

Formulario de aprobación de curso de posgrado/educación permanente

Asignatura:

Construcción Formal de Programas en Teoría de Tipos

Modalidad:

(posgrado, educación permanente o ambas)

Posgrado x

Educación permanente

Profesor de la asignatura ¹:

(título, nombre, grado o cargo, instituto o institución)

Dr. Ing. Gustavo Betarte, Profesor Titular (Gr. 5), InCo.

Dr. Lic. Carlos Luna Profesor Agregado (Gr. 4), InCo.

Profesor Responsable Local ¹:

(título, nombre, grado, instituto)

Otros docentes de la Facultad:

(título, nombre, grado, instituto)

Docentes fuera de Facultad:

(título, nombre, cargo, institución, país)

¹ Agregar CV si el curso se dicta por primera vez.

(Si el profesor de la asignatura no es docente de la Facultad se deberá designar un responsable local)

Programa(s) de posgrado:

Maestría y Doctorado del Área Informática del Pedeciba

Instituto o unidad:

Instituto de Computación

Departamento o área:

Horas Presenciales: 56

Nº de Créditos: 10

Público objetivo:

Estudiantes avanzados de Ingeniería en Computación.

Cupos:

No hay cupos para estudiantes de posgrado.

Objetivos:

Iniciar al estudiante al uso de métodos formales para la producción de software correcto por construcción.

Conocer los sistemas de autenticación más importantes identificando sus características

Conocimientos previos exigidos: Cálculo Proposicional y de Predicados. Deducción natural.

Conjuntos y pruebas inductivas. Funciones de alto orden.

Conocimientos previos recomendados: Cálculos funcionales. Sistemas de Tipos. Programación Funcional. Programación Lógica, procedimientos de unificación de términos.

Metodología de enseñanza:

Descripción de la metodología:

Presentación de conceptos teóricos fundamentales en clases presenciales, con frecuencia de un dictado de 2 horas por semana. Realización de 7 trabajos práctico/laboratorios donde se realizan ejercicios y tareas usando el asistente de pruebas Coq.

Detalle de horas:

- Horas clase (teórico):20
- Horas clase (práctico):
- Horas clase (laboratorio):15
- Horas consulta:15
- Horas evaluación:6
 - Subtotal horas presenciales:56
- Horas estudio: 30
- Horas resolución ejercicios/prácticos:70
- Horas proyecto final/monografía:
 - Total de horas de dedicación del estudiante:156

Forma de evaluación:

La asignatura se evaluará por medio de dos parciales y trabajos de laboratorio. El nivel mínimo de suficiencia en los trabajos de laboratorio es eliminatorio, ya que esta parte del trabajo del curso no puede ser evaluada mediante exámenes. Por otra parte, dependiendo de las condiciones de dictado del curso, el trabajo de laboratorio se evalúa según las opciones aprobado/no aprobado, o con puntaje diferenciado en el caso de aprobación. En este último caso, el puntaje del laboratorio se integraría al puntaje total del curso, prorrateándose en los de las pruebas parciales.

En todos los casos de los resultados obtenidos surgen dos posibilidades:

1. Exoneración del curso
2. Insuficiencia en el curso; el estudiante reprueba el curso

Se presenta a continuación el esquema de evaluación del curso

Exoneración. El estudiante debe cumplir los siguientes requisitos:

1. llegar al nivel mínimo y en cada uno de los trabajos de laboratorio, y
2. reunir al menos el 60% del puntaje de parciales,
3. obtener al menos el 25% en cada prueba parcial
4. Presentación de trabajo final (preparación y presentación de artículos científicos)

Insuficiencia. El estudiante no obtiene los puntajes de ninguna de las franjas anteriores.

-
1. Asistentes de pruebas para lógicos y matemáticos: Una presentación formal de la lógica proposicional y de primer orden.
 2. Asistentes de Pruebas para programadores: El calculo lambda como lenguaje de programación funcional.
 3. Identificación de pruebas y programas: Isomorfismo de Curry Howard
 4. Recursión: Definiciones Inductivas, Principios de Inducción, Esquemas de Recursión.
 5. Extracción de programas a partir de pruebas. Construcción de Pruebas a partir de programas.
 6. Construcción de programas certificados usando Coq.

Bibliografía:

Interactive Theorem Proving and Program Development. Y. Bertot, P. Casteran, Springer-Verlag, 2004.

Computation and Reasoning: A Type Theory for Computer Science. Volume 11 of International Series of Monographs on Computer Science. Oxford Science Publications, 1994. ISBN: 0198538359.

The Coq Proof Assistant. Reference Manual. B. Barras et al. Rapport de Recherche INRIA. Disponible en: <http://pauillac.inria.fr/coq/doc-eng.html>

A Tutorial on Recursive Types in Coq. Eduardo Giménez, Rapport de Recherche INRIA. Disponible en: <http://pauillac.inria.fr/coq/doc-eng.html>