



Universidad Nacional de Río Cuarto  
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

## FORMULARIO PARA LA PRESENTACIÓN DE PROGRAMAS DE ASIGNATURAS

Año Lectivo: 2024

**UNIVERSIDAD NACIONAL DE RÍO CUARTO**  
**FACULTAD DE CIENCIAS EXACTAS, FÍSICO-QUÍMICAS Y NATURALES**  
**DEPARTAMENTO DE COMPUTACIÓN**

**CARRERA/S:** Licenciatura en Ciencias de la Computación

**PLAN DE ESTUDIOS:** 2023

**ASIGNATURA:** Algoritmos y Estructura de Datos I

**CÓDIGO:** 3378

**MODALIDAD DE CURSADO:** Presencial

**DOCENTE RESPONSABLE:** Pablo Castro, Dr., Profesor Adjunto, Dedicación Simple.

**EQUIPO DOCENTE:** Cecilia Kilmurray. Dra., JTP, Dedicación Exclusiva.

Laura Tardivo. Dra. Aux. de 1ra, Dedicación SemiExclusiva

Nicolás Streri. Lic. Aux. de 1ra. Dedicación SemiExclusiva.

Agustín Nolasco. Lic. Aux. de 1ra. Dedicación Simple.

**RÉGIMEN DE LA ASIGNATURA:** Cuatrimestral

**UBICACIÓN EN EL PLAN DE ESTUDIO:** Primer año, Segundo Cuatrimestre

**RÉGIMEN DE CORRELATIVIDADES:**

Asignaturas aprobadas: Ninguna

Asignaturas regulares: Introducción a los Algoritmos (3375), Lógica y Resolución de Problemas (3377).

**CARÁCTER DE LA ASIGNATURA:** Obligatoria

**CARGA HORARIA TOTAL:** 112 horas

<b>Teóricas:</b>	<b>56 hs</b>	<b>Prácticas:</b>	<b>28 hs</b>	<b>Teóricas -Prácticas:</b>	<b>.... hs</b>	<b>Laboratorio:</b>	<b>28 hs</b>
------------------	--------------	-------------------	--------------	---------------------------------	----------------	---------------------	--------------

**CARGA HORARIA SEMANAL:** horas

<b>Teóricas:</b>	<b>4 hs</b>	<b>Prácticas:</b>	<b>2 hs</b>	<b>Teóricas -Prácticas:</b>	<b>.... hs</b>	<b>Laboratorio:</b>	<b>2 hs</b>
------------------	-------------	-------------------	-------------	---------------------------------	----------------	---------------------	-------------



## 1. CONTEXTUALIZACIÓN DE LA ASIGNATURA

La materia provee a los estudiantes con conceptos de programación avanzados tales como el desarrollo de programas correctos tanto en programación funcional como en programación imperativa. Así como también algunas técnicas básicas de programación. Estas nociones son necesarias para las materias de tercer año que complementan estos contenidos con técnicas de ingeniería de software y paradigmas de programación. Se hará hincapié en la programación utilizando estructuras de datos básicas. La materia tiene un uso importante de lógica, sobre todo para el razonamiento sobre programas, así como de nociones de programación básicas. Estos conceptos proveen la teoría necesaria para que los graduados puedan desarrollarse en el ámbito profesional con metodologías adecuadas para el desarrollo de software, así como les permite comprender conceptos fundamentales de calidad de software.

## 2. OBJETIVOS PROPUESTOS

Que los alumnos sean capaces de:

- Desarrollar habilidades para el desarrollo de programas correctos,
- Comparar críticamente los paradigmas de programación funcional e imperativo, desde el punto de vista de los métodos rigurosos de desarrollo de programas.
- Desarrollar especificaciones de programas simples.
- Comprender los conceptos básicos de la teoría de autómatas y lenguajes.
- Comprender una teoría básica de estructuras de datos y su utilización para realizar programas simples.

## 3. EJES TEMÁTICOS ESTRUCTURANTES DE LA ASIGNATURA Y ESPECIFICACIÓN DE CONTENIDOS

### 3.1. Contenidos mínimos (según plan de estudio vigente)

Los contenidos de la materia incluyen las nociones básicas para que los alumnos puedan comprender, y usar en la práctica, el concepto de programa correcto. Para este propósito, como primer objetivo, se introduce a los alumnos a un lenguaje lógico que después se utiliza para la demostración de propiedades sobre programas. En particular, se ve un cálculo para la lógica proposicional y de primer orden. haciendo hincapié en su utilización para la resolución de problemas lógicos.

Como segundo tema se aborda el paradigma de programación funcional, los alumnos son introducidos a los conceptos básicos de la programación funcional: modelo computacional, órdenes de reducción, evaluación lazy, el lenguaje Haskell. Además, se hace hincapié en el método de inducción sobre listas que permite que el alumno realice demostraciones básicas de corrección de programas funcionales. Para esto, se hace trabajar al estudiantado con problemas de mediana complejidad, lo cual posibilita que se puedan desarrollar las habilidades matemáticas necesarias para aplicar la teoría en la práctica; además, se requiere a los estudiantes la resolución de un trabajo práctico que incluye la implementación de un programa en un lenguaje funcional o



imperativo avanzado, lo que permite brindar al alumno una perspectiva clara de la utilización de los lenguajes funcionales en la actualidad.

Como tercer tema se aborda la lógica de Hoare y la derivación de programas imperativos correctos. Con este fin, se introducen los conceptos de pre/postcondición, invariantes, aserciones y el transformador de predicados WP. Para lograr que los estudiantes obtengan una perspectiva adecuada de la utilización de dichas técnicas para la programación, se utilizan problemas de mediana complejidad para los cuales se obtienen programas correctos utilizando las técnicas enseñadas. En este tema se utiliza el lenguaje Dafny, un lenguaje que incorpora nociones como pre/post-condiciones e invariantes.

### 3.2. Ejes temáticos o unidades

Lógica y Sistemas Formales, Expresiones booleanas, cálculo proposicional, cálculo de primer orden, resolución de problemas lógicos. Inducción y recursión.

Programación Funcional, Formalismo básico. Modelo computacional. Especificación de programas funcionales. Tipos de datos: listas, árboles y pilas. Órdenes de Reducción. Evaluación Perezosa. Funciones de Orden Superior. Programación básica

en el lenguaje Haskell. Verificación y Especificación de Programas Imperativos, Lógica de Hoare. Construcción de Programas Correctos. El transformador de predicados WP. Un lenguaje simple con guardas. Invariantes. Derivación de Ciclos. Metodología de Programación Dijkstra-Gries. Estructuras de datos básicas, pilas, colas, árboles.

## 4. ACTIVIDADES A DESARROLLAR

**CLASES TEÓRICAS:** Las clases teóricas planificadas (ver más abajo) se dictarán de forma presenciales. 4 horas de clases semanales. también se proveerá a los estudiantes con videos y material audiovisual para el seguimiento de la clase por medio de aulas virtuales.

**CLASES PRÁCTICAS:** Las clases prácticas planificadas (ver más abajo) se dictarán de forma presencial. 2 comisiones de 2 horas de clases semanales.

**CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO:** Las clases de laboratorio planificadas (ver más abajo) se dictarán de forma presencial. 2 comisiones de 2 horas de clases semanales.

**OTRAS:**

## 5. PROGRAMAS Y/O PROYECTOS PEDAGÓGICOS INNOVADORES E INCLUSIVOS



## 6. CRONOGRAMA TENTATIVO DE CLASES E INSTANCIAS EVALUATIVAS

Semana	Día/H oras	Actividad: tipo y descripción*
Semana 1		Repaso de lógica
Semana 2		Introducción Formalismo Funcional
Semana 3		Introducción Haskell
Semana 4		Tipos de Datos en Haskell, Polimorfismo y Clases
Semana 5		Noción de Programas Correctos en Haskell
Semana 6		Inducción
Semana 7		Testing sobre Programas Funcionales – Primer Parcial
Semana 8		Repaso de Programación Imperativa
Semana 9		Nociones de Pre y Post Condiciones
Semana 10		Invariantes, Ejemplos
Semana 11		El Teorema del DO
Semana 12		El lenguaje Dafny – Segundo Parcial
Semana 13		Testing en Lenguajes Imperativos
Semana 14		OO en Dafny

\*Teóricos, teóricos-prácticos, trabajos de laboratorios, salidas a campo, seminarios, talleres, coloquios, instancias evaluativas, consultas grupales y/o individuales, otras.

## 7. BIBLIOGRAFÍA

### 7.1. Bibliografía obligatoria y de consulta

Bibliografía Obligatoria:

- Program Construction, Calculating Implementations from Specifications. Roland Backhouse. Wiley&Sons.
- Cálculo de Programas. Javier Blanco, Silvina Smith y Damián Barsotti. Facultad de Matemática, Astronomía y Física. Universidad Nacional de Córdoba.
- Program Proofs, Rustan Leino, MIT Press, 2023



- Aprende Haskell por el Bien de Todos, libro online, disponible en: <http://aprendehaskell.es/>

Bibliografía de Consulta:

- Introduction to Functional Programming using Haskell. Richard Bird. Prentice Hall Series in Computer Science.

## 7.2. Otros: materiales audiovisuales, enlaces, otros.

Se utilizará la plataforma Google MEET para las clases virtuales. Como aula virtual se utilizará google Classroom, y para repositorios de código se utilizará Github Classroom. Las clases teóricas serán grabadas y estarán disponibles en el aula virtual, de la misma manera la resolución de problemas prototípicos serán grabadas y estarán disponibles en el aula virtual, para comunicación asincrónica se utilizará la aplicación Slack.

## 8. DÍA Y HORARIOS DE CLASES

**Teóricos(C1,C2,C3 yC4):**

**Miércoles 14-16hs**

**Viernes 10-12hs**

**Prácticos:**

**C1 - Jueves 8 a 10 hr**

**C2 - Viernes 12 a 14 hr**

**C3 - Martes 18 a 20 hr**

**C4 - Jueves 12 a 14 hrs**

**Laboratorios:**

**C1 - Viernes 12 a 14 hr**

**C2 - Jueves 8 a 10 hr**

**C3 - Miércoles 16 a 18 hr**

**C4 - Viernes 16 a 18 hrs**

## 9. DÍA Y HORARIO DE CLASES DE CONSULTAS

Los horarios de consulta se acordarán con los estudiantes.

## 10. REQUISITOS PARA OBTENER LA REGULARIDAD Y LA PROMOCIÓN

**CONDICIÓN DE REGULARIDAD:** Aprobación de dos parciales (o sus respectivos recuperatorios). Aprobación de los trabajos prácticos de la materia, los cuales consisten de 2 trabajos prácticos grupales.



Universidad Nacional de Río Cuarto  
Facultad de Ciencias Exactas, Físico-Químicas y Naturales

**CONDICIÓN DE PROMOCIÓN:** Aprobación de los parciales (o sus respectivos recuperatorios) con nota mínima mayor a 6. Aprobación de los trabajos prácticos con nota mayor a 7. Promedio total mayor a 7.

## 11. CARACTERÍSTICAS, MODALIDAD Y CRITERIOS DE LAS INSTANCIAS EVALUATIVAS

En cada parcial se evaluarán los conocimientos obtenidos durante la materia. El examen final podrá ser escrito u oral, el cual será comprensivo y se evaluarán los conceptos dados en la materia. En caso de que sea necesario los docentes podrán pedir que los alumnos resuelvan ejercicios de programación en los lenguajes de programación vistos en la materia.

Debido a que la materia tiene un alto contenido de trabajos de laboratorios que consisten en proyectos de programación con un seguimiento semanal de los profesores de la asignatura. La materia no se puede rendir de forma libre.

Firma Profesor/a Responsable

Firma Secretario/a Académico/a