



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALS EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)
Asignatura: 609024 - Teoría de lenguajes de programación	Abrev: TL	6 ECTS
Asignatura en Inglés: Theory of Programming Languages	Carácter: Obligatoria	
Materia: Métodos Formales Fundamentales		18 ECTS
Otras asignaturas en la misma materia: Análisis estático de programas y resolución de restricciones Modelos de la concurrencia		6 ECTS 6 ECTS
Módulo: Métodos Formales Fundamentales		
Departamento: Sistemas Informáticos y Computación		Coordinador: Riesco Rodríguez, Adrián

Descripción de contenidos mínimos:

Lambda cálculo
Sistemas de tipos, cálculos de secuentes, reglas de deducción
Máquinas abstractas de reducción
Reescritura
Semánticas de lenguajes: axiomática, operacional, denotacional

Lambda calculus
Type systems, sequent calculus, deduction rules
Abstract reduction machines
Rewriting
Semantics of languages: axiomatic, operational, denotational

Programa detallado:

1. Introducción y preeliminarios
2. Reducción y reescritura
 - Sistemas abstractos de reducción
 - Sistemas de reescritura
3. Semántica de lenguajes
 - Semántica operacional
 - Teoría de dominios y semántica denotacional
4. Lambda cálculo
 - Lambda cálculo sin tipos
 - Lambda cálculo con tipos simples
5. Sistemas de tipos
 - Deducción natural e isomorfismo de Curry-Howard
 - Polimorfismo
 - Recursión
 - Subtipado

Programa detallado en inglés:

1. Introduction and review of background.
2. Reduction and rewriting
 - Abstract reduction systems
 - Term rewriting systems
3. Semantics
 - Operational semantics
 - Domains and denotational semantics
4. Lambda calculi
 - The untyped lambda calculus
 - The simply typed lambda calculus
5. Type systems
 - Natural deduction and the Curry-Howard isomorphism
 - Polymorphism
 - Recursion
 - Subtyping

Competencias de la asignatura:**Generales:**

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

- CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.
- CE_MF6-Capacidad para utilizar modelos de cómputo alternativos a los convencionales, tales como la computación cuántica, los sistemas de reescritura, las máquinas abstractas, la programación con restricciones o los algoritmos bio-inspirados
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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
- CB_MF7-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
- CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
- CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

- Capacidad para comprender los sistemas de tipos de diferentes lenguajes de programación
- Capacidad para diseñar un sistema de tipos
- Capacidad para expresar formalmente las diferentes semánticas de un lenguaje de programación
- Capacidad para deducir las condiciones de verificación que ha de satisfacer un programa
- Capacidad para diseñar máquinas abstractas
-
- Ability to understand type systems from different programming languages
- Ability to design type systems
- Ability to formally express different semantics of a programming language
- Ability to deduce verifying conditions that programs must satisfy
- Ability to design abstract machines

Evaluación:

- Exámenes presenciales 30- 70%
- Entrega de trabajos individuales o colaborativos 0- 40%
- Participación en las clases prácticas 0-40%

Evaluación detallada:

In the regular assessment period the assessment method will consist of two parts [A] and [B]. Students shall hand assignments or projects throughout the course.

- [A] Continuous evaluation.
40% - Assignments and small projects.
20% - participation in practical lectures

Fecha: ____ de _____ de _____

Firma del Director del Departamento:



[B] Exam.
40% - In-class exam.

In the extraordinary assessment sitting, students can only be graded on the basis of the in-class exam (method [B]).

Actividades formativas:

Clases teóricas , presencialidad 100%
Clases prácticas: de problemas, de laboratorio, presencialidad 100%
Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
Actividades de evaluación: exámenes presenciales, presencialidad 100%
Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

Reparto de créditos:
Teoría: 3,00
Problemas: 1,50
Laboratorios: 1,50

Otras actividades:
Face-to-face lectures.
Guided exercises and brief case studies.
Discussion of case studies.
Individual assignments and projects.
Tutorials.

Clases de teoría.
Resolución guiada de ejercicios y casos de estudio.
Discusión de casos de estudio.
Ejercicios y prácticas individuales.
Tutorías.

Bibliografía:

Hanne Riis Nielson, Flemming Nielson. Semantics with Applications : An Appetizer; Undergraduate Topics in Computer Science; Springer: New York, 2007.

Glynn Winskel. The Formal Semantics of Programming Languages : An Introduction, 4th.ed. ed.; Foundations of Computing; MIT Press: Cambridge, Mas, 1997.

Richard Ford, K. Rustan M. Leino. Dafny Reference Manual. 2014.

Franz Baader, Tobias Nipkow. Term rewriting and all that. Cambridge University Press, 1999.

Manuel Clavel et al. All about Maude - A high performance logical framework. Lecture Notes in Computer Science 4350. Springer, 2007.

Henk P. Barendregt. The Lambda Calculus: Its Syntax and Semantics. 6th imprinting. Studies in Logic, V. 40. London: College Publications, 2012.

Foundations for Programming Languages. John C. Mitchell. The MIT Press.

Handbook of Theoretical Computer Science. Volume B: Formal Models and Semantics. Janvan Leeuwen (ed.) The MIT Press

Term Rewriting Systems. Terese. Cambridge Tracts in Theoretical Computer Science.

Basic Simple Type Theory. Roger Hindley. Cambridge Tracts in Theoretical Computer Science.

Proofs and Types. Girard, Lafont and Taylor. Cambridge Tracts in Theoretical Computer Science.

Types and Programming Languages. Benjamin Pierce. The MIT Press

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALS EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)
Asignatura: 609025 - Análisis estático de programas y resolución de restricciones	Abrev: AERR Carácter: Obligatoria	6 ECTS
Asignatura en Inglés: Static analysis of programs and constraint solving		
Materia: Métodos Formales Fundamentales		18 ECTS
Otras asignaturas en la misma materia:		
Modelos de la concurrencia		6 ECTS
Teoría de lenguajes de programación		6 ECTS
Módulo: Métodos Formales Fundamentales		
Departamento: Sistemas Informáticos y Computación		Coordinador: Montenegro Montes, Manuel

Descripción de contenidos mínimos:

Interpretación abstracta
Dominios poliédricos
Análisis basados en tipos
Resolutores SAT y SMT
Programación con restricciones
Problemas de satisfacción y de optimización
Resolutores de restricciones

Abstract interpretation
Polyedric domains
Analyses based on types
SAT and SMT solvers
Constraint programming
Satisfaction and optimization problems
Constraint solvers

Programa detallado:

1. Introducción al análisis estático de programas.
2. Análisis basados en tipos: tipado de expresiones aritméticas, programas orientados a objetos, referencias, subtipado estructural y nominal, seguridad de tipos, sistemas de tipos y efectos.
3. Interpretación abstracta: conexiones de Galois, teoría de puntos fijos, operadores de ensanchamiento y estrechamiento.
4. Dominios abstractos: análisis de signo, intervalos y dominios poliédricos.
5. Resolutores SAT y SMT: propagación de restricciones booleanas, DPLL(T), lógica con igualdad, funciones no interpretadas, teoría de arrays.
6. Programación con restricciones.
7. Problemas de satisfactibilidad y optimización.
8. Resolución de restricciones.

Programa detallado en inglés:

1. Introduction to static program analysis.
2. Type-based analysis: typed arithmetic expressions and typed object-oriented programs, references, structural and nominal subtyping, type safety, type and effect systems.
3. Abstract interpretation: Galois connections, fixed point theory, widening and narrowing.
4. Abstract domains: sign analysis, intervals, polyhedral domains.
5. SAT and SMT solvers: boolean constraint propagation, DPLL(T), equality logic, uninterpreted functions, array theory.
6. Constraint programming.
7. Satisfaction and optimization problems.
8. Constraint solvers.

Competencias de la asignatura:**Generales:**

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
- CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



- CE_MF2-Capacidad para utilizar de forma competente las herramientas existentes de demostración automática y asistida de teoremas y de propiedades matemáticas.
- CE_MF4-Capacidad para utilizar y desarrollar herramientas que analizan automáticamente propiedades de los programas, utilizando tan solo el texto fuente de los mismos.
- CE_MF6-Capacidad para utilizar modelos de cómputo alternativos a los convencionales, tales como la computación cuántica, los sistemas de reescritura, las máquinas abstractas, la programación con restricciones o los algoritmos bio-inspirados
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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
- CB_MF7-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, y a sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
- CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
- CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

- Capacidad para diseñar análisis estáticos basados en interpretación abstracta
- Capacidad para diseñar análisis estáticos basados en tipos
- Capacidad para expresar un problema como un conjunto de restricciones
-
- Ability to design static analyses based on abstract interpretation
- Ability to design static analyses based on types
- Ability to express problems as a set of constraints

Evaluación:

- Exámenes presenciales 30- 70%
- Entrega de trabajos individuales o colaborativos 0- 40%
- Participación en las clases prácticas 0-40%

Evaluación detallada:

Students will be graded on the basis of class assignments and projects that will be proposed throughout the course. Students are expected to hand in two individual projects in order to obtain a passing grade. The final grade will be computed as follows:

- [65%] Assignments and a project on static analysis and SAT/SMT solvers. Students are expected to design and implement a given analysis and prove its correctness. Students might be required to present their work in class.
- [35%] Assignments and a project on constraint solving and optimization. Students are expected to express a given problem as a set of constraints, apply the necessary tools for solving it, and assess the results. Students might be required to present their work in class.

A face-to-face exam will take place at the end of the course. It consists in presenting and defending the work done throughout the course. For each of the two modules described above, 30% of the grade will be given on the basis of these presentations.

In the extraordinary assessment sitting, a new deadline will be set for those students that obtained a failing grade in the regular assessment period. Students will be required to (re)submit any assignments and projects that failed to obtain a passing grade.

Actividades formativas:

- Clases teóricas, presencialidad 100%
- Clases prácticas: de problemas, de laboratorio, presencialidad 100%

Fecha: ____ de _____ de _____

Firma del Director del Departamento:



Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
Actividades de evaluación: exámenes presenciales, presencialidad 100%
Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

Reparto de créditos:

Teoría: 3,00
Problemas: 1,50
Laboratorios: 1,50

Otras actividades:

Clases presenciales.
Resolución guiada de ejercicios y casos de estudio.
Discusión de casos de estudio.
Ejercicios y prácticas individuales.
Tutorías.

Face-to-face lectures.
Guided exercises and brief case studies.
Discussion of case studies.
Individual exercises.
Tutorials.

Bibliografía:

- Benjamin C. Pierce. Types and Programming Languages. Cambridge, Mass.; MIT Press, 2002.
- Flemming Nielson, Hanne Riis Nielson, and Chris Hankin. Principles of Program Analysis. Berlin: Springer, 2005.
- Daniel Kroenig, Ofer Strichman. Decision Procedures: An Algorithmic Point of View (2nd ed.). Texts in Theoretical Computer Science. Springer, 2016.
- Richard L. Ford, K. Rustan M. Leino. Dafny Reference Manual. 2017.
- Francesca Rossi, Peter van Beek, Toby Walsh. Handbook of Constraint Programming. Elsevier Science, 2006.
- Krzysztof Apt. Principles of Constraint Programming. Cambridge Press, 2003.
- Kimbal Marriott, Peter Stuckey. Programming with Constraints: An Introduction. MIT Press, 1998.
- Slim Abdennadher, Thom Frühwirth. Essentials of Constraint Programming. Springer, 2003.

Ficha docente guardada por última vez el 25/09/2018 10:11:00 por el usuario: Coordinador MMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)
Asignatura: 609026 - Modelos de la concurrencia	Abrev: MC	6 ECTS
Asignatura en Inglés: Concurrency models	Carácter: Obligatoria	
Materia: Métodos Formales Fundamentales		18 ECTS
Otras asignaturas en la misma materia:		
Análisis estático de programas y resolución de restricciones		6 ECTS
Teoría de lenguajes de programación		6 ECTS
Módulo: Métodos Formales Fundamentales		
Departamento: Sistemas Informáticos y Computación		Coordinador: Frutos Escrig, David de

Descripción de contenidos mínimos:

Álgebras de procesos
Relaciones de bisimulación
Lógica temporal
Model checking
Redes de Petri

Process algebras
Bisimulation relations
Temporal logics
Model checking
Petri nets

Programa detallado:

Álgebra de procesos.
Semánticas operacional, denotacional, axiomática.
Bisimulación. Equivalencias semánticas y órdenes.
Lógicas
Herramientas para la modelización, análisis y verificación de propiedades (model checking): Mcl2, Concurrency WorkBench (CAAL)
Redes de Petri
Clases de redes.
Propiedades básicas y técnicas para su análisis
Introducción a las herramientas basadas en redes de Petri para la modelización y análisis de sistemas concurrentes

Programa detallado en inglés:

Process algebras
Operational, denotational and axiomatic semantics
Bisimulation. Semantic equivalences and orders
Logics: specifying and checking properties of systems
Introduction to modelling, analysis and verification tools (model checking): Mcl2, Concurrency WorkBench (CAAL)
Petri nets
Classes of nets
Basic properties and analysis techniques
Petri net tools for modelling and analysis of concurrent systems

Competencias de la asignatura:**Generales:**

CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



CE_MF3-Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.

CE_MF9-Capacidad para realizar un trabajo individual que recoja la integración de conocimientos adquiridos en la totalidad del máster y capacidad para defenderlo públicamente ante un tribunal

CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

CB_MF6-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

CB_MF7-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

CB_MF10-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.

CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.

CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.

CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

Capacidad para modelizar un problema concurrente como un álgebra de procesos

Capacidad para demostrar formalmente la equivalencia de dos procesos

Capacidad para expresar formalmente las propiedades temporales que ha de satisfacer un sistema concurrente

Capacidad para utilizar competentemente una herramienta de model checking

Capacidad para modelizar un problema concurrente como una red de Petri

Ability to model a concurrent program as a process algebra

Ability to formally prove the equivalence of two processes

Ability to formally express the temporal properties that a concurrent system must satisfy

Ability to competently use tools for model checking

Ability to model a concurrent problem as a Petri net

Evaluación:

Exams (30%-70%)

Assignments (0%-40%)

Participation during the course (0%-40%)

Evaluación detallada:

Regular examination: Exams 40%; Assignments 40%; Participation during the course 20%.

Extraordinary examination: Exam 40%; Assignments 40%; Participation during the course 20%

In the extraordinary assessment sitting, a new deadline will be set for those students that obtained a failing grade in the regular assessment period. Students will be required to (re)submit any assignments that failed to obtain a passing grade. The mark corresponding to participation in the extraordinary examination will be equal to the mark obtained for the regular examination.

Actividades formativas:

Clases teóricas , presencialidad 100%

Clases prácticas: de problemas, de laboratorio, presencialidad 100%

Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros

Actividades de evaluación: exámenes presenciales, presencialidad 100%

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Estudio personal de material básico y complementario de la asignatura	
Actividades docentes:	
Reparto de créditos:	Otras actividades:
Teoría: 6,00	No tiene
Problemas: 0,00	
Laboratorios: 0,00	
Bibliografía:	
L. Aceto, A. Ingólfssdóttir, K. Larsen, and J. Srba. Reactive Systems: Modelling, Specification and Verification. Cambridge University Press, 2007.	
J.F. Groote and M.R. Mousavi. Modeling and analysis of communicating systems. The MIT press. 2014.	
J. Andersen et al. CAAL: Concurrency Workbench, Aalborg Edition, Theoretical Aspects of Computing - ICTAC 2015, pp. 573--582	
J. Desel, W. Reisig, and G. Rozenberg (Eds.). Advances in Petri Nets, Lecture Notes in Computer Science, vol. 3098, Springer-Verlag, 2004.	
W. Reisig. Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies. Springer 2013.	

Ficha docente guardada por última vez el 25/09/2018 10:11:00 por el usuario: Coordinador MMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (2C)
Asignatura: 609027 - Verificación asistida de programas	Abrev: VA	6 ECTS
Asignatura en Inglés: Computer-aided Program Verification	Carácter: Optativa	
Materia: Análisis de la Corrección de los Sistemas		18 ECTS
Otras asignaturas en la misma materia: Análisis de sistemas concurrentes y distribuidos Métodos formales de testing		6 ECTS 6 ECTS
Módulo: Métodos Formales Complementarios		
Departamento: Sistemas Informáticos y Computación		Coordinador: Peña Marí, Ricardo

Descripción de contenidos mínimos:

Asistentes de demostración
Análisis de terminación de programas
Plataformas de verificación asistida
Sistemas de inferencia de tipos cualificados
Casos de estudio

Proof assistants
Analysis of program termination
Assisted platforms for verification
Inference systems for qualified types
Case studies

Programa detallado:

1. La plataforma Dafny de verificación asistida
 - 1.1 El lenguaje de programación Dafny
 - 1.2 Verificación de programas con vectores
 - 1.3 Tipos algebraicos y estructuras de datos
 - 1.4 Marcos dinámicos
2. Programación con tipos refinados: Liquid Haskell
 - 2.1 Introducción a los tipos refinados
 - 2.2 Polimorfismo
 - 2.3 Tipos algebraicos refinados
 - 2.4 Medidas
3. Análisis de terminación
4. Introducción a los asistentes de demostración

Programa detallado en inglés:

1. The Dafny verification platform:
 - 1.1. The basics of the Dafny language
 - 1.2. Verification of programs on arrays
 - 1.3. Algebraic data types and data structures
 - 1.4. Dynamic frames
2. Programming with refinement types: Liquid Haskell
 - 2.1. Introduction to refinement types
 - 2.2. Polymorphism
 - 2.3. Refined datatypes
 - 2.4. Measures
3. Termination analysis
4. Introduction to proof assistants

Competencias de la asignatura:**Generales:**

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
- CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Específicas:

- CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.
- CE_MF2-Capacidad para utilizar de forma competente las herramientas existentes de demostración automática y asistida de teoremas y de propiedades matemáticas.
- CE_MF3-Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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
- CB_MF7-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
- CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
- CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

- Capacidad para utilizar competentemente herramientas de demostración asistida
- Capacidad para diseñar análisis de terminación de programas
- Capacidad para verificar formalmente programas utilizando plataformas de verificación asistida
- Capacidad para especificar programas utilizando tipos cualificados
- Capacidad para utilizar las herramientas de inferencia de tipos cualificados
-
- Ability to competently use proof assistants
- Ability to design analyses of program termination
- Ability to formally verify program using assisted platforms for verification
- Ability to specify program using qualified types
- Ability to use tools for inferring qualified types

Evaluación:

- Exámenes presenciales 30%-70%
- Entrega de trabajos individuales o colaborativos 0% -30%
- Participación en las clases prácticas 0%-30%
- Evaluación de prácticas desarrolladas por el alumno 0%-30%

Evaluación detallada:

- . During the course, the student must prepare practical or/and theoretical assignments. The mark obtained here will be kept up to the extraordinary examination and there will not be extra assignments after the ordinary examination. (50%)
- . There will be an ordinary examination and an extraordinary one (50%)

Actividades formativas:

- Clases teóricas, presencialidad 100%
- Clases prácticas: de problemas, de laboratorio, presencialidad 100%

Fecha: ____ de _____ de _____

Firma del Director del Departamento:



Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
Actividades de evaluación: exámenes presenciales, presencialidad 100%
Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

Reparto de créditos:	Otras actividades:
Teoría: 6,00	No tiene
Problemas: 0,00	
Laboratorios: 0,00	

Bibliografía:

- K. Rustan M. Leino. Dafny: An Automatic Program Verifier for Functional Correctness. In LPAR-16, volume 6355 of LNCS, pages 348-370. Springer, 2010.
- K. Rustan M. Leino. Specification and verification of object-oriented software. In Engineering Methods and Tools for Software Safety and Security, volume 22 of NATO Science for Peace and Security Series D: Information and Communication Security, pages 231-266. IOS Press, 2009
- Online Dafny tutorial, <https://rise4fun.com/Dafny/tutorial/guide>
- Dafny reference manual, <https://github.com/Microsoft/dafny/blob/master/Docs/DafnyRef/out/DafnyRef.pdf>, 2018.
- P. M. Rondon, M. Kawaguchi, R. Jhala. Liquid types. In Proceedings of the 29th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI'08, pages 159-16, 2008.
- M. Kawaguchi, P. M. Rondon, R. Jhala. Type-based data structure verification. In Proceedings of the 30th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI'09, pages 304-315.
- N. Vazou, E.L. Seidel, R. Jhala, S. Peyton-Jones. Refinement Types for Haskell. In Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming, ICFP'14, pages 269-282, 2014.
- N. Vazou, E.L. Seidel, R. Jhala. LiquidHaskell: experience with refinement types in the real world. In Proceedings of the 2014 ACM SIGPLAN Symposium on Haskell, Haskell'14, pages 39-51, 2014.
- N. Vazou, E.L. Seidel, R. Jhala. Abstract refinement types. In 22nd European conference on Programming Languages and Systems, ESOP'13, pages 209-228, 2013.
- Online Liquid Haskell tutorial, <http://ucsd-progsys.github.io/lh-workshop/>.
- Online Proving Theorems in Lean, https://leanprover.github.io/theorem_proving_in_lean/index.html

Ficha docente guardada por última vez el 25/09/2018 10:12:00 por el usuario: Coordinador MMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)	
Asignatura: 609028 - Métodos formales de testing		Abrev: MFT	6 ECTS
Asignatura en Inglés: Formal methods in testing		Carácter: Optativa	
Materia: Análisis de la Corrección de los Sistemas		18 ECTS	
Otras asignaturas en la misma materia: Análisis de sistemas concurrentes y distribuidos Verificación asistida de programas		6 ECTS 6 ECTS	
Módulo: Métodos Formales Complementarios			
Departamento: Sistemas Informáticos y Computación		Coordinador: García Merayo, M ^a . de las Mercedes	

Descripción de contenidos mínimos:

Testing formal de máquinas de estados finitos
Relaciones de conformidad
Combinando testing y model checking
Testing formal en entornos distribuidos y en la nube
Herramientas para realizar testing formal

Formal testing of finite state machines
Conformance relations
Combining testing and model checking
Formal testing of distributed and cloud environments
Tools for formal testing

Programa detallado:

1. Introducción al testing de software.
2. Introducción a los métodos formales de testing.
3. Testing de sistemas basados en estados.
4. Testing de sistemas temporizados y probabilísticos.
5. Testing de sistemas distribuidos y asíncronos.

Programa detallado en inglés:

1. Introduction to software testing.
2. Introduction to formal methods in testing.
3. Testing from state-based systems.
4. Testing timed and probabilistic systems.
5. Testing distributed and asynchronous systems.

Competencias de la asignatura:**Generales:**

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
- CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

- CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.
- CE_MF3-Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.
- CE_MF5-Capacidad para utilizar y desarrollar herramientas que analizan propiedades de los programas, mediante su ejecución en un conjunto de casos cuidadosamente seleccionado.
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



- CB_MF7-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
- CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
- CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

- Capacidad para especificar propiedades de testing de sistemas concurrentes
- Capacidad para utilizar competentemente herramientas de verificación de propiedades de testing
- Capacidad para utilizar competentemente herramientas de verificación de propiedades temporales mediante model checking
- Capacidad para especificar propiedades de testing de sistemas distribuidos
-
- Ability to specify testing properties for concurrent systems
- Ability to competently use tools for verifying testing properties
- Ability to competently use tools for verifying temporal properties using model checking
- Ability to specify testing properties for distributed systems

Evaluación:

- Exámenes presenciales 30- 70%
- Entrega de trabajos individuales o colaborativos 0- 40%
- Participación en las clases prácticas 0-40%

Evaluación detallada:

Students have to present at least two research papers, in a fixed date (during the time when the subject is taught, it will be fixed with enough time and it will be announced both during the lectures and in the virtual campus). They will receive a mark between 0 and 10 points. If a student does not present the paper in the assigned date, then the mark corresponding to this presentation, both in the ordinary and extraordinary examination, is equal to zero. All the students must attend the lectures when other students are presenting a paper.

Students will be encouraged to participate in the regular lectures: answering questions and assignments posed by the lecturers.

Regular examination: Presentations 90%; Participation during the lectures 10%.

Extraordinary examination: Exam 50%; Presentations 40%; Participation during the lectures 10%.

The marks corresponding to presentations and participation in the extraordinary examination will be equal to the marks obtained for the regular examination.

Actividades formativas:

- Clases teóricas, presencialidad 100%
- Clases prácticas: de problemas, de laboratorio, presencialidad 100%
- Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
- Actividades de evaluación: exámenes presenciales, presencialidad 100%
- Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

- | | |
|----------------------|--------------------|
| Reparto de créditos: | Otras actividades: |
| Teoría: 6,00 | No tiene |
| Problemas: 0,00 | |
| Laboratorios: 0,00 | |

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Bibliografía:

Manfred Broy, Bengt Jonsson, Joost-Pieter Katoen, Martin Leucker, Alexander Pretschner: Model-Based Testing of Reactive Systems. Lecture Notes in Computer Science 3472, Springer 2005.

Robert M. Hierons, Jonathan P. Bowen, Mark Harman: Formal Methods and Testing, An Outcome of the FORTEST Network, Revised Selected Papers. Lecture Notes in Computer Science 4949, Springer 2008.

David Lee, Mihalis Yannakakis. Principles and methods of testing finite state machines - a survey. Proceedings of the IEEE 84 (8), 1090-1123, 1996.

Ficha docente guardada por última vez el 25/09/2018 10:12:00 por el usuario: Coordinador MMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (2C)
Asignatura: 609029 - Análisis de sistemas concurrentes y distribuidos	Abrev: ASCD	6 ECTS
Asignatura en Inglés: Analysis of Concurrent and Distributed Systems	Carácter: Optativa	
Materia: Análisis de la Corrección de los Sistemas		18 ECTS
Otras asignaturas en la misma materia: Métodos formales de testing Verificación asistida de programas		6 ECTS 6 ECTS
Módulo: Métodos Formales Complementarios		
Departamento: Sistemas Informáticos y Computación		Coordinador: Correas Fernández, Jesús

Descripción de contenidos mínimos:

Semántica de programas concurrentes y distribuidos
Propiedades básicas: terminación y consumo finito de recursos
Propiedades de vitalidad: ausencia de bloqueo e inanición
Verificación basada en análisis estático
Validación basada en testing
Implementación y herramientas existentes

Semantics of concurrent and distributed systems
Basic properties: termination and resource consumption
Liveness properties: absence of locks and starvation
Verification based on static analysis
Validation based on testing
Implementation and existing tools

Programa detallado:

No se imparte en el curso 2018/19

Programa detallado en inglés:

This subject will not be taught in the course 2018/19

Competencias de la asignatura:**Generales:**

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
- CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

- CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.
- CE_MF3-Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.
- CE_MF4-Capacidad para utilizar y desarrollar herramientas que analizan automáticamente propiedades de los programas, utilizando tan solo el texto fuente de los mismos.
- CE_MF5-Capacidad para utilizar y desarrollar herramientas que analizan propiedades de los programas, mediante su ejecución en un conjunto de casos cuidadosamente seleccionado.
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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
- CB_MF7-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

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



CB_MF10-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.

CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.

CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.

CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

Capacidad para especificar la semántica formal de un sistema distribuido

Capacidad para especificar propiedades de seguridad y vitalidad de sistemas distribuidos

Capacidad para desarrollar análisis estáticos de sistemas distribuidos

Capacidad para utilizar competentemente herramientas de análisis y validación de sistemas distribuidos

Ability to specify the formal semantics of a distributed system

Ability to specify safety and liveness properties of distributed systems

Ability to develop static analysis for distributed systems

Ability to competently use tools for analyzing and validating distributed systems

Evaluación:

Exámenes presenciales 30%-70%

Entrega de trabajos individuales o colaborativos 0%-30%

Participación en las clases prácticas 0%-30%

Evaluación de prácticas desarrolladas por el alumno 0%-30%

Evaluación detallada:

Actividades formativas:

Clases teóricas, presencialidad 100%

Clases prácticas: de problemas, de laboratorio, presencialidad 100%

Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros

Actividades de evaluación: exámenes presenciales, presencialidad 100%

Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

Reparto de créditos:	Otras actividades:
Teoría: 3,00	No tiene
Problemas: 0,00	
Laboratorios: 3,00	

Bibliografía:

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALS EN INGENIERÍA INFORMÁTICA		Curso: 1º (2C)
Asignatura: 609030 - Diseño de sistemas correctos por construcción	Abrev: DSCC	6 ECTS
Asignatura en Inglés: Design of correct-by-construction systems	Carácter: Optativa	
Materia: Diseño y construcción Rigurosa de Sistemas		18 ECTS
Otras asignaturas en la misma materia:		
Desarrollo formal de software dirigido por modelos		6 ECTS
Diseño de algoritmos bioinspirados		6 ECTS
Módulo: Métodos Formales Complementarios		
Departamento: Sistemas Informáticos y Computación		Coordinador: Peña Marí, Ricardo

Descripción de contenidos mínimos:

Requisitos y refinamiento incremental
Especificación formal de requisitos
Obligaciones de demostración
Desarrollo directo de sistemas secuenciales
Desarrollo con refinamiento de sistemas secuenciales
Desarrollo con refinamiento de sistemas concurrentes
Generación de código

Requirements and incremental refinement
Formal requirement specification
Proof obligations
Direct development of sequential systems
Development with refinement of sequential systems
Development with refinement of concurrent systems
Code generation

Programa detallado:

No se imparte en el curso 2018/19

Programa detallado en inglés:

This subject will not be taught in the course 2018/19

Competencias de la asignatura:**Generales:**

CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.
CE_MF2-Capacidad para utilizar de forma competente las herramientas existentes de demostración automática y asistida de teoremas y de propiedades matemáticas.
CE_MF3-Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.
CE_MF8-Capacidad para la dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, con garantía de la seguridad para las personas y bienes, y la calidad final de los productos.
CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

CB_MF6-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

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



- CB_MF7-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
- CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
- CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

- Capacidad para especificar formalmente los requisitos de un sistema
- Capacidad para refinar incrementalmente los requisitos de un sistema
- Capacidad para establecer las obligaciones de demostración que se deducen de una construcción incremental
- Capacidad de utilizar competentemente herramientas de ayuda al desarrollo formal incremental de sistemas
- Capacidad para aplicar las mismas técnicas a sistemas concurrentes
-
- Ability to formally specify system requirements
- Ability to incrementally refine system requirements
- Ability to establish the proof obligations inferred from an incremental construction
- Ability to use competently computer-aided tools for formal, incremental system development
- Ability to apply the same techniques to concurrent systems

Evaluación:

- Exámenes presenciales 30%-70%
- Entrega de trabajos individuales o colaborativos 0%-40%
- Participación en las clases prácticas 0%-30%
- Evaluación de prácticas desarrolladas por el alumno 0%-40%

Evaluación detallada:

No tiene

Actividades formativas:

- Clases teóricas, presencialidad 100%
- Clases prácticas: de problemas, de laboratorio, presencialidad 100%
- Trabajos desarrollados por los alumnos:
problemas, prácticas y resúmenes de artículos o libros
- Actividades de evaluación: exámenes presenciales, presencialidad 100%
- Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

- | | |
|----------------------|--------------------|
| Reparto de créditos: | Otras actividades: |
| Teoría: 6,00 | No tiene |
| Problemas: 0,00 | |
| Laboratorios: 0,00 | |

Bibliografía:

No tiene

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)
Asignatura: 609031 - Desarrollo formal de software dirigido por modelos	Abrev: DDM	6 ECTS
Asignatura en Inglés: Formal model-driven software development	Carácter: Optativa	
Materia: Diseño y construcción Rigurosa de Sistemas		18 ECTS
Otras asignaturas en la misma materia: Diseño de algoritmos bioinspirados Diseño de sistemas correctos por construcción		6 ECTS 6 ECTS
Módulo: Métodos Formales Complementarios		
Departamento: Sistemas Informáticos y Computación		Coordinador: Peña Marí, Ricardo

Descripción de contenidos mínimos:

Introducción al desarrollo de software dirigido por modelos.
Modelización y meta-modelización, modelización multi-nivel.
Lenguajes de dominio específico: sintaxis y semántica.
Transformación de modelos, reescritura de grafos.
Análisis de modelos. Model finders. Uso de redes de Petri y de model checking

Introduction to model-driven software development
Modeling and meta-modeling, multi-level modeling
Domain specific languages: syntax and semantics
Model transformation, graph rewriting
Model analysis. Model finders. Petri nets and model checking

Programa detallado:

- Introducción al desarrollo dirigido por modelos
 - Conceptos
 - Tecnologías
 - Aplicaciones y escenarios
- Modelización del software y meta-modelizado
 - UML
 - OCL
 - Meta-modelizado
 - Meta-modelizado multinivel
- Lenguajes de modelizado específicos del dominio (DSML)
 - Sintaxis concreta gráfica
 - Sintaxis concreta textual
 - DSMLs en la práctica: xText, Sirius
- Generación de código e Ingeniería inversa
 - Generación de código: transformación modelo-a-texto
 - Generación de código: lenguajes y herramientas
 - Ingeniería inversa y modernización basada en modelos
- Transformaciones de modelos
 - Transformaciones in-place
 - Transformaciones Modelo-a-Modelo
 - Lenguajes y herramientas de transformación en la práctica
 - Transformaciones de grafos

Programa detallado en inglés:

- Introduction to Model-Driven Software Development
 - Concepts
 - Technologies
 - Applications and Scenarios
- Software Modelling and Meta-Modelling
 - UML
 - OCL
 - Meta-modelling

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



d. Multi-level meta-modelling

3. Domain-Specific Modelling Languages

- a. Graphical Concrete Syntax
- b. Textual Concrete Syntax
- c. Defining DSLs in practice: xText, Sirius

4. Code Generation and Reverse Engineering

- a. Code generation principles: Model-to-text transformations
- b. Code generation languages and tools
- c. Reverse Engineering and Model-based modernization

5. Model Transformations

- a. In-place Transformations
- b. Model-to-Model Transformations
- c. Transformation Languages and Tools in practice
- d. Graph Transformation

Competencias de la asignatura:

Generales:

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
- CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

- CE_MF1-Capacidad para expresar los requisitos y propiedades que ha de satisfacer un sistema informático, en una variedad de lenguajes formales y a diferentes niveles de detalle.
- CE_MF2-Capacidad para utilizar de forma competente las herramientas existentes de demostración automática y asistida de teoremas y de propiedades matemáticas.
- CE_MF3-Capacidad para utilizar técnicas y herramientas avanzadas, automáticas y asistidas, para verificar formalmente que un programa o sistema informático satisface las propiedades lógicas previamente especificadas.
- CE_MF8-Capacidad para la dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, con garantía de la seguridad para las personas y bienes, y la calidad final de los productos.
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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
- CB_MF7-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
- CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
- CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

Fecha: ____ de _____ de _____

Firma del Director del Departamento:



Capacidad para aplicar el desarrollo dirigido por modelos a sistemas de software
Capacidad para especificar sistemas en los lenguajes específicos del desarrollo dirigido por modelos
Capacidad para transformar modelos a diferentes niveles de detalle
Capacidad para especificar y verificar propiedades formales de los modelos
Capacidad para utilizar competentemente herramientas de asistencia al desarrollo dirigido por modelos

Ability to apply model-driven development to software systems
Ability to specify systems in model-driven development languages
Ability to transform models to different detail levels
Ability to specify and verify formal properties on models
Ability to use competently computer-aided tools for model-driven development

Evaluación:

Exámenes presenciales 30%-70%
Entrega de trabajos individuales o colaborativos 0%-40%
Participación en las clases prácticas 0%-30%
Evaluación de prácticas desarrolladas por el alumno 0%-40%

Evaluación detallada:

The course involves lectures with (at least) one assignment per unit (but not more than one per week), lab sessions and a practical research project. The assignments are to be solved individually. The research project is to be done in teams of two people, and involves the use of the studied technologies to solve a practical problem (of the student choice or proposed by the professors). The project is to be presented by the members of the team, who should also deliver a short document (in the form of a research paper) and the code of the project.

In the ordinary exam period, the evaluation will be made according to the following scheme:

40 % exercises.
60 % final project

The exam will consist of the public presentation of the project

Actividades formativas:

Clases teóricas, presencialidad 100%
Clases prácticas: de problemas, de laboratorio, presencialidad 100%
Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
Actividades de evaluación: exámenes presenciales, presencialidad 100%
Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

Reparto de créditos:	Otras actividades:
Teoría: 6,00	No tiene
Problemas: 0,00	
Laboratorios: 0,00	

Bibliografía:

Bettini, L. "Implementing Domain-Specific Languages with Xtext and Xtend" (<https://www.packtpub.com/application-development/implementing-domain-specific-languages-xtext-and-xtend>). Packt Publishing (2013).
Brambilla, M., Cabot, J., Wimmer, M. "Model-Driven Software Engineering in Practice". Synthesis Lectures on Software Engineering, Morgan & Claypool Publishers 2012.
Clayberg, E., Rubel, D. "Eclipse Plug-ins". Addison-Wesley Professional, 3rd Edition (2008).
Ehrig, H., Ehrig, K., Prange, U., Taentzer, G. 2006. "Fundamentals of Algebraic Graph Transformation". Springer.
Kelly, S., Tolvanen, J.-P. "Domain-Specific Modeling. Enabling Full Code Generation". IEEE CS, 2008.
Raistrick, C. "Model driven architecture with executable UML". Cambridge University Press. 2004.
Stahl, T., Völter, M. "Model-Driven Software Development". Wiley, 2006
Steinberg, D., Budinsky, F., Paternostro, M., Merks, E. "EMF: Eclipse Modeling Framework". Addison-Wesley Professional, 2nd Edition (2008).
Völter, M. "DSL Engineering - Designing, Implementing and Using Domain-Specific Languages". dslbook.org 2013.

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)
Asignatura: 609032 - Diseño de algoritmos bioinspirados	Abrev: DABI	6 ECTS
Asignatura en Inglés: Design of bio-inspired algorithms	Carácter: Optativa	
Materia: Diseño y construcción Rigurosa de Sistemas		18 ECTS
Otras asignaturas en la misma materia:		
Desarrollo formal de software dirigido por modelos		6 ECTS
Diseño de sistemas correctos por construcción		6 ECTS
Módulo: Métodos Formales Complementarios		
Departamento: Sistemas Informáticos y Computación		Coordinador: Peña Marí, Ricardo

Descripción de contenidos mínimos:

Introducción a los modelos de cómputo bioinspirados

Modelos clásicos: autómatas celulares, gramáticas de derivación paralela, sistemas inspirados en ADN.

Computación evolutiva, desde los algoritmos genéticos a la programación genética con representaciones formales complejas de las poblaciones

Otros modelos de cómputo: basados en membranas, en redes de procesadores que evolucionan, etc.

Introduction to bio-inspired computing models

Classical models: cellular automata, parallel derivation grammars, DNA-inspired systems

Evolutionary computation, from genetic algorithms to genetic programming using genetic programming with complex, formal representations of populations

Other computing models: membrane-based, networks of processors that evolve, etc.

Programa detallado:

IMPORTANTE: No se imparte el curso 2018/19

1. Bio-inspiración para proponer computadores no convencionales

1.1 Mecanismo de cómputo abstracto a partir de fenómenos naturales

1.2 Algunos computadores naturales clásicos (tales como los basados en gramáticas, ADN y células)

1.3 Estado del arte en computadores naturales

2. Computación evolutiva

2.1 Métodos heurísticos y meta-heurísticos modernos

2.2 Introducción a los algoritmos genéticos (GA), estrategias evolutivas, algoritmos evolutivos (EA) y programación genética

2.3 Representaciones. Diseño de operadores e individuos (fenotipos vs genotipos)

2.4 Algoritmos genéticos: operadores de selección, mutación y reemplazamiento

2.5 Teoría básica de GA. Limitaciones de los EA simples

2.6 Representaciones formales complejas de individuos en EA (corrección sintáctica y semántica)

3. Inteligencia de los enjambres

3.1 Introducción a los modelos y métodos de inteligencia de los enjambres

3.2 Optimización basada en colonias de hormigas y enjambres de partículas

Programa detallado en inglés:

IMPORTANT: This subject will not be taught in the course 2018/19

1. Bio-inspiration for proposing natural non conventional computers

1.1 Abstracting computing mechanisms from natural phenomena

1.2 Some classic natural computers (such as those based on grammars, DNA or cells)

1.3 State of the art natural computers

2. Evolutionary Computation

2.1 Modern heuristic and metaheuristic methods

2.2 Introduction to Genetic algorithms, evolution strategies, evolutionary programming, genetic programming

2.3 Representations. Design of operators and individuals (phenotypes vs genotypes).

2.4 Genetic Algorithms: Selection, Replacement, and Mutation Operators

2.5 Basic GA theory. Limitations of simple EAs.

2.6 Complex formal representations of individuals in evolutionary computation (syntactic and semantic correctness)

3. Swarm Intelligence

3.1 Introduction to Swarm intelligence models and methods

3.2 Ant Colony Optimization and Particle Swarm Optimization

Competencias de la asignatura:**Generales:**

CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.

CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

CE_MF6-Capacidad para utilizar modelos de cómputo alternativos a los convencionales, tales como la computación cuántica, los sistemas de reescritura, las máquinas abstractas, la programación con restricciones o los algoritmos bio-inspirados

CE_MF8-Capacidad para la dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, con garantía de la seguridad para las personas y bienes, y la calidad final de los productos.

CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

CB_MF6-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

CB_MF7-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

CB_MF10-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.

CT_MF1-Capacidad para trabajar en equipo, y a sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.

CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.

CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

Capacidad para diseñar algoritmos bioinspirados basados en autómatas celulares

Capacidad para diseñar algoritmos bioinspirados basados en ADN

Capacidad para diseñar algoritmos bioinspirados basados en poblaciones

Capacidad para diseñar algoritmos bioinspirados basados en membranas

Ability to design bio-inspired algorithms based on cellular automata

Ability to design bio-inspired algorithms based on DNA

Ability to design bio-inspired algorithms based on populations

Ability to design bio-inspired algorithms based on membranes

Evaluación:

Exámenes presenciales 30%-70%

Entrega de trabajos individuales o colaborativos 0%-40%

Participación en las clases prácticas 0%-30%

Evaluación de prácticas desarrolladas por el alumno 0%-40%

Evaluación detallada:

Trabajos prácticos para la comprensión de los modelos básicos 30%

Proyecto para la solución de un problema real utilizando un modelo 30%

Pruebas individuales y colaborativas a lo largo de las diferentes sesiones 40%

Actividades formativas:

Clases teóricas, presencialidad 100%

Fecha: ____ de _____ de _____

Firma del Director del Departamento:



Clases prácticas: de problemas, de laboratorio, presencialidad 100%	
Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros	
Actividades de evaluación: exámenes presenciales, presencialidad 100%	
Estudio personal de material básico y complementario de la asignatura	
Actividades docentes:	
Reparto de créditos:	Otras actividades:
Teoría: 4,00	No tiene
Problemas: 1,00	
Laboratorios: 1,00	
Bibliografía:	
[1] Andries P. Engelbrecht. Computational Intelligence: An Introduction. Willey.	
[2] Marco Dorigo and Thomas Stützle. Ant Colony Optimization. The MIT Press.	
[3] Riccardo Poli, James Kennedy, Tim Blackwell. Particle Swarm Optimization. Swarm Intelligence, Vol 1, Issue 1, pp. 33-57, 2007.	
[4] A. E. Eiben, J. E. Smith: Introduction to Evolutionary Computing, Springer, 2003.	
[5] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989	
[6] T. Bäck, D. B. Fogel, Z. Michalewicz. Evolutionary Computation 1: Basic Algorithms and Operators. IoP, 2000.	
[7] T. Bäck, D. B. Fogel, Z. Michalewicz. Evolutionary Computation 2: Advanced Algorithms and Operators. IoP, 2000.	

Ficha docente guardada por última vez el 25/09/2018 10:14:00 por el usuario: Coordinador MMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)	
Asignatura: 609033 - Aprendizaje automático		Abrev: AA	
Asignatura en Inglés: Machine learning		Carácter: Optativa	
Materia: Técnicas Matemáticas Especializadas		18 ECTS	
Otras asignaturas en la misma materia:			
Computación cuántica		6 ECTS	
Diseño y análisis de protocolos de seguridad		6 ECTS	
Módulo: Métodos Formales Complementarios			
Departamento: Sistemas Informáticos y Computación		Coordinador: Peña Marí, Ricardo	

Descripción de contenidos mínimos:

Conceptos básicos de aprendizaje automático

Preprocesamiento de los datos

Aprendizaje supervisado: árboles decisión, validación de modelos, redes neuronales, máquinas de soporte vectorial

Aprendizaje no supervisado: modelos paramétricos y no paramétricos

Basic concepts of machine learning

Data preprocessing

Supervised learning: decision trees, model validation, neural networks, support vector machines

Unsupervised learning: parametric and non-parametric models

Programa detallado:

1. Introducción al aprendizaje automático
2. Preprocesamiento de los datos
3. Reducción de la dimensionalidad
4. Clasificación y regresión
5. Teoría Bayesiana de la decisión
6. Estimación de la densidad
7. Árboles de decisión
8. Métodos conjuntistas
9. Máquinas vectoriales de soporte
10. Modelos paramétricos
11. Modelos no paramétricos
12. Validación de agrupamientos

Programa detallado en inglés:

PART I: Introduction

1. Introduction to Machine Learning

- Basic concepts (definitions, types of approaches)
- Machine Learning applications
- Components and stages in a Machine Learning project
- Evaluation, regularization and model selection (error measures, confusion matrix, ROC analysis, PR, validation)
- Introduction to the Machine Learning software Weka

2. Data pre-processing

- Normalization and standardization
- Discretization
- Processing of special variables: dates, series, categorical and nominal attributes
- Missing values and outliers

3. Dimensionality reduction

- The curse of dimensionality
- Variable selection (filter and wrapper methods)
- Variable construction (LDA, PCA, spectral methods)

PART II: Supervised Learning

4. Classification and regression

- Linear regression (LMS)
- Logistic regression and classification
- Underfitting and overfitting
- Generalized Linear Models

5. Bayesian decision theory

- Likelihood ratio test
- Bayes risk
- MAP and ML criteria

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



- Generative Learning Algorithms, Gaussian Discriminant Analysis
- Naïve Bayes
- 6. Density estimation
 - Parametric (Maximum Likelihood, EM)
 - Non-parametric (histograms, Parzen Windows, kernels)
- 7. Decision trees
 - Definitions
 - Decision tree construction. Division criteria (expected error, Gini index, information gain)
 - Pruning
 - Definition of ensemble learning. Precision and diversity assumptions
 - Statistical, numerical and representational problems of single classifiers
 - Bagging and boosting
 - The bias-variance tradeoff
- 9. Support Vector Machines
 - The Vapnik-Chervonenkis dimension
 - Structural Risk Minimization
 - Maximum margin classifier, optimal separating hyperplane
 - Separable and non-separable cases
 - Nonlinear problems. Cover's theorem. Kernels
 - The SMO algorithm

PART III: Unsupervised Learning

- 10. Parametric models
 - Model based clustering and density estimation
 - Maximum Likelihood. The EM algorithm
 - Gaussian Mixtures
 - Validation criteria. AIC and BIC, cross-validation
- 11. Non-parametric models
 - Unsupervised classification (clustering)
 - K-means algorithm
 - Fuzzy c-means
 - Hierarchical clustering
- 12. Cluster validation
 - Geometric indices
 - Information and likelihood based indices
 - Negentropy based indices

Competencias de la asignatura:

Generales:

- CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
- CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
- CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

- CE_MF7-Capacidad para aplicar técnicas matemáticas a problemas informáticos relevantes, tales como el diseño de protocolos criptográficos seguros, la construcción de modelos formales del software, o el diseño de sistemas que aprenden automáticamente durante su ejecución.
- CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

- CB_MF6-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
- CB_MF10-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.
- CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

- Capacidad para aplicar distintos modelos de aprendizaje al diseño de sistemas
- Capacidad para diseñar sistemas de redes neuronales
- Capacidad para utilizar competentemente las herramientas de ayuda al diseño de sistemas de aprendizaje
-
- Ability to apply different machine learning models to system design
- Ability to design neural networks
- Ability to competently use tools for designing machine learning systems

Evaluación:

- Exámenes presenciales 30%-70%
- Entrega de trabajos individuales o colaborativos 0%-40%
- Participación en las clases prácticas 0%-30%
- Evaluación de prácticas desarrolladas por el alumno 0%-40%

Evaluación detallada:

In the ordinary exam period, the evaluation will be made according to the following scheme:

- . 50 % Lab assignments
- . 50 % Final exam

It is necessary to have a pass grade (≥ 5) in both the lab assignments and the final exam to pass the course. This applies also to the extraordinary exam period.

The grades of the individual parts are kept for the extraordinary exam period. In case of a fail grade in the ordinary exam period, in the extraordinary exam period the student has the opportunity to turn in all the lab assignments with corrections.

The grade will be determined by:

- . 50 % Lab assignments [if the lab assignments are not turned in, the grade obtained in the ordinary exam period will be used]
- . 50 % Final exam [mandatory]

Actividades formativas:

- Clases teóricas, presencialidad 100%
- Clases prácticas: de problemas, de laboratorio, presencialidad 100%
- Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
- Actividades de evaluación: exámenes presenciales, presencialidad 100%
- Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

- | | |
|----------------------|--------------------|
| Reparto de créditos: | Otras actividades: |
| Teoría: 6,00 | No tiene |
| Problemas: 0,00 | |
| Laboratorios: 0,00 | |

Bibliografía:

1. T. Hastie, R. Tibshirani, J.H. Friedman. The Elements of Statistical Learning. Springer 2009
2. C.M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006
3. R.O. Duda, P.E. Hart. D.G. Stork; Pattern Classification; Wiley, 2000
4. T.M. Mitchell. Machine Learning. McGraw-Hill, 1997

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (A)	
Asignatura: 609034 - Diseño y análisis de protocolos de seguridad		Abrev: DAPS	6 ECTS
Asignatura en Inglés: Design and analysis of security protocols		Carácter: Optativa	
Materia: Técnicas Matemáticas Especializadas		18 ECTS	
Otras asignaturas en la misma materia: Aprendizaje automático Computación cuántica		6 ECTS 6 ECTS	
Módulo: Métodos Formales Complementarios			
Departamento: Sistemas Informáticos y Computación		Coordinador: Peña Marí, Ricardo	

Descripción de contenidos mínimos: No tiene
Programa detallado: ESTA ASIGNATURA NO SE IMPARTE DURANTE EL CURSO 2018-19
Programa detallado en inglés: This subject will not be taught in the course 2018/19
Competencias de la asignatura:
Generales: CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática. CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos. CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.
Específicas: CE_MF7-Capacidad para aplicar técnicas matemáticas a problemas informáticos relevantes, tales como el diseño de protocolos criptográficos seguros, la construcción de modelos formales del software, o el diseño de sistemas que aprenden automáticamente durante su ejecución. CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.
Básicas y Transversales: CB_MF6-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 CB_MF10-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. CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas. CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo. CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.
Resultados de aprendizaje: Capacidad para elegir el método criptográfico más apropiado para una aplicación Capacidad para diseñar protocolos cifrados de comunicación Capacidad para evaluar protocolos de comunicación Capacidad para implementar protocolos y servicios de seguridad ----- Ability to select the cryptographic method best suited for an application Ability to design encryption communication protocols

Fecha: ____ de _____ de _____

Firma del Director del Departamento:



Ability to evaluate communication protocols	
Ability to implement security protocols and services	
Evaluación: No tiene	
Evaluación detallada: No tiene	
Actividades formativas: No tiene	
Actividades docentes: Reparto de créditos: Teoría: 6,00 Problemas: 0,00 Laboratorios: 0,00	Otras actividades: No tiene
Bibliografía: No tiene	

Ficha docente guardada por última vez el 25/09/2018 10:15:00 por el usuario: Coordinador MMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



Ficha del curso: 2018-2019

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (2C)	
Asignatura: 609035 - Computación cuántica		Abrev: CC	6 ECTS
Asignatura en Inglés: Quantum computing		Carácter: Optativa	
Materia: Técnicas Matemáticas Especializadas		18 ECTS	
Otras asignaturas en la misma materia:			
Aprendizaje automático		6 ECTS	
Diseño y análisis de protocolos de seguridad		6 ECTS	
Módulo: Métodos Formales Complementarios			
Departamento: Análisis Matemático y Matemática Aplicada		Coordinador: Peña Marí, Ricardo	

Descripción de contenidos mínimos:

Fundamentos de información cuántica
Computación cuántica
Algoritmos de Shor y Grover
Criptografía cuántica
Algoritmo BB84

Quantum information foundations
Quantum computing
Shor and Grover algorithms
Quantum cryptography
BB84 algorithm

Programa detallado:

Fundamentos de información cuántica:
Postulados de la mecánica cuántica y notación.
Teleportación y codificación superdensa.
Computación cuántica:
Circuitos cuánticos.
Algoritmos cuánticos.
Algoritmos básicos de búsqueda.
Algoritmos de Shor y Grover:
Transformada de Fourier cuántica.
Algoritmo de Shor.
Algoritmo de Grover.
Criptografía cuántica:
Fundamentos.
Principio de incertidumbre.
Teorema de no clonación.
Algoritmo BB84:
El algoritmo.
Prueba simplificada de seguridad.

Programa detallado en inglés:

Foundations of quantum information:
Postulates of quantum mechanics and basic notation.
Quantum teleportation and superdense coding.
Quantum Computation: Quantum circuits.
Quantum algorithms.
Some basic searching algorithms.
Shor's and Grover's algorithms:
Quantum Fourier transform.
Shor's algorithm.
Grover's algorithm.
Quantum cryptography:
Foundations.
Heisenberg's uncertainty principle.
No cloning Theorem.
BB84 Algorithm:
The algorithm.
Simplified proof of security.

Competencias de la asignatura:**Generales:**

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



CG_MF1-Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática.
CG_MF5-Capacidad para la aplicación de los conocimientos adquiridos para resolver problemas en entornos nuevos o poco conocidos dentro de contextos amplios y multidisciplinares, siendo capaces de integrar dichos conocimientos.
CG_MF7-Capacidad para comprender trabajos de investigación y para crear nuevo conocimiento en el área de los métodos formales aplicados a la Ingeniería Informática.

Específicas:

CE_MF6-Capacidad para utilizar modelos de cómputo alternativos a los convencionales, tales como la computación cuántica, los sistemas de reescritura, las máquinas abstractas, la programación con restricciones o los algoritmos bio-inspirados
CE_MF10-Capacidad para la modelización matemática, el cálculo y la simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería en Informática.

Básicas y Transversales:

CB_MF6-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
CB_MF10-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.
CT_MF1-Capacidad para trabajar en equipo, ya sea como un miembro más o realizando la labor de dirección del mismo, promoviendo el libre intercambio de ideas.
CT_MF2-Capacidad para fomentar la creatividad tanto propia como la de los restantes miembros del equipo.
CT_MF3-Capacidad de razonamiento crítico como vía para mejorar la generación y desarrollo de ideas en un contexto profesional o de investigación.

Resultados de aprendizaje:

Capacidad para analizar algoritmos de computación cuántica
Capacidad para diseñar variantes de algoritmos de computación cuántica
Capacidad para diseñar protocolos criptográficos cuánticos

Ability to analyze quantum computing algorithms
Ability to design variants of quantum computing algorithms
Ability to design quantum cryptographic protocols

Evaluación:

Exámenes presenciales 30%-70%
Entrega de trabajos individuales o colaborativos 0%-40%
Participación en las clases prácticas 0%-30%
Evaluación de prácticas desarrolladas por el alumno 0%-40%

Evaluación detallada:

30% exámenes presenciales,
30% participación en las clases prácticas,
40% prácticas desarrolladas por el alumno.

Actividades formativas:

Clases teóricas, presencialidad 100%
Clases prácticas: de problemas, de laboratorio, presencialidad 100%
Trabajos desarrollados por los alumnos: problemas, prácticas y resúmenes de artículos o libros
Actividades de evaluación: exámenes presenciales, presencialidad 100%
Estudio personal de material básico y complementario de la asignatura

Actividades docentes:

Reparto de créditos: Otras actividades:

Fecha: ____ de _____ de ____
Firma del Director del Departamento:



Teoría: 6,00
Problemas: 0,00
Laboratorios: 0,00

No tiene

Bibliografía:

Básica:

M. A. Nielsen and I. L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, 2000.

De consulta:

J. Preskill, Quantum computation: lecture notes. Available at: <http://www.theory.caltech.edu/%7Epreskill/ph219/index.html#lecture>

R. de Wolf, Quantum Computing: lecture notes. Available at: <https://homepages.cwi.nl/~rdewolf/qcnotes.pdf>

M.M. Wilde, From Classical to Quantum Shannon Theory, available at: <https://arxiv.org/pdf/1106.1445.pdf>

Ficha docente guardada por última vez el 25/09/2018 10:10:00 por el usuario: CoordinadorMMF

Fecha: ____ de _____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento: