



Ficha del curso: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)	Idioma: English
Asignatura: 609024 - Teoría de lenguajes de programación Asignatura en Inglés: Theory of Programming Languages		Abrev: TL Carácter: Obligatoria	6 ECTS
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: Fabregas Álfaro, Ignacio	

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 preliminares
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.

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_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 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.

50% - Assignments and small projects.

[B] Exam.

50% - In-class exam.

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



In the extraordinary assessment sitting, students' grades from "Assignments and small projects" in the ordinary sitting will be kept (method [A] - 50%);
the rest of the mark will be graded on the basis of the in-class exam (method [B] - 50%).

Actividades docentes:

Reparto de créditos:

Teoría: 3,00
Problemas: 3,00
Laboratorios: 0,00

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:

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

Henk P. Barendregt. The Lambda Calculus: Its Syntax and Semantics: v.103. Studies in Logic and the Foundations of Mathematics. North Holland, 2014.

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

Jean-Yves Girard, Yves Lafont, Paul Taylor. Proofs and Types. Cambridge Tracts in Theoretical Computer Science, 7. Cambridge University Press, 1989

Roger Hindley. Basic Simple Type Theory. Cambridge Tracts in Theoretical Computer Science, 42. Cambridge University Press, 1997.

Janvan Leeuwen (editor). Handbook of Theoretical Computer Science. Volume B: Formal Models and Semantics. The MIT Press, 1991.

John C. Mitchell. Foundations for Programming Languages. Foundation of computing series. MIT Press, 1996.

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

Benjamin Pierce. Types and Programming Languages. MIT Press, 2002.

Terese. Term Rewriting Systems. Cambridge Tracts in Theoretical Computer Science, 55. Cambridge University Press, 2003.

Glynn Winskel. The Formal Semantics of Programming Languages: An Introduction, 4th. ed.; Foundations of Computing. MIT Press, 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: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)	Idioma: English
Asignatura: 609025 - Análisis estático de programas y resolución de restricciones Asignatura en Inglés: Static analysis of programs and constraint solving		Abrev: AERR Carácter: Obligatoria	6 ECTS
Materia: Métodos Formales Fundamentales		18 ECTS	
Otras asignaturas en la misma materia: Modelos de la concurrencia Teoría de lenguajes de programación		6 ECTS 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, 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 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 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.
- [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.

An 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 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.

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



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 22/07/2020 18:17:00 por el usuario: Vic. Estudios

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: 2022-2023

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

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): Mcrl2, 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): Mcrl2, 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 precesses

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 detallada:

Regular examination:

There are two tracks of evaluation:

[A] Assignments 80%; Participation during the course 20%

Those students failing evaluation track [A] can follow track [B]:

[B] Assignments 40%; Exam 40%; Participation during the course 20%

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

In the extraordinary assessment sitting, students may be required to submit new assignments before the exam, for which a new deadline will be set. The mark corresponding to participation in the extraordinary examination will be equal to the mark obtained for the regular examination.

Actividades docentes:

Reparto de créditos:

Teoría: 4,50

Problemas: 1,50

Laboratorios: 0,00

Otras actividades:

No tiene

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



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 23/06/2022 10:13:00 por el usuario: Vic. Ordenación Académica

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: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (2C)	Idioma: English
Asignatura: 609027 - Verificación asistida de programas Asignatura en Inglés: Computer-aided Program Verification		Abrev: VA Carácter: Optativa	6 ECTS
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: Segura Díaz, Clara Mª	

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 Asistencia a las demostraciones de terminación y de lemas
 - 1.3 Verificación de programas con vectores
 - 1.4 Tipos algebraicos y estructuras de datos
 - 1.5 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

Programa detallado en inglés:

1. The Dafny verification platform:
 - 1.1. The basics of the Dafny language
 - 1.2 Assisted proofs of termination and lemmas
 - 1.3. Verification of programs on arrays
 - 1.4. Algebraic data types and data structures
 - 1.5. 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

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 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 docentes:

- | | |
|----------------------|--------------------|
| Reparto de créditos: | Otras actividades: |
| Teoría: 4,00 | No tiene |
| Problemas: 0,00 | |
| Laboratorios: 2,00 | |

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



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://dafny.org/dafny/OnlineTutorial/guide>
- Dafny reference manual, <https://github.com/dafny-lang/dafny/blob/master/docs/DafnyRef/out/DafnyRef.pdf>, 2022. Online version (<https://dafny.org/dafny/DafnyRef/DafnyRef>)
- 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 24/06/2022 13:44:00 por el departamento: Sistemas Informáticos y Computación

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: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)	Idioma: English
Asignatura: 609028 - Métodos formales de testing Asignatura en Inglés: Formal methods in testing		Abrev: MFT Carácter: Optativa	6 ECTS
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: Requeno Jarabo, José Ignacio	

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 distribuidos y asíncronos.
5. Verificación de sistemas en tiempo de ejecución.

Programa detallado en inglés:

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

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:

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



- 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 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 detallada:

Students have to present a certain number of research papers. The number of research papers 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 docentes:

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

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.

Yliès Falcone, Klaus Havelund, Giles Reger: A Tutorial on Runtime Verification. Engineering Dependable Software Systems 2013: 141-175

Yliès Falcone, Srdan Krstic, Giles Reger, Dmitriy Traytel: A taxonomy for classifying runtime verification tools. Int. J. Softw. Tools Technol. Transf. 23(2): 255-284 (2021)

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.

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



Ficha del curso: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (2C)	Idioma: English
Asignatura: 609029 - Análisis de sistemas concurrentes y distribuidos Asignatura en Inglés: Analysis of Concurrent and Distributed Systems		Abrev: ASCD Carácter: Optativa	6 ECTS
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: Albert Albiol, Elvira María	

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:

1. Semántica de programas concurrentes y distribuidos
2. Análisis dinámico de sistemas concurrentes y distribuidos
3. Análisis estático de sistemas concurrentes y distribuidos
 - 3.1. Propiedad básicas y de vitalidad
 - 3.2. Terminación y consumo de recursos
4. Análisis y verificación de contratos inteligentes
5. Pruebas de sistemas concurrentes basadas en propiedades

Programa detallado en inglés:

1. Semantics of Concurrent and Distributed Programs
2. Dynamic analysis of concurrent and distributed systems
3. Static analysis of concurrent and distributed systems
 - 3.1. Basic and liveness properties
 - 3.2. Termination and resource consumption
4. Analysis and verification of smart contracts
5. Property-based testing 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_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
- 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 detallada:

- Examen final 60%
- Presentación al grupo de un artículo de investigación 20%
- Participación en las clases 20%
- Habrà un examen extraordinario por el 60%. Las notas obtenidas por la presentación del artículo y la participación en clase se conservarán
- Final examination 60%
- Public presentation of a research paper 20%
- Participation in lectures 20%
- There will be an extraordinary examination for 60%. The marks obtained for the paper presentation and the participation in lectures are preserved

Actividades docentes:

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

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



Bibliografía:

- Susan Owicki and Leslie Lamport. 1982. Proving Liveness Properties of Concurrent Programs. ACM Trans. Program. Lang. Syst. 4, 3 (July 1982), 455-495. DOI=<http://dx.doi.org/10.1145/357172.357178>
- Gregory R. Andrews. 1991. Concurrent Programming: Principles and Practice. Benjamin-Cummings Publ. Co., Inc., Redwood City, CA, USA.
- M. Ben-Ari. 1990. Principles of Concurrent and Distributed Programming. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- Byron Cook, Andreas Podelski, and Andrey Rybalchenko. 2007. Proving thread termination. SIGPLAN Not. 42, 6 (June 2007), 320-330. DOI: <https://doi.org/10.1145/1273442.1250771>
- Cormac Flanagan, Patrice Godefroid: Dynamic partial-order reduction for model checking software. POPL 2005: 110-121
- Parosh Aziz Abdulla, Stavros Aronis, Bengt Jonsson, Konstantinos Sagonas: Source Sets: A Foundation for Optimal Dynamic Partial Order Reduction. J. ACM 64(4): 25:1-25:49 (2017)
- Krzysztof R. Apt, Frank S. de Boer, Ernst-Rüdiger Olderog: Verification of Sequential and Concurrent Programs. Texts in Computer Science, Springer 2009, ISBN 978-1-84882-744-8, pp. i-xxiii, 1-502
- C Barrett, R Sebastiani, S Seshia, and C Tinelli, "Satisfiability Modulo Theories." In Handbook of Satisfiability, vol. 185 of Frontiers in Artificial Intelligence and Applications, (A Biere, M J H Heule, H van Maaren, and T Walsh, eds.), IOS Press, Feb. 2009, pp. 825-885.

Ficha docente guardada por última vez el 22/06/2022 15:02:00 por el departamento: **Sistemas Informáticos y Computación**

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: 2022-2023

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

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:

1. Introducción: Demostración de corrección de programas.
2. Fundamentos: Especificación, Lógica de primer orden, Demostraciones, Programas.
3. Event-B: Conceptos básicos. La Rodin Tool.
4. Sistemas secuenciales.
5. Event-B: Conjunto de herramientas matemáticas y sus aplicaciones.
6. Sistemas reactivos: Concurrencia y Distribución.
7. De la deducción automática a la programación con lógica.
8. Semántica y características avanzadas.
9. CLP y verificación de programas mediante interpretación abstracta.

Programa detallado en inglés:

1. Introduction: Proving Programs Correct
2. Fundamentals: Specification, First-Order Logic, Proofs, Programs.
3. Event-B: Basics and the Rodin Tool.
4. Sequential Systems.
5. Event-B: Mathematical Toolkit and Applications.
6. Reactive Systems: Concurrency and Distribution.
7. From Automated Deduction to Programming with Logic.
8. Semantics and Advanced Features.
9. CLP and Program Verification via Abstract Interpretation.

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_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_MF5-Capacidad para utilizar y desarrollar herramientas que analizan propiedades de los programas, mediante su ejecución en un conjunto de casos cuidadosamente seleccionado.
- 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_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_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:

- 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 detallada:

Recommended option

Homework: 60%

Exam: Presentation of a personal project, 40%

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



Students will be evaluated based on their performance in the course homework / quizzes and the project. In the presentation, the quality of the information and the ability to answer questions on the decision designs will be taken into account. All students participating in a project are expected to also present part of the project and be able to answer questions to any part of the project.

There is an option of a final examination for 100%

Failing students will have the opportunity of an extraordinary examination under the same premises than the ordinary one.

Actividades docentes:

Reparto de créditos:

Teoría: 4,00

Problemas: 2,00

Laboratorios: 0,00

Otras actividades:

No tiene

Bibliografía:

- Event B development environment.
- Modeling in Event-B: System and Software Engineering. Jean-Raymond Abrial. Cambridge University Press.
- <http://wiki.event-b.org/>
- Seven Myths of Formal Methods. Anthony Hall. IEEE Software, September 1990
- Seven More Myths of Formal Methods. Jonathan P. Bowen, Michael G. Hinchey. IEEE Software, July 1995.
- Coq .
- Ciao/CiaoPP.

Ficha docente guardada por última vez el 27/06/2022 12:57:00 por el departamento: Sistemas Informáticos y Computación

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: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (1C)	Idioma: English
Asignatura: 609031 - Desarrollo formal de software dirigido por modelos Asignatura en Inglés: Formal model-driven software development		Abrev: DDM Carácter: Optativa	6 ECTS
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: Rubio Gimeno, Albert	

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-modelado
 - 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
- Validación y Verificación de modelos (V&V)
 - V&V estructural: búsqueda de modelos.
 - V&V de comportamiento: redes de Petri.

Programa detallado en inglés:

- Introduction to Model-Driven Software Development
 - Concepts
 - Technologies
 - Applications and Scenarios

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



2. Software Modelling and Meta-Modelling
 - a. UML
 - b. OCL
 - c. Meta-modelling
 - d. Multi-level meta-modelling
3. Domain-Specific Modelling Languages
 - a. Graphical Concrete Syntax
 - b. Textual Concrete Syntax
 - c. Defining DSMLs 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
6. Model Validation and Verification (V&V)
 - a. Structural V&V: model searching
 - b. Behavioral V&V: Petri Nets

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.

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



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 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 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 docentes:

Reparto de créditos:

Teoría: 6,00

Problemas: 0,00

Laboratorios: 0,00

Otras actividades:

No tiene

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: 2022-2023

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

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:

1. Bio-inspiración para modelos de cómputo

1.1 Mecanismo de cómputo abstracto a partir de fenómenos naturales

1.2 Inspiración en razonamiento humano.

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

1.4 Estado del arte en computadores naturales: tisulares, basados en grafos, simulación del descubrimiento, etc.

2. Computación evolutiva

2.1 Introducción a los algoritmos genéticos y computación evolutiva

2.2 Representaciones formales complejas de individuos en programación automática evolutiva (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:

1. Bio-inspiration for models of computing

1.1 Abstracting computing mechanisms from natural phenomena

1.2 Human reasoning inspiration

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

1.4 State of the art in natural computers: tissue based, graph based, discovery simulation, etc.

2. Evolutionary Computation

2.1 Introduction to genetic algorithms and evolutionary computation

2.2 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.

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_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, 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 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 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 docentes:

Reparto de créditos:

Teoría: 4,00

Problemas: 1,00

Laboratorios: 1,00

Otras actividades:

No tiene

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.

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



Ficha del curso: 2022-2023

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

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:

PARTE I: Introducción
1. Introducción al aprendizaje automático
2. Preprocesamiento de los datos
3. Reducción de la dimensión

PARTE II: Aprendizaje supervisado
4. Clasificación y regresión
5. Árboles de decisión
6. Validación de modelos
7. Redes neuronales artificiales
8. Máquinas de soporte vectorial
9. Ensembles

PARTE III: Aprendizaje no supervisado
10. Modelos paramétricos
11. Modelos no paramétricos

Programa detallado en inglés:

PART I: Introduction
1. Introduction to Machine Learning
2. Data pre-processing
3. Dimensionality reduction

PART II: Supervised Learning
4. Classification and regression
5. Decision trees
6. Model validation
7. Artificial neural nets
8. Support Vector Machines
9. Ensembles

PART III: Unsupervised Learning
10. Parametric models
11. Non-parametric models

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_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_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 detallada:

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

. 40 % Lab assignments

. 60 % Final exam

This scheme will be the same for the extraordinary exam period. The student has the opportunity to improve the already presented lab assignments

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 docentes:

Reparto de créditos:

Teoría: 6,00

Problemas: 0,00

Laboratorios: 0,00

Otras actividades:

No tiene

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



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
5. I. Goodfellow, Y. Bengio, A. Courville. Deep Learning. MIT Press 2016. <http://www.deeplearningbook.org>

Ficha docente guardada por última vez el 28/07/2020 17:55:00 por el usuario: Vic. Estudios

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: 2022-2023

Grado: MÁSTER EN MÉTODOS FORMALES EN INGENIERÍA INFORMÁTICA		Curso: 1º (A)	Idioma: English
Asignatura: 609034 - Diseño y análisis de protocolos de seguridad Asignatura en Inglés: Design and analysis of security protocols		Abrev: DAPS Carácter: Optativa	6 ECTS
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: Rubio Gimeno, Albert	

Descripción de contenidos mínimos:

No tiene

Programa detallado:

- Seguridad perfecta y seguridad computacional:
Noción de encriptación, libreta de un solo uso
Nociones de seguridad
Funciones unidireccionales y funciones unidireccionales trampa, ejemplos matemáticos
Seguridad IND-CPA
- Pseudoaleatoriedad y criptografía de clave simétrica:
Predicados de núcleo duro para funciones unidireccionales
Generadores pseudoaleatorios y funciones pseudoaleatorias
Códigos de autenticación de mensajes
Encriptación de clave simétrica
Modos de operación
- Criptografía de clave pública:
Intercambio de claves
Encriptación de clave pública
Cifrado ElGamal
Cifrado RSA
Firma digital
- Protocolos de seguridad avanzados:
Pruebas de conocimiento nulo
Secreto compartido
Computación multipartita segura

Programa detallado en inglés:

- Perfect security and computational security:
Notion of encryption, one-time pad
Notions of security
One-way functions and one way trapdoor functions, mathematical examples
IND-CPA security
- Pseudorandomness and symmetric-key cryptography:
Hardcore predicates for one-way functions
Pseudorandom generators and pseudorandom functions
Message authentication codes
Symmetric key encryption
Modes of operation
- Public key cryptography:
Key Exchange
Public key encryption
El Gamal encryption
RSA encryption
Digital signatures
- Advanced protocols:

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



Zero knowledge proofs
Secret sharing
Secure multiparty computation

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
- Ability to evaluate communication protocols
- Ability to implement security protocols and services

Evaluación detallada:

El método de evaluación consistirá de dos partes:

- Examen final en el aula [60%].
- Tareas a desarrollar durante el curso en clase y fuera de clase [40%].

The assessment method will consist of two parts:

- In-class final exam [60%].
- Assignments to be developed during the course period at home or in class [40%].

Actividades docentes:

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



Reparto de créditos: Teoría: 4,50 Problemas: 1,50 Laboratorios: 0,00	Otras actividades: No tiene
Bibliografía: Basic: -Goldwasser, Bellare: Lecture Notes on Cryptography, https://cseweb.ucsd.edu/~mihir/papers/gb.pdf -Boneh, Shoup: A Graduate Course in Applied Cryptography, http://toc.cryptobook.us/ Additional reading: -Katz, Lindell: Introduction to Modern Cryptography, Second Edition, CRC Press -Barak: An Intensive Introduction to Cryptography, https://intensecrypto.org/public/index.html -Rosulek: The Joy of Cryptography, https://web.engr.oregonstate.edu/~rosulekm/crypto/	

Ficha docente guardada por última vez el 07/09/2021 10:21:00 por el departamento: Sistemas Informáticos y Computación

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: 2022-2023

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

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.
Algoritmo de Shor:
Transformada de Fourier cuántica.
Algoritmo de Shor.
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 algorithm:
Quantum Fourier transform.
Shor'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:



UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE INFORMATICA

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: Comprender los fundamentos de la computación cuántica y sus principales algoritmos Desarrollar algoritmos cuánticos nuevos, o variantes de los existentes 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 detallada: 30% exámenes presenciales, 30% participación en las clases prácticas, 40% prácticas y entregas.									
Actividades docentes: <table><tr><td>Reparto de créditos:</td><td>Otras actividades:</td></tr><tr><td>Teoría: 6,00</td><td>No tiene</td></tr><tr><td>Problemas: 0,00</td><td></td></tr><tr><td>Laboratorios: 0,00</td><td></td></tr></table>		Reparto de créditos:	Otras actividades:	Teoría: 6,00	No tiene	Problemas: 0,00		Laboratorios: 0,00	
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:

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 23/06/2021 10:51:00 por el departamento: **Análisis Matemático y Matemática Aplicada**

Fecha: ____ de ____ de ____

Firma del Director del Departamento:



UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMATICA

Fecha: ____ de _____ de ____

Firma del Director del Departamento: