

**Guía docente de la asignatura**

<b>Asignatura</b>	Planificación y Diseño de Sistemas Computacionales		
<b>Materia</b>	3.3: Computación		
<b>Módulo</b>	3: Tecnologías Específicas		
<b>Titulación</b>	Graduado en Ingeniería Informática (Mención Computación)		
<b>Plan</b>	545	<b>Código</b>	46955
<b>Periodo de impartición</b>	1	<b>Tipo/Carácter</b>	OB
<b>Nivel/Ciclo</b>	Grado	<b>Curso</b>	4
<b>Créditos ECTS</b>	6		
<b>Lengua en que se imparte</b>	Español		
<b>Profesor/es responsable/s</b>	Yania Crespo González-Carvajal, Pablo de la Fuente Redondo		
<b>Datos de contacto (E-mail, teléfono...)</b>	<a href="mailto:yania@infor.uva.es">yania@infor.uva.es</a> Teléfono: 5695 <a href="mailto:pfuente@infor.uva.es">pfuente@infor.uva.es</a> Teléfono: 5611		
<b>Horario de tutorías</b>	Acceder a la web de la Universidad ( <a href="http://www.uva.es">www.uva.es</a> ) y seguir navegando por Centros, elegir el Campus de Valladolid y seleccionar a continuación E.T.S. de Ingeniería Informática. Acceder a Profesorado y localizar el nombre del profesor. Al final de la página aparecen las tutorías correspondientes a dicho profesor.		
<b>Departamento</b>	Informática		

**1. Situación / Sentido de la Asignatura****1.1 Contextualización**

Esta asignatura está integrada en el cuarto curso del grado de ingeniería informática (mención Computación). La asignatura tiene un doble objetivo, por una parte, se tiene el objetivo de presentar los conocimientos precisos para realizar la gestión de un proyecto, tanto de desarrollo de software, como de consecución de un servicio. Para ello, seguirá las pautas proporcionadas por el Instituto de Gestión de Proyectos (PMI) lo que permitirá dotar al alumno de las competencias básicas para poder llevar a cabo la tarea de gestión de proyectos. Por otra parte, se tiene el objetivo de presentar un cuerpo de conocimientos básicos para realizar la actividad de diseño de software basada en principios y patrones. Se estudiarán principios básicos del diseño de software, así como el concepto de arquitectura de software y se presentarán un conjunto de patrones arquitectónicos fundamentales. Por último, se estudiarán un conjunto bien conocido de patrones de micro-arquitectura o patrones de diseño.

**1.2 Relación con otras materias**

Tanto la Planificación y Gestión de Proyectos como el Diseño de Software son áreas de las definidas en el Cuerpo de conocimiento de Ingeniería de software (SWEBOK). Por tanto, esta asignatura está relacionada con diversas asignaturas de la materia Ingeniería de Software.

El Diseño de Software es una actividad integradora y muy técnica que incluye aspectos de conexión con bases de datos, diseño de interfaces gráficas, selección de estructuras de datos, etc., por lo que se beneficia de los conocimientos adquiridos por el alumno en otras asignaturas. Todas estas asignaturas se mencionan en el siguiente apartado de esta guía docente.

### 1.3 Prerrequisitos

Se supone que el alumno tiene conocimiento de los conceptos fundamentales de la ingeniería y del proceso de desarrollo del software.

En particular, el alumno debe tener los conocimientos aportados por las asignaturas Programación Orientada a Objetos (Segundo Curso, Primer Cuatrimestre), Fundamentos de Ingeniería del Software (Segundo Curso, Segundo Cuatrimestre) y Análisis y Diseño de Bases de Datos (Tercer Curso, Primer Cuatrimestre).

Serán de utilidad también los conocimientos adquiridos en las asignaturas Estructuras de Datos y Algoritmos (Segundo Curso, Primer Cuatrimestre) e Interfaz Persona-Computadora (Segundo Curso, Segundo Cuatrimestre) para la realización del trabajo práctico.

Se requiere una buena disposición para el trabajo en equipo pues será fundamental en el desarrollo del trabajo práctico de la asignatura.

## 2. Competencias

### 2.1 Generales

Código	Descripción
CG1	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
CT7	Toma de decisiones
CT9	Trabajo en equipo
CT12	Liderazgo
CT13	Capacidad de aplicar los conocimientos en la práctica

### 2.2 Específicas

Código	Descripción
CI2	Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.
CO1	Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.
CO3	Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.
CI8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CI16	Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la Ingeniería de Software.

### 3. Objetivos

Código	Descripción
	Reconocer y valorar la importancia y necesidad de la gestión de proyectos.
	Conocer las tareas más importantes de un responsable de proyecto y utilizar herramientas de soporte a la gestión de proyectos.
	Identificar y analizar los riesgos en el proceso de desarrollo de software.
	Ser capaces de planificar algún proyecto concreto como caso de estudio, señalando etapas, asignando costes, etc.
	Conocer los principios y conceptos fundamentales de la arquitectura y el diseño de sistemas software.
	Ser capaces de elaborar los modelos de diseño de un producto software ajustándose a un conjunto de requisitos funcionales y no funcionales.
	Ser capaces de aplicar patrones arquitectónicos y de diseño en el proceso de desarrollo de aplicaciones TI.

#### 4. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teóricas (T/M)	28	Estudio y trabajo autónomo individual	40
Clases prácticas (A)	22	Estudio y trabajo autónomo grupal	50
Laboratorios (L)			
Prácticas externas, clínicas o de campo			
Seminarios (S)	6		
Tutorías Grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	2		
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>

#### 5. Bloques temáticos

##### Bloque Organización y gestión de los proyectos informáticos (Planificación, Gestión de P1: Riesgos, etc.).

Carga de trabajo en créditos ECTS: 

##### a. Contextualización y justificación

Como se ha indicado anteriormente, la Planificación y Gestión de Proyectos (se supone de software), esta considerada en distintos puntos del SWEBOK. Por otra parte el PMBOK (cuerpo de conocimiento de gestión de proyectos) engloba todos los aspectos importantes en la realización de un proyecto de cualquier tipo. Este bloque pretende dar a conocer al alumno qué es un proyecto, qué lo caracteriza y la importancia que tienen la existencia de estándares en el desarrollo de software.

##### b. Objetivos de aprendizaje

- Conocer perfectamente las etapas de desarrollo de un proyecto, así como las ideas generales de algunos estándares de Gestión de Proyectos, como PMBOK, los capítulos adecuados de SWEBOK, etc.
- Conocer las técnicas de planificación y seguimiento de proyectos
- Elaborar la planificación temporal y de costes de un proyecto de prueba
- Conocer la incidencia que los recursos humanos tienen en la planificación de un proyecto.
- Identificar y gestionar los riesgos en el desarrollo de un proyecto software
- Elaborar los aspectos correspondientes de un Plan de Desarrollo de Software

##### c. Contenidos

- Qué es un proyecto. Tipos de proyectos y ejemplos.
- Elementos a considerar en el desarrollo de un proyecto con incidencia en los proyectos de software.



- Planificación de proyectos
  - Gestión del tiempo
  - Gestión de costes
  - Gestión de riesgos
  - Seguimiento y control de la evolución de un proyecto
- Plan de Desarrollo de Software.
- 

#### d. Métodos docentes

---

En este primer bloque se utilizarán las clases magistrales para introducir los conceptos planteando a los alumnos cuestiones relativas a las razones del fracaso en el cumplimiento de las expectativas de tiempo y coste de los proyectos software.

En la clase de prácticas se les presentaran las características generales de un software de planificación de proyectos como MS Project, tanto en cuanto a la gestión del tiempo como a la de recursos.

Por último, dado que una parte de las prácticas es orientada a proyectos se definirán grupos de trabajo y se planteará el proyecto a realizar.

#### e. Plan de trabajo

---

Ver cronograma al final de la guía

#### f. Evaluación

---

La prueba tipo test correspondiente a los contenidos de este bloque se realizará en los primeros 20 minutos de la última sesión de teoría correspondiente a este bloque.

#### g. Bibliografía básica

---

- Bob Hughes and Mike Cotterell. *Software Project Management*. McGraw Hill, 2002. ISBN 0 07 709834 X
- *A guide to Project Management Body of Knowledge*. PMI. Existe una versión en español de la tercera edición.

#### h. Bibliografía complementaria

---

- Walter Royce. *Software Project Management*. Addison-Wesley. 1998. ISBN 0-201-30958-0
- Elaine M. Hall. *Managing Risk*. Addison-Wesley. 1998. ISBN 0201255928

#### i. Recursos necesarios

---

Se introducen los conceptos básicos de la práctica de gestión de proyectos. Se dispondrá de una herramienta de planificación como MS-Project o similar.

- Damith C. Rajapakse. *Tips for succeed in Software Engineering Students projects*.  
<http://www.comp.nus.edu.sg/~damithch/guide/>

**Bloque P2: Introducción a los métodos ágiles y Gestión de las Configuraciones**Carga de trabajo en créditos ECTS: **a. Contextualización y justificación**

La gestión de configuraciones es un área de conocimiento de SWEBOK. Por otro lado, la utilización de un sistema de gestión de cambios es fundamental en el desarrollo de cualquier proyecto.

**b. Objetivos de aprendizaje**

- Entender la diferencia entre control de versiones y gestión de configuraciones
- Valorar la importancia que la gestión de configuraciones tiene en el éxito de los proyectos de software.
- Conocer los estándares de la IEEE correspondientes a gestión de configuraciones.
- Elaborar los aspectos correspondientes de un Plan de Desarrollo de Software

**c. Contenidos**

- Modelos ágiles de proceso. SCRUM
- Gestión de Configuraciones. Conceptos generales
- Operaciones de la gestión de configuraciones de software

**d. Métodos docentes**

Los conceptos básicos se seguirán presentando en clases magistrales planteando la participación del alumno en base a cuestiones puntuales relativas a los conceptos tratados.

En la parte práctica se exigirá que los grupos utilicen métodos ágiles para el desarrollo de su proyecto y alguna herramienta de software libre para la gestión de configuraciones.

**e. Plan de trabajo**

Ver cronograma al final de la guía

**f. Evaluación**

La prueba tipo test correspondiente a los contenidos de este bloque se realizará en los primeros 20 minutos de la última sesión de teoría correspondiente a este bloque.

**g. Bibliografía básica**

- S.A. Dart, *Concepts in Configuration Management Systems*, Proceedings of the third International Workshop on Software Configuration Management, ACM 1991, pag. 1-18
- Anne Mette Jonasses Hass. *Configuration Management. Principles and Practice*. Pearson Education 2003. ISBN 0-321-11766-2

**h. Bibliografía complementaria**

- *A guide to Project Management Body of Knowledge*. PMI. Existe una versión en español de la tercera edición.
- Casi todos los libros clásicos de Ingeniería de software (Pressman, Sommerville, etc.) contienen un capítulo dedicado a gestión de configuraciones.
- En cuanto a métodos ágiles y SCRUM se indicarán las direcciones web desde donde obtener el material adecuado.

#### **i. Recursos necesarios**

---

Alguno de los paquetes de control de versiones como CVS, Subversión, GIT, etc.

### **Bloque D1: Arquitectura del Software. Patrones arquitectónicos**

---

Carga de trabajo en créditos ECTS:

#### **a. Contextualización y justificación**

---

Una de las actividades fundamentales relacionadas con el diseño de software en un proceso de desarrollo es la definición de la arquitectura. La arquitectura de software es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de este para satisfacer lo que se conoce como los atributos de calidad del sistema. La arquitectura de software juega un papel fundamental para guiar el desarrollo. Una de las múltiples estructuras que la componen se enfoca en partir el sistema en componentes que serán desarrollados por individuos o grupos de individuos. La identificación de esta estructura de asignación de trabajo es esencial para apoyar las tareas de planificación y gestión del proyecto.

#### **b. Objetivos de aprendizaje**

---

- Conocer los objetivos y principios del Diseño de Software
- Conocer las arquitecturas del software y su papel dentro del proceso de desarrollo
- Elaborar y documentar la arquitectura de un producto software

#### **c. Contenidos**

---

- El diseño en el proceso de desarrollo. Artefactos de entrada y de salida.
- Principios de Diseño
- Arquitectura del Software. Catálogo de patrones arquitectónicos

#### **d. Métodos docentes**

---

Los conceptos básicos se seguirán presentando en clases magistrales planteando la participación del alumno en base a cuestiones puntuales relativas a los conceptos tratados.

En la parte práctica se exigirá que los grupos definan la arquitectura del sistema a construir aplicando los principios de diseño enunciados.

#### **e. Plan de trabajo**

---

Ver cronograma al final de la guía

#### f. Evaluación

---

La prueba tipo test correspondiente a los contenidos de este bloque se realizará en los primeros 20 minutos de la sesión del martes correspondiente a la semana 6 del curso.

#### g. Bibliografía básica

---

**Pattern-Oriented Software Architecture Volume 1: A System of Patterns.** Frank Buschmann , Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Wiley, ISBN-13 978-0471958697

**UML 2.** Jim Arlow, Ila Neustadt. Anaya Multimedia. ISBN-13: 978-8441520332

#### h. Bibliografía complementaria

---

**El Lenguaje Unificado de Modelado. Guía del usuario.** Booch, G., Jacobson, I., Rumbaugh, J. Addison-Wesley/Diaz de Santos, 2o edición, 2005

**El Lenguaje Unificado de Modelado. Manual de referencia.** Rumbaugh, J., Jacobson, I., Booch, G. 2o edición. Pearson, 2007

**Patterns of Enterprise Application Architecture.** Fowler, M., Addison-Wesley Professional; 1era ed, 2002

**Software Architecture in Practice.** Len Bass, Paul Clements, Rick Kazman, Addison-Wesley Professional; 2da ed, 2003

**Essential Software Architecture.** Gorton, I. Springer, 2da, ed., 2011

**Documenting Software Architectures: Views and Beyond.** Clements, P. et al, Springer, 2da ed., 2010

#### i. Recursos necesarios

---

Herramientas de modelado UML, Astah Profesional o Visual Paradigm (ambas con licencias para la Universidad de Valladolid)

### Bloque Patrones de Diseño. Diseño detallado D2:

---

Carga de trabajo en créditos ECTS:

#### a. Contextualización y justificación

---

Los patrones de diseño son una guía para resolver con buenas características de calidad problemas habituales. El conocimiento sobre patrones de diseño ayuda a tener un lenguaje común con otros diseñadores de software y por tanto facilita la comunicación en un equipo de trabajo. Por otra parte, el conocimiento sobre un catálogo esencial de patrones de diseño



ayuda a acortar los tiempos de desarrollo permitiendo la rápida aplicación de soluciones preconcebidas de éxito conocido.

---

**b. Objetivos de aprendizaje**

---

- Evaluar diferentes alternativas de diseño en base a los principios del diseño de software
- Aplicar patrones de diseño adecuados para la construcción de una aplicación software.
- Desarrollar y probar un proyecto software que cumple con el diseño propuesto.
- Documentar el diseño detallado de un diseño detallado

---

**c. Contenidos**

---

- Realización en diseño de casos de uso.
- Patrones de micro-arquitectura.
- Diseño detallado.

---

**d. Métodos docentes**

---

Los conceptos básicos se seguirán presentando en clases magistrales planteando la participación del alumno en base a cuestiones puntuales relativas a los conceptos tratados.

Resolución de problemas en clase.

En la parte práctica se exigirá que los grupos desarrollen la realización en diseño de un sistema pequeño basado en casos de uso, documenten el diseño detallado y desarrollen dicho sistema conforme al diseño propuesto.

---

**e. Plan de trabajo**

---

Ver cronograma al final de la guía

---

**f. Evaluación**

---

La prueba tipo test correspondiente a los contenidos de este bloque se realizará en los primeros 20 minutos de la sesión del martes correspondiente a la semana 13 del curso.

---

**g. Bibliografía básica**

---

**Design Patterns: Elements of Reusable Object-Oriented Software.** Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, Prentice Hall; ISBN-13 978-0201633610

**UML y Patrones. Introducción al Análisis y Diseño Orientado a Objetos y al Proceso Unificado.** Larman, C. Prentice Hall, 2002. (2a ed.)

---

**h. Bibliografía complementaria**

---

**Head First Design Patterns.** Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra. O'Reilly Media. ISBN-13: 978-0596007126

**Pattern Hatching: Design Patterns Applied.** John M. Vlissides. Addison Wesley. ISBN-13: 978-0201432930

**Patrones De Diseño En Java. Los 23 Modelos De Diseño. Descripción Y Solución Ilustradas En UML 2 y Java.** Laurent Debrauwer. ENI. ISBN-13: 978-2746086456

#### i. Recursos necesarios

Herramientas de modelado UML, Astah Profesional o Visual Paradigm (ambas con licencias para la Universidad de Valladolid). Entorno de desarrollo integrado Netbeans.

#### 6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Gestión de proyectos de software. Introducción, Planificación y Control	2	Semana 1 a 10 (50%)
Métodos ágiles. Gestión de las Configuraciones.	1	Semana 11 a 15 (50%)
Arquitectura del Software. Patrones Arquitectónicos	1	Semana 1 al 5 (50%)
Patrones de Diseño. Diseño Detallado	2	Semana 6 al 15 (50%)

#### 7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Examen de teoría	40%	Para poder compensar se deberá obtener una nota de 4,5 o superior en la parte teórica.
Valoración de la práctica	40%	Será necesaria la defensa, con asistencia de todos los miembros del grupo de la práctica. Para poder considerar la nota obtenida en la parte práctica será necesario haber alcanzado, al menos, un 4,5 en la calificación de dicha prueba. En dicha calificación se tendrá en cuenta, tanto los aspectos de funcionalidad, como la calidad de la documentación aportada

		y la defensa realizada.
Pruebas tipo test	10%	Se realizarán hasta 4 pruebas a lo largo del desarrollo de la asignatura
Valoración de las tareas y de la aportación en seminarios y tutorías	10%	A lo largo del curso se propondrán al alumno la realización de 1 seminario y 1 tarea entregable que serán valoradas en esta parte de la evaluación.

CRITERIOS DE CALIFICACIÓN
<ul style="list-style-type: none"> <li> <b>Convocatoria ordinaria:</b>                      Los criterios se han indicado en el punto de sistema de calificaciones. Conviene recordar la necesidad de superar un 4,5 en el examen de teoría y de práctica para poder compensar la otra parte (práctica o teoría). En el caso en que algún alumno no superara un 4,5 en la parte de teoría y de práctica la nota que se reflejaría sería el mínimo de entre la obtenida utilizando la suma ponderada indicada antes (práctica*0,4+examen final*0,4+pruebas tipo test*0,1+ Valoración*0,1) y 4,0.                 </li> <li> <b>Convocatoria extraordinaria:</b>                      Si la práctica ya está superada se realizará un examen de teoría con el mismo formato que en la convocatoria ordinaria. Si, por el contrario, la práctica no estuviera superada deberá volver a presentar la práctica (en una única entrega) y realizar un examen de teoría con el mismo formato que en la convocatoria ordinaria. En ambos casos las pruebas se considerarán con similar peso y fórmula de cálculo para la nota final será (práctica*0,5+examen final*0,5).                 </li> </ul>

### Consideraciones finales

- Un 20% corresponderá a la valoración de los seminarios, pruebas tipo test y de la participación de los alumnos en las discusiones planteadas.
- Calificación de la convocatoria extraordinaria: Si la práctica ya está superada se realizará un examen con el mismo formato que en la convocatoria ordinaria. Si, por el contrario, la práctica no estuviera superada deberá volver a presentar la práctica (en una única entrega) y realizar un examen con el mismo formato que en la convocatoria ordinaria. En ambos casos las pruebas se considerarán con similar peso y fórmula de cálculo para la nota final que en la convocatoria ordinaria.

### 8. Anexo: Métodos docentes

Actividad	Metodología
<b>Clase de teoría</b>	<ul style="list-style-type: none"> <li>• Clase magistral participativa</li> <li>• Estudio de casos en aula</li> </ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"> <li>• Clase magistral participativa</li> <li>• Realización de un proyecto guiado por el profesor, que encargará y guiará el trabajo que se realizará en grupos (4 alumnos), siguiendo un enfoque colaborativo.</li> </ul>



<b>Seminarios</b>	<ul style="list-style-type: none"> <li>Talleres de aprendizaje</li> </ul>
<b>Tutoría</b>	<ul style="list-style-type: none"> <li>Evaluación de los contenidos teóricos y de los proyectos</li> </ul>

### 9. Anexo: Cronograma de actividades previstas

Semana	Tiempo	Hora	Tipo Actividad	Contenido
1	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	Presentación de MS Project y otras herramientas de Planificación
1	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
1	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
2	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	MS Project. Gestión de recursos.
2	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
2	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
3	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	Discusión sobre el trabajo de la práctica
3	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
3	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
4	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
4	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
4	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
5	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
5	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
5	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
6	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
6	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	Test del Bloque Arquitectura del Software
6	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
7	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
7	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
7	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
8	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
8	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
8	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
9	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	No lectivo
9	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	No lectivo
9	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	No lectivo
10	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
10	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
10	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
11	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
11	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
11	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
12	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
12	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	





12	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
13	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
13	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	Test del Bloque Patrones de Diseño
13	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	
14	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	No lectivo
14	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	No lectivo
14	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	No lectivo
15	Lunes (2 horas)	11:00-13:00	Lab./Sem./Tuto.	
15	Martes (1 hora)	10:00-11:00	Teoría (Diseño)	
15	Martes (1 hora)	12:00-13:00	Teoría (Planificación)	

