

**Proyecto/Guía docente de la asignatura**

Asignatura	PLANIFICACIÓN Y DISEÑO DE SISTEMAS COMPUTACIONALES		
Materia	COMPUTACIÓN		
Módulo	TECNOLOGÍAS ESPECÍFICAS		
Titulación	PLAN DE ESTUDIOS CONJUNTO DE GRADO EN ESTADÍSTICA Y DE GRADO EN INGENIERÍA INFORMÁTICA (INdat)		
Plan	51	Código	46955
Periodo de impartición	1	Tipo/Carácter	OB
Nivel/Ciclo	GRADO	Curso	5º
Créditos ECTS	6		
Lengua en que se imparte	ESPAÑOL		
Profesor/es responsable/s	YANIA CRESPO GONZÁLEZ-CARVAJAL Y CÉSAR LLAMAS BELLO		
Datos de contacto (E-mail, teléfono...)	yania_at_infor.uva.es Teléfono: 983 185695 cllamas_at_infor.uva.es Teléfono: 983 185610		
Departamento	INFORMÁTICA (ATC, CCIA y LSI)		



1. Situación / Sentido de la Asignatura

1.1 Contextualización

Esta asignatura está integrada en el quinto curso Programa de Estudios Conjunto del Grado de Estadística y 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 la comunidad internacional de ingenieros, entre las que se encuentra el Project Management Institute (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 presupone 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

- 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

- 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

- 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.

4. Contenidos y/o bloques temáticos

La asignatura se compone de dos partes que abordan los siguientes contenidos, globalmente:

Bloque 1: Organización y Gestión de Proyectos Informáticos.

Bloque 2: Arquitectura del Software. Patrones Arquitectónicos y de Diseño.

Bloque 1: Planificación y Gestión de Proyectos Informáticos (Plan, Gestión de Riesgos, etc.).

Carga de trabajo en créditos ECTS: 3

a. Contextualización y justificación

Este bloque de contenidos se refiere principalmente a los contenidos relativos a la actividad de Planificación y Gestión de Proyectos Informáticos a los que se refiere el planteamiento general indicado en el punto 1.1.

La construcción de software es una actividad compleja que involucra diversas actividades y que por lo general requiere de la colaboración de varias personas y diversas entidades, mucho más allá de la actividad individual de un analista, programador o diseñador. La gestión de proyectos estudia las actividades adyacentes a los métodos de análisis, diseño, prueba del software como pueden ser la gestión de personas, la gestión de los riesgos, la estimación del trabajo y el coste, la planificación de las actividades, la gestión de los contratos y el seguimiento de los proyectos.

Aún existiendo diversas aproximaciones tradicionales y novedosas (ágiles), el alumno tendrá la ocasión de entrar en contacto con este cuerpo de doctrina y que presupone cierto nivel de conocimientos que todo ingeniero debe tener y que le permitirán entrar a participar en grupos de trabajo bajo la supervisión de gestores de proyectos con experiencia.

b. Objetivos de aprendizaje

- Conocer las actividades básicas relativas a la planificación y a la gestión de las tareas relativas a los proyectos software, para adquirir una base común y sólida de gestión de proyectos.
- Aplicar los conocimientos básicos de planificación y gestión a proyectos sencillos y de tipo ágil para poder integrarse con facilidad en un equipo de desarrollo software.
- Ser capaz de gestionar proyectos software de nivel inicial mediante las herramientas más comunes.

c. Contenidos

1. Conceptos básicos sobre gestión de proyectos software.
2. Evaluación de proyectos y gestión del programa.
3. Introducción a la planificación de un proyecto informático.
4. Aproximaciones de proyectos informáticos.
5. Introducción a los métodos ágiles.
6. Estimación del esfuerzo software.
7. Planificación de actividades.
8. Gestión de riesgos.
9. Asignación de recursos.
10. Seguimiento y monitorización de proyectos.

d. Métodos docentes

Modelo de docencia inversa: El profesor ofrece bien transparencias, píldoras de vídeo, textos y material didáctico online, acompañado de material on-line elaborado por el profesor, que servirá de apoyo para ilustrar en las clases presenciales aquellos puntos con alguna dificultad que serán planteados por los alumnos o suscitados mediante ejercicios planteados por el profesor.

Tutorías activas: donde el profesor plantea ejercicios mediante el campus virtual de la Uva, y que se desarrollarán posteriormente en el aula estudiando aquellos puntos que hayan demostrado especial dificultad a la vista de las evaluaciones.

Prácticas guiadas: donde el profesor mediante material on-line y vídeo expondrá la parte básica del uso de aplicaciones de laboratorio o temas eminentemente prácticos con anterioridad suficiente a las clases prácticas. El alumno podrá así completar dichas actividades (guiones) de laboratorio que consistirán en supuestos prácticos y preguntas, lo cual que permite evaluar el desempeño del alumno.

Práctica de proyecto: en la que de común acuerdo con el Bloque 2 de la asignatura, se tratará de conseguir los objetivos y habilidades en la realización de un proyecto sencillo software en lo que respecta a la planificación y gestión de proyectos.

e. Plan de trabajo

Este bloque de la asignatura se realizará en primer lugar y ocupará las primeras semanas del cuatrimestre en exclusiva, de modo equitativo con el Bloque 2.

- La parte teórica de este bloque de la asignatura se desarrollará en primer lugar, ocupando para ello el tiempo asignado al aula de laboratorio y aula de teoría por la Escuela de Ingeniería Informática (E.I. Informática), lo cual es posible dado que se cuenta con un sólo grupo de práctica.
- Las prácticas guiadas de este bloque se realizarán posteriormente ocupando las horas correspondientes al desglose de teórico-práctico de la asignatura, tanto en laboratorio como en aula aprovechando las nuevas tecnologías proporcionadas por la E.I. Informática.

f. Evaluación

Este bloque de la asignatura se adhiere a un modelo de “**evaluación continuada**”, de modo que el alumno podrá demostrar su desempeño en tres tipos de actividades, en lo concerniente a este bloque de la asignatura:

- Se realizarán **dos tutorías activas** hacia la mitad de la primera parte y como conclusión de la primera parte de la asignatura. La primera tutoría comprenderá los temas 1 al 5. La segunda tutoría comprenderá los temas 6 al 10. Las tutorías tendrán el formato de *Prueba de opción múltiple*. El alumno deberá realizar un desempeño suficiente (5 puntos) en cada parte.
- Se llevarán a cabo diversas **prácticas guiadas**, cuyo número podrá variar en función de la disponibilidad de aulas para los alumnos y que podrán realizarse presencialmente según la capacidad de los laboratorios. Tendrán el formato de *pruebas de opción múltiple y envío de tareas*. Serán evaluadas y formarán parte la nota práctica de este bloque de la asignatura.
- Se evaluará conjuntamente con el Bloque 2 el desempeño realizado en la realización de un **proyecto software sencillo** en lo concerniente al Bloque 1 de Planificación y Gestión.

g Material docente



g.1 Bibliografía básica

- Bob Hughes y Mike Cotterell, “Software Project Management”, 5ªEd. McGraw Education, 2009.

g.2 Bibliografía complementaria

- Project Management Institute, “A guide to the Project Management Body of Knowledge (PMBOK guide) & Agile practice guide bundle” 6th Ed. PMI. 2020
- Lasa Gómez y otros, “Métodos Ágiles. Scrum Kanban, Lean” (
- Project Management Institute, “A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - The Standard for Project Management”. Project Management Institute. 2021
- Project Management Institute, “Guía Práctica de Ágil”, Project Management Institute

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

- Así mismo se proporcionarán enlaces a material on-line estable proporcionado por asociaciones profesionales y artículos de consulta pública.
- Además el profesor elaborará podcast y guías de laboratorio.
- El portal web de la Uva permitirá la realización de pruebas, consultas y el seguimiento de tareas.

h. Recursos necesarios

- El alumno deberá poder tener acceso al software MICROSOFT PROJECT, que se encontrará instalado en los ordenadores de laboratorio, y de los que el alumnado dispone de una licencia de campus para el caso de que se vea en la necesidad de realizar el laboratorio remotamente.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1,5 Teoría	Desde la semana 1 a la 4 del cuatrimestre. 1. Prueba teórica con un peso del 10 % de la asignatura en la 4ª semana del cuatrimestre. 2. Prueba teórica con un peso del 10 % de la asignatura en la 7ª semana del cuatrimestre.
1,5 Práctica	Desde la semana 4 hasta la 15 del cuatrimestre en paralelo con el Bloque 2 de la asignatura. Encargo práctico de la asignatura mediante laboratorio y desarrollo de un proyecto en conjunción con el bloque 2 en la 15ª semana del cuatrimestre.

Bloque 2: Arquitectura del Software. Patrones Arquitectónicos y de Diseño.

Carga de trabajo en créditos ECTS: 3

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 descomponer el sistema en partes que serán desarrolladas 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.

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

- 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.
- 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 una aplicación.

c. Contenidos

1. El diseño en el proceso de desarrollo. Artefactos de entrada y de salida.
2. Principios de Diseño.
3. Arquitectura del Software. Catálogo de Estilos y Patrones Arquitectónicos.
4. Realización en diseño de casos de uso.
5. Patrones de microarquitectura.
6. Diseño detallado

d. Métodos docentes

Los conceptos básicos se seguirán presentando en clases magistrales participativas, planteando la participación del alumno en base a cuestiones puntuales relativas a los conceptos tratados y a los ejemplos y casos de estudio tratados.

Resolución de problemas en clase.

Tutorías activas: donde el profesor plantea ejercicios mediante el campus virtual de la Uva, y que se desarrollarán posteriormente en el aula analizando aquellos puntos que hayan demostrado especial dificultad a la vista de las evaluaciones.

Aprendizaje Basado en Proyecto (ABP): Los alumnos realizarán un pequeño proyecto software en equipo, planificado en el Bloque 1, el que se hará énfasis en la parte de diseño arquitectónico, diseño del comportamiento y diseño detallado, aplicando los principios y patrones explicados

e. Plan de trabajo

Este bloque de la asignatura se realizará a continuación del Bloque 1.

- La parte teórica de este bloque de la asignatura se desarrollará en primer lugar, ocupando para ello el tiempo asignado al aula de laboratorio y aula de teoría por la Escuela de Ingeniería Informática (E.I. Informática), lo cual es posible dado que se cuenta con un sólo grupo de práctica.



- Las prácticas guiadas de este bloque se realizarán posteriormente ocupando las horas correspondientes al desglose de teórico-práctico de la asignatura, tanto en laboratorio como en aula aprovechando las nuevas tecnologías proporcionadas por la E.I. Informática.

f. Evaluación

Este bloque de la asignatura se adhiere a un modelo de “**evaluación continuada**”, de modo que el alumno podrá demostrar su desempeño en tres tipos de actividades, en lo concerniente a este bloque de la asignatura:

- Se realizará **una tutoría activa** hacia la mitad de la segunda parte del bloque 2 el alumno deberá realizar un desempeño suficiente en esta parte.
- Se realizará un ejercicio evaluable de aplicación de patrones de diseño, conjuntamente con una prueba teórica en la convocatoria ordinaria de la asignatura.
- Se evaluará conjuntamente con el Bloque 1 de Planificación y Gestión el desempeño realizado en la realización de un **proyecto software sencillo**.

g Material docente

g.1 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
- *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.)

g.2 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
- *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



g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), ...)

El alumno contará con algunos videotutoriales grabados por el profesorado. El campus virtual de la Uva permitirá la realización de pruebas, consultas y el seguimiento de tareas. Los alumnos utilizarán el gitlab de la Escuela (gitlab.inf.uva.es) para realizar la gestión de la configuración del proyecto software y el seguimiento de las tareas y la carga de horas del proyecto.

h. Recursos necesarios

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

Entornos de desarrollo integrado (IDEs) a decisión de los estudiantes.

Lenguajes de programación, frameworks y bibliotecas a decisión de los estudiantes.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
1,5	Parte teórica del Bloque 2 de la semana 8 a la 10.
1,5	Parte práctica del Bloque 2 mediante laboratorio y desarrollo de un proyecto en conjunción con el Bloque 1 de la semana 11 a la 15.

5. Métodos docentes y principios metodológicos

Véase el apartado (d) Métodos docentes de los Bloques 1 y 2.

**6. Tabla de dedicación del estudiante a la asignatura**

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA ⁽¹⁾	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases Teóricas	24	Estudio y trabajo autónomo individual	44
Clases Prácticas	25	Estudio y trabajo autónomo grupal	54
Evaluación fuera de periodo de exámenes.	2		
Presentación del proyecto final software e la asignatura	1		
Total presencial	52	Total no presencial	98
TOTAL presencial + no presencial			150

(1) Actividad presencial a distancia es cuando un grupo sigue una videoconferencia de forma síncrona a la clase impartida por el profesor.

7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
2 Pruebas de tipo test sobre la parte teórica del Bloque 1	20 %	El estudiante deberá conseguir una calificación de 5 puntos en cada prueba, o compensar con 4,5 una de ellas con la otra.
4 Prácticas de laboratorio sobre la parte práctica del Bloque 1	20 %	Del total de las seis prácticas previstas, el alumno deberá realizar al menos las 4 últimas prácticas con un desempeño suficiente, globalmente consideradas.
Realización y seguimiento del proyecto software.	10 %	Conjuntamente con el Bloque 2 se deberá demostrar un desempeño suficiente en los temas correspondientes al Bloque 1 de la asignatura en el proyecto software final.
Proyecto software	20 %	Conjuntamente con el Bloque 1 se evaluará la parte de diseño arquitectónico y diseño e implementación del proyecto.
Presentación y defensa del proyecto software.	5 %	Se evaluará de forma integral la presentación del proyecto, las respuestas a las preguntas realizadas por los profesores y la claridad de la presentación
2 Pruebas de tipo test sobre la parte teórica del Bloque 2.	10 %	El estudiante deberá obtener una calificación de 5 puntos en cada prueba o compensar con 4,5 una de ellas con la otra.
Ejercicio práctico sobre aplicación de patrones de diseño.	15 %	Evaluación sumativa.



CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
 - El alumno deberá demostrar un conocimiento suficiente en cada bloque de la asignatura (Bloque 1 y 2). Véase la tabla de Instrumentos de Evaluación.
 - El alumno deberá presentar y defender junto con su equipo el proyecto software realizado.
- **Convocatoria extraordinaria:**
 - Se conservarán los porcentajes de la evaluación aplicada en la convocatoria ordinaria.
 - El alumno podrá recuperar por separado los temas de cada una de las dos pruebas teóricas del Bloque 1, conservando la nota que hubiera obtenido en la evaluación continuada. Las pruebas tendrán el mismo formato que en la evaluación continuada.
 - Con respecto a la parte práctica del Bloque 1, el alumno podrá recuperar la parte práctica entregando un trabajo de planificación y seguimiento de un supuesto que será enunciado por el profesor y que puede ser evaluado por los profesores de la asignatura mediante entrevista. Este trabajo podrá estar conectado con la recuperación del Proyecto global enunciado anteriormente.
 - Los alumnos podrán trabajar por equipo o individualmente en el proyecto software para mejorar la nota obtenida en la convocatoria ordinaria. De ser así, deberán tener una sesión de presentación y defensa de las mejoras realizadas.
 - Podrá realizarse una recuperación del ejercicio de patrones de diseño, la nota de este ejercicio obtenida en la convocatoria extraordinaria reemplaza la nota de la convocatoria ordinaria.

8. Consideraciones finales

