



Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2026)
(Programa en trámite de aprobación)
(Presentado el 12/08/2026 09:27:09)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACIÓN I	LIC.CS.COMP.	RD-3 -1/20 23	2026	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BAIGORRIA FERNANDEZ, LORENA S.	Prof. Responsable	P.Adj Exc	40 Hs
ABDELAHAD, CORINA NATALIA	Prof. Colaborador	P.Adj Exc	40 Hs
GAUNA, DARIO EMILIO	Auxiliar de Práctico	A.1ra Simp	10 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	Teóricas	Prácticas de Aula	Práct. de lab/ camp/ Resid/ PIP, etc.	Total
2 Hs	Hs	2 Hs	3 Hs	7 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	2° Cuatrimestre

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
03/08/2026	13/11/2026	15	105

IV - Fundamentación

Introducir al estudiante en el paradigma de la programación imperativa, abordando todos los temas de este paradigma, tales como estructuras de control a nivel de sentencias, variables simples y estructuradas, tipos de datos predefinidos y definidos por el usuario y sus usos en resolución de problemas, registros, archivos, estructuras de control a nivel de unidades, parámetros formales y actuales, así como su aplicación en programas en un lenguaje que responda al paradigma, además de la práctica a través de laboratorios de implementación y ejecución de programas cortos permitirá al estudiante desarrollar las habilidades necesarias para poder construir programas en cualquier lenguaje de este paradigma

V - Objetivos / Resultados de Aprendizaje

Al finalizar el curso se espera que el/la estudiante sea capaz de:

- Utilizar eficaz y eficientemente un enfoque de programación basado en el paradigma imperativo.
- Aplicar los principales conceptos del paradigma como por ejemplo los vinculados a las estructuras de control a nivel de sentencias y subprogramas, tipos de datos y control de datos.
- Diseñar, codificar y depurar programas de complejidad moderada utilizando un lenguaje representativo del paradigma.
- Diseñar y codificar soluciones recursivas usando un lenguaje imperativo.

Durante el dictado de la asignatura se abordan los siguientes ejes transversales:

- Identificación, formulación y resolución de problemas de informática.
- Utilización de técnicas y herramientas de aplicación en la informática.
- Fundamentos para el desempeño en equipos de trabajo.
- Fundamentos para la comunicación efectiva.
- Fundamentos para la acción ética y responsable.
- Fundamentos para el aprendizaje continuo.

VI - Contenidos

Paradigmas y lenguajes de programación. Programación imperativa. Datos elementales. Estructuras de datos. Tipo de datos predefinidos y definidos por el usuario y sus usos para la resolución de problemas. Ambientes de referenciación. Estructuras de control. Modularidad, parámetros formales y reales. Recursividad. Representación de datos en memoria. Estrategias de Implementación. Manejo de Memoria en ejecución. Eficiencia, legibilidad y reusabilidad de algoritmos. Entrada/Salida. Tipo de datos abstractos.

Estos contenidos mínimos se desglosan en las siguientes unidades:

Unidad I Datos elementales. Estructuras de datos y Operadores

Paradigma de programación Imperativa. Concepto de lenguaje de programación. Introducción conceptual al Paradigma de Programación Imperativa utilizando el lenguaje C. Historia y características del lenguaje C. Fundamentos de C. Operadores y expresiones. Tipos de Datos. Variables simples y estructuradas: registros y arreglos. Tipos de datos predefinidos y definidos por el usuario y sus usos en resolución de problemas. Enfoque conceptual y aspectos operativos de las sentencias de control.

Unidad II Estructuras de Control

Programación estructurada. Estructuras de control básicas y compuestas. Estructuras de control a nivel de unidades. Programación estructurada utilizando C. Modularidad. Concepto de módulo. Definición y uso de funciones en C. Distintos tipos de funciones. Pasaje de parámetros. Parámetros formales y reales. Ambientes de referenciación.

Unidad III Representación de datos en memoria. Estrategias de Implementación. Manejo de Memoria en ejecución

Punteros. Naturaleza conceptual de un puntero. Conceptos de “apunta a” y “apuntado por”. Funcionamiento de punteros en C. Arreglos y asignaciones estáticas de memoria. Arreglos en C. Arreglos y punteros en C. Concepto de “cadena de caracteres”. Manejo conceptual de las cadenas de caracteres. Singularidades del manejo de cadenas de caracteres en C.

Unidad IV Recursión

Eficiencia, legibilidad y reusabilidad de algoritmos. Concepto de recursividad. Asignación de memoria (stack) durante la ejecución de un algoritmo recursivo. Recursividad en C.

Unidad V Tipo de Datos Abstracto

Concepto de “tipo de dato abstracto”. Ejemplo de “tipo de dato abstracto”. Manejo de tipos abstractos de datos en C. Listas como ejemplo de asignación dinámica de memoria. Asignación de memoria en el cúmulo (heap). Listas en lenguaje C. Modificaciones de listas. Pilas y sus aplicaciones. Colas y sus aplicaciones.

Unidad VI Entrada Salida

Entrada/Salida. Entrada Salida Standard. Salida con formato. Argumentos de longitud variable. Entrada con formato. Manejo de Archivos. Manejo de errores. Descriptores de archivos. Operaciones de acceso a archivos.

VII - Plan de Trabajos Prácticos

Metodología de enseñanza:

Para cada unidad se deja disponible el material correspondiente a los contenidos de la unidad, las diapositivas de clase, el apunte teórico y su correspondiente trabajo práctico en el sitio web de la materia.

Trabajo Práctico I: Operadores y expresiones. Tipos de Datos. Variables simples. Estructuras de control básicas y compuestas.

Trabajo Práctico II: Modularidad. Definición y uso de funciones en C. Pasaje de parámetros. Punteros en C.

Trabajo Práctico III: Arreglos y punteros en C.

Trabajo Práctico IV: Registros en C.

Trabajo Práctico V: Tipos de datos abstractos

Trabajo Práctico VI: Recursividad.

Trabajo Práctico VII: Datos recursivos.

Trabajo Práctico VIII: Entrada/Salida.

Práctico Integrador realizado en grupos de dos o tres integrantes

Abordaje de los Ejes transversales:

El eje “Identificación, formulación y resolución de problemas de informática” es abordado desde el practico 1 hasta el práctico integrador donde el estudiante debe comprender, consultar y realizar un bosquejo de la solución propuesta en papel para luego codificarlo en lenguaje C. Esto se realiza en forma incremental y con la “Utilización de técnicas y herramientas de aplicación en la informática” provistas por el lenguaje C y los conceptos adquiridos en la materia. Se realiza el seguimiento continuo de los estudiantes a través de la corrección de actividades y la participación en la resolución de ejercicios en la pizarra por parte de los estudiantes.

Para la resolución de los trabajos prácticos los estudiantes pueden revisar tanto la bibliografía proporcionada por la materia, los materiales y videos explicativos, así como utilizar material adicional de sitios referenciados por los docentes, impulsando la búsqueda y análisis de información de manera tal que antes de utilizarlo, se verifique la fuente e incluso pueda ser validada por los docentes. Esto colabora y promueve el aprendizaje continuo.

Así mismo se realiza el práctico integrador de laboratorio el cual es obligatoriamente grupal. De esta manera se intenta que los estudiantes se integren y formen equipos de trabajo con distribución de tareas. Para esto, los docentes de la materia realizan el seguimiento del avance del proyecto y del trabajo en equipo colaborando en la resolución de posibles conflictos y brindándole “Fundamentos para el desempeño en equipos de trabajo” así como es posible evaluar la aplicación de “Fundamentos para la acción ética y responsable”, al solicitar las entregas de los controles correspondientes al práctico integrador, esto se ve reflejado en el desarrollo exitoso del mismo.

El coloquio de promoción o el examen final (salvo causa mayor) se evalúa de manera oral, para esto en clases de consulta se proporciona las guías de cómo preparar el examen dándole la oportunidad de preparar un tema inicial para comenzar la evaluación. Luego de la evaluación se realiza una devolución al estudiante indicándole fortalezas y debilidades con el fin de mejorar su desempeño y mejorar la “comunicación efectiva”

VIII - Regimen de Aprobación

Condiciones para regularizar la materia:

- Aprobar los dos exámenes parciales o sus respectivas recuperaciones con nota mayor o igual a 6 (seis).
- Contar con al menos el 70% de asistencia a clases.
- Aprobar el práctico integrador con nota mayor o igual a 6 (seis)

Condiciones para promocionar la materia:

Regularizar la materia con las siguientes condiciones adicionales:

- Aprobar los dos exámenes parciales o sus respectivas recuperaciones con nota mayor o igual a 7 (siete).
- Contar con al menos el 80% de asistencia a clases.
- Aprobar el práctico integrador con nota mayor o igual a 7 (siete)
- Aprobar un coloquio oral con nota mayor o igual que 7 (siete)

En caso de no promocionar el alumno deberá rendir un examen final oral.

No se admiten exámenes libres debido al carácter eminentemente práctico de la materia.

Nota: Se otorga, tal como lo expresa la reglamentación vigente, dos recuperaciones por parcial a todos los estudiantes.

IX - Bibliografía Básica

[1] Aristides Dasso & Ana Funes, Introducción a la Programación: Notas de clase para el curso Programación I, 2014.

[2] Modern C (2nd Edition); Jens Gustedt; Manning Publications; 2019.

[3] Effective C: An Introduction to Professional C Programming; Robert C. Seacord; No Starch Press; 2020.

[4] C How to Program (9th Edition); Paul Deitel & Harvey Deitel; Pearson; 2022.

[5] JTC1/SC22/WG14 - C Enlace: <https://www.open-std.org/jtc1/sc22/wg14/> Sitio Estándare

[6] Tutorial de C online <https://www.w3schools.com/c/index.php>

[7] Presentaciones y videos explicativos realizados por el equipo docente disponibles en proguno.unsl.edu.ar

X - Bibliografía Complementaria

[1] Data structures and program design in C; Kruse, Robert; Ed. Pearson; 2007.

[2] Algoritmos + Estructuras de Datos = Programas; N. Wirth; Ed. Del Castillo; 1999.

XI - Resumen de Objetivos

Conocer los principales conceptos del paradigma de programación imperativa y desarrollar una experiencia de programación donde el alumno sea capaz de diseñar, codificar y depurar programas de complejidad moderada utilizando un lenguaje de programación representativo del paradigma imperativo.

XII - Resumen del Programa

Concepto de lenguaje de programación. Paradigma de Programación Imperativa. Operadores y expresiones. Variables. Estructuras de control básicas y compuestas. Tipos de Datos. Arreglos. Registros. Programación estructurada. Modularidad. Ambientes de referenciación. Punteros. Asignación estática y dinámica de la memoria. Tipo de dato abstracto. Listas, pilas y colas. Recursividad. Entrada/Salida.

XIII - Imprevistos

Sitio web:

<http://proguno.unsl.edu.ar>

Contacto:

programacion1.unsl@gmail.com

Puede encontrar a los docentes de la materia: en oficina 1, 2 y 27 del dpto de informática.
1er piso, bloque 2.

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	