



**FORMULARIO PARA LA PRESENTACIÓN DE PROGRAMAS DE ASIGNATURAS en
el CONTEXTO DE PANDEMIA por Covid-19¹**

Año Lectivo: 2021

**UNIVERSIDAD NACIONAL DE RÍO CUARTO
FACULTAD DE CIENCIA EXACTAS, FÍSICO-QUÍMICAS y NATURALES
DEPARTAMENTO DE COMPUTACIÓN**

CARRERA/S: Licenciatura en Ciencias de la Computación - Analista en Computación

PLAN DE ESTUDIOS: 1999 (versión vigente)

ASIGNATURA: Programación Avanzada **CÓDIGO:** 1948

MODALIDAD DE CURSADO: Virtual

DOCENTE RESPONSABLE: Dr. Pablo Castro, Prof.Adj. Ded.Simple (con reducción por licencia)

EQUIPO DOCENTE: Lic. Marta Novaira (JTP) Full, Lic. Luciano Putruele Ay. 1ra Simple, Dr. Simon Gutierrez, Ay. 1ra Simple, Prof. E. Cerdá, Ay. de Primera semi.

RÉGIMEN DE LA ASIGNATURA: Cuatrimestral

UBICACIÓN EN EL PLAN DE ESTUDIO: Primer Cuatrimestre, Segundo Año

RÉGIMEN DE CORRELATIVIDADES:

Asignaturas aprobadas: Lógica Matemática Elemental

Asignaturas regulares: Introducción a la Programación y Algorítmica

CARÁCTER DE LA ASIGNATURA: Obligatoria

CARGA HORARIA TOTAL: 168 horas

Teóricas:	56 hs	Prácticas:	56hs	Teóricas -Prácticas: hs	Laboratorio:	56 hs
------------------	--------------	-------------------	-------------	---------------------------------	----------------	---------------------	--------------

CARGA HORARIA SEMANAL: 8 horas (según el plan de estudio vigente)

Teóricas:	4hs	Prácticas:	2hs	Teóricas -Prácticas: hs	Laboratorio:	2 hs
------------------	------------	-------------------	------------	---------------------------------	----------------	---------------------	-------------

¹ Res. CS 120/2017 y Res. CD 049/2020



1. CONTEXTUALIZACIÓN DE LA ASIGNATURA

La materia se encuentra en el segundo año de las carreras de Analista en Computación y Licenciatura en Ciencias de la Computación.

2. OBJETIVOS PROPUESTOS

Que los alumnos sean capaces de:

- Desarrollar habilidades para el desarrollo formal de programas.
- 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 introduce un cálculo de programas funcionales, que permite que el alumno se inicie en el mundo de las derivaciones de programas correctos. 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 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.

Finalmente, se brinda una introducción a los temas básicos de lenguajes y autómatas: Autómatas y lenguajes, Autómatas finitos, expresiones regulares, gramáticas y las nociones básicas de computabilidad. Estos temas proveen a los estudiantes de una visión más amplia sobre la teoría de la computación.

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. Autómatas y Lenguajes, Autómatas Finitos. Expresiones Regulares. Gramáticas. Jerarquía de Chomsky. Nociones básicas de computabilidad.

4. ACTIVIDADES A DESARROLLAR

4.1. Actividades en modalidad virtual (modalidades alternativas a la presencialidad).



CLASES TEÓRICAS: Las clases teóricas planificadas (ver más abajo) se dictarán de forma virtual utilizando la plataforma meet, las mismas se grabarán para que los alumnos que no puedan asistir tengan acceso a las mismas. 4 horas de clases semanales.

CLASES PRÁCTICAS: Las clases prácticas planificadas (ver más abajo) se dictarán de forma virtual utilizando la plataforma meet, las mismas se grabarán para que los alumnos que no puedan asistir tengan acceso a las mismas. 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 virtual utilizando la plataforma meet, las mismas se grabarán para que los alumnos que no puedan asistir tengan acceso a las mismas. 2 comisiones de 2 horas de clases semanales.

OTRAS: Se tomarán dos parciales que deberán ser resueltos de forma presencial, cada uno con un recuperatorio, y un trabajo práctico para resolver en grupo.

4.2. Actividades en la presencialidad

No hay planificadas actividades presenciales.

CLASES TEÓRICAS:-

CLASES PRÁCTICAS:-

CLASES DE TRABAJOS PRÁCTICOS DE LABORATORIO:-

OTRAS:

5. PROGRAMAS Y/O PROYECTOS PEDAGÓGICOS INNOVADORES E INCLUSIVOS

6. CRONOGRAMA TENTATIVO DE CLASES E INSTANCIAS EVALUATIVAS a realizar en la virtualidad y en la presencialidad

6.1. Cronograma de clases e instancias evaluativas a realizar en la virtualidad.

Semana	Dia	Fecha	Hora	Actividad (Tipo y Descripción)
1	Lunes	05/04/21	14hs	Práctico 1: Nociones básicas, recursión
1	Martes	06/04/21	10hs	Laboratorio: Repaso
1	Martes	06/04/21	16hs	Laboratorio: Repaso



1	Miércoles	07/04/21	14hs	Teórico: Nociones básicas
1	Jueves	08/04/21	8hs	Práctico 1: Nociones básicas, recursión
1	Viernes	10/04/21	10hs	Teórico: Nociones básicas, Programación funcional
2	Lunes	12/04/21	16hs	Práctica 2: Paradigma Funcional y Recursión
2	Martes	13/04/21	10hs	Laboratorio: Programación con Recursión: Ejercicios Básicos
2	Martes	13/04/21	16hs	Laboratorio: Programación con Recursión: Ejercicios Básicos
2	Miércoles	14/04/21	14hs	Prog. Funcional, Haskell
2	Jueves	15/04/21	8hs	Práctica 2: Paradigma Funcional y Recursión
2	Viernes	16/04/21	10hs	Teórico: Haskell
3	Lunes	19/04/21	14hs	Practica 3: Haskell
3	Martes	20/04/21	10hs	Laboratorio: Introducción Recursión en Haskell
3	Martes	20/04/21	16hs	Laboratorio: Introducción Recursión en Haskell
3	Miércoles	21/04/21	14hs	Teórico: Lógica
3	Jueves	22/04/21	8hs	Practica 3: Haskell
3	Viernes	23/04/21	10hs	Teórico: Lógica, Especificaciones
4	Lunes	26/04/21	14hs	Practica 3: Haskell
4	Martes	27/04/21	10hs	Laboratorio: Programación en Haskell: Funciones de Alto Orden y Listas
4	Martes	27/04/21	16hs	Laboratorio: Programación en Haskell: Funciones de Alto Orden y Listas
4	Miércoles	28/04/21	14hs	Teórico: Inducción
4	Jueves	29/04/21	8hs	Practica 3: Haskell
4	Viernes	30/04/21	10hs	Teórico: Derivación de Programas Funcionales
5	Lunes	03/05/21	14hs	Practica 4: Lógica y Especificaciones
5	Martes	04/05/21	10hs	Laboratorios: Programación en Haskell: Funciones de Alto Orden y Listas
5	Martes	04/05/21	16hs	Laboratorios: Programación en Haskell: Funciones de Alto Orden y Listas
5	Miércoles	05/05/21	14hs	Teórico: Derivación de Programas Funcionales



5	Jueves	06/05/21	8hs	Practica 4: Lógica y Especificaciones
5	Viernes	07/05/21	10hs	Teórico: Derivación de Programas Funcionales
6	Lunes	10/05/21	14hs	Parcial
6	Martes	11/05/21	10hs	Laboratorio: Programación en Haskell: Definiciones de Tipos Nuevos
6	Martes	11/05/21	16hs	Laboratorio: Programación en Haskell: Definiciones de Tipos Nuevos
6	Miércoles	12/05/21	14hs	Teórico: Derivación de Programas Funcionales
6	Jueves	13/05/21	8hs	Practica 4: Lógica y Especificaciones
6	Viernes	14/05/21	10hs	Teórico: Programación Imperativa, Pre y PostCondiciones
7	Lunes	17/05/21	14hs	Practica 5: Derivación de Programas Funcionales
7	Martes	18/05/21	10hs	Laboratorio: Lógica en Haskell
7	Martes	18/05/21	16hs	Laboratorio: Lógica en Haskell
7	Miércoles	19/05/21	14hs	Recuperatorio Parcial
7	Jueves	20/05/21	8hs	Práctica 5: Derivación de Programas Funcionales
7	Viernes	21/05/21	10hs	Segundo Parcial
8	Lunes	24/05/21	14hs	Práctica 5: Derivación de Programas Funcionales
8	Martes	25/05/21	10hs	Laboratorio: Derivación de Programas en Haskell
8	Martes	25/05/21	16hs	Laboratorio: Derivación de Programas en Haskell
8	Miércoles	26/05/21	14hs	Teórico: Programación Imperativa, Desarrollo de Invariantes
8	Jueves	27/05/21	8hs	Práctica 5: Derivación de Programas Funcionales
8	Viernes	28/05/21	10hs	Recuperatorio Segundo Parcial
9	Lunes	31/05/21	14hs	Práctica 5: Derivación de Programas Funcionales
9	Martes	01/06/21	10hs	Laboratorio: Derivación de Programas en Haskell
9	Martes	01/06/21	16hs	Laboratorio: Derivación de Programas en Haskell
9	Miércoles	02/06/21	14hs	Teórico: Programación Imperativa, Derivación de Programas
9	Jueves	03/06/21	8hs	Práctica 5: Derivación de Programas Funcionales



9	Viernes	04/06/21	10hs	Teórico: Programación Imperativa, Derivación de Programas
10	Lunes	07/06/21	14hs	Práctica 6: Programación Imperativa, Pre y PostCondiciones
10	Martes	08/06/21	10hs	Laboratorio: Programación Imperativa, Pre y PostCondiciones
10	Martes	08/06/21	16hs	Laboratorio: Programación Imperativa, Pre y PostCondiciones
10	Miércoles	09/06/21	14hs	Teórico: Programación Imperativa, Derivación, Ejemplos
10	Jueves	10/06/21	8hs	Práctica 6: Programación Imperativa, Pre y PostCondiciones
10	Viernes	11/06/21	10hs	Teórico: Programación Imperativa, Derivación Ejemplos
11	Lunes	14/06/21	14hs	Practica 7: Programación Imperativa, Invariantes
11	Martes	15/06/21	10hs	Laboratorio: Programación Imperativa, Invariantes
11	Martes	15/06/21	16hs	Laboratorio: Programación Imperativa, Invariantes
11	Miércoles	16/06/21	14hs	Teórico: Programación Imperativa, Variantes y Cotas
11	Jueves	17/06/21	8hs	Practica 7: Programación Imperativa, Invariantes
11	Viernes	18/06/21	10hs	Teórico: Programación Imperativa, Fortalecimiento de Invariantes
12	Lunes	21/06/21	14hs	Practica 7: Programación Imperativa, Invariantes
12	Martes	22/06/21	10hs	Laboratorio: Programación Imperativa, Derivaciones de Programas
12	Martes	22/06/21	16hs	Laboratorio: Programación Imperativa, Derivaciones de Programas
12	Miércoles	23/06/21	14hs	Teórico: Introducción a Autómatas y Lenguajes
12	Jueves	24/06/21	8hs	Practica 7: Programación Imperativa, Invariantes
12	Viernes	25/06/21	10hs	Teórico: Introducción a Autómatas y Lenguajes
13	Lunes	18/06/21	14hs	Practica 7:Autómatas y Lenguajes
13	Martes	29/06/21	10hs	Laboratorio: Trabajo Práctico
13	Martes	29/06/21	16hs	Laboratorio: Trabajo Práctico
13	Miércoles	30/06/21	14hs	Teórico: Introducción a Autómatas y Lenguajes



13	Jueves	01/07/21	8hs	Practica 7:Autómatas y Lenguajes
13	Viernes	02/07/21	10hs	Teórico: Introducción a Computabilidad
14	Lunes	05/07/21	14hs	Practica 7:Autómatas y Lenguajes
14	Martes	06/07/21	10hs	Laboratorio: Herramientas para el procesamiento de Lenguajes
14	Martes	06/07/21	16hs	Laboratorio: Herramientas para el procesamiento de Lenguajes
14	Miércoles	07/07/21	14hs	Teórico: Introducción a Computabilidad
14	Jueves	08/07/21	8hs	Practica 7:Autómatas y Lenguajes
14	Viernes	08/07/21	10hs	Repaso de los temas vistos.

6.2. Cronograma de clases e instancias evaluativas a realizar en la presencialidad.

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.
- Introduction to the Theory of Computation. Michael Sipser. PWS Pub. Co.
- 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. Plataformas/herramientas virtuales; 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 VIRTUALES y PRESENCIALES

Clases Teóricas:

- **Miércoles 14-16hs**
- **Viernes 10-12hs**

Clases Prácticas:

- **Lunes 14-16hs**
- **Jueves 8-10hs**

Laboratorios:

- **Martes 10-12hs**
- **Martes 16-18hs**

9. DÍA Y HORARIO DE CLASES DE CONSULTAS VIRTUALES y PRESENCIALES

Se dan consultas virtuales los siguientes días:

Lunes 18hs

Martes 12hs

Miércoles 12hs

Jueves 12hs

Viernes 18:30hs

10. REQUISITOS PARA OBTENER LA REGULARIDAD Y LA PROMOCIÓN

Evaluaciones Parciales: 2 parciales.

- **Evaluación Final: El oral consta de una evaluación oral o escrita.**
- **CONDICIONES DE REGULARIDAD: Aprobación de los parciales y el trabajo práctico con nota de más de 5 (cinco).**



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 que sea necesario los docentes podrán pedir que los alumnos resuelvan ejercicios de programación en los lenguajes de programación vistos en la materia. La materia se puede rendir de forma libre, para la cual se le podrá pedir al alumno que resuelva un trabajo práctico para demostrar el manejo de las herramientas de programación vistas en clase.

Firma Profesor/a Responsable

Firma Secretario/a Académico/a