CUS	2	
Especificación detallada del 20% de CU de la arquitectura; lista requisitos funcionales y no funcionales; interpreta Benchmarking Plan de alto nivel del proyecto y plan del 2do entregable	3	
<b>Evaluación 1</b>	<b>4</b>	
Plantear la arquitectura del sistema (vistas); ajustar el alcance; Plan final del proyecto y plan del 3er entregable	5	
Completar (100%) la especificación de CU de la arquitectura: (para cada CU) diseño de interfaz de usuario, análisis de los requisitos, modelamiento de BD, diseño, programación, prueba unitaria, integrar al sistema, prueba integral	6	
<b>Evaluación 2</b>	<b>7</b>	
<b>EXÁMENES PARCIALES</b>	<b>8</b>	
Continuar desarrollo de CUS al 25%	9	
Continuar desarrollo de CUS al 50%; plan del 4to entregable	10	
<b>Evaluación 3</b>	<b>11</b>	
Continuar desarrollo de CUS al 75%	12	
Completar desarrollo de CUS al 100%	13	
<b>Evaluación 4</b>	<b>14</b>	
<b>Evaluación 5 - Sustentación Final</b>	<b>15</b>	
<b>EXÁMENES FINALES</b>	<b>16</b>	
<b>EXÁMENES SUSTITUTORIOS</b>	<b>17</b>	

Ciclo 1 – Fases de Inicio y Elaboración (1ra. Iteración)

Ciclo 2 – Fases de Elaboración (2da. Iteración) y Construcción (1ra. Iteración)

Ciclo 3 – Fase de Construcción (2da. Iteración)

Ciclo 4 – Fase de Transición

## 7. METODOLOGÍA

La metodología del curso está estrictamente orientada a la adquisición de habilidades, por lo cual los talleres son reforzados en gran medida por la discusión de casos que son elaborados por los estudiantes. El profesor expondrá inicialmente los casos a desarrollarse y proseguirá con el desarrollo de aplicaciones, los cuales irán progresivamente incrementando su funcionalidad.

## 8. EVALUACIÓN

### Principios

- La evaluación será individual y realizada teniendo en cuenta el conocimiento y la calidad mostrado en el desempeño de los roles de los estudiantes.
- La evaluación tendrá en cuenta la asistencia y puntualidad a clases como una forma de garantizar crear los hábitos de la autodisciplina y la responsabilidad en los futuros profesionales.
- La evaluación tendrá en cuenta el desempeño en el proyecto y los conocimientos adquiridos y aplicados.
- La evaluación tendrá en cuenta el trabajo en equipo de los estudiantes.

La nota de cada evaluación parcial estará compuesta de las siguientes componentes:

1. Presentación y respuesta a las preguntas.
2. Documento/programación entregados.
3. Desempeño en clases.
4. Asistencia y puntualidad.

Los dos primeros aspectos tienen un valor de 80% (Presentación y respuesta a las preguntas y Documento/programación entregados).

El 20% corresponde con el desempeño en clases y la asistencia y puntualidad.

### Hitos y forma de evaluación

Semana	Día de sustentación	Iteración	Entregable y evaluación	Revisión	%
4	Jueves	1	1	Docente del grupo	5%
7	Jueves	2	2	Docente del grupo	10%
11	Jueves	3	3	Docente del grupo	20%
14	Jueves	4	4	Docente del grupo	15%
15	Jueves		5	Jurado	50%

## 9. REGLAMENTO

### Sobre los equipos y alumnos

- Los equipos se conformarán con alumnos de todos los niveles (cursos), que asumirán sus roles.
- Todos los alumnos del equipo serán responsables del desarrollo programación de por lo menos un (1) CUS (caso de uso del sistema), el mismo que será parte de su evaluación final en forma individual.
- El control de asistencia (30% máximo de ausencias injustificadas) es permanente:
  - En el laboratorio por el docente; y
  - Fuera del laboratorio por el gerente.
  - El alumno desaprueba la asignatura si tiene ausencias injustificadas por mas del 30% de las horas efectivas de clases del curso durante el semestre.
- El Gerente del proyecto está en la obligación de participar en las reuniones de revisión y control, solicitadas por los alumnos de Control de Proyectos asignados, proporcionándoles la información de avance y entregables correspondientes, firmar las actas de control y disponer la ejecución de las acciones correctivas que permitan levantar las incidencias de control detectadas.
- En la presentación del proyecto no se considerará ni evaluará el paquete de seguridad (acceso a la aplicación), si se adopta en forma de componente.
- En cualquier sustentación, si no se presenta el trabajo, el equipo (todos los integrantes) tienen cero y no tendrán derecho a ninguna otra evaluación. Por tanto desaprueban la asignatura.
- Si un alumno, individualmente, no sustenta un entregable tiene cero para ese entregable.
- La no entrega del informe (documento y/o CD), en cualquier entregable, implica la pérdida de 5 puntos por día.
- Para aprobar la asignatura totalmente es indispensable la defensa de cada miembro del equipo en la sustentación final y aprobarla. Esto es independiente del acumulado que tenga el estudiante.
- Las fechas de presentación de los entregables son impostergables.
- La presentación se inicia con los alumnos que estén presentes. Los que no lleguen a tiempo tiene cero en ese entregable.

### Sobre los proyectos

Los proyectos pueden ser:

- Construcción de software.
- Adaptación/Personalización de software de algún producto comercial (SAP, Dynamic, BizAgi, etc.) y/o de software libre (Open Bravo, Joomla, Moodle, Drupal, Intallio, etc.).

Para cada proyecto:

- Se desarrollará o usará componentes, los mismos que deben permitir crear y mantener una "Biblioteca de Componentes".
- El componente de Control de Acceso (login) no se evaluará.
- Puede tener una duración de varios semestres académicos.
- Se debe establecer/definir los estándares de desarrollo (base de datos, programación, pantallas, botones, etc.), desde el inicio de clases.

### Sobre las sustentaciones

- Las sustentaciones se realizarán desde el servidor/repositorio de Laboratorio y de ser el caso con el producto desplegado.
- La realización de las sustentaciones de los entregables está a cargo de los propios docentes de cada grupo expositor.
- La realización de la sustentación final de taller es realizada ante un Jurado donde también participa el docente de cada grupo. De esta manera se puede lograr una mejor evaluación individual al mismo tiempo que se tienen en cuenta criterios diversos, y no sólo los del docente.
- Para cada entregable se hará llegar una lista de comprobación para ser utilizada en la calificación. Esta lista servirá para la realización de una revisión técnica formal de la documentación y solución técnica de los proyectos, de manera de tamizar aquellos trabajos que no utilizan las mejores prácticas.
- Los documentos en los discos correspondientes de cada proyecto de la sustentación final serán depositados en el buzón de la OSA el viernes de la semana 14, a más tardar.

## 10. REFERENCIAS BIBLIOGRÁFICAS Y OTRAS FUENTES

### Libros

- SOMMERVILLE, I. Ingeniería de Software. Addison Wesley 7<sup>ma</sup>. Edición, 2005.
- CONALLEN, J., Building Web Application with UML. Addison-Wesley Object Technology Series, 2<sup>nd</sup>. Ed., 2003
- CERAMI, Ethan. Web services Essentials (O'Reilly XML). O'Reilly Media, Inc.; 1<sup>st</sup>. Ed., 2002.
- RUMBAUGH, J. JACOBSON, I. BOOCH, G. El Lenguaje Unificado de Modelado. Manual de Referencia. Pearson Educación S.A. Madrid, 2000.
- RUMBAUGH, J. JACOBSON, I. BOOCH, G. The Unified Software Development Process. Addison Wesley Longman Inc. ISBN 0-201-57169-2, 1999.

### Referencias electrónicas

- <http://sce.uhcl.edu/helm/rationalunifiedprocess/> RUP
- <http://www.sei.cmu.edu/> Software Engineering Institute Of Carnegie Mellon University
- <http://www.isr.uci.edu/> Software Research (ISR).University California, Irvine
- <http://www-01.ibm.com/software/rational/> IBM-Rational
- <http://www.uml.org/> Lenguaje Uml
- <http://www.omg.org/> OMG-Object Management Group
- <http://www.bpmn.org> OMG-Business Process Management Initiative
- <http://qse.ifs.tuwien.ac.at/index.htm> Quality Software Engineering (QSE) Research at the TU Wien, Institute of Software Technology and Interactive Systems (IFS)
- <http://www.ifs.tuwien.ac.at/> Center of Information and Software Engineering
- <http://www2.umassd.edu/secenter/saresources.html> Sitios de Recursos de Arquitectura de Software
- <http://www.usernomics.com/user-interface-design.html> Principios de diseño de interfaces de usuario
- <http://www.ambysoft.com/essays/userInterfaceDesign.html> User Interface Design Tips, Techniques, and Principles
- <http://www.loop11.com/> Usabilidad y experiencia de usuarios
- <http://www.webstandards.org/learn/tutorials/> Tutoriales de diseño Web