



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY

Programa de Construcción Formal de Programas en Teoría de Tipos

1. NOMBRE DE LA UNIDAD CURRICULAR

Construcción Formal de Programas en Teoría de Tipos (CFPTT)

2. CRÉDITOS

12 créditos

3. OBJETIVOS DE LA UNIDAD CURRICULAR

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

4. METODOLOGÍA DE ENSEÑANZA

- Horas clase (teórico): 26
- Horas clase (práctico y laboratorio): 15
- Horas consulta: 15
- Horas evaluación: 6
 - Subtotal horas presenciales: 62
- Horas estudio: 45
- Horas resolución ejercicios/prácticos: 73
- Horas proyecto final/monografía:
 - Total de horas de dedicación del estudiante: 180

5. TEMARIO

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 cálculo 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. Síntesis y Verificación de programas.
6. Construcción de programas certificados usando Coq.

6. BIBLIOGRAFÍA

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

6.1 Básica

1. Interactive Theorem Proving and Program Development. Y. Bertot, P. Casteran, Springer-Verlag, 2004. (<http://www.labri.fr/perso/casteran/CoqArt/>)
2. 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.

6.2 Complementaria

3. The Coq Proof Assistant. Reference Manual. B. Barras et al. Rapport de Recherche INRIA. Disponible en: <https://coq.inria.fr/distrib/current/refman/>.
4. A Tutorial on Recursive Types in Coq. Eduardo Giménez, Rapport de Recherche INRIA. Disponible en: <http://www.labri.fr/perso/casteran/RecTutorial.pdf>.
5. Materiales y bibliografía disponibles en el sitio oficial de Coq: <https://coq.inria.fr/documentation>.

7. CONOCIMIENTOS PREVIOS EXIGIDOS Y RECOMENDADOS

7.1 Conocimientos Previos Exigidos: Cálculo Proposicional y de Predicados. Deducción natural. Conjuntos y pruebas inductivas. Funciones de alto orden.

7.2 Conocimientos Previos Recomendados: Cálculos funcionales. Sistemas de Tipos. Programación Funcional. Programación Lógica, procedimientos de unificación de términos.

ANEXO A
Para todas las Carreras

A1) INSTITUTO

Instituto de Computación.

A2) CRONOGRAMA TENTATIVO

Consiste en un cronograma de avance semanal con detalle de las horas de clase asignadas a cada tema.

Semana 1	Introducción al curso y a los Métodos Formales (3hs)
Semana 2	Cálculo proposicional de alto orden en Coq (3hs)
Semana 3	Cálculo de predicados de alto orden en Coq (3hs)
Semana 4	Introducción al Cálculo Lambda Tipado Simple (3hs)
Semana 5	Introducción al Cálculo de Construcciones (3hs)
Semana 6	Conjuntos Inductivos, Principios de Inducción (3hs)
Semana 7	Conjuntos Inductivos, Recursión (3hs)
Semana 8	Inversión (3hs)
Semana 9	Isomorfismo de Curry Howard (3hs)
Semana 10	Síntesis de programas (3hs)
Semana 11	Verificación de Programas (3hs)
Semana 12	Presentación de Caso de Estudio (3hs)
Semana 13	Discusión de Caso de Estudio (3hs)
Semana 14	Caso de Estudio (1h)
Semana 15	Caso de Estudio (1h)

A3) MODALIDAD DEL CURSO Y PROCEDIMIENTO DE EVALUACIÓN

La unidad curricular 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:

- Exoneración del curso.
- Insuficiencia en el curso; el estudiante reprueba el curso.

Se presenta a continuación el esquema de evaluación del curso

ANEXO B para la(s) carrera(s) Ingeniería en Computación (plan 87)

B1) ÁREA DE FORMACIÓN

... No corresponde

B2) UNIDADES CURRICULARES PREVIAS

Para el Curso: Previas comunes a las electivas y
Examen de Lógica y
Curso de Int. A la Programación Funcional

... Para el Examen: No aplica

Observación: Esta unidad curricular se corresponde con una electiva