



Universidad de Oviedo



# Arquitectura del Software Presentación



ARQUITECTURA  
DEL SOFTWARE

2023-24

Jose E. Labra Gayo  
Pablo González  
Cristian Augusto  
Jorge Álvarez Fidalgo

# Arquitectura del Software

## **Grado Ingeniería Informática del Software**

**Tipo:** Obligatoria, tercer año

**Créditos ECTS:** 6

**Periodo:** 2º Semestre

**Idioma:** Inglés/Español

### Recursos:

- Página web del curso: <https://arquisoft.github.io/>  
Trasparencias y contenido público
- Campus virtual: <https://www.campusvirtual.uniovi.es/>  
Comunicaciones internas (notas y similares)

# Profesores

Jorge Álvarez Fidalgo

[alvarezfjorge@uniovi.es](mailto:alvarezfjorge@uniovi.es)

Cristian Augusto

[augustocristian@uniovi.es](mailto:augustocristian@uniovi.es)

Pablo González González

[gonzalezgpablo@uniovi.es](mailto:gonzalezgpablo@uniovi.es)

Jose Emilio Labra Gayo

[labra@uniovi.es](mailto:labra@uniovi.es) (Coordinador)

# Dedicación temporal

6 créditos ECTS  $\approx$  150 horas de trabajo

60 horas presenciales, 90 horas no presenciales

Organización (*por semana*)

2h clases de teoría (21h total)

1h seminarios (7h total)

2h prácticas de laboratorio (28h total)

2h tutorías grupales (bajo demanda)

**7,5h no presenciales (90h total)**

# Competencias y Resultados de aprendizaje

# Competencias generales

## Habilidades metodológicas

<b>CG-1</b>	Competencia para el diseño de soluciones a problemas complejos humanos
-------------	--

# Competencias específicas

## Comunes a rama informática

<b>Com.1</b>	Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
<b>Com.8</b>	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
<b>Com.11</b>	Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.

# Competencias específicas

## Ingeniería del Software

<b>ISW.1</b>	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.
<b>ISW.3</b>	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles
<b>ISW.4</b>	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.



# Resultados de aprendizaje

<b>RA.IS-1.</b>	Realizar Proyectos de Ingeniería del Software complejos que den solución a problemas reales y solucionarlos mediante técnicas y tecnologías relacionadas con los procesos de fabricación de software incluyendo <i>frameworks</i> , patrones arquitectónicos, patrones de diseño y de integración persiguiendo el desarrollo de software de calidad
<b>RA.IS-3.</b>	Aplicar distintas técnicas de construcción de software en el diseño de bajo nivel
<b>RA.IS-4.</b>	Desarrollar diseños y programación orientados a objetos con un elevado nivel de competencia
<b>RA.IS-5.</b>	Evolucionar y refactorizar diseños existentes ante cambios en los requisitos
<b>RA.IS-6.</b>	Determinar el grado de mantenibilidad, fiabilidad y eficiencia de diseños software
<b>RA.IS-7</b>	Diseñar e implementar software aplicando diferentes tecnologías middleware
<b>RA.IS-9</b>	Diseñar y llevar a cabo controles e inspecciones eficientes y efectivas de validación, verificación y calidad, planes de pruebas
<b>RA.IS-10</b>	Analizar estadísticamente la densidad de defectos y probabilidad de fallos
<b>RA.IS-11</b>	Evaluar la calidad de un proceso de software desde el punto de vista de la calidad del producto

# Evaluación

# 3 posibilidades

Evaluación continua

Evaluación diferenciada

Evaluación extraordinaria

# Evaluación continua

$$Final = Teoría \times 40\% + Práctica \times 60\%$$

donde:

$$Teoría = Examen \times 70\% + Pruebas parciales \times 30\%$$

$$Práctica = Grupo \times 70\% + Individual \times 30\%$$

## Requisitos:

Asistencia mínima (80%)

Nota mínima (teoría & seminario): 5

Nota mínima (prácticas): 5

# Evaluación diferenciada

Teoría: Igual que anterior

Práctica (2 posibilidades)

1) Integrarse en grupos (asistencia mínima 20%)

**Obligatorio:** Participar en sesión de presentación pública

La nota será: 70% grupo + 30% individual.

2) Trabajo individual

Durante el primer mes el estudiante será asignado un Proyecto similar a los proyectos en grupo

Presentación pública individual

# Evaluación extraordinaria

$$Final = Teoría \times 40\% + Práctica \times 60\%$$

donde:

$$Teoría = Examen + Trabajo Individual$$

$$Práctica = Proyecto individual$$

Requisitos:

Nota mínima en ambos  $\geq 5$

Presentación pública de Proyecto individual

Normalmente después del examen final

# Actividades de enseñanza

Clases de teoría

Conferencias

Clases de seminario

Clases de laboratorio

# Clases de teoría

Este año utilizaremos metodología de *clases invertidas* mixta

Clases ya han sido grabadas y están disponibles

<https://arquisoft.github.io/course2223.html>

Espero/deseo que los estudiantes veáis los vídeos antes de cada clase

Durante la clase:

1ª parte: Repaso de conceptos principales

2ª parte: Kahoot!/Quizzizz: Las puntuaciones serán parte de la nota



# Conferencias

Se organizarán varias conferencias durante el curso

Arquitectos de software o expertos de empresas

Las conferencias de otros años están disponibles en:

<https://arquisoft.github.io/#Conferences>

Asistencia obligatoria

El material de las conferencias forma parte del contenido del curso

Se pueden incluir preguntas de las conferencias en exámenes

# Seminarios

## Trabajo en equipo

Equipos de 2 a 4 personas

Temas de los trabajos serán propuestos por los profesores

El trabajo consistirá en:

Pequeño informe sobre el tema (tipo blog post)

Presentación del tema a la clase

Presentación pública durante las clases de seminario

Al menos 2 preguntas del resto de equipos

## Nota

Informe + presentación + preguntas

# Clases de laboratorio

Aprendizaje basado en proyectos + trabajo en equipo

1. Diseñar y documentar una arquitectura de software
2. Implementar prototipo
3. Presentación pública



# Evaluación

70% nota grupo + 30% nota individual

Nota grupo: Presentación + prototipo + docs + github repo

**Presentación final es Obligatoria** (igual que un examen)

Profesores eligen la(s) persona(s) que presenta

Los otros miembros del grupo pueden participar

Nota individual: contribuciones en github

Herramienta gestión del proyecto: [github.com](https://github.com)

**Importante:** Cread cuenta en github  
Si es posible, utilizad un nombre que recuerde vuestro nombre/apellido...

# Sobre los equipos

Equipos creados inicialmente por profesores

Tamaño: 5-8 personas

Los equipos trabajarán juntos todo el curso

Trabajo en equipo = habilidad fundamental!

Los miembros que abandonen suspenden evaluación continua

En caso de problemas dentro de un equipo

Intentar comprender y resolver el problema

Si no es posible resolverlo, comunicarlo a los profesores tan pronto como sea posible

Acciones posibles:

Una persona problemática, es expulsada del equipo y suspende evaluación continua

Todos los miembros del equipo suspenden la evaluación continua

El equipo podría dividido por los profesores

# Sesiones de laboratorio

## 13 sesiones de laboratorio

### Durante sesiones de laboratorio (2 horas)

- 1.- Un professor explicará algún concepto (1h aprox.)
- 2.- El equipo trabajará en la práctica

Esa hora cuenta como una reunión del grupo

Se pueden hacer todas las reuniones extra que se quieran

# Reuniones de los grupos

Obligatorio: Registrar actas de todas las reuniones

Una persona debe escribir las actas

Consejo: Rotar la persona que toma actas

Las actas deberán mantenerse en la wiki del Proyecto

Estructura general:

Fecha/hora/lugar de la reunión

Participantes

Registrar

Tareas realizadas (issues en github cerrados)

Acciones a realizar (issues en github abiertos y asignado a alguien)

Acuerdos/decisiones tomadas (mantener registro de decisiones arquitectónicas)

# 4 Entregables

Chequeo en cada entregable

1er entregable - Semana 4

Documentación 0.1

2º entregable - Semana 7

Prototipo version 0.1

3er entregable - Semana 10

Prototipo version 1.0 + Documentación 1.0

4º entregable - Semana 13

Prototipo versión 1.1 + Documentación 1.1

Presentación pública

Antes de cada entregable

- Cada estudiante creará un autoinforme
- Breve resumen de actividades realizadas
- Enlazado a issues en github
- Los compañeros deberán indicar si están o no de acuerdo



# Presentación pública

Última semana

Es igual que un examen

Participación obligatoria

Cada grupo presentará su Proyecto

Los profesores eligen la(s) persona(s) que presenta(n)



# Material para seguir el curso

Página web: Información pública y transparencias

<https://arquisoft.github.io/>

Campus virtual (información interna)

Foro

Notas

Otro material

Manuales, Tutoriales, Videos, etc...

# Práctica de este año

WIQ: **W**ikidata based **Q**uizz Show  
Inspirado por “Saber y Ganar”  
Juego de preguntas y respuestas

Enunciado:

<https://arquisoft.github.io/course2324.html#labs>



[This picture](#) unknown actor under [CC BY-NC-ND](#) licence

# Acerca de Wikidata

- <https://www.wikidata.org/>
- Base de conocimiento de propósito general
- Da soporte a Wikipedia
- Acceso libre desarrollado con código abierto
- Parte de Wikimedia Foundation
- Acceso a datos:

[https://www.wikidata.org/wiki/Wikidata:Data\\_access](https://www.wikidata.org/wiki/Wikidata:Data_access)



# Si tenéis preguntas...

## Sobre el curso...

Fechas de entrega, exámenes, tareas, etc.

Issues en <https://github.com/arquisoft/faq> o foro del Campus Virtual

El mensaje llegará al resto de estudiantes

Todos pueden ver la pregunta y la respuesta

Todos pueden incluso responder

## Sobre temas técnicos...

Utilizar sitios públicos, por ejemplo:

Stackoverflow (general): <https://stackoverflow.com/>

Solid forum (sobre solid): <https://forum.solidproject.org/>

Foro de la asignatura

## Sobre problemas personales o similares

Enviad un correo al profesor

# Fechas importantes (recordatorio)

## Entregas

1er entregable (clase 4ª de lab., 19 - 235 Feb.)

2º entregable (clase 7ª de lab. 11 - 15 Marzo)

3er entregable (clase 10ª de lab. 8 - 12 Abril)

4º entregable **OBLIGATORIO** (case 13ª, 29-Abril - 3 Mayo)

## Exámenes de teoría:

Ordinario: Mayo/Junio ?

Extraordinario: Julio ?

# ¡Primer Cuestionario!

Importante: Conectarse con el usuario UOxxx

Si se desconecta, conectarse de nuevo con UOxxx\_0, UOxxx\_1,...

La asistencia a clase se toma de los informes

**Fin de la presentación**



# Planificación

Temas	Horas totales	Actividades presenciales						No presenciales		
		Clase Teoría	Seminarios	Prácticas de laboratorio	Tutorías grupales	Sesiones de Evaluación	Total	Trabajo grupo	Trabajo autónomo	Total
1 (Conceptos)	40	7	2	4	1	1	15	15	10	25
2 (Taxonomías)	110	14	5	24	1	1	45	45	20	65
Total	150	21	7	28	2	2	60	60	30	90