24-25

GUÍA DE ESTUDIO PÚBLICA

ESPECIFICACIÓN DE LOS SISTEMAS SOFTWARE

CÓDIGO 31105024

UNED

24-25

ESPECIFICACIÓN DE LOS SISTEMAS SOFTWARE

CÓDIGO 31105024

ÍNDICE

PRESENTACIÓN Y CONTEXTUALIZACIÓN
REQUISITOS Y/O RECOMENDACIONES PARA CURSAR ESTA ASIGNATURA
EQUIPO DOCENTE
HORARIO DE ATENCIÓN AL ESTUDIANTE
COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE
RESULTADOS DE APRENDIZAJE
CONTENIDOS
METODOLOGÍA
SISTEMA DE EVALUACIÓN
BIBLIOGRAFÍA BÁSICA
BIBLIOGRAFÍA COMPLEMENTARIA
RECURSOS DE APOYO Y WEBGRAFÍA
IGUALDAD DE GÉNERO

Nombre de la asignatura	ESPECIFICACIÓN DE LOS SISTEMAS SOFTWARE
Código	31105024
Curso académico	2024/2025
Título en que se imparte	MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS
Tipo	CONTENIDOS
Nº ETCS	9
Horas	225
Periodo	ANUAL
Idiomas en que se imparte	CASTELLANO

PRESENTACIÓN Y CONTEXTUALIZACIÓN

Presentación:

Bajo el título **Especificación de los Sistemas Software** se presentan diversas metodologías y técnicas formales para especificar requisitos en distintas etapas del proceso de desarrollo de sistemas software.

Cada vez son más los desarrollos de software que requieren comprobación y verificación formal en alguna parte del producto. Sobre todo en aquellos proyectos que implican la gestión de sistemas críticos: transacciones, aviación y vuelos espaciales, plantas nucleares, etc.

Las diferentes metodologías de verificación se basan en diversas ampliaciones de la lógica clásica. El hacer uso de ellas requiere saber como expresar los requisitos a verificar en las gramáticas de estas lógicas.

Empezaremos con un repaso de las lógicas clásicas de primer orden: lógica proposicional y la lógica de predicados. Las futuras ampliaciones de estas lógicas las veremos dentro de su marco de utilización a lo largo de curso. Así, por ejemplo, estudiaremos la lógica temporal dentro de la verificación de sistemas concurrentes. En otros escenarios, en cambio, lo importante es verificar que el propio código funciona correctamente, para ello existen metodolologías y tecnologías llamadas verificadores de programas que utilizan una lógica proposicional modificada. En concreto, veremos en el curso la lógica de Hoare.

Contextualización:

La asignatura ***Especificación de los Sistemas Software*** se encuentra integrada en el Bloque de Ingeniería del Software dentro de la materia *Ingeniería del Desarrollo del Software* .

Tiene carácter anual y es de tipo optativo. El número de créditos ECTS que se consigue es 9, igual al resto de asignaturas del máster.

Sus contenidos están diseñados para capacitar la comprensión, alcance y limitación de la verificación formal de requisitos. Obteniendo así una visión general de las distintas soluciones que pudieran aparecer en los distintos ámbitos del Desarrollo del Software.

Estos métodos y soluciones específicos de la verificación dependen, principalmente, de la propia arquitectura del software, que los condicionará en gran medida. Es por ello que podemos afirmar que los contenidos de esta asignatura son completamente transversales a los del resto de asignaturas que forman la materia Ingeniería del Desarrollo de Software.

Esto significa, que sus contenidos no dependen de asignatura previas ni tiene continuación con otra asignatura. Se puede cursar y estudiar de forma independiente, ya su vez, resultará un buen complemento al resto de las asignaturas de su materia.
La introducción de métodos genéricos de especificación y verificación formal dentro del desarrollo software es relativamente nuevo. De hecho estos métodos provienen de otros campos de la ingeniería como Diseño Hardware, Ingeniería Industrial y también del propio desarrollo software pero en dominios de aplicación muy limitada. Es por ello que esta asignatura también se complementa perfectamente con otras materias del Máster como por ejemplo:
- Robótica y Percepción Visual
- Sistemas Difusos y Aplicaciones
- Sistemas Móviles

REQUISITOS Y/O RECOMENDACIONES PARA CURSAR ESTA ASIGNATURA

Para una mejor comprensión de la materia **es recomendable** que el alumno haya realizado un curso introductorio de ingeniería del software así como otro de lógica formal.
Estos conocimientos se han debido adquirir en asignaturas del grado como:

•Introducción a la Ingeniería del Software.

•Introducción a la Lógica Formal.

•Matemática Discreta

Se trata de una asignatura con contenidos matemáticos que requiere saber utilizar el formalismo de la lógica.

No obstante, la asignatura es auto contenida, esto es, todos los conceptos que damos están definidos y explicados desde su inicio.

EQUIPO DOCENTE

Nombre y Apellidos	JOAN ANTONI MASCARELL ESTRUCH (Coordinador/a de asignatura)
Correo Electrónico	jmascarell@issi.uned.es
Teléfono	
Facultad	
Departamento	

HORARIO DE ATENCIÓN AL ESTUDIANTE

CURSO VIRTUAL

La tutorización y seguimiento del alumno se hará mediante las herramientas propias de la plataforma virtual: correo, foros, pruebas de evaluación, etc...
En la metodología a distancia de la UNED, los **foros** del curso virtual son el principal recurso de atención colectiva los estudiantes. La comunicación a través de los foros tiene una doble vertiente en el aprendizaje: el enriquecimiento en el ejercicio de la dialéctica y del diálogo

entre los estudiantes, y la exposición del profesor a todos los alumnos (atención colectiva), junto con el debate que ello pueda suscitar.
En la atención colectiva en los foros del curso virtual, ante cualquier cuestión concreta, planteada sobre los contenidos o el funcionamiento de la asignatura, la respuesta será inferior a 5 días del calendario lectivo.

HORARIO ATENCIÓN ESTUDIANTE

Así mismo, la atención personal al alumno se hará en el siguiente horario de guardia.
Horario de atención presencial y telefónica (*guardia*):
Miércoles **lectivos** de 15:00 a 19:00 h.
Profesorado:
Juan Antonio Mascarell Estruch.
Telf.: +34 91398 8220
Correo: jmascarell@issi.uned.es
Dirección postal:
ETS de Ingeniería Informática de la UNED
Dpto. de Ingeniería de Software y Sistemas Informáticos. Despacho 2.22.
C/ Juan del Rosal, 16
28040 Madrid

COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Competencias Básicas:

CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el 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 Generales:

CG01 - Saber aplicar los conocimientos adquiridos y la capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares relacionados con la Ingeniería de Sistemas y la Ingeniería de Software.
CG02 - Demostrar una comprensión sistemática del campo de estudio de la Ingeniería de Software o de la Ingeniería de Sistemas, y el dominio de las habilidades y métodos de

investigación relacionados con dicho campo.
CG03 - Demostrar la capacidad de concebir, diseñar, poner en práctica y adoptar un proceso sustancial de investigación con seriedad académica.
CG04 - Ser capaz de realizar un análisis crítico, evaluación y síntesis de ideas nuevas y complejas.
CG05 - Saber comunicar sus conclusiones -y los conocimientos y razones últimas que las sustentan- a públicos especializados y no especializados, a sus colegas, a la comunidad académica en su conjunto y a la sociedad, de un modo claro y sin ambigüedades.
CG06 - Ser capaz de fomentar, en contextos académicos y profesionales, el avance tecnológico dentro de una sociedad basada en el conocimiento.
CG07 - Ser capaz 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.
CG08 - Realizar una contribución a través de una investigación original que amplíe las fronteras del conocimiento desarrollando un corpus sustancial, del que parte merezca la publicación referenciada a nivel nacional o internacional.
CG09 - Poseer las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

Competencias Específicas:
CE01 - Incorporar mejoras cualitativas sustanciales, bien sea en la elaboración de software o bien en el desarrollo e implantación de sistemas robóticos.
CE02 - Concebir, implementar implantar y supervisar nuevas soluciones a los problemas específicos que se le planteen en el ámbito de la investigación, innovación y desarrollo de software o de la robótica.

RESULTADOS DE APRENDIZAJE

RESULTADOS DE APRENDIZAJE
Los resultados de aprendizaje que se espera alcanzar con esta asignatura por parte del estudiante son:

- Comprender bien el papel de la especificación.
- Comprender la necesidad de utilizar métodos formales de especificación.
- Conocer los sistemas lógicos y tener habilidad para expresar requisitos através de ellos.
- Conocer el funcionamiento de la herramienta Alloy. Esta herramienta genera modelos para requisitos expresados en lenguaje de predicados.
- Conocer los límites de la expresividad en la lógica clásica y aprender algunas de sus extensiones: lógica temporal y modal.

CONTENIDOS

Bloque 1 : La especificación

En este bloque repasaremos los conceptos básicos de la Ingeniería del software vistos el cursos anteriores.
El objetivo es intentar recordar y esclarecer en que consiste la especificación de un Sistema Software.

Bloque 2 : Lógica clásica y su alcance

Los principales sistemas formales de verificación tienen como lenguaje de especificación alguna extensión de la lógica clásica. En esta unidad repasaremos la lógica clásica: la proposicional y de predicados. La funcionalidad de muchos sistemas o modelos vienen muy bien descritos mediante las expresiones lógicas.
Diferenciar bien los conceptos de demostración o inferencia frente a la deducción lógica o semántica es clave para entender el concepto de modelo en lógica.

Bloque 3 : Lógicas temporales

En este bloque estudiaremos los sistemas de verificación agrupados bajo el nombre de model checking. Esta técnica consiste en crear un modelo del sistema, codificar la propiedad a estudiar (requisito) en el lenguaje del entorno y ver si el modelo satisface dicha propiedad.
En general las técnicas de verificación mediante el examen de modelos utilizan un sistema formal basado en la lógica temporal.

Bloque 4 : Verificación de Programas

En los sistemas secuenciales que manipulan grandes cantidades de información con datos complejos, el posible espacio de estados que describiría el sistema en cada momento concreto (dando valor a todas la variables) sería muy grande y por tanto intratable de forma práctica. Por eso no sería posible obtener un modelo global del sistema.
Verificar una propiedad en estos sistemas se consigue construyendo una prueba de la misma dentro del un sistema de cálculo de un sistema formal. Del éxito o no de conseguir dicha prueba diremos que el sistema satisface o no la propiedad.

Bloque 5 : Lógicas modales

Por último veremos como, a partir de la lógica temporales, aparecieron otro tipo de lógicas con una semántica distinta que nos permitirán modelizar otros aspectos de interés de un sistema.

METODOLOGÍA

La docencia de esta asignatura se impartirá a distancia, siguiendo el modelo educativo propio de la UNED adaptado al EEES. El principal instrumento docente será un Curso Virtual dentro de la plataforma educativas para la enseñanza a distancia. Así mismo se complementa la docencia con la asistencia personalizada del equipo docente.
Con el fin de alcanzar los resultados de aprendizaje se han diseñado las siguientes actividades formativas:

•Actividades de contenido teórico: comprenden todo trabajo relativo a la lectura y su comprensión de los temas y material que vienen en las guías.
A través de estos contenidos teóricos se pretende que el alumno llegue a conocer los distintos sistemas lógicos que se utilizan para expresar diferentes problemas: lógica clásica, temporal, modal y lógicas de verificación de programas.

•Actividades de contenido práctico:

•Trabajo personal: al final de cada capítulo, se proponen actividades prácticas que se deben realizar mediante alguna herramienta software que implementa la técnica de verificación vista en el tema. En concreto los alumnos deben realizar los ejercicios en los siguientes entornos: Alloy, NuSMV.

•Trabajo colectivo o público: se entenderán aquellas actividades que desarrollará el alumno en el entorno virtual del curso como la participación en grupos de trabajo, el intercambio de información con otros compañeros sobre aspectos prácticos; así como la participación, argumentación y aportación constructiva en los debates en foros del curso.

Durante la realización de estas actividades el alumno entenderá el importante papel de la especificación de un problema, así como, comprender la necesidad de utilizar métodos formales de especificación en algunos supuestos o en problemas que así lo requieran.

•Trabajo autónomo: la presentación de memorias de cada práctica que realice el alumno, conllevará actividades asociadas como la búsqueda de información adicional en otros canales, la resolución eficiente y original al problema propuesto en el enunciado.

Para llevar a término todas estas actividades formativas dentro del curso virtual el alumnado dispondrá de:

•Plan de Trabajo: Información complementaria del curso, donde se establecen los objetivos concretos y los puntos de interés.
•Calendario: donde se establece una planificación temporal de las actividades. Así mismo se sugiere el Calendario de Estudio para que el estudiante se adapte a él y mantener un flujo continuo en todo el proceso de aprendizaje.
•Materiales**:**
•Documentos de interés.
•Comunicación:
•Correo electrónico para comunicaciones individuales.
•Foros de Debate donde se intercambian conocimientos y se resuelven dudas de tipo académico general.

SISTEMA DE EVALUACIÓN

TIPO DE PRIMERA PRUEBA PRESENCIAL

Tipo de examen — No hay prueba presencial

TIPO DE SEGUNDA PRUEBA PRESENCIAL

Tipo de examen2 — No hay prueba presencial

CARACTERÍSTICAS DE LA PRUEBA PRESENCIAL Y/O LOS TRABAJOS

Requiere Presencialidad — No

Descripción

Cada Activiad consiste en la realización y entrega de un trabajo a través del curso virual.

Las siguientes actividades se consideran de caracter obligatorio y se denotan como Actividad Calificable.

Actividad Calificable (AC1): Modelado del sistema a partir de requisitos no formales. Se trata de recordar la notación no formal de los DTE. Se proponen una serie de sistemas con unos requisitos funcionales y el alumno debe crear un primer modelo jerárquico del sistema utilizando alguna notación tipo DTE.

Actividad Calificable (AC2): La lógica como lenguaje de especificación. Se trata de saber expresar enunciados sencillos mediante expresiones de lógica clásica. Esto incluye la lógica proposicional y la predicativa. La instalación y uso de sencillos ejemplos con la herramienta Alloy.

Actividad Calificable (AC3): Especificación de sistemas temporales. Se tratará de expresar requisitos en algún sistema de verificación formal y saber interpretar el resultado.

Actividad Calificable (AC4): Verificación de programas. Entender el mecanismo de la verificación formal de requisitos. Consiste en la entrega de un trabajo con varios ejercicios. Se utiliza la notación de Hoare.

Existe un trabajo optativo referenciado como Actividad No Calificable

Actividad No Calificable (ANC): Lógica Modal. Se trata de comprender la sintaxis básica de estas lógicas.

Criterios de evaluación

AC1: Modelado del sistema a partir de requisitos no formales. Se evalúa la capacidad de abstracción para comprender un problema y utilizar una notación capaz de modelar el sistema acorde con la especificación dada.

AC2: La lógica como lenguaje de especificación. Se evaluará la destreza en el uso de las expresiones lógicas para transmitir y reflejar ciertas características o restricciones del modelo. Se evaluará también el uso del programa Alloy.

AC3: Especificación de sistemas temporales. Continuando con el estudio de la lógica se evaluará la capacidad y destreza del alumno para modelar mediante lógicas temporales ciertos sistemas reactivos. El uso y destreza de verificadores formales como NuSMV

AC4: Verificación de programas. Se evaluará la destreza para traducir especificaciones al formato de Hoare y saber realizar pruebas de correcciones parciales.

ANC: Lógica modal. Se pretende que el alumno adquiera destreza en la descripción de un sistema multi agente. Ello requiere saber utilizar la notación de lógicas modales.

Como criterios generales y comunes a todos los trabajos se tendrá en cuenta:

- La capacidad para argumentar y transmitir adecuadamente las ideas y conceptos de la asignatura.

- La participación activa en los foros de discusión del curso.

- La ética y honestidad a la hora de presentar trabajos originales con aporte de ideas propias que mejoren o generalicen las soluciones presentadas.

Ponderación de la prueba presencial y/o los trabajos en la nota final	Cada Actividad Calificable representa un 20% de la Calificación Final. Los trabajos optativos y la participación en los foros representa el 20% del total de la Calificación Final.
Fecha aproximada de entrega	Durante el curso hasta 15/06/2024
Comentarios y observaciones	

Si falta alguna Actividad Calificable en la convocatoria de Junio no se podrá calificar como apto en esta convocatoria.

No obstante, los trabajos que falten, o hayan sido evaluados de forma insuficiente, se podrán entregar para su evaluación en la convocatoria Extraordinaria de Septiembre.

La calificación de los trabajos aprobados se mantiene en la convocatoria extraordinaria.

PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC?	No
Descripción	
Criterios de evaluación	
Ponderación de la PEC en la nota final	
Fecha aproximada de entrega	
Comentarios y observaciones	

OTRAS ACTIVIDADES EVALUABLES

¿Hay otra/s actividad/es evaluable/s?	No
Descripción	
Criterios de evaluación	
Ponderación en la nota final	
Fecha aproximada de entrega	
Comentarios y observaciones	

¿CÓMO SE OBTIENE LA NOTA FINAL?

la Nota Final se obtiene de Actividades Calificables (80%) más la nota proveniente de los trabajos optativos y la particiapación en los foros (20%) .

BIBLIOGRAFÍA BÁSICA

ISBN(13): 9780521543101

Título: LOGIC IN COMPUTER SCIENCE (MODELLING AND REASONING ABOUT SYSTEMS) (2nd edition) edición

Autor/es: Mark Ryan; Michael Huth

Editorial: CAMBRIDGE UNIVERSITY PRESS..

El texto báse, *Logic in Computer Science* se utilizará como el material teórico de la asignatura. En concreto se se utilizarán los siguientes temas:

•Capitulo 1: Propositional Logic. Se estudiará todo menos el apartado 1.6 SAT solvers

•Capítulo 2: Predicate Logic. Entra todo.

•Capítulo 3: Verification by Model Checking. Todo excepto los apartados 3.6 y 3.7

•Capítulo 4: Program Verification. Todo menos el apartado 4.5

•Capítulo 5: Modal Logic an Agents. Se estdiarán solo los apartados: 5.1, 5.2, 5.3

Los fundamentos de Alloy y algunos ejemplos están tratados en el apartado 2.7- Micromodels of software.

El entorno NuSMV se describe así mismo en el apartado 3.3- Model checking: systems, tools and properties.

BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13): 9780262017152
Título: SOFTWARE ABSTRACTIONS: LOGIC, LANGUAGE, AND ANLYSIS 2 edición
Autor/es: Daniel Jackson
Editorial: : MIT PRESS

ISBN(13): 9780857292766
Título: SPECIFICATION OF SOFTWARE SYSTEMS 2011 edición
Autor/es: V Alagar
Editorial: Springer

El programa Alloy está extensamente explicado en *Software Abstractions* (2 ed.) de D. Jackson. Se puede consultar para ver el funcionamiento de algunos ejemplos o para profundizar más en este entorno de creación de micro-modelos de la lógica de predicados.

El libro de Alagar y Periyasamy *Specifications of Software Systems* es un estudio más extenso de la materia tratada en la asignatura. Estudia muchos de los principales sistemas de especificación que existen, así como también recoge un gran número de notaciones e implematnaciones para cada uno de estos sistemas. En concreto estudia la notación Z ampliamente extendida para la especificación de sistemas basados en el modelo. Otro aspecto a tener en cuenta es que el libro sigue otro tipo de notación (cálculo) para la lógica temporal.

RECURSOS DE APOYO Y WEBGRAFÍA

Una de las herramientas con las que más se trabaja en esta asignatura es Alloy. Desde su página se pueden encontrar ayudas y consejos:
https://alloy.mit.edu/alloy/index.html
Otra página interesante es:
https://www.doc.ic.ac.uk/project/examples/2007/271j/suprema_on_alloy/Web

IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.