



Universidad de Oviedo



Arquitectura del Software Presentación



ARQUITECTURA
DEL SOFTWARE

Curso 2018/2019

Jose Emilio Labra Gayo

Arquitectura del Software

Grado en Ingeniería Informática del Software

Tipo: Obligatoria, tercer año

Créditos ECTS: 6

Periodo: 2º Semestre

Idioma: Inglés/Español

Profesores

Jose Emilio Labra Gayo labra@uniovi.es

Noelia Rico Pachón (noeliarico@uniovi.es)

Victor Manuel Álvarez García (victoralvarez@uniovi.es)

Pablo Suárez Otero-González (suarezgpablo@uniovi.es)

Javier Escalada Gómez (escaladajavier@uniovi.es)

Oscar Jesús Cosido Cobos (oscar.cosido@uniovi.es)

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)

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 (por defecto)

Evaluación final

Evaluación diferenciada

Evaluación continua (por defecto)

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

donde:

$$Teoría = Examen \times 70\% + Seminarios \times 30\%$$

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

Requisitos:

Asistencia mínima (80% teoría, seminarios, laboratorios)

Nota mínima (teoría & seminario): 3

Nota mínima (prácticas): 5

Evaluación final solo (no continua)

SI falla evaluación continua*

Nota final

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

donde

Teoría = Examen + Trabajo Individual (seminarios)

Práctica = Proyecto Individual

Presentación en público de proyecto individual

Normalmente el día del examen teórico (al finalizar)

* Se aplicarán penalizaciones en nota prácticas si Estudiantes no son capaces de trabajar en equipo

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

Seminarios

Seminarios

Trabajos realizados en grupos de 2/3 personas

Temáticas propuestas por profesores

Analizar un Sistema de código abierto

Describir atributos de calidad, requisitos, etc.

Presentaciones públicas durante clases de seminarios

Evaluación:

Informe entregado + Presentación + Preguntas

Sobre trabajo práctico

Evaluación

70% nota grupo + 30% nota individual

Nota grupo: Día de presentación

Presentación final = Obligatoria (igual que un examen)

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

Los otros miembros del grupo pueden participar

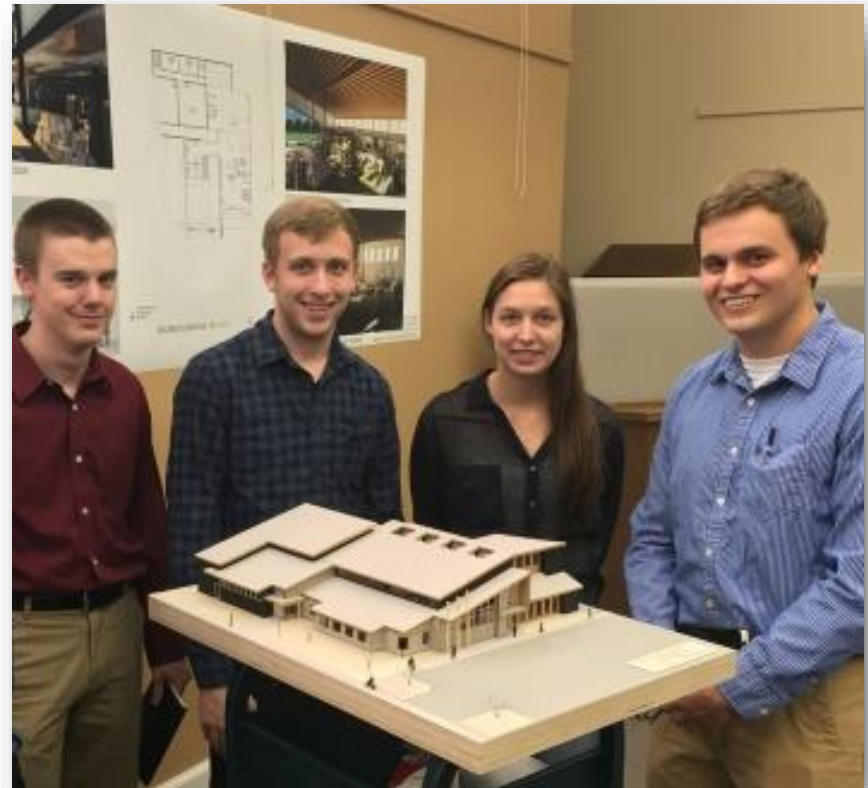
Nota individual: contribuciones en github

Herramienta gestión del proyecto: github.com

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

Proyecto en equipo

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



Equipos

Equipos creados inicialmente por profesores

Tamaño: 5-8 personas

Los equipos trabajarán juntos todo el curso

Habilidad de trabajo en equipo = fundamental!

Los miembros que abandonen presentarán individualmente

Penalización possible al individuo y al grupo

Sesiones de laboratorio

13 sesiones de laboratorio

Excepto grupos de los lunes, solo 12

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 reunion del grupo

Reuniones de los grupos

Cada sesión de laboratorio

Los grupos pueden organizar sus propias reuniones

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

Decisiones tomadas

Acciones a realizar

4 Entregables

Chequeo en cada entregable

1er entregable - Semana 4 (19 Feb - 25 Feb.)

Documentación 0.1

2º entregable - Semana 7 (12 Marzo - 18 Marzo)

Prototipo version 0.1

3er entregable - Semana 10 (2 Abril - 8 Abril)

Prototipo version 1.0 + Documentación 1.0

4º entregable - Semana 13 (29 Abril - 3 Mayo)

Prototipo versión 1.1 + Documentación 1.1

Presentación pública

Presentación pública

Última semana (29 Abril - 3 Mayo)

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)

Guía de aprendizaje

Foro

Bibliografía

Notas

Otro material

Manuales, Tutoriales, Videos, etc...

Práctica de este año



DeChat - Decentralized Chat

<https://labra.solid.community/public/SoftwareArchitecture/AssignmentDescription/index.html>

Concurso Inrupt (<https://www.inrupt.com/>)

Inrupt ha mostrado interés en la práctica

Certificado de reconocimiento al mejor proyecto

Participación voluntaria

Evento de entrega: 12th Junio

Debe basarse en plataforma SOLID

Acerca de SOLID

SOLID (SOcial LInked 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 favour: 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/>

Sobre problemas personales o similares

Enviadme un correo

Fechas importantes (recordatorio)

Entregas

1er entregable (19 Feb - 25 Feb.)

2º entregable (12 Marzo - 18 Marzo)

3er entregable (2 Abril - 8 Abril)

4º entregable **OBLIGATORIO** (29 Abril - 3 Mayo)

Exámenes de teoría:

Ordinario: 30/05/2019, 12:00 – 13:00

Extraordinario: 01/07/2019, 12:00 – 13:00