



1. Identificación

1.1. De la Asignatura

Curso Académico	2023/2024
Titulación	MÁSTER UNIVERSITARIO EN INGENIERÍA DEL SOFTWARE / MASTER IN SOFTWARE ENGINEERING
Nombre de la Asignatura	DESARROLLO FULL-STACK
Código	7802
Curso	PRIMERO
Carácter	OBLIGATORIA
N.º Grupos	1
Créditos ECTS	12
Estimación del volumen de trabajo del alumno	300
Organización Temporal/Temporalidad	Anual
Idiomas en que se imparte	ESPAÑOL

1.2. Del profesorado: Equipo Docente

Coordinación de la asignatura MARCOS MENARGUEZ TORTOSA	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS
	Categoría	PROFESORES TITULARES DE UNIVERSIDAD
	Correo Electrónico / Página web / Tutoría electrónica	marcos@um.es Tutoría Electrónica: Sí

Grupo de	Teléfono, Horario y	Duración	Día	Horario	Lugar	Observaciones
Docencia: 1 Coordinación de los grupos:1	Lugar de atención al alumnado	Segundo Cuatrimestre	Miércoles	10:00- 13:00	868884234, Facultad de Informática B1.2.043	Tutorías presenciales (despacho 2.33) o por videoconferencia. Contactar previamente por Aula Virtual o correo electrónico.
JOSE ANTONIO MIÑARRO GIMENEZ Grupo de Docencia: 1	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS				
	Categoría	PROFESOR CONTRATADO DOCTOR TIPO A (DEI)				
	Correo Electrónico / Página web / Tutoría electrónica	jose.minyarro@um.es Tutoría Electrónica: Sí				



	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar	Observaciones
		Primer Cuatrimestre	Martes	11:00- 13:30	868888556, Facultad de Informática B1.2.040	Despacho 2.32
		Primer Cuatrimestre	Miércoles	16:00- 17:00	868888556, Facultad de Informática B1.2.040	Despacho 2.32
		Segundo Cuatrimestre	Lunes	09:30- 12:00	868888556, Facultad de Informática B1.2.040	Despacho 2.32
		Segundo Cuatrimestre	Jueves	09:30- 10:30	868888556, Facultad de Informática B1.2.040	Despacho 2.32
FRANCISCO JAVIER BERMUDEZ RUIZ	Área/Departamento	LENGUAJES Y SISTEMAS INFORMÁTICOS/INFORMÁTICA Y SISTEMAS				
Grupo de Docencia: 1	Categoría	PROFESOR COLABORADOR (LICENCIADO)				
	Correo Electrónico / Página web / Tutoría electrónica	fjavier@um.es Tutoría Electrónica: Sí				

	Teléfono, Horario y Lugar de atención al alumnado	Duración	Día	Horario	Lugar	Observaciones
		Primer Cuatrimestre	Martes	12:00- 13:30	868888524, Facultad de Informática B1.2.038	PLAN A: Despacho 2.30 INFORMÁTICA PLAN B y C:Mediante herramientas online como Zoom y mensajes de AV
		Primer Cuatrimestre	Miércoles	12:00- 13:30	868888524, Facultad de Informática B1.2.038	PLAN A: Despacho 2.30 INFORMÁTICA PLAN B y C:Mediante herramientas online como Zoom y mensajes de AV
		Segundo Cuatrimestre	Martes	16:00- 18:00	868888524, Facultad de Informática B1.2.038	PLAN A: Despacho 2.30 INFORMÁTICA PLAN B y C:Mediante herramientas online como Zoom y mensajes de AV
		Segundo Cuatrimestre	Martes	10:00- 11:00		



2. Presentación

Con esta asignatura el estudiante conocerá las arquitecturas más frecuentemente usadas en las aplicaciones multiplataforma (web o dispositivo móvil), junto con las principales tecnologías para construir los dos grandes elementos de las aplicaciones web: back-end (servidor) y front-end (cliente).

3. Condiciones de acceso a la asignatura

3.1 Incompatibilidades

No consta

3.2 Recomendaciones

No hay recomendaciones ni limitaciones al acceso para los estudiantes del Máster Universitario en Ingeniería del Software.

4. Competencias

4.1 Competencias Básicas

No disponible

4.2 Competencias de la titulación

No disponible

4.3 Competencias transversales y de materia

5. Contenidos

Bloque 1: Introducción al desarrollo web

TEMA 1. Principios básicos de la web: HTML y CSS

- El lenguaje HTML.
- Presentación de sitios web con CSS.



TEMA 2. Lenguajes de programación para la web

- El lenguaje JavaScript.
- Programación en el navegador web.
- El entorno de ejecución JavaScript (NodeJS).
- El lenguaje TypeScript.

TEMA 3. Introducción al desarrollo full-stack para una aplicación web

- Introducción a los conceptos de web apps vs native apps
- Introducción a los conceptos de front-end vs back-end
- Introducción al concepto de arquitectura mvc-web para un servidor NodeJS(Express/Nest)

Bloque 2: Back-end

TEMA 4. Arquitectura del back-end

- Microservicios.
- Patrones Domain-Driven Design.
- Arquitectura Clean.
- Sistemas Reactivos: colas de mensajes.

TEMA 5. Desarrollo del back-end: API Web

- Especificación OpenAPI.
- Estilo arquitectónico REST.
- Buenas prácticas de diseño REST.
- Herramientas de prototipado y pruebas.
- Seguridad: Autenticación y autorización.
- Patrón pasarela de API.
- GraphQL.

TEMA 6. Comunicación en el back-end

- Comunicación asíncrona (colas de mensajes).
- Comunicación orientada al recurso (REST).



- Comunicación orientada al procedimiento (RPC).

Bloque 3: Front-end

TEMA 7. Diseño de Interfaces Webs Adaptativas

- Diseño web adaptativo
- Estudio de frameworks de desarrollo adaptativo basados en CSS: BootStrap y/o Tailwind

TEMA 8. Diseño de la Experiencia de usuario UX

- Introducción al concepto de UX
- Diseño de interfaces webs orientado a la UX.
- Ejemplos de principios y buenas prácticas en el diseño de la UX: Material Design

TEMA 9. Desarrollo del front-end

- Desarrollo del front-end de una aplicación web haciendo uso de la familia de tecnologías JS
- Desarrollo basado en el uso de lenguajes de plantilla para el servidor (Express-hbs/jade)
- Desarrollo modular del front-end haciendo uso de tecnologías de componentes para la web (React, Vue)

Bloque 4: Desarrollo web para dispositivos móviles

TEMA 10. Desarrollo Progresivo

- Desarrollo Progresivo con PWA

TEMA 11. Desarrollo Híbrido

- Desarrollo Híbrido usando tecnologías JS/TypeScript como ReactNative

PRÁCTICAS

Práctica 1. HTML y CSS: Relacionada con los contenidos Tema 1

- Construcción de páginas web con el lenguajes HTML
- Uso de CSS para el establecimiento de propiedades visuales

Práctica 2. JavaScript: Relacionada con los contenidos Tema 2

- Programación de aplicaciones con el lenguaje JavaScript (JS)
- Programación JS en el navegador web: browser apis.



- El entorno de ejecución JavaScript (NodeJS).

Práctica 3. TypeScript.: Relacionada con los contenidos Tema 2

El lenguaje de programación TypeScript.

Práctica 4. Diseño de servicios: Relacionada con los contenidos Tema 4

Diseño e implementación del caso de uso, en términos de la arquitectura Clean, utilizando patrones Domain-Driven Design.

Práctica 5. Diseño REST: Relacionada con los contenidos Tema 5

Aplicación de las recomendaciones y buenas prácticas de diseño REST a varios casos de estudio.

Práctica 6. OpenAPI: Relacionada con los contenidos Tema 5

Especificación de un API Web utilizando la propuesta OpenAPI y las herramientas Swagger.

Práctica 7. Pruebas de un API Web: Relacionada con los contenidos Tema 5 y Tema 6

Estudio de las herramientas cURL y Postman para probar un API Web.

Práctica 8. Framework Nest: Relacionada con los contenidos Tema 5

Estudio del framework Nest como base de la implementación de un API Web de acuerdo a los principios de la arquitectura hexagonal.

Práctica 9. Pasarela de API: Relacionada con los contenidos Tema 5

- Implementación del patrón pasarela de API con los frameworks Express y Nest de NodeJS.
- Implementación de mecanismos de autenticación y autorización de API en el framework Nest.

Práctica 10. Comunicación asíncrona: Relacionada con los contenidos Tema 6

- Implementación de sistemas reactivos basados en la comunicación asíncrona mediante colas de mensajes utilizando el protocolo AMQP y el software RabbitMQ.

Práctica 11. Comunicación síncrona: Relacionada con los contenidos Tema 6

- Uso del software gRPC para la implementación de la comunicación síncrona orientada al procedimiento.

Práctica 12. GraphQL: Relacionada con los contenidos Tema 5

- Estudio de la herramienta Apollo para construir un servidor GraphQL en JavaScript.
- Uso del cliente Apollo para la conexión a un servidor GraphQL.

Práctica 13. Desarrollo de páginas webs adaptativas: Relacionada con los contenidos Tema 7

- Diseño web adaptativo: uso de flexbox, grid
- Frameworks adaptativos basados en CSS: BootStrap y/o Tailwind

Práctica 14. Experiencia de Usuario UX: Relacionada con los contenidos Tema 8

- Introducción y diseño de la UX.
- Principios y buenas prácticas de la UX (Material Design)

Práctica 15. Desarrollo del front-end basado en plantillas: Relacionada con los contenidos Tema 9

Desarrollo del front-end sobre NodeJS/Express basado en lenguajes de plantilla (Express-hbs/jade)

Práctica 16. Desarrollo modular del front-end: Relacionada con los contenidos Tema 9

Desarrollo del front-end siguiendo un enfoque modular, basado en el uso de componentes JS: React y/o Vue

Práctica 17. Desarrollo Progresivo: Relacionada con los contenidos Tema 10

Desarrollo Progresivo usando la tecnología Progressive Web Application (PWA)

Práctica 18. Desarrollo Híbrido: Relacionada con los contenidos Tema 11

Desarrollo híbrido de aplicaciones webs haciendo uso de las tecnologías ReactNative y/o Flutter

6. Metodología Docente

Actividad Formativa	Metodología	Horas Presenciales	Trabajo Autónomo	Volumen de trabajo
AF1: Sesiones virtuales y sincrónicas	MD1.1 Actividades de clase expositiva: exposición teórica, clase magistral, proyección, dirigida al gran grupo, con independencia de que su contenido sea teórico o práctico. La modalidad de impartición estará adaptada a las posibilidades que ofrece la docencia virtual. Junto a la exposición de conocimientos, en las clases se plantean cuestiones, en ocasiones utilizando herramientas de respuesta de audiencia, se aclaran dudas, se realizan ejemplificaciones, se establecen relaciones con las diferentes actividades prácticas que se realizan y se orienta la búsqueda de información.	40		40.00



Actividad Formativa	Metodología	Horas Presenciales	Trabajo Autónomo	Volumen de trabajo
AF2: Seminarios especializados virtuales y sincrónicos	MD1.2 Seminarios: trabajo de los alumnos de profundización en una temática concreta, que puede integrar contenidos teóricos y prácticos, realizado en grupos reducidos y supervisado por el profesor, concluyendo con la elaboración y presentación escrita de un informe que, en algunos casos, puede hacerse público mediante exposición oral por parte de los alumnos y debate.	8		8.00



Actividad Formativa	Metodología	Horas Presenciales	Trabajo Autónomo	Volumen de trabajo
AF3: Prácticas de laboratorio virtuales y sincrónicas	<p>MD2.1 Actividades de clase práctica de aula: actividades prácticas de ejercicios y resolución de problemas, estudio de casos, actividades presenciales programadas en una docencia de aula invertida, aprendizaje basado en roles, contrato didáctico o de aprendizaje, exposición y análisis de trabajos, debates, simulaciones, etc. Suponen la realización de tareas por parte de los alumnos, dirigidas y supervisadas por el profesor, con independencia de que en el aula se realicen individualmente o en grupos reducidos.</p> <p>M2.2 Aprendizaje orientado a proyectos. Los estudiantes, organizados en grupos, llevan a cabo de forma colaborativa la realización de un proyecto en un tiempo determinado abordando una tarea mediante la planificación, diseño y realización de una serie de actividades, en las cuales se puede utilizar también un aprendizaje basado en roles.</p>	40		40.00
AF5: Trabajo autónomo del alumno	<p>Trabajo autónomo del estudiante para repasar y estudiar los conceptos vistos en clase, realizar las tareas asignadas y preparación de las sesiones de la asignatura.</p>	0	204	204.00



Actividad Formativa	Metodología	Horas Presenciales	Trabajo Autónomo	Volumen de trabajo
AF6: Tutorías formativas virtuales, sincrónicas y asincrónicas, individualizadas y en grupo	<p>MD3.1 Tutorías en grupo: sesiones programadas de orientación, revisión o apoyo a los alumnos por parte del profesor, realizadas en pequeños grupos, con independencia de que los contenidos sean teóricos o prácticos, pudiendo ser presenciales o virtuales.</p> <p>MD3.2 Tutorías individualizadas: sesiones de intercambio individual con el estudiante, previstas en el desarrollo de la materia, pudiendo ser presenciales o virtuales</p>	8		8.00
	Total	96	204	300

7. Horario de la asignatura

<https://www.um.es/web/master-software/2023-24#horarios>

8. Sistema de Evaluación

No disponible

Fechas de exámenes

<https://www.um.es/web/master-software/2023-24#exámenes>

9. Resultados del Aprendizaje

- R11. Diseñar arquitecturas full-stack basadas en el uso de frameworks orientados a la programación del front-end y back-end, incluyendo servicios transversales tales como seguridad, autorización y autenticación.



- R12. Desarrollar el front-end de una aplicación siguiendo un enfoque nativo, híbrido y basado en WebApps.
- R13. Desarrollar el back-end de una aplicación haciendo uso de lenguajes y tecnologías actuales.
- R14. Diseñar e implementar una arquitectura basada en servicios y microservicios.
- R15. Capacidad de aplicar la programación reactiva en el desarrollo full-stack.

10. Bibliografía

Bibliografía Básica



M. Haverbeke, Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming



Número de Título: 670621. Erl, Thomas. Service-Oriented Architecture: analysis and design for services and microservices. Prentice Hall, 2017.

Bibliografía Complementaria



.NET Microservices: Architecture for Containerized .NET Applications



D. Betts et al. Exploring CQRS and Event Sourcing: A Journey into High Scalability, Availability, and Maintainability with Windows Azure. Microsoft patterns & practices, 1st edition, 2013. [Libro gratuito]



Banks, Alex & Procello, Eve. Learning React Modern Patterns for Developing React Apps, Ed. OREILLY. ISBN: 978-1-492-05172-5



Número de Título: 681437. Aubry, Christophe. Responsive Web Design, diseños y cuadrículas: Técnicas modernas de diseño web. Ed. Cornellá de Llobregat (Barcelona) : Eni, 2017.



Número de Título: 767083. De Voil, Nick. User experience foundations. Ed. Swindon, U.K : BCS The Chartered Institute for It, 2020.



Número de Título: 402941. Eric Evans. Domain-driven design : tackling complexity in the heart of software. Addison Wesley, 2004.



11. Observaciones y recomendaciones

Proyecto de máster

El máster de Ingeniería del Software sigue una metodología de "aprendizaje basado en proyecto" por lo que el trabajo autónomo del estudiante en esta asignatura estará relacionado con el proyecto definido para el conjunto del máster.

Competencias generales del título a las que contribuye la asignatura

- CG1 Capacidad para integrarse en equipos de desarrollo de software que aplican conceptos, métodos, técnicas y tecnologías modernas que actualmente se usan en las principales empresas del mundo.
- CG2 Capacidad de producir software de calidad a través de la aplicación de principios, métodos y técnicas ampliamente aceptadas y usando herramientas extendidas.
- CG3 Capacidad de diseñar soluciones software en cualquier dominio abordando los diferentes aspectos arquitecturales.
- CG5 Habilidades para trabajo en equipo y gestión de equipos en proyectos software.
- CG6 Concienciación del papel crucial del software en la sociedad actual y cómo puede ayudar al bienestar de las personas.

Competencias específicas del título a las que contribuye la asignatura

- CE01 Capacidad para realizar la dirección, estimación y seguimiento de proyectos ágiles de software garantizando su valor de negocio e incorporando la gestión de riesgos y de la seguridad desde las primeras etapas.
- CE02 Capacidad para integrarse en la ejecución de proyectos ágiles de diferente tamaño desempeñando los distintos roles de gestión y desarrollo del proyecto.
- CE04 Capacidad para aplicar DevOps en la producción de software a través de las herramientas apropiadas.
- CE06 Capacidad para diseñar el proceso de integración de DevOps en las líneas de producción de software de una empresa.
- CE07 Diseñar y desplegar arquitecturas de procesamiento de datos escalables para proyectos de desarrollo de software intensivos en datos.



- CE08 Capacidad para usar e integrar bases de datos modernas.
- CE10 Capacidad para diseñar y ejecutar cada tipo de prueba que debe ser aplicado a un producto software.
- CE11 Capacidad para usar herramientas de prueba y su integración con otras herramientas de automatización en el desarrollo y las operaciones.
- CE12 Capacidad de diseñar un plan de pruebas y validación para un proyecto de desarrollo de una aplicación.
- CE13 Conocer y saber aplicar principales arquitecturas usadas para desarrollo Web y de aplicaciones móviles.
- CE14 Capacidad de usar frameworks del lado front-end y back-end.
- CE15 Capacidad de diseñar la arquitectura full-stack de una aplicación de acuerdo a los requisitos funcionales y no funcionales, y teniendo en cuenta la mejora de la productividad a través de reutilización y automatización y aspectos de seguridad.
- CE16 Capacidad de desarrollar una aplicación de calidad, abordando desde los requisitos hasta el despliegue en la nube, a través de prácticas ágiles y continuas, y aplicando técnicas que permitan conseguir usabilidad y la mayor productividad posible.

Instrumentos de evaluación

SE1. Entrevistas virtuales de seguimiento de prácticas.

- Peso 25%
- Criterios de evaluación: Este instrumento está relacionado con las sesiones prácticas de la asignatura, tanto las sesiones supervisadas como la práctica entregable y puntuable. Estas sesiones tendrán durante el desarrollo de la asignatura, pudiendo haber una entrevista final relacionada con la entrega final. Todas las entrevistas tendrán el mismo peso en la calificación de este instrumento. Se valorará:
 - ✂ Respuestas de los estudiantes a las preguntas haga el profesor durante las entrevistas para controlar el proceso de aprendizaje.
 - ✂ Asistencia y participación.
 - ✂ Grado de avance en la tarea asignada sometida a seguimiento.
- No se establece nota mínima para este instrumento de evaluación



SE2. Evaluación de informes escritos, trabajos y proyectos

- Peso 65%
- Criterios de evaluación: Este instrumento está relacionado con la práctica entregable y puntuable. Esta práctica se podrá entregar en cualquier convocatoria del curso. Se valorará:
 - ✂ La presentación correcta del trabajo, según las pautas marcadas por el profesorado.
 - ✂ Justificación de las decisiones tomadas.
 - ✂ Resolución eficaz y eficiente de las tareas propuestas.
 - ✂ Grado de integración de los conocimientos teóricos y prácticos de la asignatura, desde un conocimiento reflexivo y analítico.
 - ✂ Cumplimiento de plazos.
- Se establece un 5/10 como nota mínima para este instrumento de evaluación.

SE4. Procedimientos de observación del trabajo del estudiante

- Peso 10%
- Criterios de evaluación: Incluye la valoración de las siguientes actividades:
 - ✂ Respuestas de los estudiantes a las preguntas que haga el profesor durante las exposiciones teóricas para controlar el proceso de aprendizaje.
 - ✂ Asistencia y participación.
 - ✂ Realización de ejercicios propuestos.
 - ✂ Cumplimiento de plazos.
- No se establece nota mínima para este instrumento de evaluación

Derecho a prueba final

El artículo 8.6 del Reglamento de Evaluación de Estudiantes (REVA) prevé: "Salvo en el caso de actividades definidas como obligatorias en la guía docente, si el o la estudiante no puede seguir el proceso de evaluación continua por circunstancias sobrevenidas debidamente justificadas, tendrá derecho a realizar una prueba global".

Atención a la diversidad



Aquellos estudiantes con discapacidad o necesidades educativas especiales podrán dirigirse al Servicio de Atención a la Diversidad y Voluntariado (ADYV; <https://www.um.es/web/adyv/inicio>) para recibir orientación sobre un mejor aprovechamiento de su proceso formativo y, en su caso, la adopción de medidas de equiparación y de mejora para la inclusión, en virtud de la Resolución Rectoral R358/2016. El tratamiento de la información sobre este alumnado, en cumplimiento con la LOPD, es de estricta confidencialidad.

Sobre Objetivos de Desarrollo Sostenible

Esta asignatura se encuentra vinculada de forma directa con el Objetivo de Desarrollo Sostenible 9 Industria, Innovación e Infraestructuras.