

# Planificación de Programación I



Código/s: R-113

## Identificación y características del Espacio Curricular

Carrera/s:	Licenciatura en Ciencias de la Computación		
Plan de Estudios:	2010, TO2024	Carácter:	Obligatoria
Bloque/Campo:		Área:	Algoritmos y Lenguajes
Régimen de cursado:	Cuatrimestral		
Cuatrimestre:	1º [LCC], 1º [LCC]		
Carga horaria:	135 hs. / 9 hs. semanales	Formato curricular:	Asignatura
Escuela:	Ciencias Exactas y Naturales	Departamento:	Ciencias de la Computación
Docente responsable:	MANZINO, Cecilia		

## Programa Sintético

Elementos básicos de un lenguaje de programación. Definiciones y expresiones. Evaluación de expresiones. Expresiones condicionales. Tipos de datos básicos. Diseño de programas. Metodologías de diseño de funciones y buenas prácticas de programación. Programación interactiva. Eventos y manejadores de eventos. Tipos de datos compuestos: estructuras. Programación con estructuras. Datos de largo arbitrario: listas. Programación con listas. Recursión. Patrones. Números naturales como tipo de dato inductivo. Programación recursiva con naturales.

## Espacios Curriculares Relacionados

Previos Aprobados:

Simultaneos Recomendados:

Posteriores: R-123 - Programación II

## Vigencia desde 2024

\_\_\_\_\_  
Firma Profesor

\_\_\_\_\_  
Fecha

\_\_\_\_\_  
Firma Aprob. Escuela

\_\_\_\_\_  
Fecha

Con el aval del Consejo Asesor:

## Fundamentación

La resolución de problemas de programación es un área fundamental para una carrera de Ciencias de la Computación. En particular, en esta asignatura se utiliza un enfoque fuertemente práctico en la resolución de problemas seleccionados para cada unidad temática. Como paradigma de programación se presenta la programación funcional, utilizando DrRacket como lenguaje de programación, el cual es un lenguaje diseñado para la enseñanza.

Se pone especial atención en técnicas de diseño sistemático de programas y buenas prácticas de programación, de manera que el estudiante pueda incorporarlas en una etapa temprana de su carrera.

Los aportes concretos de la asignatura dentro de la carrera son:

- Iniciación en la aplicación de técnicas de resolución de problemas a problemas de programación (descomposición, abstracción, generalización).
- Diseño de programas.
- Comprensión de los elementos básicos de la programación funcional.
- Incorporación de buenas prácticas de programación.
- Manejo de estructuras de datos fundamentales.
- Identificación de estado de un programa interactivo. Uso de manejadores de eventos como transformadores de estado.

La asignatura se encuentra en el primer cuatrimestre de primer año,

## Resultados del aprendizaje

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

RA1 Aplicar distintas estrategias para la resolución de problemas simples bajo el paradigma funcional.
RA2 Emplear técnicas básicas de diseño de programas.
RA3 Aplicar buenas prácticas de programación en contextos sencillos.
RA4 Elegir y manipular estructuras de datos adecuadas para representar información del mundo real.
RA5 Diseñar programas interactivos simples que permitan comunicarse con los diferentes dispositivos de entrada y salida de una computadora (mouse, teclado, pantallas, sistema de archivos).
RA6 Utilizar la recursión para manipular números naturales y listas.
RA7 Utilizar patrones de programación como mecanismo de abstracción.
RA8 Desarrollar trabajos en grupos de manera colaborativa, aportando ideas y habilidades para contribuir a un resultado colectivo

## 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	Bajo	RA1, RA5, RA6 y RA7
CGT4-Utilización de técnicas y herramientas de aplicación en la informática	Bajo	RA2, RA3, RA4
CGS1-Fundamentos para el desempeño en equipos de trabajo	Bajo	RA8

## Programa Analítico

En sintonía con las expectativas de logro, resultados de aprendizaje y ejes transversales señalados, consignar los contenidos que se prevén desarrollar en el espacio curricular. Indicarlos mediante el siguiente formato:

### Unidad 1

- 1.1. Elementos de un lenguaje de programación.
- 1.2. Definiciones y expresiones.
- 1.3. Evaluación de expresiones.
- 1.4. Introducción a un lenguaje de programación funcional: sintaxis, entorno de ejecución, formas de evaluación.
- 1.5. Tipos de datos básicos: numéricos, strings, imágenes, valores booleanos.
- 1.6. Predicados.
- 1.7. Expresiones condicionales. Condicionales simples y compuestos. Orden de evaluación para expresiones condicionales no determinísticas.

### Unidad 2

- 2.1. Diseño de programas.
- 2.2. Presentación de una metodología de diseño de programas.
- 2.3. Buenas prácticas de programación.
- 2.4. Documentación, descomposición, abstracción, diseño para el cambio.

### Unidad 3

- 3.1. Programación interactiva.
- 3.2. Eventos. Diferentes clases de eventos: paso del tiempo, dispositivos de entrada (teclado, mouse, touchpad).
- 3.3. Concepto de estado de un programa interactivo. Manejadores de eventos como transformadores de estado. Asociación entre eventos y manejadores.

### Unidad 4

- 4.1. Tipos de datos compuestos.
- 4.2. Estructuras. Constructores y selectores. Predicados. Operaciones derivadas.
- 4.3. Diseño de programas con estructuras. Programación interactiva con estructuras como estado.

### Unidad 5

- 5.1. Datos de largo arbitrario. Motivación.
- 5.2. Definición de listas. Constructores y selectores. Predicados. Otras operaciones.
- 5.3. Programación con listas.
- 5.4. Recursión.
- 5.5. Patrones de diseño como mecanismo de abstracción: los patrones map, fold y filter.
- 5.6. Programación interactiva con listas como parte del estado.

### Unidad 6

- 6.1. Números naturales. Constructores y selectores. Predicados. Otras operaciones.
- 6.2. Programación con naturales.
- 6.3. Recursión.
- 6.4. Generación de datos de largo arbitrario a partir de números naturales.

### Unidad 7

- 7.1. Problemas integradores. Diseño de programas de complejidad media: método de Montecarlo, test de primalidad, elementos básicos de criptografía, aproximación de funciones, satisfacibilidad en Lógica

Proposicional y experimentos aleatorios.

### Modalidades de enseñanza

Los contenidos teóricos del curso se presentan en una clase semanal en el aula, mientras que las y los estudiantes resuelven problemas de programación en dos clases semanales de laboratorio con el apoyo de los docentes y una clase práctica en aula, donde se resuelven problemas seleccionados de las prácticas.

Se fija por parte de cada docente una hora semanal de consulta.

En las clases de laboratorio las y los estudiantes pasan gran parte del cursado de la materia resolviendo problemas de acuerdo a la temática dada, esta modalidad de enseñanza se relaciona con los resultados de aprendizajes RA1, RA2, RA3, RA4 y RA5. Mientras que las clases de práctica y teoría se relacionan más con los resultados RA3, RA4, RA5 y RA6.

La última práctica de la materia contiene problemas integradores de complejidad media, para rendir el final de la materia las y los estudiantes deben presentar en forma de trabajo práctico grupal alguno de los problemas propuestos en esta práctica, el resultado de aprendizaje RA8 está relacionado con la elaboración de dicho trabajo práctico.

### Recursos

Se utiliza un proyector en las clases teóricas, en la clase de práctica en aula y en los laboratorios. Para las clases de laboratorio se utilizan 3 laboratorios informáticos a razón de una pc cada dos o tres estudiantes.

El material de estudio, que incluye apuntes teóricos, prácticas y trabajos prácticos se publica en el campus virtual (Comunidades) de la materia. Se utiliza el campus virtual para la entrega de trabajos prácticos y de exámenes en laboratorio.

Se usa un canal de Telegram para la comunicación con las y los estudiantes.

### Actividades de Formación Práctica

Nº	Título	Descripción
1	Práctica 0	Ejercicios sobre definiciones de expresiones en Racket, tipos de datos básicos y operaciones sobre los mismos.
2	Práctica 1	Ejercicios sobre expresiones condicionales simples y compuestas. Evaluación de expresiones condicionales y predicados.
3	Práctica 2	Ejercicios sobre diseño de programas.
4	Práctica 3	Ejercicios sobre programas interactivos.
5	Práctica 4	Ejercicios sobre estructuras y programas interactivos que utilizan estructuras como estado.
6	Práctica 5	Ejercicios sobre listas, recursión sobre listas, patrones de listas y Programación interactiva con listas como estado.
7	Práctica 6	Ejercicios sobre números naturales como tipo de datos inductivo y programación recursiva con naturales.
8	Práctica 7	Problemas integradores. Ejercicios con formato de trabajo práctico que requieren la aplicación de todos los conceptos dados en la materia para resolver un problema concreto.

## Evaluación

La evaluación durante el cursado se realiza mediante dos exámenes parciales. El primer parcial corresponde a las unidades 1 a 3 y el segundo a las unidades 4 y 5.

Durante la última semana de clases se contempla un examen recuperatorio con carácter sustitutivo de uno de los parciales (aquel donde el estudiante hubiera obtenido menor nota).

El estudiante obtiene la condición de promovido si el promedio de sus notas en los exámenes parciales (considerando el recuperatorio si existiese) es al menos 8 y ninguna de dichas notas es inferior a 6.

El estudiante obtiene la condición de regular si el promedio de sus notas en los exámenes parciales (considerando el recuperatorio si existiese) es al menos 6, ninguna de dichas notas es inferior a 4 y no obtuvo la condición de promovido.

Quienes no cumplan estos requisitos obtienen la condición de libre.

El examen final consiste en la entrega de un trabajo práctico grupal, el cual consiste en la resolución de un problema integrador de la práctica 7, una evaluación práctica en laboratorio y un examen teórico en papel que incluye una defensa escrita del trabajo práctico entregado.

La diferencia entre de los exámenes para las distintas condiciones es la longitud del examen práctico de laboratorio. Quienes hayan alcanzado la condición de promovido sólo rinden los temas de las prácticas no evaluadas en los parciales, es decir las prácticas 6 y 7, mientras que quienes estén en condición regular o libre rinden los temas dados en todas las prácticas, siendo más corto el examen de regular.

Resultado de Aprendizaje	Actividades/Modalidad de Enseñanza	Modalidad de Evaluación
RA1	Clases teóricas, resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico.
RA2	Clases teóricas, resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico, examen escrito de teoría y resolución del tp.
RA3	Resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico y resolución del tp.
RA4	Resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico.
RA5	Clases teóricas, resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico.
RA6	Clases teóricas, resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico, examen escrito de teoría y resolución del tp.
RA7	Clases teóricas, resolución de problemas en clases de práctica en aula y práctica en laboratorio.	Ejercicios en laboratorio en parciales y examen final práctico, examen escrito de teoría y resolución del tp.
RA8	Práctica en laboratorio.	Examen final teórico escrito.

## Bibliografía básica

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

Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi	2018 (second Edition)	How to design programs	MIT Press	<a href="https://htdp.org/2003-09-26/">https://htdp.org/2003-09-26/</a>
Matthew Flatt y Robert Bruce Findler		The Racket Guide		<a href="https://docs.racket-lang.org/guide/">https://docs.racket-lang.org/guide/</a>

### Bibliografía complementaria

Autores (Apellido, Inicial nombre)	Año de edición	Título de la obra	Editorial o Revista	Ejemplares disponibles o sitio web
Harold Abelson, Gerald Jay Sussman, Julie Sussman	1996 (Second Edition)	Structure and interpretation of computer programs	MIT Press	1

### Distribución de la carga horaria

#### Presenciales

Teóricas				54 Hs.
Prácticas			Formación Experimental	
			Resolución de Problemas vinculados a la Profesión	
			Resolución de Problemas y Ejercicios	75 Hs.
			Actividades de Proyecto y Diseño	
			Formación en la Práctica Profesional	
Evaluaciones				6 Hs.
			<b>Total</b>	<b>135 Hs.</b>

#### Dedicadas por el alumno fuera de clase

			Preparación Teórica	20 Hs.
			Preparación Práctica	50 Hs.
			Elaboración y redacción de informes, trabajos, presentaciones, etc.	10 Hs.
			<b>Total</b>	<b>80 Hs.</b>

### Cronograma de actividades

Semana	Unidad	Tema	Actividad
1	1	Elementos de un lenguaje de programación. Definiciones y expresiones. Evaluación de expresiones. Introducción a un lenguaje de programación funcional: sintaxis, entorno de ejecución, formas de evaluación. Tipos de datos básicos. (1.1. a 1.5.)	Práctica 0
2	1	Predicados. Expresiones condicionales. (1.6 y 1.7)	Práctica 1

3	2	Diseño de programas. Presentación de una metodología de diseño de programas. (2.1 y 2.2)	Práctica 2
4	2	Buenas prácticas de programación. Documentación, descomposición, abstracción, diseño para el cambio. (2.3. y 2.4.)	Práctica 2
5	3	Programación interactiva. Eventos. Diferentes clases de eventos: paso del tiempo, dispositivos de entrada. Concepto de estado de un programa interactivo. Manejadores de eventos como transformadores de estado. Asociación entre eventos y manejadores. (3.1. a 3.3.)	Práctica 3
6	3	1° Parcial	Práctica 3
7	4	Tipos de datos compuestos. Estructuras. Constructores y selectores. Predicados. Operaciones derivadas. Diseño de programas con estructuras. Programación interactiva con estructuras como estado. (4.1. a 4.3.)	Práctica 4
8	5	Datos de largo arbitrario. Motivación. Definición de listas. Constructores y selectores. Predicados. Otras operaciones. Programación con listas. Recursión. (5.1. a 5.4.)	Prácticas 4 y 5
9	5	Patrones de diseño como mecanismo de abstracción: los patrones map, fold y filter. (5.5.)	Práctica 5
10	5	Programación interactiva con listas como parte del estado. (5.6.)	Práctica 5
11	5	2° Parcial	Práctico 5
12	6	Números naturales. Constructores y selectores. Predicados. Otras operaciones. Programación con naturales. Recursión. Generación de datos de largo arbitrario a partir de números naturales. (6.1. a 6.4.)	Práctica 6
13	7	Problemas integradores. (7.1.)	Prácticas 6 y 7
14	7	Más sobre problemas integradores. (7.1.)	Práctica 7
15	7	Recuperatorios	