

Guía Docente

Modalidad Presencial

Planificación, Verificación y Validación del Software

Curso 2018/19

Grado en Ingeniería de
Sistemas de Información



UCAV

www.ucavila.es

Nombre:	PLANIFICACIÓN, VERIFICACIÓN Y VALIDACIÓN DEL SOFTWARE
Carácter:	OBLIGATORIO
Código:	40204GH
Curso:	4º
Duración (Semestral/Anual):	SEMESTRAL
Nº Créditos ECTS:	6
Prerrequisitos:	CONOCIMIENTOS DE INGENIERIA DEL SW I Y II
Responsable docente:	SORAYA ANDALUZ DELGADO GRADUADA EN INGENIERIA DE SISTEMAS DE INFORMACIÓN
Email:	soraya.andaluz@ucavila.es
Departamento (Área Departamental):	TECNOLÓGICO
Lengua en la que se imparte:	CASTELLANO
Módulo:	CONTENIDOS ESPECIFICOS A LA INGENIERIA INFORMATICA
Materia:	SISTEMAS DE INFORMACIÓN EMPRESARIAL

2.1. COMPETENCIAS BÁSICAS Y GENERALES

- CB1. Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

- CB2. Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.
- CB3. Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
- CB4. Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
- CB5. Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
- G1. Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del Anexo II de la Resolución de 8 de junio de 2009, de la Secretaría General de Universidades, por la que se da publicidad al Acuerdo del Consejo de Universidades, por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la Ingeniería Informática, Ingeniería Técnica Informática e Ingeniería Química, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas.
- G2. Capacidad para dirigir las actividades objeto de los proyectos del ámbito de la informática de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del Anexo II de la Resolución de 8 de junio de 2009, de la Secretaría General de Universidades, por la que se da publicidad al Acuerdo del Consejo de Universidades, por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la Ingeniería Informática, Ingeniería Técnica Informática e Ingeniería Química.

- G3. 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.
- G5. Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del Anexo II de la Resolución de 8 de junio de 2009, de la Secretaría General de Universidades, por la que se da publicidad al Acuerdo del Consejo de Universidades, por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la Ingeniería Informática, Ingeniería Técnica Informática e Ingeniería Química.
- G8. 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.
- G9. 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, tanto en lengua nativa como en lengua inglesa.
- G10. Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en el apartado 5 del Anexo II de la Resolución de 8 de junio de 2009, de la Secretaría General de Universidades, por la que se da publicidad al Acuerdo del Consejo de Universidades, por el que se establecen recomendaciones para la propuesta por las universidades de memorias de solicitud de títulos oficiales en los ámbitos de la Ingeniería Informática, Ingeniería Técnica Informática e Ingeniería Química.

2.2. COMPETENCIAS ESPECÍFICAS

- E7. Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.
- E19. Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.
- E21. Capacidad para elaborar el pliego de condiciones técnicas de una instalación informática que cumpla los estándares y normativas vigentes.
- E24. Capacidad de integrar soluciones de Tecnologías de la Información y las Comunicaciones y procesos empresariales para satisfacer las necesidades de información de las organizaciones, permitiéndoles alcanzar sus objetivos de forma efectiva y eficiente, dándoles así ventajas competitivas.
- E25. Capacidad para determinar los requisitos de los sistemas de información y comunicación de una organización atendiendo a aspectos de seguridad y cumplimiento de la normativa y la legislación vigente.
- E27. Capacidad para comprender y aplicar los principios de la evaluación de riesgos y aplicarlos correctamente en la elaboración y ejecución de planes de actuación.
- E29. Capacidad de documentar de forma correcta todas las etapas del ciclo de vida del software.
- E33. Capacidad para participar activamente en la especificación, diseño, implementación y mantenimiento de los sistemas de información y comunicación.

2.3. RESULTADOS DE APRENDIZAJE

- Estimaciones de costes y esfuerzos necesarios por medio de mediciones del software.
- Conocimiento de los modelos de estimación.
- Conocimiento de las Métricas del software.
- Ser capaz de realizar la gestión de riesgos.
- Conocimiento de las herramientas para la planificación y gestión de productos software.
- Comprobación de los requerimientos del software en su desarrollo.
- Aprendizaje de las estrategias y técnicas de prueba del software.
- Conocer y diferenciar entre validación y verificación del software.
- Conocer el concepto de pruebas, situarlas dentro del ciclo de vida de desarrollo de un producto y relacionarla con el proceso y sistema de pruebas.
- Saber definir una serie de estrategias a seguir durante el proceso de pruebas, proporcionando pautas que pueden ser útiles para la elección de la mejor estrategia.
- Saber que existen distintos niveles y tipos de pruebas, y en qué consisten.
- Aprender una serie de buenas prácticas a seguir durante el proceso de pruebas.
- Aprender qué son las revisiones, tarea más orientada al proceso de verificación, y la importancia de las mismas.
- Analizar las distintas técnicas de prueba de software, con sus debilidades y sus puntos fuertes.
- Comprender las distintas técnicas de prueba dinámicas, cómo y donde deben aplicarse.
- Profundizar en el conocimiento de las revisiones, como técnica de prueba estática, y aprender en qué consiste el análisis estático.
- Conocer los beneficios que pueden aportar herramientas dedicadas a pruebas, así como los riesgos que supone el uso de herramientas en las empresas.
- Clasificar los distintos tipos de herramientas de pruebas en base de su funcionalidad.



3.1. PROGRAMA

1. Administración y gestión de proyectos software
 - 1.1. Introducción a la gestión de proyectos
 - 1.2. La Medición del Software
 - 1.3. Las Métricas del software
 - 1.4. Estimación de proyectos
 - 1.5. Gestión de riesgos
 - 1.6. Planificación de proyectos software
 - 1.7. Estándares de gestión de proyectos
2. Planificación temporal y seguimiento del proyecto
 - 2.1. Introducción a la planificación temporal
 - 2.2. Técnicas de planificación temporal
 - 2.2.1. Diagrama de Gantt
 - 2.2.2. Método PERT
 - 2.2.3. Método CPM
 - 2.3. Herramientas informáticas de planificación temporal
3. Procesos de validación y verificación
 - 3.1. Validación y verificación
 - 3.2. Actividades de validación y verificación
 - 3.2.1. Pruebas
 - 3.2.2. Estrategias de pruebas
 - 3.2.3. Niveles de pruebas
 - 3.2.4. Tipos de pruebas
 - 3.2.5. Revisiones
 - 3.3. Validación y verificación en distintos modelos del ciclo de vida
4. Técnicas y herramientas de validación y verificación
 - 4.1. Técnicas de prueba
 - 4.1.1. Técnicas de prueba dinámicas
 - 4.1.2. Técnicas de prueba estáticas
 - 4.2. Herramientas de prueba

3.2. BIBLIOGRAFÍA

- Abran, A.; Desharnais J. M.; Maya M.; St-Pierre D.; Bourque P. 1998. Design of a Functional size Measurement for Real-Time Software. Research Report N° 13. Software Engineering Management Research Laboratory, University du Québec a Montreal, Canada.
- AENOR. 2011. Gestión de riesgos. Normas UNE. AENOR
- Albrecht, A. J. 1979. Measuring Application Development Productivity. Proc Of IBM applications. Development Joint SHARE/GUIDE Symposium, Monterrey, Páginas 83-92.
- Albrecht, A. J. & Gaffney J.E. 1983. Software function, source lines of code and development effort prediction: a software science validation. IEEE Trans. Software Engineering. Pp. 639 - 648.
- Alonso, F & Martínez, L & Segovia, F.J. 2005. *Introducción a la ingeniería del software: modelo de desarrollo de programas*. Delta.
- Andaluz, S & Morales, A & Díaz, L. 2012. *Ingeniería del Software I*. Universidad Católica de Ávila.
- Andaluz, S & Morales, A & Díaz, L. 2012. *Ingeniería del Software II*. Universidad Católica de Ávila.
- Arthur, L.J. 1985. Measuring Programmer Productivity and Software Quality. Wiley Interscience.
- Beizer, B., *Software Testing Techniques*, Segunda Edición, 1990.
- Black, R., *Managing the Testing Process*, Segunda Edición, 2001.
- Boehm, B. W. 1989. Software Risk Management. IEEE Computer Society Press.
- Boehm, B., *Software Engineering Economics*, 1981.
- Brian Hambling, Peter Morgan, Angelina Samaroo, Geoff Thompson y Peter Williams, *Software Testing - An ISEB Foundation*, 2007.
- Buchtik L. 2013. Secretos para dominar la gestión de riesgos. *Buchtik global*.
- Cerrada, J.A. 2000. *Introducción a la Ingeniería del Software*. Editorial Centro de Estudios Ramón Areces, S.A.
- Cos, M. 1997. Teoría General del proyecto, Ed. Síntesis.
- Dijkstra, Edsger, *Notes on Structured Programming*, 1972.

- Down, A. & Coleman, M. & Absolon, P. Risk Management for Software Projects. McGraw-Hill Book Company.
- Funston F. & Wagner, S. 2010. Sobrevivir y prosperar en la incertidumbre: Cómo crear una empresa con inteligencia de riesgos. John Wiley & Sons, Inc.
- Gilb T. 1988. Principles of Software Engineering Management. AddisonWesley.
- Goethert, W. y Sivi, J. (2004). Applications of the Indicador Template for Measurement and Analysis. Technical Note CMU/SEI-2004-TN-024. Software Engineering Institute, Septiembre 2004.
- Hampton, J. 2009. Fundamentos de administración de riesgo empresarial: Cómo las mejores compañías evalúan riesgos, manejan la exposición y aprovechan oportunidades. Amacom.
- IEEE “Standard for Software Verification and Validation”, IEEE 1012, 2004.
- IEEE Computer Society, IEEE Std. 829-1998 “Software test documentation”, IEEE, 1998.
- IEEE Software Engineering Standards, Std 610.12 - 1990, pp: 47-48.
- IEEE 1998. IEEE Std 1061-1998 IEEE Standard for a Software Quality Metrics Methodology.
- INTECO, 2009. “Guía de Validación y Verificación”. Laboratorio Nacional de Calidad del Software.
- International Function Point users Group. Function Point Counting Practices Manual. Release 4.0. 1994.
- ISO 31000:2009. Risk management - Principles and guidelines.
- ISO/IEC. 2002. ISO 15939: Software Engineering - Software Measurement Process.
- Jones, C. 1987. A Short History of Function Points and Feature Points. Software Productivity Research Inc. USA.
- McGarry, J., Card, D., Jones, C., Layman, B., Clark, E., Dean, J. y Hall, F. (2002). Practical Software Measurement. Objective Information for Decision Makers. Addison-Wesley.
- Ould, M. Managing Software Quality and Business Risk. 1999. John Wiley & Sons.
- Ould, M. Strategies For Software Engineering: The Management of Risk and Quality. 1990. John Wiley & Sons.

- Park, R., Goethert, W., Florac, W. (1996). Goal-Driven Software Measurement - A Guidebook. Handbook CMU/SEI-96-HB-002, Software Engineering Institute, Agosto 1996.
- Piattini, M & Calvo-Manzano, J.A & Cervera, J & Fernández, L. 2007. *Análisis y diseño detallado de Aplicaciones Informáticas de Gestión*. RA-MA.
- Piattini, M., García, F., Garzás, J., Genero, M. 2008. *Medición y Estimación del Software: Técnicas y Métodos para Mejorar la Calidad y Productividad*. Ra-Ma.
- Pressman, R. 2010. *Ingeniería del software, un enfoque práctico*. Mc Graw Hill.
- Ragland, B. 1995. Measure, metric or indicator: What's de difference?. Crosstalk, vol 8, nº 3, pp: 29-30.
- Rex Black, *Pragmatic Software Testing: Becoming an effective and efficient test professional*, 2007.
- Roger S. Pressman, *Software Engineering. A practitioner's Approach*, Quinta Edición, 2001.
- Romero, C. 1997. *Técnicas de programación y control de proyectos*, Pirámide.
- Ron Burbach, *Software Engineering Methodology*, 1998.
- Scott, G .2001. Strategic Planning for High-Tech Product Development. *Technology Analysis & Strategic Management*, Vol. 13, No. 3.
- Sommerville, I. 2005. *Ingeniería de software*. Addison Wesley.
- Sommerville, Ian, *Software Engineering 07 Edition*, 2005.
- Symons, C. R. 1998. Function Point Analysis Difficulties and Improvements. *IEEE Transactions on Software Engineering*. Páginas 2-11.
- Van Solingen, R. y Berghout, E. 1999. *The Goal/Question/Metric Method, A Practical Guide for Quality Improvement of Software Development*. London, England: McGraw-Hill International (UK),
- Whitmire, S. A. 1992. 3D Function Points: Scientific and Real-Time Extensions to Function Points. *Pacific Northwest Software Quality Conference*.
- Williams, R.C. & Walker, J.A. & Dorofee, A.J. 1997. *Putting Risk Management into Practice*. IEEE Software.

La asignatura se desarrollará a través de los siguientes métodos y técnicas generales, que se aplicarán diferencialmente según las características propias de la asignatura:

- **Exposición del profesor:** mediante la clase magistral el profesor de la asignatura expondrá y explicará a los alumnos los contenidos principales de la misma, fomentando la participación y la opinión crítica de los alumnos. Estas sesiones tienen como objetivo transmitir conocimiento y activar procesos cognitivos en el estudiante. Durante las sesiones de teoría se podrán plantear preguntas o situaciones problemáticas sobre un tema, ocasionar debate individual o en grupo y resolver dudas que se puedan plantear.
- **Estudio personal dirigido:** el alumno acometerá de forma individual el estudio de la asignatura de modo que le permita adquirir las competencias de la misma. Para ello contará con la tutorización personalizada del profesor de la asignatura, como principal responsable docente.
- **Ejercicios y problemas prácticos:** tienen como objetivo la aplicación directa de los conocimientos adquiridos durante la clase magistral. Se propondrá al alumno la realización de ejercicios y casos prácticos para que resuelva y lo confronte con las soluciones dadas por el profesor.
- **Prácticas con ordenador:** estas sesiones tendrán lugar en el aula de informática. El alumno realizará individualmente las diferentes prácticas según las competencias y actividades correspondientes que deba trabajar durante el semestre. Para ello contará con la tutorización personalizada del profesor de la asignatura, como principal responsable docente.
- **Realización de examen escrito sobre materia:** El alumno realizará un examen final que junto con los trabajos obligatorios servirán para la evaluación de éste en la materia.
- **Revisión:** El profesor establecerá un horario para que el alumno pueda ponerse en contacto con él para poder revisar la evaluación de la asignatura.

La evaluación es un componente fundamental de la formación del alumno. Está compuesta por un examen final escrito y la evaluación continua, que consta de ejercicios y actividades evaluables.

La evaluación de esta asignatura se realiza mediante la media del examen (valorado en un 60%) y la evaluación continua (con valor del 40%).

➤ Examen (60 % de la nota final)

La superación de dicho examen constituye un requisito indispensable para la superación de la asignatura. El alumno deberá tener en el examen al menos un 5 para poder realizar la ponderación de notas. El alumno con nota inferior se considerará suspenso. El alumno dispondrá de dos convocatorias de examen por curso académico. No se guardará la nota del examen, si éste estuviera aprobado, para una convocatoria posterior.

➤ Evaluación continua (40% de la nota final)

La evaluación continua está formada por un trabajo obligatorio y el seguimiento y la participación activa del alumno en la asignatura. El trabajo obligatorio pondera un 20%. El seguimiento y la participación activa del alumno ponderan el otro 20% restante sobre la nota final de la asignatura. Las fechas de entrega del trabajo obligatorio se indicarán al alumno con suficiente antelación a la entrega del mismo en cada una de las convocatorias del curso académico.

No se admitirán trabajos fuera de estas fechas límite de entrega. Con la no presentación del trabajo obligatorio en fecha, se considerará una puntuación de cero en esta parte a ponderar, con independencia de la nota obtenida en el examen.

La superación del trabajo no es un requisito indispensable para la superación de la asignatura. La ponderación de la nota del trabajo obligatorio puede ser establecida sin necesidad de la entrega del mismo; es decir, no es necesaria su presentación, así como tampoco su superación, para la realización del examen final.

En el caso de tener el trabajo obligatorio superado y no aprobar el examen, se guardará su nota hasta la segunda convocatoria de examen perteneciente al curso académico actual. Si el alumno lo desea, puede presentar un nuevo trabajo obligatorio para intentar mejorar la nota del anterior. Esto solamente será válido para la segunda convocatoria de examen del curso académico.

Si acercada la fecha de cualquiera de las convocatorias, el alumno observara que no le da tiempo a realizar la entrega del trabajo en tiempo y forma, puede solicitar al profesor una ampliación del plazo establecido. Esta concesión será como máximo de una semana más a partir de la fecha límite de entrega y llevará consigo una penalización en la nota, de tal manera que, como máximo, el alumno obtendrá un 5 como nota final en la asignatura. En ningún caso supondrá el suspenso de la asignatura si el examen y el trabajo obligatorio estuvieran aprobados.

EJERCICIOS Y ACTIVIDADES EVALUABLES	PROPORCIÓN
Examen final escrito	60%
Evaluación continua	40%
TOTAL	100%

Los criterios para la evaluación del trabajo obligatorio se presentan en la siguiente tabla, donde se resumen los aspectos a valorar y el porcentaje que representa cada uno de los mismos:

COMPONENTES EVALUABLES	PROPORCIÓN
Gestión y administración del proyecto	30
Planificación temporal de proyecto	30
Realización de batería de pruebas	20
Utilización de herramientas informáticas	20
TOTAL	100%

Para el apoyo tutorial, el alumno tendrá a su disposición un equipo docente encargado de acompañar al alumno durante toda su andadura en el proceso formativo, prestando una atención personalizada al alumno. Las dos figuras principales son:

- **Profesor docente:** encargado de resolver todas las dudas específicas de la asignatura y de informar al alumno de todas las pautas que debe seguir para realizar el estudio de la asignatura.
- **Tutor personal o de grupo:** asignado al alumno al iniciar los estudios de Grado y que orienta al alumno tanto en cuestiones académicas como personales.

La información sobre el horario de tutorías la encontrará el alumno en la plataforma virtual. El horario de tutorías podrá estar sujeto a cambios puntuales durante el curso académico. Prevalcerán los horarios publicados en la plataforma virtual.

Horario de Tutorías: LUNES de 13:00 14:00 horas.

MARTES de 11:00 12:00 horas y de 16:00 18:00 horas.

MIÉRCOLES de 10:00 a 11 horas y de 13:00 14:00 horas.

Horario de la asignatura: **MARTES DE 09:00 a 11:00 horas y
JUEVES DE 09:00 a 11:00 horas.**

El peso de cada unidad formativa dentro de cada asignatura queda determinado según la siguiente tabla o cronograma por el tiempo dedicado a la misma. **Las unidades de tiempo y las horas de dedicación son orientativas, pudiendo variar ligeramente, dependiendo de la evolución del alumno.**

UNIDADES DIDÁCTICAS	UNIDAD DE TIEMPO	HORAS DEDICACIÓN
Unidad 1. Administración y gestión de proyectos software	20	30 HORAS
Unidad 2. Planificación temporal y seguimiento del proyecto	20	30 HORAS
Unidad 3. Procesos de validación y verificación	20	30 HORAS
Unidad 4. Técnicas y herramientas de validación y verificación	20	30 HORAS
Trabajo Obligatorio	20	30 HORAS
TOTAL	100	150

Las sesiones se desarrollarán según la siguiente tabla, en la que se recogen las semanas, contenidos, metodología y actividad. **El plan de trabajo y las semanas son orientativos, pudiendo variar ligeramente, dependiendo de la evolución del alumno durante las distintas sesiones.**

SEMANAS	CONTENIDOS	METODOLOGIA Y ACTIVIDAD
1	Unidad 1	Exposición del profesor y estudio personal dirigido
2	Unidad 1	Exposición del profesor y estudio personal dirigido
3	Unidad 2	Exposición del profesor, estudio personal dirigido, ejercicios y problemas prácticos y prácticas con ordenador
4	Unidad 2	Exposición del profesor, estudio personal dirigido, ejercicios y problemas prácticos y prácticas con ordenador
5	Unidad 2	Exposición del profesor, estudio personal dirigido, ejercicios y problemas prácticos y prácticas con ordenador
6	Unidad 2	Exposición del profesor, estudio personal dirigido, ejercicios y problemas prácticos y prácticas con ordenador
7	Unidad 2	Exposición del profesor, estudio personal dirigido, ejercicios y problemas prácticos y prácticas con ordenador
8	Unidad 2	Exposición del profesor, estudio personal dirigido, ejercicios y problemas prácticos y prácticas con ordenador
9	Unidad 3	Exposición del profesor, estudio personal dirigido y ejercicios y problemas prácticos
10	Unidad 3	Exposición del profesor, estudio personal dirigido y ejercicios y problemas prácticos

SEMANAS	CONTENIDOS	METODOLOGIA Y ACTIVIDAD
11	Unidad 3	Exposición del profesor, estudio personal dirigido y ejercicios y problemas prácticos
12	Unidad 4	Exposición del profesor, estudio personal dirigido y ejercicios y problemas prácticos
13	Unidad 4	Exposición del profesor, estudio personal dirigido y ejercicios y problemas prácticos
14	Unidad 4	Exposición del profesor, estudio personal dirigido y ejercicios y problemas prácticos

*Esta tabla se presenta a título informativo y orientativo. La estructura exacta de las clases dependerá del normal desarrollo de las mismas.