



Universidad
de Alcalá

GUÍA DOCENTE

PATRONES SOFTWARE

Grado en Ingeniería Informática
Grado en Ingeniería de Computadores
**Grado en Ingeniería en Sistemas de
Información**

Universidad de Alcalá

Curso Académico 2019/2020
Curso 4^o – Cuatrimestre 1^o

GUÍA DOCENTE

Nombre de la asignatura:	Patrones Software
Código:	780042
Titulación en la que se imparte:	Grado en Ingeniería Informática Grado en Ingeniería de Computadores Grado en Ingeniería en Sistemas de Información
Departamento y Área de Conocimiento:	Departamento Ciencias de la Computación
Carácter:	Optativa
Créditos ECTS:	6
Curso y cuatrimestre:	4º Curso / 1º Cuatrimestre
Profesorado:	Salvador Otón Tortosa Consultar en la página web del departamento
Horario de Tutoría:	El horario de Tutorías se indicará el primer día de clase
Idioma en el que se imparte:	Español / English Friendly

1. PRESENTACIÓN

La asignatura Patrones Software pretende enseñar a los alumnos la utilización de patrones software en el desarrollo de aplicaciones informáticas como parte fundamental de la ingeniería del software. La asignatura se centra en el uso de los patrones de diseño que representan una solución a problemas que se dan de una forma habitual en la creación de software.

Prerrequisitos y Recomendaciones

Es obligatorio haber cursado las asignaturas relacionadas con programación e ingeniería del software.

1.b PRESENTATION

The Software Patterns subject aims to teach students the use of standards in the development of software applications as an essential part of software engineering. The subject focuses on the use of design patterns that represent a solution to problems that occur in a normal way in creating software.

2. COMPETENCIAS

Competencias generales:

CG4 Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de la resolución BOE-A-2009-12977.

CG5 Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de la resolución BOE-A-2009-12977.

CG8 Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

CG9 Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.

CG12 Conocimiento y aplicación de elementos básicos de economía y de gestión de recursos humanos, organización y planificación de proyectos, así como la legislación, regulación y normalización en el ámbito de los proyectos informáticos, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 de la resolución BOE-A-2009-12977.

Competencias específicas:

CIS1 Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.

CIS2 Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.

CIS3 Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

CIS4 Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

CIS5 Capacidad de identificar, evaluar y gestionar los riesgos potenciales asociados que pudieran presentarse.

CIC4 Capacidad de diseñar e implementar software de sistema y de comunicaciones.

CC1 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.

CC4 Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.

CSI3 Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

Resultados de aprendizaje esperados

Los resultados de aprendizaje esperados, determinados a partir de las competencias específicas incluidas en la memoria verificada de la titulación como competencias específicas, son los siguientes:

- RA1. Definir los distintos tipos de patrones software y sus características principales. Presentar el concepto de patrón de diseño y sus características.
- RA2. Identificar los patrones fundamentales en los que se basan los patrones de diseño.
- RA3. Describir las tres familias en las que se clasifican los patrones de diseño e identificar el uso de cada uno de los patrones. Determinar cuál es el patrón de diseño más adecuado a un problema concreto.
- RA4. Saber interpretar el modelo gráfico que representa la estructura de un patrón de diseño y poder adaptarlo al dominio de un problema concreto.
- RA5. Analizar y diseñar aplicaciones en las que se aplican los patrones de diseño.
- RA6. Desarrollar aplicaciones en lenguajes de programación orientados a objetos que incluyan patrones de diseño.
- RA7. Saber combinar diversos patrones de diseño para resolver un problema software complejo.
- RA8. Definir que es un framework y cuál es su aplicación en la construcción de aplicaciones software. Mostrar la relación que existe entre los patrones de diseño y los frameworks.

3. CONTENIDOS

1. Introducción a los patrones software: Orígenes, características, clasificación, Patrones vs Frameworks.
2. Patrones Fundamentales: Delegation, Interface, Abstract Superclass, Interface and Abstract Class, Immutable, Marker Interface, Proxy.
3. Patrones de Creación: Introducción, Patrón Abstract Factory, Patrón Builder, Patrón Factory Method, Patrón Prototype, Patrón Singleton.
4. Patrones Estructurales: Patrón Adapter, Patrón Bridge, Patrón Composite, Patrón Decorador, Patrón Facade, Patrón Flyweight, Patrón Proxy.
5. Patrones de Comportamiento: Patrón Chain of responsibility, Patrón Command, Patrón Interpreter, Patrón Iterator, Patrón Mediator, Patrón Memento, Patrón Observer, Patrón State, Patrón Strategy, Patrón Template Method, Patrón Visitor.
6. Otros Patrones: Patrones GRASP, Antipatrones.
7. Frameworks: Características de los frameworks, Tipos de frameworks, Ejemplos de frameworks.

Bloques de contenido (se pueden especificar los temas si se considera necesario)	Total de clases, créditos u horas
Introducción a los patrones software	4 horas
Patrones Fundamentales	4 horas
Patrones de Creación	10 horas
Patrones Estructurales	12 horas
Patrones de Comportamiento	18 horas
Otros Patrones	4 horas
Frameworks	4 horas

PECS

4 horas

4. METODOLOGÍAS DE ENSEÑANZA-APRENDIZAJE.- ACTIVIDADES FORMATIVAS

La asignatura Patrones Software se organiza como una asignatura cuatrimestral de 6 ECTS (150 horas).

En el proceso de enseñanza-aprendizaje de los contenidos anteriormente reseñados se emplearán las siguientes actividades formativas:

- Clases Teóricas presenciales.
- Clases Prácticas: resolución de problemas presenciales.
- Prácticas en Laboratorio presenciales.
- Tutorías: individuales y/o grupales.

Además, en función de la naturaleza de las distintas partes de la materia objeto de estudio, se podrán utilizar, entre otras, las siguientes actividades formativas:

- Elaboración de trabajos con responsabilidad individual pero con gestión de la información como equipo.
- Puesta en común de la información, problemas y dudas que aparezcan en la realización de los trabajos.
- Organización y realización de jornadas públicas con presentaciones orales y discusión de resultados.
- Utilización de Plataforma de Aula Virtual.

Actividades presenciales:

1. En el aula: exposición y discusión de los conocimientos básicos de la asignatura. Planteamiento y resolución teórica de ejercicios y supuestos relacionados. Orientadas a la enseñanza de las competencias específicas de la asignatura, especialmente las relacionadas con los conocimientos y las técnicas de utilización de los patrones software en la creación de aplicaciones informáticas.
2. En el laboratorio: planteamiento y desarrollo de ejercicios prácticos que permitan solventar problemas y analizar hipótesis y contribuyan al desarrollo de la capacidad de análisis de resultados, razonamiento crítico y comprensión de los métodos de resolución planteados. Servirán como base para la adquisición de las competencias genéricas descritas en el apartado 2.

Actividades no presenciales:

1. Análisis y asimilación de los contenidos de la materia, resolución de problemas, consulta bibliográfica, preparación de trabajos individuales y grupales, realización de exámenes presenciales y autoevaluaciones. Orientadas

especialmente al desarrollo de métodos para la autoorganización y planificación del trabajo individual y en equipo.

2. Tutorías: asesoramiento individual y en grupos durante el proceso de enseñanza-aprendizaje, bien en forma presencial o a distancia.

4.1. Distribución de créditos (especificar en horas)

Número de horas presenciales:	56 horas + 4 horas de examen de evaluación
Número de horas del trabajo propio del estudiante:	90 horas
Total horas	150 horas

4.2. Estrategias metodológicas, materiales y recursos didácticos

Materiales y recursos:

- Bibliografía de referencia sobre la asignatura.
- Ordenadores personales.
- Entornos de desarrollo y manuales de uso de los mismos.
- Conexión a Internet.
- Plataforma de Aula Virtual y manuales de uso de las mismas.
- Proyector.

5. EVALUACIÓN: Procedimientos, criterios de evaluación y de calificación

En la convocatoria ordinaria el método de evaluación por defecto es la “evaluación continua”, con características de evaluación formativa para servir de realimentación en el proceso de enseñanza-aprendizaje por parte del alumno.

Opcionalmente, y de manera justificada, el alumno podrá solicitar ante el director del centro la evaluación mediante prueba única, lo que deberá de ser solicitado por escrito y en los plazos reglamentados. Este método de evaluación consiste en un examen teórico y una práctica de laboratorio.

En la convocatoria extraordinaria la evaluación se basará en una prueba única, consistente en un examen teórico y una práctica de laboratorio, en la que se determinará el grado de dominio de las competencias de la asignatura.

Criterios de Evaluación

El rendimiento de los alumnos se evaluará atendiendo a los conocimientos y destrezas adquiridas. Los métodos a emplear serán: la resolución de casos prácticos y trabajos planteados, defensa pública de determinados trabajos, realización de una práctica final, así como la realización de exámenes para comprobar los conocimientos teóricos de la asignatura.

Cada una de las pruebas de evaluación continua constará de dos partes Teoría y Laboratorio. Distribución de la calificación en las Pruebas de Evaluación Continua (PEC's):

- La parte teórica se valorará en un 50% y la de prácticas de laboratorio en otro 50%, para obtener el 100% de la nota de la asignatura.
- Para la parte teórica se realizarán dos PEI (Prueba de Evaluación Intermedia) representando cada una el 50% de la nota total de teoría. (PEI1 – PEI2)
- Para la parte de prácticas de laboratorio se realizarán dos PL (Prueba de Laboratorio), teniendo un peso del 10% y el 90% (PL1 – PL2) respectivamente.

En la Convocatoria Extraordinaria la evaluación se basará en dos pruebas, consistentes en un examen teórico y una práctica de laboratorio (con una valoración de un 50% cada una), en la que se determinará el grado de dominio de las competencias de la asignatura.

Los estudiantes que se acojan a Pruebas de Evaluación Final deberán realizar dos pruebas, consistentes en un examen teórico y una práctica de laboratorio (con una valoración de un 50% cada una), en la que se determinará el grado de dominio de las competencias de la asignatura.

Se establecen los siguientes criterios de evaluación específicos para la asignatura:

- CE1. El alumno conoce los distintos tipos de patrones software y sus características, así como el concepto de patrón de diseño.
- CE2. El alumno sabe identificar los patrones fundamentales.
- CE3. El alumno conoce las tres familias de patrones de diseño y comprende la aplicabilidad de cada uno de los patrones.
- CE4. El alumno ha adquirido los conocimientos necesarios para adaptar la estructura general de un patrón de diseño en un problema concreto.
- CE5. El alumno ha adquirido los conocimientos necesarios para aplicar patrones de diseño en el desarrollo de una aplicación informática.
- CE6. El alumno sabe plasmar en un lenguaje orientado a objetos la aplicación de los patrones de diseño.
- CE7. El alumno comprende que es un framework y su relación con los patrones de diseño.

En la siguiente tabla se indica el peso en la calificación (entre 0 y 100) de cada prueba, y su relación con los criterios de evaluación, resultados de aprendizaje y competencias generales.

Competencias	Resultado Aprendizaje	Criterio de Evaluación	Instrumento de Evaluación	Peso en la calificación
CG4, CG5, CG8, CG9, CG12. CIS1, CIS2, CIS3, CIS4, CIS5, CIC4, CC1, CC4, CSI3	RA1, RA2, RA3, RA4, RA5, RA8	CE1, CE2, CE3, CE4, CE5	PEI1, PEI2	50
	RA4, RA5, RA6, RA7	CE5, CE6, CE7	PL1, PL2	50

Para la evaluación final y extraordinaria se tendrán que realizar dos pruebas, una con los contenidos teóricos (PT) y otra con los de laboratorio (PL) teniendo un peso del 50% cada una de ellas.

Competencias	Resultado Aprendizaje	Criterio de Evaluación	Instrumento de Evaluación	Peso en la calificación
CG4, CG5, CG8, CG9, CG12. CIS1, CIS2, CIS3, CIS4, CIS5, CIC4,	RA1, RA2, RA3, RA4, RA5, RA8	CE1, CE2, CE3, CE4, CE5	PT	50
	RA4, RA5, RA6, RA7	CE5, CE6, CE7	PL	50

CC1, CC4, CSI3				
-------------------	--	--	--	--

6. BIBLIOGRAFÍA

Bibliografía Básica

Patrones de Diseño

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Addison Wesley, 2003

Patrones de diseño aplicados a Java

Stephen Steling, Olav Maassen
Pearson Education / Sun Microsystems, 2003

Head First Design Patterns

Eric Freeman, Elisabeth Freeman
O'Reilly, 2004

Software Architecture Design Patterns in Java

Partha Kuchana
Auerbach, 2004

Patterns in Java, volumen 1 y 2

Mark Grand
Wiley computer publishing, 1998

Java Design Patterns: A Tutorial

James W. Cooper
Addison-Wesley, 2000

Patrones de diseño en Java

Laurent Debrauwer
Ediciones Eni, 2013

Bibliografía Complementaria

Programación orientada a objetos con Java.

Salvador Otón; J.J. Martínez; J.R. Hilera.
Sº Publ. Universidad de Alcalá, 2000.

Construcción de Software Orientado a Objetos. Segunda Edición.

Bertrand Meyer. Prentice Hall, 1998.

Core J2EE™ Patterns: Best Practices and Design Strategies
Deepak Alur, John Crupi, Dan Malks. Prentice Hall / Sun Microsystems Press,
2001

TEACHING GUIDE

SOFTWARE PATTERNS

**Bachelor's Degree in
Computer Engineering
Computer Science
Information Systems Engineering**

University of Alcalá

**Academic Year 2019/2020
4th Course - 1st Semester**

TEACHING GUIDE

Subject:	Software Patterns
Code:	780042
Degree in which it is taught:	Degree in Computer Engineering Degree in Computer Science Degree in Information Systems Engineering
Department and Area of:	Sciences Department Computer
Character:	Optional
ECTS Credits:	6
Course and semester:	4th year / 1st semester
Faculty:	Salvador Otón Tortosa See the department web site
hours:	Tutorials hours will be indicated the first day of class
Language in which it is taught:	English Friendly/Spanish

1. INTRODUCTION

The subject software patterns aims teach students to use patterns in developing software applications as essential part of software engineering. The course focuses on the use of design patterns that represent a solution to problems that occur in a normal way in creating software.

Prerequisites and Recommendations

Is mandatory have studied subjects related to programming and software engineering.

1.b PRESENTACIÓN

La asignatura Patrones Software pretende enseñar a los alumnos la utilización de patrones software en el desarrollo de aplicaciones informáticas como parte fundamental de la ingeniería del software. La asignatura se centra en el uso de los patrones de diseño que representan una solución a problemas que se dan de una forma habitual en la creación de software.

2. COMPETENCIES

General skills:

CG4 Ability to define, evaluate and select hardware and software platforms for the development and implementation of systems, services and applications, according to the knowledge acquired as provided in paragraph 5 of resolution BOE-A-2009-12977.

CG5 Ability to conceive, develop and maintain systems, services and applications using the methods of software engineering as a tool for quality assurance, according to the knowledge acquired as provided in paragraph 5 of resolution BOE-A- 2009-12977.

CG8 Knowledge of basic materials and technologies that enable learning and development of new methods and technologies, as well as to equip them with great versatility to adapt to new situations.

CG9 Ability to solve problems with initiative, decision making, autonomy and creativity. Ability to communicate and transmit knowledge and skills of the profession of Technical Engineer.

CG12 Knowledge and application of basic elements of economics and human resource management, organization and project planning, as well as legislation, regulation and standardization in the field of IT projects, according to the knowledge acquired as provided in paragraph 5 resolution BOE-a-2009-12977.

Specific skills:

CIS1 Ability to develop, maintain and evaluate software services and systems that meet all user requirements and behave reliably and efficiently, are affordable to develop and maintain and meet quality standards, applying theories, principles, methods and practices of Software Engineering.

CIS2 Ability to assess the needs of the client and specify the software requirements to meet these needs, reconciling conflicting objectives through the search for acceptable commitments within the limitations derived from cost, time, the existence of already developed systems and their own organizations

CIS3 Ability to solve integration problems based on available strategies, standards and technologies.

CIS4 Ability to identify and analyze problems and design, develop, implement, verify and document software solutions based on an adequate knowledge of current theories, models and techniques.

CIS5 Ability to identify, evaluate and manage potential associated risks that may arise.

CIC4 Ability to design and implement software and communications system.

CC1 Ability to have a thorough understanding of the fundamental principles and models of computation and know how to apply to interpret, select, assess, model, and create new concepts, theories, uses and technological developments related to computer science.

CC4 Ability to know the basics, paradigms and own techniques of intelligent systems and analyze, design and build systems, services and applications that use these techniques in any scope.

CSI3 Ability to actively participate in the specification, design, implementation and maintenance of information systems and communication.

Learning

The expected learning outcomes, determined from the specific competences included in the verified memory of the degree as specific competences, are the following:

- RA1. Define the different types of software patterns and its main features. Introduce the concept of design pattern and features.
- RA2. Identify the fundamental patterns on which the design patterns are based.
- RA3. Describe the three families in which the design patterns are classified and identify the use of each of the patterns. Determine what is the most appropriate design pattern for a particular problem.
- RA4. Know how to interpret the graphic model that represents the structure of a design pattern and be able to adapt it to the domain of a specific problem.
- RA5. Analyze and design applications in which design patterns are applied.
- RA6. Develop applications in object-oriented programming languages that include design patterns.
- RA7. Knowing how to combine different design patterns to solve a complex software problem.
- RA8. Define what is a framework and what is its application in the construction of software applications. Show the relationship that exists between design patterns and frameworks.

3. CONTENTS

1. Introduction to software patterns: Origins, characteristics, classification, Patterns vs Frameworks.
2. Fundamental patterns: Delegation, Interface, Abstract Superclass, Interface and Abstract Class, Immutable, Marker Interface, Proxy.
3. Creating patterns: Introduction, Abstract Factory, Builder, Factory Method, Prototype, Singleton.
4. Structural Patterns: Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Proxy.
5. Behavior Patterns: Chain of Responsibility, Command, Interpreter, Iterator, Mediator, Memento, Observer, State, Strategy, Template Method, Visitor.
6. Other patterns: GRASP patterns, antipatterns.
7. Frameworks: Frameworks features, frameworks types, examples of frameworks.

Content blocks (you can specify the topics if deemed necessary)	Total class, credits or hours
Introduction to Software Patterns	4 hours
Fundamental Patterns	4 hours
Creational Patterns	10 hours
Structural Patterns	12 hours
Behavior Patterns	18 hours
Other Patterns	4 hours
Frameworks	4 hours
PECS	4 hours

4. TRAINING ACTIVITIES

The subject Software Patterns is organized as a four-month course of 6 ECTS (150 hours).

In the teaching-learning process of the aforementioned contents, the following training activities will be used:

- Classroom theoretical classes.
- Practical classes: solving classroom problems.
- Classroom laboratory practices.
- Tutorials: individual and / or group.

In addition, depending on the nature of the different parts of the subject matter, the following training activities may be used, among others:

- Preparation of works with individual responsibility but with information management as a team.
- Put in common the information, problems and doubts that appear in the realization of the works.
- Organization and realization of public days with oral presentations and discussion of results.
- Use of eLearning platform (Aula Virtual).

Classroom activities:

1. In the classroom: exhibition and discussion of the basic knowledge of the subject. Approach and theoretical resolution of exercises and related assumptions. Oriented to the teaching of the specific competences of the subject, especially those related to knowledge and techniques of using software patterns in the creation of computer applications.
2. In the laboratory: planning and development of practical exercises to solve problems and analyze hypotheses and contribute to the development of the ability to analyze results, critical reasoning and understanding of the proposed resolution methods. They will serve as the basis for the acquisition of the generic competences described in section 2.

Not in-person activities:

1. Analysis and assimilation of the contents of the subject, resolution of problems, bibliographical consultation, preparation of individual and group work, realization of face-to-face exams and self-evaluations. Specially oriented to the development of methods for self-organization and planning of individual and team work.
2. Tutorials: individual and group counseling during the teaching-learning process, either in person or remotely.

4.1. Credit distribution (specify in hours)

Classroom hours:	56 hours + 4 hours of assessment
Homework hours:	90 hours

Total hours.	150 hours
--------------	-----------

4.2 Methodological strategies, teaching materials and resources

Materials and Resources:

- Reference bibliography on the subject.
- Personal computers.
- Development environments and user manuals for them.
- Internet connection.
- Virtual Classroom platform and user manuals for them.
- Projectors.

5. Evaluation: Procedures, assessment criteria and rating

In the ordinary call the default evaluation method is the "continuous evaluation", with characteristics of formative evaluation to serve as feedback in the teaching-learning process by the student.

Optionally, and in a justified manner, the student may request the evaluation from the director of the center, by means of a single exam, which must be requested in writing and within the prescribed time limits. This method of evaluation consists of a theoretical examination and a laboratory practice.

In the extraordinary period, the evaluation will be based on a single exam, consisting of a theoretical exam and a laboratory practice, in which the degree of mastery of the competences of the subject will be determined.

Evaluation Criteria

The performance of the students will be evaluated according to the knowledge and skills acquired. The methods to be used will be: the resolution of practical cases and proposed works, public defense of certain works, realization of a final practice, as well as the performance of exams to check the theoretical knowledge of the subject.

Each of the continuous assessment exams will consist of two parts Theory and Laboratory. Distribution of the qualification in the Continuous Evaluation Exams (PEC's):

- The theoretical part will be valued at 50% and that of laboratory practices at another 50%, to obtain 100% of the grade of the subject.
- For the theoretical part, two PEI (Intermediate Assessment Test) will be carried out, each representing 50% of the total theory mark. (PEI1 - PEI2)
- For the part of laboratory practices will be made two PL (Laboratory Test), having a weight of 10% and 90% (PL1 - PL2) respectively.

In the Extraordinary Evaluation, the evaluation will be based on two exams, consisting of a theoretical exam and a laboratory practice (with an assessment of 50% each), in which the degree of mastery of the competences of the subject will be determined.

Students who take part in Final Evaluation must complete two exams, consisting of a theoretical exam and a laboratory practice (with an assessment of 50% each), in which the degree of mastery of the competences of the subject will be determined.

The following specific evaluation criteria are established for the subject:

- CE1. The student knows the different types of software patterns and their characteristics as well as the concept of design pattern.
- CE2. The student knows how to identify the fundamental patterns.

- CE3. The student knows the three families of design patterns and understands the applicability of each of the patterns.
- CE4. The student has acquired the necessary knowledge to adapt the general structure of a design pattern to a specific problem.
- CE5. The student has acquired the necessary knowledge to apply design patterns in the development of a computer application.
- CE6. The student knows how to apply the design patterns in an object-oriented language.
- CE7. The student understands what is a framework and its relation with the design patterns.

In the following table the weight in the qualification (between 0 and 100) of each exam is indicated, and its relation with the evaluation criteria, learning results and general competences.

Competences	Outcome Learning	Evaluation Criteria	Assessment Instrument	Weight in qualifying
CG4, CG5, CG8, CG9, CG12.	RA1, RA2, RA3, RA4, RA5, RA8	CE1, CE2, CE3, CE4, CE5	PEI1, PEI2	50
CIS1, CIS2, CIS3, CIS4, CIS5, CIC4, CC1, CC4, CSI3	RA4, RA5, RA6, RA7	CE5, CE6, CE7	PL1, PL2	50

For the final and extraordinary evaluation will have to perform two exams, one with the theoretical content (PT) and another with the laboratory (PL) having a weight of 50% each.

Competences	Outcome Learning	Evaluation Criteria	Assessment Instrument	Weight in qualifying
CG4, CG5, CG8, CG9, CG12.	RA1, RA2, RA3, RA4, RA5, RA8	CE1, CE2, CE3, CE4, CE5	PT	50
CIS1, CIS2, CIS3, CIS4, CIS5, CIC4, CC1, CC4, CSI3	RA4, RA5, RA6, RA7	CE5, EC6, EC7	PL	50

6. Bibliography

Basic Bibliography

Design Patterns

Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
Addison Wesley 2003

Design Patterns applied to Java

Stephen Steling, Olav Maassen
Pearson Education / Sun Microsystems, 2003

Head First Design Patterns

Eric Freeman, Elisabeth Freeman
O'Reilly, 2004

Software Architecture Design Patterns in Java

Partha Kuchana
Auerbach, 2004

Patterns in Java, volume 1 and 2

Grand Mark
Wiley Computer Publishing, 1998

Java Design Patterns: A Tutorial

James W. Cooper
Addison-Wesley, 2000

Design Patterns in Java

Laurent Debrauwer
Editions Eni, 2013

Complementary Bibliography

Object Oriented Programming with Java.

Salvador Otón; J.J. Martínez; J.R. Hilera.
Sº Publ. University of Alcalá, 2000.

Construction of Object Oriented Software. Second Edition.

Bertrand Meyer.
Prentice Hall, 1998.

Core J2EE™ Patterns: Best Practices and Design Strategies

Deepak Alur, John Crupi, Dan Malks

Prentice Hall / Sun Microsystems Press, 2001