



## Proyecto/Guía docente de la asignatura

<b>Asignatura</b>	DESARROLLO BASADO EN COMPONENTES Y SERVICIOS		
<b>Materia</b>	INGENIERÍA DE SOFTWARE		
<b>Módulo</b>	Tecnologías Específicas		
<b>Titulación</b>	Grado en INGENIERÍA INFORMÁTICA		
<b>Plan</b>	545	<b>Código</b>	46954
<b>Periodo de impartición</b>	Primer cuatrimestre	<b>Tipo/Carácter</b>	Obligatoria (Mención IS)
<b>Nivel/Ciclo</b>	Grado	<b>Curso</b>	4º
<b>Créditos ECTS</b>	6		
<b>Lengua en que se imparte</b>	Español		
<b>Profesor/es responsable/s</b>	Yania Crespo González-Carvajal (Coordinadora de la parte teórica) Carlos E. Vivaracho Pascual (Coordinador de la parte práctica) Irene Lavín Perrino (Profesora de prácticas)		
<b>Datos de contacto (E-mail, teléfono...)</b>	yania[at]infor.uva.es, cevp[at]infor.uva.es, irene[at]infor.uva.es		
<b>Departamento</b>	Informática		

### 1. Situación / Sentido de la Asignatura

#### 1.1 Contextualización

Esta asignatura se encuentra situada en el cuarto curso de Ingeniería Informática, junto a otras que conforman la materia "Ingeniería de Software" (ver apartado 1.2), y es obligatoria en la mención de ingeniería de software de la titulación. En la actualidad los productos y sistemas software están formados por un gran número de componentes heterogéneos que interactúan en entornos distribuidos basados en la Web. La programación de este tipo de sistemas requiere conocimientos de herramientas de programación específicas. Por otro lado, la formación básica en diseño de arquitecturas y aplicaciones debe ser complementada con una introducción al modelado de procesos de negocio.

La asignatura aborda de forma integrada el desarrollo basado en componentes y en servicios Web, tanto desde el punto de vista del servidor como de los clientes que utilicen esos componentes o servicios. Por un lado, presentará los principios y características de la Ingeniería del Software basada en componentes, junto con las técnicas relacionadas desde los modelos de requisitos y arquitectura a las herramientas y plataformas necesarias para su implementación. Por otro, junto con un repaso del desarrollo básico de aplicaciones basadas en la Web, se centrará en los aspectos específicos de los servicios Web, su desarrollo y técnicas de coordinación para su integración en sistemas empresariales.



## 1.2 Relación con otras materias

La asignatura está planteada como una de las disciplinas que componen la materia “Ingeniería de Software”. Está situada en el primer semestre de cuarto curso, por tanto, después de las asignaturas de MODELADO DE SISTEMAS SOFTWARE, ANALISIS Y DISEÑO DE BASES DE DATOS, DISEÑO DE SOFTWARE (obligatorias) y SERVICIOS Y SISTEMAS WEB (optativa), de modo que utilizará y complementará los conocimientos adquiridos en estas asignaturas. Por otro lado, la asignatura de PROGRAMACIÓN ORIENTADA A OBJETOS de segundo curso proporcionó al alumno las habilidades de programación necesarias.

## 1.3 Prerrequisitos

Aunque no se han establecido prerrequisitos, es recomendable que el alumno haya cursado TODAS las asignaturas mencionadas en el punto anterior. Se requiere una buena disposición para el trabajo en equipo. Es recomendable disponer de un nivel de inglés que permita al estudiante leer bibliografía de consulta.

## 2. Competencias

### 2.1 Generales

Código	Descripción
G02	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática
G03	Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan.
G05	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad
G06	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes
G08	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.
G09	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.



G10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática
-----	---

## 2.2 Específicas

Código	Descripción
IS3	Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.
CI11	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.

## 3. Objetivos

Código	Descripción
IS3.1	Conocer y aplicar los principios de la construcción del software basada en componentes.
IS3.2	Usar entornos de middleware en el desarrollo de aplicaciones o servicios Web.
IS3.3	Aplicar aproximaciones de componentes a sistemas que incluyen, por ejemplo, concurrencia, comunicaciones y acceso a bases de datos.
CI11.1	Saber aplicar técnicas de construcción de software específicas para sistemas heterogéneos hardware-software, middleware y sistemas distribuidos
CI11.2	Conocer y aplicar herramientas que permitan realizar todas estas tareas de forma productiva

## 4. Contenidos y/o bloques temáticos

### Bloque 1: Componentes, servicios y microservicios: Conceptos y técnicas de modelado y diseño

Carga de trabajo en créditos ECTS: 

3
---

#### a. Contextualización y justificación

El bloque 1 se desarrolla en el aula y presenta los conocimientos teóricos de la asignatura: los conceptos generales, las técnicas de modelado y diseño de componentes y servicios Web, y los aspectos específicos del proceso de desarrollo de este tipo de sistemas. La primera parte aborda los aspectos específicos del desarrollo de los **servicios** Web y el diseño de sistemas que los integran y su coordinación y composición. Posteriormente se abordan las particularidades de las arquitecturas basadas en **microservicios**. De forma intercalada y al final del curso, se introduce al alumno en el proceso y las técnicas relacionadas con el desarrollo de software basado en **componentes**, incluyendo la identificación y especificación de componentes y servicios, así como el modelado de la **arquitectura** y el **despliegue** de este tipo de sistemas.



## b. Objetivos de aprendizaje

Código	Descripción
IS3.1	Conocer y aplicar los principios de la construcción del software basada en componentes.
CI8.1	Saber aplicar técnicas de construcción de software específicas para sistemas heterogéneos, middleware, distribuidos y basados en servicios.
CI14.1	Aplicar aproximaciones de componentes a sistemas que incluyen, por ejemplo, concurrencia, comunicaciones y acceso a bases de datos.

## c. Contenidos

- Conceptos básicos del Desarrollo Basado en Componentes y Servicios
  - Reutilización de software
  - Fundamentos y definiciones
  - Interfaces y Contratos: IDL y UML
- Ingeniería de Software basada en Servicios Web
  - Introducción a los servicios web: Tecnología
  - SOAP, WSDL y UDDI
  - Ingeniería de Servicios Web
  - Ingeniería de Software orientada a Servicios: SOA
- Ingeniería de Software basada en Servicios REST
  - Servicios RESTful
  - REST: Detalles de implementación
  - Estilos arquitectónicos y estilos de interacción
  - Ingeniería de Servicios REST: API REST
  - Ingeniería de Software orientada a Recursos: ROA
- Micro-Arquitecturas y computación en la nube
  - Microservicios
  - Domain Driven Design
  - Cloud Computing
  - SaaS, "Software as a service"
- Modelos de Componentes
  - Modelos básicos de componentes: COM, .NET, JavaBeans, Android
  - El problema de la heterogeneidad: El papel del middleware
- Tecnologías y marcos de trabajo para desarrollo de sistemas de componentes distribuidos.
  - Tecnología Java: EJB y JEE
  - Tipos de EJB
  - Persistencia: JPA y Entities
  - Despliegue de Aplicaciones JEE
- Ingeniería de Software basada en Componentes
  - CBSE: Desarrollo para y con reutilización
  - Diseño de componentes con JEE
  - Patrones de diseño
  - Diseño basado en componentes con UML
  - Despliegue de componentes

## d. Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none"><li>• Clase magistral participativa complementada con material disponible en el aula virtual</li><li>• Estudio de casos en aula</li><li>• Resolución de problemas (identificación de requisitos, construcción de modelos, etc., en pequeños grupos y discusión)</li></ul>



---

### e. Plan de trabajo

---

Ver apartado 7. La fecha concreta de cada actividad se publicará en la web de la asignatura

---

### f. Evaluación

---

Ver apartado 7

---

### g Material docente

---

---

#### g.1 Bibliografía básica

---

##### Conceptos generales y plataformas de componentes/servicios:

- Sommerville, I. "Ingeniería del software" Pearson, 2011 (9ª ed.)
- Andy Ju An Wang, Kai Qian, Component-Oriented Programming, Wiley, 2005, ISBN: 978-0-471-64446-0
- D. Panda, R. Rahman, D. Lane. EJB3 in Action. 2aEd Manning. 2014. ISBN: 978-1-935182-99-3.
- Leonard Richardson, Sam Ruby. RESTful Web Services, Web services for the real world. O'Reilly, 2007.

##### Diseño:

- John Cheesman & John Daniels. UML components: a simple process for specifying component-based software. Boston, MA, USA, Addison-Wesley Longman Publishing Co., Inc., 2000
- Robert Daigneau. "Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services", Addison 2012

---

#### g.2 Bibliografía complementaria

---

- Clemens Szyperski "Component Software - Beyond Object-Oriented Programming" – Second Edition, Addison-Wesley / ACM Press, 2002 .ISBN 0-201-74572-0
- Martin Fowler. Patterns of Enterprise Application Architecture. Addison Wesley, 2003.
- Arlow, Jim, Neustadt, Ila. "UML 2", Anaya Multimedia, 2006

---

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

---

- AWS: <https://aws.amazon.com/es/education/awseducate/>

---

### h. Recursos necesarios

---

- Aula virtual de la asignatura.

---

### i. Temporalización

---

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
3	Semanas 1..15

**Bloque 2: Implementación y Despliegue de Servicios y Microservicios**Carga de trabajo en créditos ECTS: **a. Contextualización y justificación**

El bloque se desarrolla en el laboratorio y está dedicado a aplicar los conocimientos teóricos de la asignatura. Dada su amplia aplicación práctica actual nos vamos a centrar en la parte de Servicios y Microservicios. Dentro de las limitaciones que se tienen al empezar desde cero con muchas de las tecnologías que veremos, se intentará que sean lo más actuales posibles y de amplio uso real. Aun centrándonos en la parte de Servicios, no nos olvidaremos de la parte de Componentes, ya que muchas de las tecnologías y herramientas que usaremos están basadas en ellos.

**b. Objetivos de aprendizaje**

Código	Descripción
IS3.2	Usar entornos de middleware en el desarrollo de aplicaciones o servicios Web.
CI11.1	Conocer las técnicas de construcción de software específicas para sistemas basados en Internet, incluyendo frameworks y servicios Web.
CI13.1	Aplicación de las técnicas necesarias para el procesamiento y acceso a los Sistemas basados en servicios web.
CI13.2	Conocer y aplicar herramientas que permitan realizar todas estas tareas de forma productiva.

**c. Contenidos**

- Parte I: Servicios Web tipo REST
  - Servidor (API REST) mediante Spring Boot
  - Acceso a datos mediante JPA (Java Persistence API)
  - Cliente Angular
- Parte II: Contenedores y Despliegue en Servidor Real
  - Empaquetamiento del servicio en contenedores usando tecnología Docker
  - Despliegue en un servidor real
- Parte III: Microservicios y Gateway
  - Creación de microservicios
  - Acceso mediante Gateway

**d. Métodos docentes**

Actividad	Metodología
<b>Seminarios</b> <b>Tutorías</b>	<ul style="list-style-type: none"><li>• Talleres de aprendizaje del manejo de herramientas en sesiones específicas.</li><li>• Seguimiento de las prácticas desarrolladas en grupo.</li></ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"><li>• Aprendizaje autónomo del alumno/a. Se favorecerá la metodología basada en el modelo de aprendizaje inverso, de manera que la hora de laboratorio sea más un seguimiento del trabajo de alumno y un seminario de dudas y guía en el aprendizaje, que un momento de clase magistral.</li></ul>



	<ul style="list-style-type: none"><li>• Aprendizaje basado en problemas. Los distintos conceptos involucrados, así como las herramientas, se aprenderán mediante la realización de ejemplos y problemas.</li><li>• Aprendizaje basado en proyectos. El alumno/a tendrá que realizar varios proyectos a lo largo de la asignatura</li><li>• Aprendizaje colaborativo. El trabajo se realizará en grupos de alumnos/as.</li></ul>
--	---

### e. Plan de trabajo

---

La temporalización y plan de trabajo de cada semana tanto dentro del laboratorio como fuera, así como las actividades de evaluación con sus fechas límite de entrega, se publicarán al principio del curso en la web de la asignatura

### f. Evaluación

---

Ver apartado 7

### g Material docente

---

#### g.1 Bibliografía básica

---

- Deinum, Marten., Daniel. Rubio, and Josh. Long. Spring 5 Recipes A Problem-Solution Approach. 4th ed. 2017. Berkeley, CA: Apress, 2017.
- Bucchiarone, Antonio. et al. Microservices Science and Engineering. 1st ed. 2020. Cham: Springer International Publishing, 2020.
- Jangla, Kinnary. Accelerating Development Velocity Using Docker Docker Across Microservices. 1st ed. 2018. Berkeley, CA: Apress, 2018.
- Freeman, Adam. Pro Angular 6. 3rd ed. 2018. Berkeley, CA: Apress, 2018

#### g.2. Bibliografía complementaria

---

- Redmond, Paul. Lumen Programming Guide Writing PHP Microservices, REST and Web Service APIs. 1st ed. 2016. Berkeley, CA: Apress, 2016.
- Leonard Richardson, Sam Ruby. RESTful Web Services, Web services for the real world. O'Reilly, 2007.
- Carneiro Jr., Cloves., and Tim. Schmelmer. Microservices From Day One Build Robust and Scalable Software from the Start. 1st ed. 2016. Berkeley, CA: Apress, 2016.
- Macero, Moises. Learn Microservices with Spring Boot A Practical Approach to RESTful Services Using RabbitMQ, Eureka, Ribbon, Zuul and Cucumber. 1st ed. 2017. Berkeley, CA: Apress, 2017.
- Boada Oriols, Miquel, and Juan Antonio Gómez Gutiérrez. El gran libro de Angular: [100 ejercicios prácticos] . Barcelona: Marcombo, 2018
- Prasad Reddy, K. Siva. Beginning Spring Boot 2: Applications and Microservices with the Spring Framework. Berkeley, CA: Apress L. P, 2017.
- Trabajos Fin de Grado sobre servicios REST y Microservicios realizados en la Escuela de Ingeniería Informática de la UVa. Ejemplos:
  - Aguilera Gómez, Jorge. Arquitectura basada en Microservicios. Implementación de un sistema cloud políglota para una franquicia de tiendas de ropa <https://uvadoc.uva.es/handle/10324/47126>
  - Maza Valles, Andrés de la. API REST para la gestión de partidas multijugador de un juego serio. N.p., 2020. <http://uvadoc.uva.es/handle/10324/44417>
  - Gómez Pedriza, David. Una arquitectura de microservicios Python como soporte a un diccionario de datos en un sistema de aseguramiento de la calidad y asistencia al desarrollo de IAMs por grandes equipos globalmente distribuidos. <https://uvadoc.uva.es/handle/10324/44278>



- Gómez Escribano, Cristina. Desarrollo de aplicación para la generación de juegos geográficos a medida. <https://uvadoc.uva.es/handle/10324/38747>

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

- Kong: <https://docs.konghq.com/>. Última consulta: 15-07-2021
- Kong: <https://microservices.io/patterns/apigateway.html>. Última consulta: 15-07-2021
- Angular: <https://angular.io/docs>
- Java frameworks.
  - Spring Boot: <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>. Última consulta: 15-07-2021
- Node frameworks Express: <https://expressjs.com/es/>
- Python frameworks.
  - Flask: <https://flask.palletsprojects.com/>
  - PyMS: <https://py-ms.readthedocs.io/>
- Go frameworks
  - Echo: <https://github.com/labstack/echo>
  - go-kit: <https://github.com/go-kit/kit>

#### i. Recursos necesarios

- Aula virtual de la asignatura.
- Software (en la web de la asignatura se actualizarán los enlaces a las webs de descarga correspondientes):
  - IDE. Se recomienda Visual Studio Code o IntelliJ
  - MySQL
  - Angular
  - Kong
  - Spring Boot Framework
  - Docker
  - Flask

#### j. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
3	Semanas 1..15

## 5. Métodos docentes y principios metodológicos

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none"> <li>• Clase magistral participativa complementada con material disponible en el aula virtual</li> <li>• Estudio de casos en aula</li> </ul>



	<ul style="list-style-type: none"> <li>Resolución de problemas (identificación de requisitos, construcción de modelos, etc. en pequeños grupos y discusión)</li> </ul>
<b>Seminarios Tutorías</b>	<ul style="list-style-type: none"> <li>Talleres de aprendizaje del manejo de herramientas en sesiones específicas.</li> <li>Seguimiento de las prácticas desarrolladas en grupo.</li> </ul>
<b>Clase práctica</b>	<ul style="list-style-type: none"> <li>Aprendizaje autónomo del alumno/a. Se favorecerá la metodología basada en el modelo de aprendizaje inverso, de manera que la hora de laboratorio sea más un seguimiento del trabajo de alumno y un seminario de dudas y guía en el aprendizaje, que un momento de clase magistral.</li> <li>Aprendizaje basado en problemas. Los distintos conceptos involucrados, así como las herramientas, se aprenderán mediante la realización de ejemplos y problemas.</li> <li>Aprendizaje basado en proyectos. El alumno/a tendrá que realizar varios proyectos a lo largo de la asignatura</li> <li>Aprendizaje colaborativo. El trabajo se realizará en grupos de alumnos/as.</li> </ul>

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

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA <sup>(1)</sup>	HORA S	ACTIVIDADES NO PRESENCIALES	HORA S
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	60
Laboratorios (L)	30	Estudio y trabajo autónomo grupal	30
Total presencial	<b>56</b>	Total no presencial	<b>90</b>
TOTAL presencial + no presencial			<b>150</b>

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

## 7. Sistema y características de la evaluación

Criterio: cuando al menos el 50% de los días lectivos del cuatrimestre transcurran en normalidad, se asumirán como criterios de evaluación los indicados en la guía docente. Se recomienda la evaluación continua ya que implica minimizar los cambios en la agenda.

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Dos cuestionarios realizados a través del aula virtual	10%	5% cada cuestionario
Examen escrito sobre un supuesto práctico, del tipo de los realizados en aula	20 %	Aproximadamente Semana 9
Examen escrito sobre un supuesto práctico, del tipo de los realizados en aula	20 %	Periodo de exámenes
Entrega práctica 1L (Servicios REST y cliente Angular)	15%	Aproximadamente Semana 8
Entrega práctica 2L (Dockerizar y desplegar servicios)	10 %	Aproximadamente Semana 10
Entrega práctica 3L (microservicios y cliente)	15%	Aproximadamente Semana 15
Realización de los ejercicios propuestos en la parte de laboratorio	10%	Semanas 1 a 15



Criterios de calificación
<ul style="list-style-type: none"><li>• <b>Calificación final:</b> Suma ponderada de los cuestionarios (10%), prácticas en parejas (50%, con un mínimo de 4/10 para aprobar) y exámenes escritos (40%, con una nota media mínima de 4/10 para aprobar), debiendo obtener una suma igual o mayor a 5.<ul style="list-style-type: none"><li>○ Si nota(exámenes) <math>\geq 4</math> y nota(prácticas) <math>\geq 4</math>, Nota final = Suma ponderada</li><li>○ Si nota(exámenes) <math>&lt; 4</math> o nota(prácticas) <math>&lt; 4</math>, Nota final = mínimo (Suma ponderada; 4,5)</li></ul></li><li>• <b>Calificación de la convocatoria extraordinaria:</b> se utilizará la misma fórmula de cálculo de la nota final, pudiendo recuperar cada una de las partes por separado a excepción de los cuestionarios y los ejercicios planteados en el laboratorio durante el curso<ul style="list-style-type: none"><li>○ Opcionalmente, el alumno podrá presentarse a uno o dos exámenes escritos con el mismo formato que en la convocatoria ordinaria</li><li>○ Opcionalmente, el alumno podrá volver a presentar cada una de las entregas prácticas que sustituyen a las entregadas durante el curso</li></ul></li><li>• <b>Calificación de la convocatoria extraordinaria fin de carrera (noviembre):</b> se utilizará la misma fórmula de cálculo de la nota final, teniendo que realizar obligatoriamente cada una de las partes, incluido un cuestionario global</li></ul>

## 8. Consideraciones finales

----