



**UNIVERSIDAD NACIONAL DE RÍO CUARTO**

**FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES**

**DEPARTAMENTO DE COMPUTACIÓN**

**CARRERA/S:** Licenciatura en Ciencias de la Computación (Cód. 14)

**PLAN DE ESTUDIOS:** Año 1999 – Versión 1

**ASIGNATURA:** Construcción Formal de Programas en Teoría de Tipos

**CÓDIGO:**

**DOCENTE RESPONSABLE:** Carlos Luna

**EQUIPO DOCENTE:** Carlos Luna, Dr.

Dante Zanarini, Licenciado en Ciencias de la Computación

**AÑO ACADÉMICO:** 2015

**REGIMEN DE LA ASIGNATURA:** cuatrimestral

**RÉGIMEN DE CORRELATIVIDADES:** (para cursado)

<i>Aprobada</i>	<i>Regular</i>
1948 - 3301	3302

**CARGA HORARIA TOTAL:** 112

**TEÓRICAS:** 40 hs    **PRÁCTICAS:** 40 hs    **LABORATORIO:** 32 hs

**CARÁCTER DE LA ASIGNATURA:** Optativa

**A. CONTEXTUALIZACIÓN DE LA ASIGNATURA**

Segundo cuatrimestre de cuarto año.

**B. OBJETIVOS PROPUESTOS**

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

Proponemos un curso/taller que tiene como meta la adquisición de destreza en el uso de herramientas de razonamiento fundamentales: la inducción matemática y la deducción lógica aplicadas a la construcción y verificación de programas. Los objetivos centrales que se persiguen pueden resumirse en la frase “programar rigurosamente sobre la base de argumentos matemáticos”. Esto es, fortalecer la noción de que junto con la construcción de los algoritmos existe la obligación de la verificación rigurosa (formal) de su corrección y que los programas son objetos matemáticos plausibles de ser tratados con argumentos lógico-matemáticos. Para lograr estos objetivos presentamos un taller para apoyar la enseñanza de métodos formales en una currícula de grado, usando el asistente de pruebas Coq y conceptos del área de Teoría de Tipos. El taller abarca la especificación, construcción y verificación de sistemas, y en particular el análisis de sistemas críticos.

**C. CONTENIDOS BÁSICOS DEL PROGRAMA A DESARROLLAR**

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

#### D. FUNDAMENTACIÓN DE LOS CONTENIDOS

Los profesionales de computación deben estar capacitados para estudiar los fundamentos de su disciplina. El núcleo central de las Ciencias de la Computación está constituido en buena parte por la matemática discreta y la lógica matemática. En consecuencia, un especialista en computación debe estar en condiciones de usar las herramientas básicas y las técnicas de dichas áreas. Esto le permitirá una adecuación rápida y eficaz a los acelerados cambios tecnológicos, que son una constante en la disciplina.

Proponemos un curso taller que tiene como meta la adquisición de destreza en el uso de herramientas de razonamiento fundamentales: la inducción matemática y la deducción lógica aplicadas a la construcción y verificación de programas. Los objetivos centrales que se persiguen pueden resumirse en la frase “programar rigurosamente sobre la base de argumentos matemáticos”. Esto es, fortalecer la noción de que junto con la construcción de los algoritmos existe la obligación de la verificación rigurosa (formal) de su corrección y que los programas son objetos matemáticos plausibles de ser tratados con argumentos lógico-matemáticos. Para lograr estos objetivos presentamos un taller para apoyar la enseñanza de métodos formales en una currícula de grado, usando el asistente de pruebas Coq y conceptos del área de Teoría de Tipos. El taller abarca la especificación, construcción y verificación de sistemas, y en particular el análisis de sistemas críticos.

El asistente de pruebas Coq es una herramienta muy potente (de porte industrial, que trasciende el ámbito exclusivamente académico), utilizada en formalizaciones importantes en diferentes áreas; en particular en el área lógico-matemática y en el área informática. Asimismo, ha sido utilizada en la enseñanza de métodos formales. Información actualizada sobre Coq está disponible en su sitio web oficial: <https://coq.inria.fr/>.

#### E. ACTIVIDADES A DESARROLLAR

**Clases teóricas:** cada 15 días, exposición a cargo de los docentes del curso, siguiendo material de lectura sugerido.

**Clases prácticas:** discusión de temas y ejercicios de los trabajos prácticos.

**Clases de trabajos prácticos de laboratorio:** realización de trabajos prácticos obligatorios que deben entregarse en fechas estipuladas durante el curso.

**CARGA HORARIA TOTAL: 112**

**TEÓRICAS: 40 hs      PRÁCTICAS: 40 hs      LABORATORIO: 32 hs**

#### F. NÓMINA DE TRABAJOS PRÁCTICOS

- Trabajo Práctico N°1: Asistentes de pruebas para lógicos y matemáticos. Cálculo proposicional.
- Trabajo Práctico N°2: Asistentes de pruebas para lógicos y matemáticos. Cálculo de predicados y cálculo de predicados con igualdad.

- Trabajo Práctico N°3: Asistentes de prueba para programadores. Cálculo de construcciones: primera parte.
- Trabajo Práctico N°4: Asistentes de prueba para programadores. Definiciones inductivas: primera parte.
- Trabajo Práctico N°5: Asistentes de prueba para programadores. Definiciones inductivas: segunda parte.
- Trabajo Práctico N°6: Verificación de programas funcionales.
- Trabajo Práctico N°7: Especificación y verificación de un caso de estudio.

**HORARIOS DE CLASES:** Jueves y Viernes cada 15 días en horarios a definir con los alumnos.

**HORARIO DE CLASES DE CONSULTAS:** A determinar con los alumnos.

## G. MODALIDAD DE EVALUACIÓN:

- **Evaluaciones Parciales:** 2 parciales con 30 y 40 puntos, respectivamente.
- **Evaluación Final:** entrega de un trabajo práctico final.
- **CONDICIONES DE REGULARIDAD:** el curso sólo admite la opción de promoción.
- **CONDICIONES DE PROMOCIÓN:**
  - llegar al nivel mínimo en los trabajos de laboratorio. Son 7 entregas, una por cada trabajo práctico, incluyendo la entrega del caso de estudio (práctico 7).
  - reunir al menos el 50% del puntaje de los parciales.
  - obtener al menos el 50% en cada prueba parcial

## PROGRAMA ANALÍTICO

### A. CONTENIDOS

semanas 1 y 2: Asistentes de pruebas para lógicos y matemáticos.

semanas 3 y 4: Asistentes de Pruebas para programadores.

semanas 5 y 6: Identificación de pruebas y programas.

semanas 7 a 9: Recursión.

semanas 10 y 11: Extracción de programas a partir de pruebas.

semanas 12 y 13: Construcción de programas certificados usando Coq.

semana 14: Desarrollo de un caso de estudio.

### B. CRONOGRAMA DE CLASES Y PARCIALES

Semana	Día/ Fecha	Teóricos	D/ F	Prácticos	D/ F	Laboratorios	Parcial / Recup.
1 y 2		Asistentes de pruebas para lógicos y matemáticos.		Cálculo proposicional.		Cálculo proposicional.	
3 y 4		Asistentes de Pruebas para programadores.		Cálculo de predicados y cálculo de predicados con igualdad.		Cálculo de predicados y cálculo de predicados con igualdad.	
5 y 6		Identificación de pruebas y programas.		Cálculo de construcciones		Cálculo de construcciones	1 <sup>er</sup> Parcial
7, 8, y 9		Recursión.		Definiciones inductivas		Definiciones inductivas	
10 y 11		Extracción de programas a partir de pruebas.		Verificación de programas funcionales.		Verificación de programas funcionales.	
12 y 13		Construcción de programas certificados usando Coq.				Especificación y verificación de un caso de estudio.	2 <sup>do</sup> Parcial

14		Desarrollo de un caso de estudio.			Especificación y verificación de un caso de estudio.	Entrega Proyecto
----	--	-----------------------------------	--	--	--	------------------

### C. BIBLIOGRAFÍA

De lectura obligatoria:

- Material de la cátedra.
- Interactive Theorem Proving and Program Development. Y. Bertot, P. Casteran, Springer-Verlag. 2004. Material disponible en: <http://www.labri.fr/perso/casteran/CoqArt/>.
- The Coq Proof Assistant. Reference Manual. Disponible en: <https://coq.inria.fr/distrib/current/refman/>.

De consulta:

- Software Foundations. Benjamin Pierce et al, 2007. Disponible en: <http://www.cis.upenn.edu/~bcpierce/sf/current/index.html>
- Certified Programming with Dependent Types. Adam Chlipala, 2008. Disponible en: <http://adam.chlipala.net/cpdt/>
- Computation and Reasoning: A Type Theory for Computer Science. Volume 11 of International Series of Monographs on Computer Science. Oxford Science Publications, 1994.
- Material adicional disponible en: <https://coq.inria.fr/documentation>.