

Experto en Arquitectura y Desarrollo Software DISEÑO ORIENTADO A OBJETOS

Guía de Aprendizaje Información al estudiante

Datos Descriptivos

Centro responsable	Escuela Técnica Superior de Ingeniería de Sistemas Informáticos	
Titulación:	Experto en Arquitectura y Desarrollo Software	
Asignatura:	Diseño Orientado a Objetos	
Carácter:	Obligatoria	
Curso académico:	2015/2016	
Curso/semestre:	1º Curso / 2º Semestre	
Créditos Europeos	5	
Idioma impartición:	Español	
Departamento:	Sistemas Informáticos	
Profesorado (c = coordinador)	Despacho	Correo electrónico
(c) Luis Fernández Muñoz	D-1128	setillo@etsisi.upm.es

Contenidos de la asignatura

Resultados de aprendizajes	
Código	Descripción
Al terminar con éxito esta asignatura, los estudiantes serán capaces de:	
RA1	Conocer, comprender, analizar y sintetizar los fundamentos del diseño orientado a objetos
RA2	Conocer, comprender, analizar y sintetizar los principios del diseño orientado a objetos
RA3	Conocer, comprender, analizar y sintetizar los patrones del diseño orientado a objetos
RA4	Conocer, comprender, analizar y sintetizar el diseño dirigido por responsabilidades
RA5	Conocer, comprender, analizar y sintetizar el diseño dirigido por interfaces
RA6	Conocer, comprender, analizar y sintetizar la disciplina de análisis orientado a objetos
RA7	Conocer, comprender, analizar y sintetizar la disciplina de diseño orientado a objetos
RA8	Conocer, comprender y analizar los antipatrones del desarrollo software

Contenidos formativos		
Tema	Contenidos específicos	Horas
T1	Fundamentos de Diseño Orientado a Objetos Sistemas complejos. Complejidad del Software. Fundamentos del Diseño Orientado a Objetos. Clases de Calidad. Relaciones entre Clases. Disciplinas de Análisis y Diseño Orientado a Objetos. Estrategias de Análisis Orientado a Objetos.	6
T2	Diseño orientado a Interfaces Diseño por Contrato. Leyes de los Interfaces. Características de los Interfaces. Ingredientes de los Interfaces. Interfaces Remotos. Jerarquías de Interfaces.	4
T3	Principios de Diseño Orientado a Objetos Principio "No te repitas!". Principio "Mantenlo simple, estúpido!" Principio de Única Responsabilidad. Principio Abierto/Cerrado. Principio de Sustitución de Liskov. Principio de Segregación de Interfaces. Principio de Inversión de Dependencias. Principio de Equivalencia de Reutilización de Entregas. Principio de Cierre Común. Principio de Reutilización Común. Principio de Dependencias Acíclicas. Principio de Dependencias Estables. Principio de Abstracciones Estables.	4
T4	Patrones Generales de Software para Asignación de Responsabilidades Patrón Experto en Información. Patrón Creador. Patrón Controlador. Patrón Alta cohesión. Patrón Bajo acoplamiento. Patrón Polimorfismo. Patrón Fabricación pura. Patrón Indirección. Patrón Variaciones Protegidas	4
T5	Patrones de Diseño Definiciones y Objetivos. Categorías de Patrones. Clasificación de Patrones de Diseño. Elementos de Patrones de Diseño. Antipatrones de Desarrollo Software.	2
T6	Catálogo de Patrones de Diseño Para la creación de objetos especificando su clase explícitamente. Para la dependencia de operaciones concretas. Para la dependencia de plataformas hardware o software. Para la dependencia de las representaciones o implementaciones de objetos. Para la dependencias algorítmicas. Para el fuerte acoplamiento. Para la adición de funcionalidad mediante la herencia. Para la incapacidad de modificar las clases convenientemente.	30

Evaluación

Código	Descripción	Valor en %	Calificación mínima
Evaluación Continuada			
EA	Asistencia y participación en el aula	10%	80%
ET	Evaluación de Test (sin apoyo documental)	15%	30%
EP1	Evaluación Práctica: Fundamentos de Diseño Orientado a Objetos y a Interfaces	20%	
EP2	Evaluación Práctica: Principios y Patrones Generales de Diseño Orientado a Objetos	25%	
EP3	Evaluación Práctica: Patrones de Diseño Orientado a Objetos	30%	
En la convocatoria ordinaria, el alumno utilizará el sistema de evaluación continuada. No obstante, si no pudiera realizarlo, podrá informar por escrito al coordinador de la asignatura para optar por la evaluación mediante sólo prueba final			
Evaluación Prueba Final			
ETF	Evaluación de Test (sin apoyo documental)	15%	30%
EPF	Evaluación Práctica: Diseño Orientado a Objetos	25%	
EEF	Examen escrito	60%	

Alineación de Contenidos		
Resultados de aprendizaje	Temas	Evaluación
RA1	T1	EA, ET y EP1
RA2	T3	EA, ET y EP2
RA3	T4, T5 y T6	EA, ET, EP2 y EP3
RA4	T1	EA, ET y EP1
RA5	T2	EA, ET y EP1
RA6	T1	EA, ET y EP1
RA7	T1	EA, ET y EP1
RA8	T5	EA, ET y EP3

Breve descripción de las modalidades organizativas utilizadas y métodos de enseñanzas empleados	
Clases de teoría	Se sigue el método expositivo / lección magistral. El profesor expone verbalmente los conceptos de la materia en cada uno de los temas.
Clases problemas	Se sigue el método de resolución de problemas en clase. Se plantea un problema que los estudiantes tienen que resolver desarrollando estrategias nuevas a partir de los conocimientos de la clase magistral.
Prácticas	Al final de cada unidad, se plantean problemas que el alumno deberá resolver y presentar.
Tutorías	No hay tutorías grupales en la asignatura. Las tutorías son individuales y los estudiantes son atendidos en los horarios establecidos para las tutorías académicas.

Cronograma de trabajo de la asignatura: Evaluación Continua		
Día	Actividades aula	Actividades Evaluación
1	T1	EA
2	T1 y T2	EA
3	T3 y T4	EA y EP1
4	T4 y T5	EA

Cronograma de trabajo de la asignatura: Evaluación Continua		
Día	Actividades aula	Actividades Evaluación
5	T6	EA y EP2
6	T6	EA
7	T6	EA
8	T6	EA
9	T6	EA
10	T6	EA
(siguiente sesión)		ET y EP3

Recursos didácticos

Recursos didácticos	
Bibliografía básica	<p>Object-Oriented Analysis and Design with Applications. Booch, G.; et al. Addison-Wesley Professional; 3 edition, 2007</p> <p>The Unified Software Development Process. Jacobson, I. Addison-Wesley Professional; 1999</p> <p>Interface Oriented Design: With Patterns (Pragmatic Programmers). Pugh, K. Pragmatic Bookshelf; 2006</p> <p>Agile Software Development, Principles, Patterns, and Practices. Martir, R. Prentice Hall; 2002</p> <p>Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Larman, C. Prentice Hall; 3 edition, 2004</p> <p>Design Patterns: Elements of Reusable Object-Oriented Software. Gamma, E. et al. Addison-Wesley Professional; 1994</p>
Recursos Web	<p>oodesign.com. http://www.oodesign.com/</p> <p>PrinciplesOfOod. http://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod</p>
Bibliografía avanzada	<p>An Introduction to Object-Oriented Programming. Budd, T. Addison-Wesley; 3 edition, 2001</p> <p>Clean Code: A Handbook of Agile Software Craftsmanship. Martin, R.C. Prentice Hall; 2008</p> <p>Code Complete: A Practical Handbook of Software Construction. McConnell, S. Microsoft Press; 2004</p> <p>The Pragmatic Programmer: From Journeyman to Master. Hunt, A. Addison-Wesley Professional; 1999</p> <p>The Software Craftsman: Professionalism, Pragmatism, Pride. Mancuso, S. Prentice Hall; 2014</p> <p>Wiki: List of software development philosophies. http://en.wikipedia.org/wiki/List_of_software_development_philosophies</p>
Equipamiento	<p>Aula Ericsson de ETSISI</p> <p>20 puestos</p> <p>Pizarra y cañón de video</p>

Experto en Arquitectura y Desarrollo Software PROGRAMACIÓN EXTREMA (XP)

Guía de Aprendizaje Información al estudiante

Datos Descriptivos

Centro responsable	Escuela Técnica Superior de Ingeniería de Sistemas Informáticos	
Titulación:	Experto en Arquitectura y Desarrollo Software	
Asignatura:	Programación Extrema (XP)	
Carácter:	Obligatoria	
Curso académico:	2015/2016	
Curso/semestre:	1º Curso / 2º Semestre	
Créditos Europeos	5	
Idioma impartición:	Español	
Departamento:	Sistemas Informáticos	
Profesorado (c = coordinador)	Despacho	Correo electrónico
(c) Micael Gallego	-	micael.gallego@urjc.es

Contenidos de la asignatura

Resultados de aprendizaje	
Código	Descripción
Al terminar con éxito esta asignatura, los estudiantes serán capaces de:	
RA1	Conocer, comprender, analizar y sintetizar los valores y principios de la programación extrema (XP)
RA2	Conocer, comprender, analizar y sintetizar el diseño e implementación de pruebas de código
RA3	Conocer, comprender, analizar y sintetizar las refactorizaciones de código
RA4	Conocer, comprender, analizar y sintetizar los malos olores en el código
RA5	Conocer, comprender, analizar y sintetizar las métricas orientadas a objetos
RA6	Conocer, comprender, analizar y sintetizar la disciplina del desarrollo guiado por pruebas

Contenidos formativos		
Tema	Contenidos específicos	Horas
T1	Programación Extrema (XP)	3
	Introducción a XP. Objetivo, valores, roles y principios de XP. Buenas prácticas de XP.	
T2	Diseño e Implementación de Pruebas	7
	Introducción. Construcción de tests. Dobles. Tests de calidad.	
T3	Refactorización de código	5
	Introducción a las refactorizaciones. Catálogo de refactorizaciones. Composición de métodos. Moviendo funcionalidades entre objetos. Organizando los datos. Simplificando las expresiones condicionales. Simplificando las llamadas a métodos. Generalizando. Conclusiones.	
T4	Malos olores en el código	5
	Introducción. Código duplicado. Método grande. Clase grande. Demasiados parámetros. Envidia de características. Intimidad inadecuada. Herencia rechazada. Clase perezosa / gorrón. Complejidad artificiosa. Identificadores excesivamente largos. Identificadores excesivamente cortos. Excesivo uso de literales. Supercallback	
T5	Métricas Orientadas a Objetos	5
	Introducción. Medidas y visualizaciones. Caracterizando y evaluando el diseño. Disonancia propia. Disonancia en la colaboración. Disonancia en la clasificación.	
T6	Desarrollo guiado por pruebas (TDD)	15
	Introducción. Un primer ejemplo. Las bases del TDD. Diseño incremental. TDD en proyectos heredados. Manteniendo TDD.	

Evaluación			
Código	Descripción	Valor en %	Calificación mínima
Evaluación Continuada			
EA	Asistencia y participación en el aula	10%	80%

Evaluación			
Código	Descripción	Valor en %	Calificación mínima
Evaluación Continuada			
ET	Evaluación de Test (sin apoyo documental)	15%	30%
EP1	Evaluación Práctica: Diseño e implementación de pruebas	20%	
EP2	Evaluación Práctica: Refactorización del código	25%	
EP3	Evaluación Práctica: Desarrollo guiado por pruebas	30%	
En la convocatoria ordinaria, el alumno utilizará el sistema de evaluación continuada. No obstante, si no pudiera realizarlo, podrá informar por escrito al coordinador de la asignatura para optar por la evaluación mediante sólo prueba final			
Evaluación Prueba Final			
ETF	Evaluación de Test (sin apoyo documental)	15%	30%
EPF	Evaluación Práctica: Programación Extrema (XP)	25%	
EEF	Examen escrito	60%	

Alineación de Contenidos		
Resultados de aprendizaje	Temas	Evaluación
RA1	T1	EA, ET
RA2	T2	EA, ET y EP1
RA3	T3	EA, ET y EP2
RA4	T4	EA, ET
RA5	T5	EA, ET
RA6	T6	EA, ET y EP3

Breve descripción de las modalidades organizativas utilizadas y métodos de enseñanzas empleados	
Clases de teoría	Se sigue el método expositivo / lección magistral. El profesor expone verbalmente los conceptos de la materia en cada uno de los temas.
Clases problemas	Se sigue el método de resolución de problemas en clase. Se plantea un problema que los estudiantes tienen que resolver desarrollando estrategias nuevas a partir de los conocimientos de la clase magistral.
Prácticas	Al final de cada unidad, se plantean problemas que el alumno deberá resolver y presentar.
Tutorías	No hay tutorías grupales en la asignatura. Las tutorías son individuales y los estudiantes son atendidos en los horarios establecidos para las tutorías académicas.

Cronograma de trabajo de la asignatura: Evaluación Continua		
Día	Actividades aula	Actividades Evaluación
1	T1 y T2	EA
2	T2	EA
3	T3	EA y EP1
4	T4	EA
5	T5	EA y EP2
6	T6	EA
7	T6	EA
8	T6	EA y EP3

Recursos didácticos

Recursos didácticos	
Bibliografía básica	<p>Extreme Programming Explained: Embrace Change. Kent Beck, K. and Andres, C. Addison-Wesley. 2nd edition, 2004.</p> <p>Scrum and Xp from the Trenches. Kniberg, H. Lulu.com, 2007. Disponible la versión castellano en: http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-las-trincheras.pdf</p> <p>Modern C++ Programming with Test-Driven Development: Code Better, Sleep Better. Langr, J., Pragmatic Bookshelf, 2013.</p> <p>Refactoring. Improving the Design of Existing Code. Fowler, M; et. al. Addison-Wesley Professional, 1999</p> <p>Object-Oriented Metrics in Practice: Using Software Metrics to Characterize, Evaluate, and Improve the Design of Object-Oriented Systems. Lanza, M; et. al. Springer, 2010.</p>
Recursos Web	<p>Manifiesto Ágil: http://agilemanifesto.org/</p> <p>Catálogos de refactorizaciones: http://www.refactoring.com/catalog/, https://refactoring.guru/catalog, http://industriallogic.com/xp/refactoring/catalog.html</p> <p>TDD Katas: https://github.com/garora/TDD-Katas, http://www.codekatas.org/</p>
Bibliografía avanzada	<p>Test Driven Development. By Example. Beck, K. Addison-Wesley, 2008.</p> <p>Refactoring to Patterns. Kerievsky, J. Addison Wesley, 2004.</p> <p>Effective Unit Testing: A guide for Java. Koskela, L. Manning Publications, 2013.</p>
Equipamiento	<p>Aula Ericsson de ETSISI</p> <p>20 puestos</p> <p>Pizarra y cañón de video</p>

Experto en Arquitectura y Desarrollo Software ARQUITECTURAS SOFTWARE

Guía de Aprendizaje Información al estudiante

Datos Descriptivos

Centro responsable	Escuela Técnica Superior de Ingeniería de Sistemas Informáticos	
Titulación:	Experto en Arquitectura y Desarrollo Software	
Asignatura:	Arquitecturas Software	
Carácter:	Obligatoria	
Curso académico:	2015/2016	
Curso/semestre:	1º Curso / 2º Semestre	
Créditos Europeos	5	
Idioma impartición:	Español	
Departamento:	Sistemas Informáticos	
Profesorado (c = coordinador)	Despacho	Correo electrónico
(c) Francisco Gortázar	-	Francisco.gortazar@urjc.es

Contenidos de la asignatura

Resultados de aprendizajes	
Código	Descripción
Al terminar con éxito esta asignatura, los estudiantes serán capaces de:	
RA1	Conocer, comprender, analizar y sintetizar los fundamentos de las arquitecturas software
RA2	Conocer, comprender, analizar y sintetizar los principios de reusabilidad del software
RA3	Conocer, comprender, analizar y sintetizar los patrones y estilos de arquitecturas software
RA4	Conocer, comprender, analizar y sintetizar las decisiones arquitectónicas
RA5	Conocer, comprender, analizar y sintetizar las vistas de las arquitecturas software

Contenidos formativos		
Tema	Contenidos específicos	Horas
T1	Fundamentos de Arquitecturas Software	3
	Definición de Arquitectura Software. Diseño y Arquitectura Software. Actores. Actividades: Análisis, Diseño, Evaluación, Documentación. Topologías.	
T2	Reusabilidad del Software	7
	Principios. Equivalencia Versión-Reutilización. Cierre común. Reutilización común. Dependencias acíclicas. Dependencias estables. Abstracciones estables.	
T3	Patrones y Estilos de Arquitecturas Software	30
	Cliente/Servidor o estilo por capas. Dirigido por eventos. Tuberías y filtros. Pizarra. Basado en componentes. Centrado en los datos. Punto a punto. Orientado a Servicios. Dirigido por reglas. No compartición. Antipatrones.	
T4	Documentación de Arquitecturas Software	5
	Modelo 4+1 vistas. Vista lógica. Vista de procesos. Vista de despliegue. Vista física. Vista de escenario.	
T5	Vistas	5
	MVC. MVP. MVVM.	

Evaluación			
Código	Descripción	Valor en %	Calificación mínima
Evaluación Continuada			
EA	Asistencia y participación en el aula	10%	30%
ET	Evaluación de Test (sin apoyo documental)	15%	
EP1	Evaluación Práctica: Reusabilidad del código	20%	
EP2	Evaluación Práctica: Patrones y estilos de arquitecturas software	35%	
EP3	Evaluación Práctica: Documentación de Arquitecturas	20%	
En la convocatoria ordinaria, el alumno utilizará el sistema de evaluación continuada. No obstante, si no pudiera realizarlo, podrá informar por escrito al coordinador de la asignatura para optar por la evaluación mediante sólo prueba final			
Evaluación Prueba Final			
ETF	Evaluación de Test (sin apoyo documental)	15%	30%
EPF	Evaluación Práctica: Documentación y patrones de arquitecturas software	35%	

Evaluación			
Código	Descripción	Valor en %	Calificación mínima
Evaluación Continuada			
EEF	Examen escrito	50%	

Alineación de Contenidos		
Resultados de aprendizaje	Temas	Evaluación
RA1	T1	EA y ET
RA2	T2	EA, ET y EP1
RA3	T3	EA, ET y EP2
RA4	T3,T4	EA, ET y EP3
RA5	T5	EA y ET

Breve descripción de las modalidades organizativas utilizadas y métodos de enseñanzas empleados	
Clases de teoría	Se sigue el método expositivo / lección magistral. El profesor expone verbalmente los conceptos de la materia en cada uno de los temas.
Clases problemas	Se sigue el método de resolución de problemas en clase. Se plantea un problema que los estudiantes tienen que resolver desarrollando estrategias nuevas a partir de los conocimientos de la clase magistral.
Prácticas	Al final de cada unidad, se plantean problemas que el alumno deberá resolver y presentar.
Tutorías	No hay tutorías grupales en la asignatura. Las tutorías son individuales y los estudiantes son atendidos en los horarios establecidos para las tutorías académicas.

Cronograma de trabajo de la asignatura: Evaluación Continua		
Día	Actividades aula	Actividades Evaluación
1	T1 y T2	EA
2	T2	EA y EP1
3	T3	EA y EP2
4	T3	EA
5	T3	EA
6	T3	EA
7	T3	EA
8	T3	EA, EP3
9	T4	EA, EP3
10	T5	EA y ET

Recursos didácticos

Recursos didácticos	
Bibliografía básica	<p>Patterns of Enterprise Application Architecture. Fowler, M.; et al. Addison-Wesley Professional; 1st edition, 2003</p> <p>The Art of Software Architecture: Design Methods and Techniques. Albin, S. Wiley & Sons; 2003</p> <p>Software Systems Architecture. Rozanski, N., Woods, E. Addison-Wesley Professional; 2005</p> <p>Documenting Software Architectures. Clements, P; et al. Addison-Wesley Professional; 2nd edition, 2010</p> <p>Enterprise Integration Patterns. Hoppe, G; Woolf, B. Addison-Wesley Professional; 2003</p>
Recursos Web	PrinciplesOfOod. http://www.butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod
Bibliografía avanzada	Cloud Architecture Patterns. Wilder, B. O'Reilly; 2012
Equipamiento	Aula Ericsson de ETSISI
	20 puestos Pizarra y cañón de video

Experto en Arquitectura y Desarrollo Software PROCESO UNIFICADO DE DESARROLLO

Guía de Aprendizaje Información al estudiante

Datos Descriptivos

Centro responsable	Escuela Técnica Superior de Ingeniería de Sistemas Informáticos	
Titulación:	Experto en Arquitectura y Desarrollo Software	
Asignatura:	Proceso Unificado	
Carácter:	Obligatoria	
Curso académico:	2015/2016	
Curso/semestre:	1º Curso / 2º Semestre	
Créditos Europeos	5	
Idioma impartición:	Español	
Departamento:	Sistemas Informáticos	
Profesorado (c = coordinador)	Despacho	Correo electrónico
(c) Luis Fernández Muñoz	D-1128	setillo@etsisi.upm.es

Contenidos de la asignatura

Resultados de aprendizajes	
Código	Descripción
Al terminar con éxito esta asignatura, los estudiantes serán capaces de:	
RA1	Conocer, comprender, aplicar, analizar y sintetizar las técnicas de requisitos basadas en casos de uso
RA2	Conocer, comprender, aplicar, analizar y sintetizar las técnicas de análisis basadas en MVC
RA3	Conocer, comprender, aplicar, analizar y sintetizar las técnicas de diseño basadas en MVC
RA4	Conocer, comprender, aplicar, analizar y sintetizar las técnicas de prueba basadas en caja blanca y caja negra
RA5	Conocer, comprender, aplicar, analizar y sintetizar las técnicas de gestión del proyecto

Contenidos formativos		
Tema	Contenidos específicos	Horas
T1	Introducción	2
	Definiciones de RUP: Actividad, Rol y Artefacto. Claves de RUP: Disciplinas y UML. Historia de RUP.	
T2	Proceso Dirigido por Casos de Uso	13
	Disciplina del Modelo del Dominio Disciplina de Requisitos	
T3	Proceso Centrado en la Arquitectura	20
	Disciplina de Análisis Disciplina de Diseño Disciplina de Implementación Disciplina de Pruebas	
T4	Proceso Iterativo e Incremental	15
	Disciplina de Despliegue Disciplina de Control de Versiones Disciplina de Entorno Disciplina de Gestión	

Evaluación			
Código	Descripción	Valor en %	Calificación mínima
Evaluación Continuada			
EA	Asistencia y participación en el aula	10%	80%
ET	Evaluación de Test (sin apoyo documental)	15%	
EP	Evaluación Práctica: Proyecto UP	20%	
En la convocatoria ordinaria, el alumno utilizará el sistema de evaluación continuada. No obstante, si no pudiera realizarlo, podrá informar por escrito al coordinador de la asignatura para optar por la evaluación mediante sólo prueba final			
Evaluación Prueba Final			
ETF	Evaluación de Test (sin apoyo documental)	15%	30%
EPF	Evaluación Práctica: Proyecto UP	25%	
EEF	Examen escrito	60%	

Alineación de Contenidos		
Resultados de aprendizaje	Temas	Evaluación
RA1	T1 y T2	EA, ET/ETF, EP/EPF y EEF
RA2	T1 y T3	EA, ET/ETF, EP/EPF y EEF
RA3	T1 y T3	EA, ET/ETF, EP/EPF y EEF
RA4	T1 y T3	EA, ET/ETF, EP/EPF y EEF
RA5	T1 y T4	EA, ET/ETF, EP/EPF y EEF

Breve descripción de las modalidades organizativas utilizadas y métodos de enseñanzas empleados	
Clases de teoría	Se sigue el método expositivo / lección magistral. El profesor expone verbalmente los conceptos de la materia en cada uno de los temas.
Clases problemas	Se sigue el método de resolución de problemas en clase. Se plantea un problema que los estudiantes tienen que resolver desarrollando estrategias nuevas a partir de los conocimientos de la clase magistral.
Prácticas	Al final de cada unidad, se plantean problemas que el alumno deberá resolver y presentar.
Tutorías	No hay tutorías grupales en la asignatura. Las tutorías son individuales y los estudiantes son atendidos en los horarios establecidos para las tutorías académicas.

Cronograma de trabajo de la asignatura: Evaluación Continua		
Día	Actividades aula	Actividades Evaluación
1	T1 y T2	EA
2	T2	EA
3	T2	EA
4	T3	EA
5	T3	EA
6	T3	EA
7	T3	EA
8	T4	EA
9	T4	EA
10	T4	EA, ET
(15 días después)		EP

Recursos didácticos

Recursos didácticos	
Bibliografía básica	Jacobson, I.; Booch, G.; Rumbaugh, J. The Unified Software Development Process . Addison Wesley, 1999
Recursos Web	Agile Unified Process. http://www.ambysoft.com/unifiedprocess/agileUP.html Unified Process. https://en.wikipedia.org/wiki/Unified_Process
Bibliografía avanzada	Kruchten, P. The Rational Unified Process: An Introduction . Addison Wesley, 2003 Kroll, P.; Kruchten, P.; Booch, G. The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP . Addison Wesley, 2003

Recursos didácticos	
Equipamiento	Aula Ericsson de ETSISI
	20 puestos
	Pizarra y cañón de video