

Curso Académico: ( 2024 / 2025 )

Fecha de revisión: 25-04-2024

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: LLORENS MORILLO, JUAN BAUTISTA

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 4 Cuatrimestre : 2

## REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Ingeniería del software (Segundo curso - Primer cuatrimestre)  
Desarrollo del software (Segundo curso - Segundo cuatrimestre)  
Dirección de proyectos de desarrollo software (Cuarto curso - Primer cuatrimestre, recomendable)

## COMPETENCIAS Y RESULTADOS DEL APRENDIZAJE

- ¿ Conocer los tipos de sistemas de información corporativos en relación con los modelos de negocio
- ¿ Conocer y aplicar la gestión integral de proyectos, la estimación de recursos y la gestión de calidad.
- ¿ Conocer y aplicar la gestión del cambio y mantenimiento así como procesos de auditoría.

## OBJETIVOS

El objetivo de la asignatura es obtener los conocimientos y habilidades necesarias para Diseñar, planificar, construir, desplegar y operar un sistema software asegurando los requisitos de calidad en los diferentes entornos y arquitecturas definidos para la construcción del mismo.

## DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

- Fundamentos y conceptos del desarrollo y operación de sistemas software
- Planificación de sistemas software
- Arquitectura y construcción de sistemas software
- Despliegue de sistemas software
- Operación y monitorización de sistemas software
- Aseguramiento de calidad de sistemas software

## ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

Clases Teóricas-Prácticas: 2 ECTS

- Trabajo preparativo previo

Clases Prácticas de Laboratorio: 1 ECTS

- Resolución de ejercicios
- Exposición parcial de la práctica

Trabajo en Grupo: 1 ECTS

- Desarrollo de la práctica
- Presentación y revisión de la prácticas

Trabajo Individual: 1 ECTS

- Contribución a la práctica del equipo
- Ejercicios individuales
- Estudio y preparación del examen

Tutorías: 1 ECTS

- Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor

## SISTEMA DE EVALUACIÓN

|  |     |
|--|-----|
| <b>Peso porcentual del Examen Final:</b>           | 0   |
| <b>Peso porcentual del resto de la evaluación:</b> | 100 |

#### EVALUACIÓN CONTINUA (100%)

La evaluación de la asignatura se basa en la realización de un proyecto en equipo aplicando los conceptos teóricos y las herramientas utilizadas en las prácticas semanales para construir un sistema con elementos software y hardware.

Durante la ejecución del curso se realizarán dos revisiones del estado de desarrollo del mismo. La primera revisión demandará el dominio de la tecnología a utilizar.

La segunda y última revisión verificará que el sistema desarrollado está preparado para su utilización operativa el día del examen.

Dichas verificaciones se evalúan mediante la técnica Go/No Go.

El día del examen final se realizará una validación final del sistema, así como una demostración funcional bajo un sistema competitivo. El sistema debe proporcionar la funcionalidad demanda en los requisitos, y debe realizarla sin fallos (pruebas adecuadas).

La calificación del equipo se basa en los siguientes criterios:

- Sistema verificado y validado (funcionalmente correcto): aprobado
- Verificación final del Sistema antes del combate 10%
- Clasificación en la competición 80%
- Rol de cada miembro del equipo y Evaluación 360º entre el equipo y el líder de equipo. Este último criterio permite establecer el desempeño individual y discernir la evaluación individual. 10%.

#### EVALUACIÓN FINAL

-Examen final sólo si no se participa en el desarrollo del proyecto o los responsables de los equipos expulsan a una persona del equipo (bajo los criterios claramente delimitados en el Project Charter). (100%)

#### BIBLIOGRAFÍA BÁSICA

- Christof Ebert; Gorka Gallardo; Josune Hernantes; Nicolas Serrano DevOps, IEEE Software, 2016
- D. Farley Modern software engineering: doing what really works to build better software faster, Addison-Wesley, 2021
- G. Kim, K. Behr, and G. Spafford The phoenix project: a novel about IT, DevOps, and helping your business win, Portland, OR: IT Revolution, 2018
- J. Davis and K. Daniels Effective devOps: building a culture of collaboration, affinity, and tooling at scale, O¿Reilly, 2016
- M. Richards and N. Ford Fundamentals of software architecture: an engineering approach, O¿Reilly Media, 2020
- Martin Eigner System Lifecycle Management, Springer, 2021
- N. Ford, M. Richards, P. J. Sadalage, and Z. Dehghani Software architecture: the hard parts: modern trade-off analysis for distributed architectures, O¿Reilly Media, 2021
- N. Forsgren, J. Humble, and G. Kim Accelerate: the science behind DevOps: building and scaling high performing technology organizations, Portland, Oregon: IT Revolution, 2018
- R. C. Martin and R. C. Martin Clean architecture: a craftsman¿s guide to software structure and design, Prentice Hall, 2018
- Thomas M. Shortell INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, Willey, 2015

## BIBLIOGRAFÍA COMPLEMENTARIA

- B. Beyer, C. Jones, J. Petoff, and N. R. Murphy Site reliability engineering: how Google runs production systems, O'Reilly, 2016
- C. Rosenthal and N. Jones Chaos engineering: system resiliency in practice, O'Reilly Media, 2020
- K. Morris Infrastructure as code: managing servers in the cloud, O'Reilly, 2016
- M. T. Nygard Release it! design and deploy production-ready software, Pragmatic Bookshelf, 2007

## RECURSOS ELECTRÓNICOS BÁSICOS

- Amazon AWS . DevOps on AWS Specialization: <https://www.coursera.org/specializations/aws-devops>
- IBM . IBM DevOps and Software Engineering Professional Certificate: <https://www.coursera.org/professional-certificates/devops-and-software-engineering>
- Microsoft . Microsoft Certified: DevOps Engineer Expert: <https://learn.microsoft.com/en-us/certifications/devops-engineer/>
- Wilderness Labs . Meadow producer: <https://www.wildernesslabs.co/>