Ministerio de Cultura y Educación **(Programa del año 2026)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
PROGRAMACION I	TCO.UNIV.EN WEB	08/13	2026	2° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
PEREZ, NORMA BEATRIZ	Prof. Responsable	P.Adj Exc	40 Hs
CUELLO, CAROLINA YAMILE	Auxiliar de Práctico	A.1ra Semi	20 Hs
GATICA, CLAUDIA RUTH	Auxiliar de Práctico	A.1ra Semi	20 Hs
LEIVA OLGUIN, GASTON EMANUEL	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**
Hs	2 Hs	2 Hs	2 Hs	6 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
03/08/2026	13/11/2026	15	90

IV - Fundamentación

La carrera Tecnicatura Universitaria en Web, representa una instancia formativa de significativa relevancia institucional la cual permite a los egresados incorporar nuevos conocimientos tecnológicos que orienten en los adelantos y en las aplicaciones que son necesarias para el desarrollo profesional. La carrera está orientada hacia el estudio sistemático e ingenieril para el desarrollo, evaluación, distribución y mantenimiento de aplicaciones Web. La carrera tiene entre sus metas que el estudiante adquiera los conocimientos teórico-prácticos necesarios para el desarrollo y mantenimiento de aplicaciones Web. Las bases de estas tareas se centran en el desarrollo de habilidades en el manejo de técnicas y lenguajes de programación. En Programación I se desarrollan las tareas didácticas tendientes al cumplimiento de estos objetivos específico.

V - Objetivos / Resultados de Aprendizaje

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

* Comprender y aplicar los fundamentos del paradigma de programación imperativa para la resolución de problemas informáticos.
* Analizar problemas y diseñar soluciones algorítmicas utilizando adecuadamente estructuras de control, variables y tipos de datos simples y estructurados.
* Aplicar estructuras de control a nivel de sentencias y subprogramas, utilizando correctamente parámetros formales y actuales.

* Diseñar, codificar, ejecutar, probar y depurar programas de complejidad moderada mediante un lenguaje representativo del paradigma imperativo.
* Utilizar tipos de datos predefinidos y definidos por el usuario, registros y archivos, seleccionando las estructuras más apropiadas según el problema.
* Diseñar e implementar soluciones iterativas y recursivas, evaluando su pertinencia en distintos contextos.
* Desarrollar habilidades en el uso de técnicas, herramientas y lenguajes de programación que constituyan una base para asignaturas posteriores y para el desarrollo de aplicaciones.
* Formular y resolver problemas de manera sistemática, promoviendo el razonamiento lógico, la abstracción y el aprendizaje continuo.
* Fortalecer competencias para el trabajo colaborativo, la comunicación efectiva y el desempeño ético y responsable en el ámbito de la informática.

VI - Contenidos

Unidad I: Introducción al Lenguaje de Programación C.
Orígenes, objetivos. Acciones y descripción de datos. Datos: estándar, escalares. Definición y declaración. Constantes, variables, tipos. Acciones: Sentencias de asignación, iteración, selección, sentencias compuestas.

Unidad II: Modularidad.
Concepto de módulo. Lenguaje de programación C: funciones. Declaración, invocación, parámetros. Pasaje de parámetros. Resolución de problemas y uso de funciones.

Unidad III: Arreglos, estructuras y Punteros.
Estructuras de Datos el arreglo y registro (estructura). Orden de sus elementos: cronológico, no cronológico. Capacidad: dinámica, estática. Operaciones. Definición de 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. Registros (struct) y punteros.

Unidad IV: Recursividad.
Conceptos Generales, stack de ejecución y aplicaciones. C: funciones recursivas.

Unidad V: Estructuras de Datos Pilas, Filas y Listas.
Pilas, Filas y listas: orden de sus elementos, capacidad, operaciones y Predicados. Listas unidireccionales: Generalidad de las listas. Extensiones al lenguaje C para manejo de estructuras dinámicas.

Unidad VI: Implementación de Estructuras de Datos Dinámicas: PILAS, FILAS y LISTAS.
Concepto de “tipo de dato abstracto”. Ejemplo de “tipo de dato abstracto”. Manejo de tipos abstractos de datos en C. Implementación de pilas con y sin desplazamiento. Implementación de filas o colas: con y sin desplazamiento. Implementación de listas con y sin desplazamiento.

Unidad VII: Archivos.
Conceptos generales. Operaciones.

Unidad VIII: Datos recursivos.
Conceptos Generales. Implementación de estructuras de datos con datos recursivos empleando el tipo puntero de C.

La propuesta metodológica articula fundamentos teóricos, pensamiento computacional y resolución de problemas mediante actividades prácticas progresivas.
Se ponen a disposición de los estudiantes, a través del Classroom de la materia, recursos teóricos, presentaciones, material audiovisual y actividades prácticas.
El eje “Identificación, formulación y resolución de problemas de informática” se desarrolla transversalmente durante todo el cursado. Los estudiantes trabajan sobre problemas que requieren su análisis, descomposición, modelización y diseño algorítmico previo a la implementación.

Las soluciones son posteriormente codificadas, probadas, validadas y depuradas utilizando el lenguaje C y las herramientas propias del entorno de programación. Se promueve así la “Utilización de técnicas y herramientas de aplicación en la informática”, integrando conceptos y procedimientos de manera incremental.
El proceso de aprendizaje se acompaña mediante evaluación formativa, revisión de producciones, resolución colaborativa de ejercicios y retroalimentación permanente.
Se fomenta además la autonomía en la búsqueda, selección y validación de información, promoviendo el análisis crítico de fuentes y recursos digitales.
El Trabajo Práctico Integrador de Laboratorio se desarrolla en equipos y permite aplicar de manera integrada los conocimientos adquiridos durante el cursado. Esta instancia fortalece la planificación, distribución de roles, gestión de tareas y resolución colaborativa de problemas, vinculándose con los “Fundamentos para el desempeño en equipos de trabajo”.
El equipo docente realiza un seguimiento continuo del Trabajo Práctico Integrador, contemplando tanto los resultados obtenidos como el proceso de desarrollo y promoviendo prácticas responsables y criterios éticos en la implementación en lenguaje C y en el uso de tecnologías vigentes.
La presentación y defensa del Trabajo Práctico Integrador requiere comunicar y fundamentar las decisiones técnicas adoptadas, las estrategias de resolución, los conocimientos involucrados en su implementación y los resultados obtenidos, fortaleciendo la “comunicación efectiva”.
El examen final se desarrolla, salvo situaciones de fuerza mayor, mediante una instancia de evaluación oral, con orientación previa en las clases de consulta. La retroalimentación posterior permite identificar fortalezas y oportunidades de mejora, favoreciendo la consolidación de competencias, la autonomía y el aprendizaje continuo.

VII - Plan de Trabajos Prácticos

Las actividades se realizan con seguimiento continuo a través del aula virtual de la FCFMyN, uso del Meet o Zoom.
Los estudiantes realizarán la práctica de codificación en sus computadoras y/o computadoras de las salas de la UNSL guiados por los docentes de la cátedra.

Práctico 1. Definiciones y declaraciones, datos y acciones. Sentencias y estructuras de control, manejo de funciones.
Práctico 2. Modularidad: declaraciones y uso de funciones en C.
Práctico 3. Arreglos en C. Definiciones y declaraciones de registros (struct) y uso de punteros en C.
Práctico 4. Recursividad y Memoria Dinámica
Práctico 5. Estructuras de Datos: pilas, filas, listas unidireccionales. Operaciones empleando C.
Práctico 6. Datos Recursivos. Implementación de estructuras, listas unidireccionales, empleando el tipo puntero de C.
Práctico 7. Archivos.
Práctico 8. Práctico Integrador de Laboratorio realizado en grupos de tres o más integrantes según el número de asistentes a la materia.

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 el 75% de asistencia a clase.
- Aprobar el laboratorio integrador y el coloquio con nota mayor o igual a 7 (siete).

Se otorga, tal como lo expresa la reglamentación vigente, dos recuperaciones por parcial a todos los estudiantes.
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 el 80% de la asistencia a clase.
- Aprobar el laboratorio integrador con nota mayor o igual a 7 (siete)
- Aprobar el laboratorio integrador y el coloquio con nota mayor o igual a 7 (siete).

En caso de no promocionar el estudiante deberá rendir un examen final escrito u oral.
No se admiten exámenes libres debido al carácter eminentemente práctico de la materia.

IX - Bibliografía Básica

[1] Let Us C 18th Ed. Authentic guide to C programming language; ISBN 978-9391392994; Editorial BPB Puiblications; 26

[2] Noviembre 2021.
[3] Programming In Ansi C, 8Th Edition; ISBN 978-9351343202; Editorial MC GRAW HILL INDIA; 2019.
[4] How to program in C (8th edition); Paul Deitel & Harvey Deitel; Ed. Prentice Hall; 2016.
[5] Aristides Dasso & Ana Funes, Introducción a la Programación: Notas de clase para el curso Programación I, 2014.
[6] Fundamentos de Informática y Programación en C; Diego Rafael Llanos Ferraris; Ed. ParaInfo; 2010.
[7] Programación en C; Serie Schaum, Byron S. Gottfried; Ed. Mcgraw Hill / Interamericana De España; 2005.
[8] Data structures and program design in C; Kruse, Robert; Ed. Pearson; 2007.
[9] Algoritmos + Estructuras de Datos = Programas; N. Wirth; Ed. Del Castillo; 1999.
[10] El Lenguaje de Programación C; Kernighan & Ritchie; Ed. Pearson; 1991.
[11] Apuntes teoricos realizados por la responsable de la materia, 2026.
[12] Enunciados de ejercicios para prácticos y ejercicios integradores realizados por el equipo de práctica, 2026

X - Bibliografia Complementaria

[1] Programación Estructurada. Dahl, Dijkstra, Hoare. Ed. Tiempo Contemporáneo.
[2] Algoritmos + Estructuras de Datos = Programas. N. Wirth. Ed. del Castillo.
[3] El Arte de la Programación, Vol I. D. Knuth. Ed. Reverte.

XI - Resumen de Objetivos

Introducir al estudiante en el paradigma de la programación imperativa desarrollando competencias de programación donde el estudiante pueda desarrollar, diseñar, codificar, testear así como depurar programas de complejidad moderada.

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

Mail: nbperez@email.unsl.edu.ar
Oficina 2, 1° piso, Bloque 2 - Departamento de Informática- FCFMyN.

XIV - Otros