

Guía docente de la asignatura

Desarrollo del Software (M52/56/5/2)

Fecha de aprobación por la Comisión Académica: 04/07/2025

Máster

Máster Universitario en Desarrollo del Software

MÓDULO

Introducción

RAMA

Ingeniería y Arquitectura

CENTRO RESPONSABLE DEL TÍTULO

Escuela Internacional de Posgrado

Semestre

Anual

Créditos

3

Tipo

Obligatorio

Tipo de enseñanza

Presencial

PRERREQUISITOS Y/O RECOMENDACIONES

En el caso de utilizar herramientas de IA para el desarrollo de la asignatura, el estudiante debe adoptar un uso ético y responsable de las mismas. Se deben seguir las recomendaciones contenidas en el documento de "Recomendaciones para el uso de la inteligencia artificial en la UGR" publicado en esta ubicación: <https://ceprud.ugr.es/formacion-tic/inteligencia-artificial/recomendaciones-ia#contenido>

BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Máster)

- Ingeniería de requisitos.
- Arquitecturas software: fundamentos. clasificación, estilos, y casos de estudio (capas, cliente-servidor, cloud, edge, fog, pizarra, serverless).
- Tecnologías middleware.
- Definición y gestión de interfaces de programación de aplicaciones (APIs).
- Técnicas de desarrollo de software: patrones, diseño basado en modelos, etc.
- Buenas prácticas y herramientas de desarrollo: guías, repositorios, control de versiones y ticketing, documentación.
- Modelos de desarrollo de software: pipeline. Integración continua, dirigido por modelos.
- Metodologías y entornos de trabajo ágiles (Scrum, Kanban, Agile Inception...).
- Desarrollo de operaciones: Devops, configuración y despliegue.
- Desarrollo dirigido por tests. Automatización



- Requirements engineering.
- Software architectures: fundamentals, taxonomies, styles, and case studies (layers, client-server, cloud, edge, fog, slate, serverless).
- Middleware technologies.
- Definition and management of application programming interfaces (APIs).
- Software development techniques: patterns, model-driven design, etc.
- Best practices and development tools: guides, repositories, version control and ticketing, documentation.
- Software development models: pipeline. Continuous integration, model-driven.
- Agile methodologies and work environments (Scrum, Kanban, Agile Inception...).
- Operations development: Devops, configuration and deployment.
- Test-driven development. Automation

COMPETENCIAS

COMPETENCIAS BÁSICAS

- CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
- CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.
- CB8 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.
- CB9 - Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.
- CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

COMPETENCIAS

COMPETENCIAS BÁSICAS

- CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
- CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.
- CB8 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.



- CB9 - Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades.
- CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

RESULTADOS DE APRENDIZAJE (Objetivos)

Conocimientos o contenidos

C02. Identifica y comprende los conceptos clave y las principales características de los sistemas software, en cuanto a sus requerimientos, diseño o implementación, pruebas, despliegue y explotación.

C09. Identifica y comprende las estrategias, técnicas y herramientas que facilitan la digitalización y monitorización de procesos productivos y flujos de trabajo.

C11. Comprende los principios y fundamentos de diseño de software seguro, identificando las vulnerabilidades y amenazas que pueden afectar a cualquier tipo de sistema software.

C12. Conoce los fundamentos y beneficios que tiene el diseño, implementación y despliegue de servicios, microservicios y contenedores en el desarrollo de sistemas, plataformas y herramientas software.

Competencias

COM5. Identificar y valorar las propiedades del software de usabilidad, accesibilidad, seguridad, confiabilidad, rendimiento y ética informática, entre otros y analizar como afecta a la calidad de un sistema software.

Habilidades o destrezas

HD03. Aplica los modelos, métodos, técnicas, paradigmas, algoritmos, lenguajes y herramientas más apropiados para la creación, desarrollo o mantenimiento de sistemas software que cumplan con criterios de calidad, usabilidad, robustez, fiabilidad, seguridad, facilidad de implementación y despliegue en las plataformas más actuales.

HD08. Maneja metodologías, técnicas y buenas prácticas para el desarrollo de sistemas software seguros.

PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

TEÓRICO

Tema 1. Fundamentos.



- Ingeniería de requisitos.
- Arquitecturas software: fundamentos. clasificación, estilos, y casos de estudio (capas, cliente-servidor, cloud, edge, fog, pizarra, serverless).
- Tecnologías middleware.
- Técnicas de desarrollo de software: patrones, diseño basado en modelos, etc.

Tema 2. El proceso de desarrollo del software.

- Definición y gestión de interfaces de programación de aplicaciones (APIs).
- Modelos de desarrollo de software: pipeline. Integración continua, dirigido por modelos.
- Metodologías y entornos de trabajo ágiles (Scrum, Kanban, Agile Inception...).

Tema 3. Tecnologías y Herramientas para el desarrollo del software

- Buenas prácticas y herramientas de desarrollo: guías, repositorios, control de versiones y ticketing, documentación.
- Desarrollo de operaciones: Devops, configuración y despliegue.
- Desarrollo dirigido por tests. Automatización

Module 1. Foundations.

- Requirements engineering.
- Software architectures: fundamentals, classification, styles, and case studies (layers, client-server, cloud, edge, fog, slate, serverless).
- Middleware technologies.
- Software development techniques: patterns, model-driven design, etc.

Module 2. The software development process.

- Definition and management of application programming interfaces (APIs).
- Software development models: pipeline. Continuous integration, model-driven.
- Agile methodologies and work environments (Scrum, Kanban, Agile Inception...).

Module 3: Technologies and tools for software development.

- Best practices and development tools: guides, repositories, version control and ticketing, documentation.
- Operations development: Devops, configuration and deployment.
- Test-driven development. Automation

PRÁCTICO

Se realizarán distintos tipos de ejercicios prácticos para que el estudiante conozca con detalle los fundamentos y buenas prácticas del desarrollo del software, del proceso que se debe seguir para el desarrollo del software y las herramientas y tecnologías más adecuadas.

Como resultado el estudiante tendrá que montar el entorno de trabajo para el desarrollo de los proyectos software que se incluyan en su trabajo fin de máster.



The student must carry out different types of practical exercises in order to know in detail the fundamentals and good practices of software development, the process to be followed and the most appropriate tools and technologies.

As a result, the student will have to set up the working environment for the development of the software projects included in his master's thesis.

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL

- Len Bass, Paul Clements, Rick Kazman (2022): Software architecture in practice (4th edition), Addison-Wesley.
- Mark Richards (2021): Software architecture patterns (Second edition), O'Reilly Media.
- Cha, S., Taylor, R. N., & Kang, K. (Eds.) (2019): Handbook of software engineering, Springer.
- Gruhn, V., & Striemer, R. (2018): The Essence of Software Engineering, Springer Nature.
- Oquendo, F., Leite, J., & Batista, T. (2016): Software Architecture in Action, Springer.
- Martin, R. (2012) Código Limpio: Manual de estilo para el desarrollo ágil de software, Anaya Multimedia
- Martin, R. (2018) Arquitectura limpia: Guía para especialistas en la estructura y el diseño de software, Anaya Multimedia
- Rubin, K. (2012) Essential Scrum: A Practical Guide to the Most Popular Agile Process, Addison Wesley
- Lasa, C. Alvarez A., de las Heras, R. (2017) Métodos Ágiles. Scrum, Kanban, Lean, Anaya Multimedia
- Kim G., Humble J. Debois, P. Willis J. (2021) The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations
- Humble, J. and Farley D. (2010) Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Addison-Wesley
- Beck, K. (2002) Test-Driven Development: By Example, Addison-Wesley

BIBLIOGRAFÍA COMPLEMENTARIA

- Tilak Mitra (2016): Practical software architecture : moving from system context to deployment (1st edition), Pearson, IBM Press
- Kim, G., Behr, K., Spafford S. (2018) The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business. IT Revolution Press

ENLACES RECOMENDADOS

Como apoyo a la docencia se usará la Plataforma de Recursos de Apoyo a la Docencia PRADO2 de la Universidad de Granada: <https://prado.ugr.es>

EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

EVALUACIÓN ORDINARIA



El artículo 18 de la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada establece que la convocatoria ordinaria estará basada preferentemente en la evaluación continua del estudiante, excepto para quienes se les haya reconocido el derecho de evaluación única final.

Se realizará una evaluación continua del trabajo del estudiante, valorando tanto los conocimientos adquiridos como las competencias alcanzadas.

Modalidad presencial:

Para la evaluación en modalidad presencial se tendrán en cuenta los siguientes sistemas de evaluación, indicándose entre paréntesis el rango del porcentaje con respecto a la calificación final del estudiante.

SE1. Actividades realizadas durante el desarrollo del curso mediante la entrega de ejercicios, trabajos, informes, a través de la plataforma docente (60%).

SE2. Actividades realizadas después de finalizar el curso mediante la entrega de ejercicios, trabajos, informes, a través de la plataforma docente (20%).

SE3. Presentación pública de trabajos o ejercicios (10%)

SE5. Asistencia y participación activa (10%)

Se pedirá la entrega en tiempo y forma de las actividades propuestas a través de la plataforma PRADO.

Modalidad virtual:

Para la evaluación en modalidad virtual se tendrán en cuenta los siguientes sistemas de evaluación, indicándose entre paréntesis el rango del porcentaje con respecto a la calificación final del estudiante.

SE1. Actividades realizadas durante el desarrollo del curso mediante la entrega de ejercicios, trabajos, informes, a través de la plataforma docente (60%).

SE2. Actividades realizadas después de finalizar el curso mediante la entrega de ejercicios, trabajos, informes, a través de la plataforma docente (20%).

SE6. Participación activa en foros de debate o de recogida de información (10%).

SE7. Presentación pública de trabajos o ejercicios sincrónica (10%).

Se pedirá la entrega en tiempo y forma de las actividades propuestas a través de la plataforma PRADO.

EVALUACIÓN EXTRAORDINARIA

El artículo 19 de la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de



Granada establece que los estudiantes que no hayan superado la asignatura en la convocatoria ordinaria dispondrán de una convocatoria extraordinaria. A ella podrán concurrir todos los estudiantes, con independencia de haber seguido o no un proceso de evaluación continua. De esta forma, el estudiante que no haya realizado la evaluación continua tendrá la posibilidad de obtener el 100% de la calificación final.

[La evaluación en tal caso consistirá en la realización de una prueba y/o trabajo, y/o las actividades propuestas en la evaluación continua.](#)

EVALUACIÓN ÚNICA FINAL

El artículo 8 de la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada establece que podrán acogerse a la evaluación única final, el estudiante que no pueda cumplir con el método de evaluación continua por causas justificadas.

Para acogerse a la evaluación única final, el estudiante en las dos primeras semanas de impartición de la asignatura o en las dos semanas siguientes a su matriculación si ésta se ha producido con posterioridad al inicio de las clases o por causas sobrevenidas. Lo solicitará, a través del procedimiento electrónico, a la Coordinación del Máster, quien dará traslado al profesorado correspondiente, alegando y acreditando las razones que le asisten para no poder seguir el sistema de evaluación continua.

La evaluación en tal caso consistirá en la realización de una prueba y/o trabajo, y/o las actividades propuestas en la evaluación continua.

INFORMACIÓN ADICIONAL

Siguiendo las indicaciones recogidas en el artículo 15 de la Normativa de Evaluación y de Calificación de la Universidad de Granada sobre la originalidad de los trabajos presentados por los alumnos.

1. La Universidad de Granada fomentará el respeto a la propiedad intelectual y transmitirá a los estudiantes que el plagio es una práctica contraria a los principios que rigen la formación universitaria. Para ello procederá a reconocer la autoría de los trabajos y su protección de acuerdo con la propiedad intelectual según establezca la legislación vigente.
2. El plagio, entendido como la presentación de un trabajo u obra hecho por otra persona como propio o la copia de textos sin citar su procedencia y dándolos como de elaboración propia, conllevará automáticamente la calificación numérica de cero en la asignatura en la que se hubiera detectado, independientemente del resto de las calificaciones que el estudiante hubiera obtenido. Esta consecuencia debe entenderse sin perjuicio de las responsabilidades disciplinarias en las que pudieran incurrir los estudiantes que plagien.
3. Los trabajos y materiales entregados por parte de los estudiantes tendrán que ir firmados con una declaración explícita en la que se asume la originalidad del trabajo, entendida en el sentido de que no ha utilizado fuentes sin citarlas debidamente.

Información de interés para estudiantado con discapacidad y/o Necesidades Específicas de Apoyo Educativo (NEAE): [Gestión de servicios y apoyos](#) (<https://ve.ugr.es/servicios/atencion-social/estudiantes-con-discapacidad>).

