

Planificación de
**Construcción Formal de Programas en
Teoría de Tipos**

Código/s: Electiva



Identificación y características del Espacio Curricular

Carrera/s:	Licenciatura en Ciencias de la Computación		
Plan de Estudios:	2010, TO2024	Carácter:	Electiva
Bloque/Campo:		Área:	Algoritmos y Lenguajes
Régimen de cursado:	Cuatrimestral		
Cuatrimestre:	- [LCC], 1º [LCC]		
Carga horaria:	75 hs. / 5 hs. semanales	Formato curricular:	Asignatura
Escuela:	Ciencias Exactas y Naturales	Departamento:	Ciencias de la Computación
Docente responsable:	ZANARINI, Dante		

Programa Sintético

Asistentes de pruebas para lógicos y matemáticos. Asistentes de Pruebas para programadores. Identificación de pruebas y programas. Definiciones inductivas. Principios de inducción. Esquemas de Recursión. Extracción de programas a partir de pruebas. Construcción de pruebas a partir de programas. Construcción de programas certificados. Especificación y verificación de un sistema crítico.

Espacios Curriculares Relacionados

Previos Aprobados:	R-223 - Lógica, R-322 - Análisis de Lenguajes de Programación
Simultaneos Recomendados:	
Posteriores:	

Vigencia desde 2024

_____ Firma Profesor	_____ Fecha	_____ Firma Aprob. Escuela	_____ Fecha
-------------------------	----------------	-------------------------------	----------------

Con el aval del Consejo Asesor:

Fundamentación

Los métodos formales para la verificación de software son un área fundamental de las ciencias de la computación. A través de modelos lógico-matemáticos se establece un vínculo (a través de una demostración formal) que un programa cumple con una especificación. Los asistentes de pruebas son un conjunto de herramientas que hacen posible verificar la corrección de programas en un entorno de software. En general, trabajan con lógicas de alto poder expresivo, y por lo tanto no decidibles. En el curso se trabaja sobre un asistente de pruebas que permite expresar, a través de la correspondencia de Curry-Howard, pruebas y programas en el mismo formalismo matemático. A lo largo del curso se destaca la importancia de la construcción formal de programas con aplicaciones a sistemas críticos.

Resultados del aprendizaje

Al finalizar el cursado los/las estudiantes serán capaces de:

RA1 Comprender los fundamentos de la Teoría de Tipos y la correspondencia de Curry-Howard
RA2 Razonar sobre los enfoques para construir programas correctos utilizando un asistente de pruebas
RA3 Comprender el proceso de demostración de corrección de programas en sistemas formales
RA4 Analizar diferentes enfoques para la demostración de corrección de programas
RA5 Desarrollar la capacidad de obtener programas correctos por construcción
RA6 Desarrollar la capacidad de razonamiento lógico para ordenar ideas y explicar conceptos de manera precisa

Competencias / Ejes transversales y Resultados del Aprendizaje

Competencia/Eje transversal al que tributa	Nivel	Resultados del Aprendizaje
CGT1-Identificación, formulación y resolución de problemas de informática	Alto	RA1, RA2, RA3, RA5
CGT2-Concepción, diseño y desarrollo de proyectos de informática	Medio	RA4
CGT4-Utilización de técnicas y herramientas de aplicación en la informática	Medio	RA1, RA2, RA3, RA4
CGS2-Fundamentos para la comunicación efectiva	Medio	RA6

Programa Analítico

Unidad 1: Asistentes de pruebas para lógicos y matemáticos

- 1.1.- Lógica proposicional constructiva. Deducción natural.
- 1.2.- Lógica de Predicados. Deducción natural.
- 1.3.- Teorías axiomáticas.

Unidad 2: Asistentes de Pruebas para programadores

- 2.1.- El cálculo lambda como lenguaje de programación funcional.
- 2.2.- El cálculo de construcciones.

Unidad 3: Identificación de pruebas y programas

- 3.1.- Construcción de pruebas como proceso de cálculo.
- 3.2.- El Isomorfismo de Curry Howard.

Unidad 4: Recursión

- 4.1.- Definiciones inductivas.
- 4.2.- Principios de inducción.
- 4.3.- Esquemas de Recursión.
- 4.4.- Inversión.

Unidad 5: Extracción y Verificación de Programas

- 5.1.- Extracción de programas a partir de pruebas.
- 5.2.- Construcción de Pruebas a partir de programas.
- 5.3.- Construcción de programas certificados.

Unidad 6: Caso de Estudio

Especificación y verificación de un sistema crítico.

Modalidades de enseñanza

Existen diferentes formatos de clases para la actividad curricular. En el primer tipo el/la docente presenta los contenidos teórico-prácticos enumerados en el programa analítico. Se trabaja utilizando pizarrón para los fundamentos teóricos y luego estos contenidos se llevan a la práctica utilizando un asistente de pruebas. El resto de las clases ocurre en formato taller, donde los y las estudiantes trabajan grupalmente sobre los problemas propuestos y el/la docente les asiste cuando encuentran dificultades. Si la problemática resulta recurrente o se dispara una discusión sobre los fundamentos teóricos de la asignatura, se trabaja en pizarrón o mediante el uso de un proyector para trabajar colaborativamente sobre la herramienta de software.

Recursos

Las clases se desarrollan en laboratorios de informática. Los/as estudiantes utilizan su computadora personal o alguna de las disponibles en el laboratorio. De acuerdo al tema tratado, se dispone de proyector para realizar presentaciones en conjunto con la pizarra. Asimismo, se utiliza el Campus virtual de la UNR donde se encuentran disponibles:

- toda la información sobre el dictado de la asignatura (horarios, docentes, programa, avisos, etc).
- las guías de trabajos prácticos y las presentaciones trabajadas en clase.
- videos que cubren todos los temas teóricos de la asignatura,
- foro para consultas,
- soluciones a ejercicios seleccionados.

Actividades de Formación Práctica

Nº	Título	Descripción
1	Lógica Proposicional	Consiste en ejercicios donde el/la estudiante debe formalizar y probar propiedades sobre la lógica proposicional utilizando el asistente de pruebas Coq. Se trabajan los conceptos vistos en asignaturas previas de secuencia, regla de derivación, táctica, esquema de prueba, sistemas de pruebas constructivos y clásicos.

2	Lógica de Predicados	Consiste en problemas donde el/la estudiante debe formalizar y probar propiedades sobre la lógica de predicados utilizando el asistente de pruebas Coq. Se trabaja además el concepto de teoría axiomática y lógica con igualdad, mediante la formalización de los números naturales en Coq.
3	Identificación de pruebas y programas	Problemas que trabajan el isomorfismo de Curry-Howard entre programas funcionales y pruebas de propiedades en lógica constructiva. Mediante una serie de ejercicios, se busca comprender cómo se interpreta una demostración como un objeto computable, y cómo un programa funcional representa necesariamente una demostración (constructiva).
4	Definiciones Inductivas.	Se resuelven problemas de formalización de tipos de datos autorreferenciales. Se trabaja sobre la generación automática de esquemas de recursión y principios de inducción, resolviendo problemas que requieren ambos principios en su solución. Definiciones por recursión y pruebas por inducción en Coq.
5	Inversión	Se resuelven problemas que requieren para su solución esquemas de inversión. Se formalizan estructuras y relaciones complejas y se demuestran propiedades sobre las mismas utilizando dichos esquemas. Formalización de lenguajes de expresiones y sentencias en Coq.
6	Verificación de Programas.	Se resuelven problemas de verificación formal de programas. Mediante problemas integradores de las unidades anteriores se especifica, construye y verifica un conjunto representativo de problemas de programación. Se resuelven problemas mediante diferentes enfoques: - verificación de programas dados el código y su especificación - derivación de programas a partir de pruebas de existenciales.
7	Caso de estudio	Se presenta un problema de informática que representa (un subconjunto de) un sistema crítico de software y se construye un conjunto de funciones verificadas que cumplen con la especificación del problema.

Evaluación

La evaluación se realiza mediante la entrega de ejercicios seleccionados de las prácticas, parciales, presentación y defensa del trabajo práctico 7 (caso de estudio). Se toman dos parciales en laboratorio. El primero abarca los contenidos de las unidades 1 a 3, y el segundo el de las unidades 4 a 5. Para regularizar debe obtenerse nota mayor o igual a 6 en ambos parciales, así como entregar y aprobar todas las entregas. Para esto, debe cumplirse en tiempo y forma con las 6 entregas y obtener una calificación de al menos 8 en cada una de ellas.

Examen final.

Para quienes hayan obtenido la condición de regular, el examen final consiste en la evaluación y defensa oral del trabajo práctico 7, además de un coloquio oral sobre los temas del curso. Para quienes no hayan alcanzado dicha condición, además de las instancias antes descritas, se agrega un examen de práctica integrador sobre ejercicios de todas las unidades, así como un trabajo práctico adicional que consiste en una selección de problemas sobre las unidades 1 a 5.

Resultado de Aprendizaje Actividades/Modalidad de Enseñanza Modalidad de Evaluación

RA1	Clases teóricas y prácticas. Resolución de problemas	Principalmente la evaluación de los temas correspondientes a la unidad 3. Aunque al ser el isomorfismo la base de la teoría sobre la que se construye la herramienta, se trabaja a lo largo de todo el curso con diferente intensidad.
RA2	Clases teóricas y prácticas. Resolución de problemas	Todos los trabajos prácticos y parciales presentan problemas para resolver utilizando un asistente de pruebas, así como también el caso de estudio.
RA3	Clases teóricas y prácticas. Resolución de problemas	Principalmente la evaluación de los temas correspondientes a las unidades 4 a 6, que involucran la formalización y construcción de programas, junto con su verificación formal.
RA4	Clases teóricas y prácticas. Resolución de problemas	Principalmente la evaluación de los temas correspondientes a las unidades 4 a 6, que involucran la formalización y construcción de programas, junto con su verificación formal.
RA5	Clases teóricas y prácticas. Resolución de problemas	Principalmente la evaluación de los temas correspondientes a las unidades 4 a 6, que involucran la formalización y construcción de programas, junto con su verificación formal.
RA6	Resolución de problemas de todas las unidades. Trabajo práctico 7.	Se destaca La evaluación del trabajo final y el correspondiente coloquio. Sin embargo, todas las prácticas presentan problemas que requieren razonar lógicamente y exponer de forma rigurosa soluciones a problemas de informática.

Bibliografía básica

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web
Bertot, Y. Castéran, P.	2004	Interactive Theorem Prover and Program Development.	Springer	1
Barras et al.	2021	The Coq Proof Assistant. Reference Manual	INRIA	https://coq.inria.fr/refman/
Chlipala, A.	2019	Certified Programming with Dependent Types	MIT Press	http://adam.chlipala.net/cpdt/

Bibliografía complementaria

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web
------------------------------------	----------------	-------------------	---------------------	------------------------------------

Distribución de la carga horaria

Presenciales

Teóricas				30 Hs.
Prácticas			Formación Experimental	
			Resolución de Problemas vinculados a la Profesión	
			Resolución de Problemas y Ejercicios	45 Hs.
			Actividades de Proyecto y Diseño	
			Formación en la Práctica Profesional	
Evaluaciones				
			Total	75 Hs.

Dedicadas por el alumno fuera de clase

			Preparación Teórica	15 Hs.
			Preparación Práctica	25 Hs.
			Elaboración y redacción de informes, trabajos, presentaciones, etc.	15 Hs.
			Total	55 Hs.

Cronograma de actividades

Semana	Unidad	Tema	Actividad
1	1	Unidad 1: Asistentes de pruebas para lógicos y matemáticos 1.1.- Lógica proposicional constructiva. Deducción natural.	Introducción a la asignatura. Clases teóricas Práctica 1
2	1	1.2.- Lógica de Predicados. Deducción natural.	Clases teóricas Práctica 2 Primera entrega
3	1	1.3.- Teorías axiomáticas.	Clases teóricas Práctica 2
4	2	Unidad 2: Asistentes de Pruebas para programadores 2.1.- El cálculo lambda como lenguaje de programación funcional.	Clases teóricas Práctica 3 Segunda entrega
5	2	2.2.- El cálculo de construcciones. Unidad 3: Identificación de pruebas y programas 3.1.- Construcción de pruebas como proceso de cálculo.	Clases teóricas Práctica 3
6	3	3.2.- El Isomorfismo de Curry Howard.	Clases teóricas Práctica 3 Tercera entrega
7	4	Unidad 4: Recursión 4.1.- Definiciones inductivas.	Clases teóricas Práctica 4 Parcial 1

8	4	4.2.- Principios de inducción.	Clases teóricas Práctica 4
9	4	4.3.- Esquemas de Recursión.	Clases teóricas Práctica 4 Cuarta entrega
10	4	4.4.- Inversión.	Clases teóricas Práctica 5
11	5	Unidad 5: Extracción y Verificación de Programas 5.1.- Extracción de programas a partir de pruebas.	Clases teóricas Práctica 6 Quinta entrega
12	5	5.2.- Construcción de Pruebas a partir de programas.	Clases teóricas Práctica 6
13	5	5.3.- Construcción de programas certificados.	Clases teóricas Práctica 6 Sextra entrega
14	6	Unidad 6: Caso de Estudio Especificación y verificación de un sistema crítico.	Clases teóricas Práctica 7 Parcial 2
15	6		Práctica 7