



## Guía docente de la asignatura

<b>Asignatura</b>	Evaluación y Rendimiento de Sistemas Software		
<b>Materia</b>	Ingeniería de Software		
<b>Módulo</b>	Tecnologías específicas		
<b>Titulación</b>	Grado en Ingeniería Informática.		
<b>Plan</b>	<b>545</b>	<b>Código</b>	<b>46925</b>
<b>Periodo de impartición</b>	2º cuatrimestre	<b>Tipo/Carácter</b>	Obligatoria (Mención IS) Optativa-6 (Mención CO)
<b>Nivel/Ciclo</b>	Grado	<b>Curso</b>	3º - Mención IS 4º - Mención CO
<b>Créditos ECTS</b>	6		
<b>Lengua en que se imparte</b>	Castellano		
<b>Profesor/es responsable/s</b>	José Manuel Marqués Corral		
<b>Datos de contacto (E-mail, teléfono...)</b>	<a href="mailto:jmmc@infor.uva.es">jmmc@infor.uva.es</a> Teléfono: ext. – 5638		
<b>Horario de tutorías</b>	Véase el sitio web oficial de la Universidad de Valladolid <a href="http://www.uva.es">www.uva.es</a>		
<b>Departamento</b>	Informática		

### 1. Situación / Sentido de la Asignatura

#### 1.1 Contextualización

Los sistemas y aplicaciones software son un elemento esencial para el buen funcionamiento, la gestión y el apoyo a la dirección de cualquier empresa u organización. Un bajo rendimiento en el software que da soporte a su actividad, provocará retrasos y disfunciones, repercutiendo negativamente en la viabilidad y continuidad de la actividad.

Abordar los problemas de rendimiento una vez finalizado el desarrollo y despliegue de las aplicaciones, la práctica más extendida, conduce a sobrecostes de hardware y software básico, a inversión de tiempo y personal en procedimientos de reajuste y refactorización, y, en ciertas circunstancias, a un rediseño completo de la aplicación. Habitualmente, los esfuerzos en la mejora y ajuste del rendimiento de los sistemas suelen centrarse en aspectos de tecnologías de la información (IT), obviando la gestión del rendimiento de las aplicaciones. Una forma de evitar los problemas señalados consiste en incorporar metodologías de ajuste del rendimiento del software a lo largo del ciclo de vida del desarrollo.

En esta asignatura se expondrán un conjunto de métodos, técnicas y herramientas utilizadas para garantizar que los requisitos de calidad de servicio (QoS) de un producto software se cumplen una vez desarrollado. Se estudiarán métodos basados en modelos para el ajuste y dimensionamiento de aplicaciones software a lo largo de todo el ciclo de vida.



## 1.2 Relación con otras materias

Arquitectura de Computadoras, Sistemas Operativos, Ingeniería del software, Programación, Redes y Estadística.

## 1.3 Prerrequisitos

Es recomendable que el alumno haya cursado y superado las asignaturas de Fundamentos de Computadoras, Fundamentos de Sistemas Operativos, Fundamentos de Redes, Arquitectura y Organización de Computadores, Administración de Sistemas Operativos, Fundamentos de Ingeniería del Software y Estadística. Todas estas materias se tratan en asignaturas del primer y segundo curso de los estudios de Graduado en Ingeniería Informática.

## 2. Competencias

### 2.1 Generales

Código	Descripción
G02	Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas a continuación en esta sección de la memoria.
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
G04	Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas.
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 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
IS1	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 de Software
IS2	Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.
CI5	Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas.
CI13	Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web



### 3. Objetivos

Código	Descripción
RA1	Conocer los modelos y técnicas de evaluación de sistemas informáticos
RA2	Ser capaz de elaborar el modelado del comportamiento de un sistema software
RA3	Conocer los métodos y técnicas para la predicción del rendimiento y de la escalabilidad de los sistemas software desde las fases iniciales del desarrollo
RA4	Ser capaz de utilizar las técnicas y herramientas para la evaluación y optimización de rendimiento de Aplicaciones

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

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	30	Estudio y trabajo autónomo individual	45
Clases prácticas de aula (A)		Estudio y trabajo autónomo grupal	45
Laboratorios (L)	30		
Prácticas externas, clínicas o de campo			
Seminarios (S)			
Tutorías grupales (TG)			
Evaluación			
<b>Total presencial</b>	<b>60</b>	<b>Total no presencial</b>	<b>90</b>



## 5. Bloques temáticos

### 5.1 Teoría

#### Bloque 1: Evaluación y rendimiento de sistemas software

Carga de trabajo en créditos ECTS: 

6.0
-----

##### a. Contextualización y justificación

Un producto software perfecto desde el punto de vista funcional será un producto inservible si no exhibe el rendimiento adecuado. La mejora de recursos hardware y el rediseño del producto son las opciones más habituales para paliar, en la medida de lo posible, los problemas de rendimiento y escalabilidad del software en operación. Estas estrategias, como ya se indicó, suponen en el mejor de los casos un proceso complicado y costoso, cuyos resultados no son siempre óptimos. Al igual que el resto de requisitos, rendimiento y escalabilidad han de ser tomados en cuenta a lo largo de todo el ciclo de vida, incorporando la planificación y predicción del rendimiento en el proceso de diseño y desarrollo del software.

La ingeniería del rendimiento software (*Software Performance Engineering*) (*SPE*) es un conjunto de principios, técnicas y patrones que pueden ser utilizadas para la gestión de los aspectos de rendimiento en el proceso de concepción, diseño y construcción del software. Al igual que los métodos de análisis y diseño del software orientado al objeto, el *SPE* se basa en la construcción de modelos que pueden ser analizados para determinar si se cumplen los requisitos de rendimiento antes de abordar el desarrollo.

##### b. Objetivos de aprendizaje

RA1 Conocer los modelos y técnicas de evaluación de sistemas informáticos

RA2 Ser capaz de elaborar el modelado del comportamiento de un sistema software

RA3 Conocer los métodos y técnicas para la predicción del rendimiento y de la escalabilidad de los sistemas software desde las fases iniciales del desarrollo.

RA4 Ser capaz de utilizar las técnicas y herramientas para la evaluación y optimización de rendimiento de aplicaciones.

##### c. Contenidos

- 1 Introducción a la ingeniería del rendimiento.
  - a. Definiciones y Conceptos.
  - b. Métricas de rendimiento y comportamiento del sistema
- 2 Ingeniería del Rendimiento en el desarrollo de software.
  - a. Actividades de la Ingeniería del Rendimiento Software.
  - b. Métodos de la Ingeniería del Rendimiento Software.
  - c. UML para la Ingeniería del Rendimiento Software.
- 3 Modelo de ejecución del software
  - a. Modelo de Flujo de Ejecución de Programa.
  - b. Modelo basado en Grafo de Ejecución – SPE



- c. Modelo basado en Grafo de Ejecución - PRIMA-UML
- d. Modelo basado en el Diagrama de Actividad - UML
- 4 Modelo de ejecución del sistema.
  - a. Modelo de colas y de redes de colas
  - b. Construcción del modelo
  - c. Resolución del modelo

#### d. Métodos docentes

Ver Anexo: Métodos docentes

#### e. Plan de trabajo

El cronograma de actividades aparecerá en el Aula Virtual o página web asociada a la asignatura.

#### f. Evaluación

Ver apartado 7, sistema de calificaciones.

#### g. Bibliografía básica

- C. U. Smith y L. G. Williams, Performance Solutions: A Practical Guide to Creating Responsive, Scalable Software, 1 edition. Boston, MA: Addison-Wesley Professional, 2001.
- D. A. Menasce, L. W. Dowdy, y V. A. F. Almeida, Performance by Design: Computer Capacity Planning By Example, 1.ª ed. Prentice Hall, 2004

#### h. Bibliografía complementaria

- Raj Jain, The Art of Computer Systems Performance Analysis, John Wiley & Sons Publisher, 1991.
- Daniel A. Menascé and Virgilio A. F. Almeida, Capacity Planning for Web Services, Metrics, Models, and Methods, Prentice-Hall, 2002.
- Xavier Molero, C. Juiz y M. Rodeño. Evaluación y modelado del Rendimiento de los Sistemas Informáticos. Pearson-Prentice-Hall, 2004.

## 6. Temporalización (por bloques temáticos)

### 6.1 Teoría

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Evaluación y rendimiento de sistemas software.	6.0	Semanas 1 a 15



**7. Tabla resumen de los instrumentos, procedimientos y sistemas de evaluación/calificación**

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
<b>Examen final de teoría</b>	<b>70%</b>	
Prácticas de Laboratorio		
Prueba de carga (load testing).	10%	Semanas 1 a 3
Elaboración de un modelo de ejecución del software.	10%	Semanas 4 a 9
Construcción y resolución de un modelo de ejecución del sistema.	10%	Semanas 10 a 13  Las semanas 14 y 15 se reservan para corrección, mejora y defensa de los trabajos entregados.
<b>Total Prácticas de Laboratorio</b>	<b>30 %</b>	

1. La evaluación del alumno se separa en dos partes bien diferenciadas, la teórica y la práctica. Para poder superar la asignatura será necesario tener superadas individualmente las dos partes de acuerdo con los criterios que se indican en los puntos 4 y 5.
2. El examen final de teoría consistirá en la resolución de problemas en los que se desarrollen y apliquen los conocimientos vistos en las sesiones de teoría.
3. La evaluación de la parte práctica, al tener carácter de evaluación continua, solamente se puede realizar durante el periodo de actividad docente.
  - Dentro del periodo lectivo se podrán completar o subsanar trabajos ya realizados y entregados durante el periodo de actividad docente, teniendo que entregar los informes de la práctica y realizar su defensa antes de la fecha de examen.
  - La calificación de estas nuevas versiones de los trabajos elaborados no estará condicionada por la calificación que se hubiese obtenido en la versión anterior.
4. Para aprobar la asignatura es necesario haber alcanzado una nota mínima en la práctica de laboratorio y en el examen final.
  - Para la práctica de laboratorio 5 puntos sobre 10.
  - Para el examen final de teoría 4 puntos sobre 10.
5. Para establecer la calificación final de un alumno se tiene en cuenta la calificación obtenida en la práctica de laboratorio y la obtenida en el examen escrito de teoría.

Sea

- $N_{Total} = 0.3 \cdot NPL + 0.7 \cdot NEF$ ; donde NPL es la nota, sobre 10 puntos, de la práctica de laboratorio y NEF la calificación, sobre 10 puntos, del examen final.

La nota final en la asignatura, **NFinal**, para un alumno será:

- Si se cumple la condición 4, ha obtenido la calificación mínima en la práctica de laboratorio y examen final, entonces: **NFinal**=NTotal
- En caso contrario: **NFinal** = min (3.9, NTotal)



## 8. Consideraciones finales

La realización fraudulenta cualquiera de las pruebas de evaluación o de los trabajos de laboratorio (copia o trabajos no originales), supondrá automáticamente una calificación de SUSPENSO con una nota de 0.0 puntos en el acta de la asignatura.

### Anexo: Métodos docentes

#### Para las sesiones teóricas:

Descripción de los principales contenidos teóricos durante las horas presenciales de aula.

En su caso se proporcionarán a los alumnos enunciados de problemas que deberán trabajar de forma individual o en grupo para su resolución.

Se utilizarán recursos bibliográficos, notas y apuntes, enlaces a material disponible en la red, actividades individuales o en grupo.

#### Para las sesiones en el Laboratorio:

Durante la semana previa a la sesión o sesiones de prácticas de laboratorio el alumno estudiará de manera personal o en grupo la documentación relativa a las tareas correspondientes a las sesiones de laboratorio.

Las horas presenciales de laboratorio incluirán, si es necesario para su desarrollo, clase magistral participativa, y en cualquier caso, la realización de un proyecto guiado por el profesor que encargará y guiará el trabajo. Los bloques de laboratorio se realizarán en grupos de 4 alumnos, siguiendo un enfoque colaborativo.