



Universidad de Oviedo



Arquitectura del Software

Presentación



ARQUITECTURA
DEL SOFTWARE

Curso 2021/2022

Jose Emilio Labra Gayo

Arquitectura del Software

Grado Ingeniería Informática del Software

Tipo: Obligatoria, tercer año

Créditos ECTS: 6

Periodo: 2º Semestre

Idioma: Inglés/Español

Profesores

Hugo Lebrede Buján

lebredehugo@uniovi.es

Irene Cid Rico

cidirene@uniovi.es

Pablo González González

gonzalezgpablo@uniovi.es

Jose Emilio Labra Gayo

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

CG-1 Competencia para el diseño de soluciones a problemas complejos humanos

Competencias específicas

Comunes a rama informática

Com.1	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.
Com.8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
Com.11	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

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

RA.IS-1.	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 frameworks, patrones arquitectónicos, patrones de diseño y de integración persiguiendo el desarrollo de software de calidad [Com1], [Com2], [Com.8], [Com10], [Com13], [ISw.1], [ISw.4], [ISw.6], [Bas5] , [CG1], [CG3],[CG4],[CG5],[CG10], [CG22]
RA.IS-3.	Aplicar distintas técnicas de construcción de software en el diseño de bajo nivel [Com.8], [ISw.1], [ISw.4], [CG1]
RA.IS-4.	Desarrollar diseños y programación orientados a objetos con un elevado nivel de competencia [Com.8], [ISw.4], [CG1], [CG4]
RA.IS-5.	Evolucionar y refactorizar diseños existentes ante cambios en los requisitos [ISw.1], [ISw.6]
RA.IS-6.	Determinar el grado de mantenibilidad, fiabilidad y eficiencia de diseños software [Com.8], [ISw.1], [ISw.4], [CG1]
RA.IS-7	Diseñar e implementar software aplicando diferentes tecnologías middleware [Com11], [ISw3]
RA.IS-9	Diseñar y llevar a cabo controles e inspecciones eficientes y efectivas de validación, verificación y calidad, planes de pruebas [Com1], [Com16], [ISw1], [ISw4], [CG2], [CG4], [CG11], [CG28].
RA.IS-10	Analizar estadísticamente la densidad de defectos y probabilidad de fallos [Com16], [ISw4], [ISw5], [CG25]
RA.IS-11	Evaluar la calidad de un proceso de software desde el punto de vista de la calidad del producto [Com1], [Com18], [ISw1], [CG2], [CG11], [CG28].

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 ≥ 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

Todas las clases ya han sido grabadas y están disponibles

<https://arquisoft.github.io/course2122.html>

Se espera que los estudiantes veáis los vídeos antes de cada clase

Durante la clase:

Explicación de algunos conceptos con más detalle

Preguntas y respuestas

Se usará Kahoot! Para realimentación

Las puntuaciones de Kahoot! 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

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 quiean

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 asignados 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

DeDe

<https://arquisoft.github.io/course2122/labEnunciadoPractica.html>

Decentralized Delivery System

Basado en principios SOLID

Participación opcional en concurso SOLID

Este año: React + Typescript



Acerca de SOLID

SOLID (SOcial LIinked Data)

Objetivo: Social Web descentralizada

Separar datos personales de aplicaciones

Iniciado en el MIT

Utilizar varias especificaciones W3C

WebId

Web Access Control

Linked Data Platform

...

Debéis aprender/leer vosotros mismos

Mucho material disponible...



Si tenéis preguntas...

Sobre el curso...

Fechas de entrega, exámenes, tareas, etc.

Por favor: 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

Enviadme un correo

Fechas importantes (recordatorio)

Entregas

1er entregable (clase 4ª de lab., 21 - 25 Feb.)

2º entregable (clase 7ª de lab. 14 - 18 Marzo)

3er entregable (clase 10ª de lab. 4 - 8 Abril)

4º entregable **OBLIGATORIO** (clase 13ª, 3 - 9 Mayo)

Exámenes de teoría:

Ordinario: Mayo/Junio ?

Extraordinario: Julio ?

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