



Ministerio de Cultura y Educación  
Universidad Nacional de San Luis  
Facultad de Ciencias Físico Matemáticas y Naturales  
Departamento: Informatica  
Area: Area de Formacion Inicial en Informatica

(Programa del año 2026)  
(Programa en trámite de aprobación)  
(Presentado el 11/08/2026 10:01:30)

### I - Oferta Académica

| Materia                              | Carrera         | Plan  | Año  | Período         |
|--------------------------------------|-----------------|-------|------|-----------------|
| RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS | ING. EN COMPUT. | 28/12 | 2026 | 2° cuatrimestre |
| RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS | PROF.CS.COMPUT. | 02/16 | 2026 | 2° cuatrimestre |

### II - Equipo Docente

| Docente                   | Función                 | Cargo     | Dedicación |
|---------------------------|-------------------------|-----------|------------|
| ROSAS, MARIA VERONICA     | Prof. Responsable       | P.Adj Exc | 40 Hs      |
| ZUÑIGA, MARIELA ELISABETH | Responsable de Práctico | JTP Exc   | 40 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     | 3 Hs              | 1 Hs                                  | 6 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                  | 90                |

### IV - Fundamentación

La formación de un profesional en Ingeniería en Computación y en el Profesorado en Ciencias de la Computación requiere, en sus etapas iniciales, promover el desarrollo del pensamiento computacional como enfoque para la resolución de problemas. Este enfoque fortalece el pensamiento crítico mediante el uso de conceptos propios de la computación, permitiendo analizar los problemas, identificar aquellos aspectos que pueden ser abordados mediante herramientas computacionales y avanzar en la construcción de soluciones.

En este contexto, resulta necesario brindar al estudiante una formación sólida en los fundamentos de la programación y en el procesamiento de datos, ya que estos constituyen la base para el posterior aprendizaje y uso de lenguajes de programación en niveles más avanzados de la carrera.

### V - Objetivos / Resultados de Aprendizaje

Se pretende enfrentar al estudiante con la problemática de analizar y resolver problemas de carácter general y la transformación de los mismos en representaciones o modelos que serán resueltos por una computadora; al mismo tiempo que se desarrolla la conceptualización sobre los componentes fundamentales de la misma, de cómo estos interactúan entre sí y de las características a tener en cuenta en la ejecución de los programas.

Al finalizar el curso, el estudiante debe ser capaz de:

- Interpretar claramente los objetivos del problema y poder resolverlos, es decir, identificar las posibles restricciones o condiciones que deben ser consideradas en la resolución del problema.
- Reconocer, tomando como base los conocimientos previos y nuevos, los datos ingresados del problema y traducirlos a información relevante al procesarlos.
- Aplicar habilidades de Pensamiento Computacional (Abstraer, Descomponer, Generalizar y Desarrollar algoritmos) para la resolución de problemas integrando el pensamiento crítico, la creatividad y el poder de las computadoras.
- Diseñar e implementar algoritmos en un lenguaje de programación para la resolución del problema computacionales.
- Evaluar la factibilidad de las distintas alternativas o soluciones propuestas considerando las restricciones establecidas.
- Interpretar adecuadamente los resultados obtenidos al ejecutar las soluciones definidas.

Se busca además contribuir al desarrollo de competencias transversales definidas para la carrera de Ingeniería en Computación, tales como trabajo en equipo, comunicación efectiva, aprendizaje autónomo y actuación ética.

## VI - Contenidos

### Contenidos mínimos.

Introducción a los Sistemas Informáticos actuales. Introducción a la Programación y Resolución de Problemas: concepto de algoritmo, estrategias básicas de resolución de problemas, herramientas y técnicas para la abstracción de problemas. Conceptos Básicos de Algoritmos: definición y propiedades de los algoritmos, representación de un algoritmo. Estructuras de Control: secuenciales, de selección, de repetición, estructuras de control anidadas. Tipos de Datos y Operaciones Básicas: tipos de datos básicos, operaciones aritméticas, relacionales y lógicas, introducción a los tipos de datos estructurados. Estructuración de Algoritmos: modularización y diseño de funciones/procedimientos, tipos de pasaje de parámetros y retorno de valores, ámbito de variables. Laboratorio: Diseño, implementación, depuración y ejecución de programas en un lenguaje de diseño de algoritmos usando un ambiente de programación básico para dicho lenguaje. Se pretende que los/las estudiantes realicen prácticas iniciales de programación en computadora.

### Programa.

#### Unidad 1: Arquitectura de la computadora.

Sistemas informáticos. Las partes de una computadora. Esquema funcional y estructural de una computadora. Las partes Internas: Unidad Central de Procesamiento, Unidad y jerarquía de Memorias. Las partes Externas: Dispositivos Periféricos, Clasificación de los periféricos. Memoria Auxiliar. Procesadores y MicroProcesadores. Representación de los datos dentro de la computadora y tabla ASCII. Software de Base y de Aplicación. Concepto de Sistema Operativo y utilitarios.

#### Unidad 2: Introducción a la Lógica Proposicional.

Proposiciones. Variables proposicionales. Funtores de verdad (a) Conectivos: Conjunción-Disyunción, Condicional, Bicondicional. (b) No-conectivos: Negación. Interpretación y Resolución de problemas lógicos.

#### Unidad 3: Resolución de Problemas.

Estrategia de resolución. Búsqueda de soluciones a problemas: inferencia, analogía, similitud entre problemas, detección de patrones, particularización y generalización. Abstracción de los Problemas. Modelización. Representación y estructuración de los problemas. Problemas de tipo computacional. Etapas de la Resolución de Problemas: El proceso de resolución según G. Polya. Descomposición en Acciones Básicas. Técnica del Refinamiento Sucesivo.

#### Unidad 4: Algoritmos.

Concepto de algoritmo y programa. Algoritmos computacionales. Lenguajes algorítmicos, gráficos y no gráficos. Lenguaje algorítmico no gráfico: Lenguaje del problema. Acciones. Estructuras de control: Secuencia, Condicional, Repetición o iteración. Lenguaje algorítmico gráfico: Diagrama de flujo.

#### Unidad 5: Lenguaje de Diseño de Algoritmos: introducción.

Lenguaje de Diseño PSeInt. Tipos de Datos, Operaciones, Expresiones y Entrada y Salida de Datos. Sintaxis de las acciones. Estructuras de control: Secuencia, Condicional, Repetición o iteración. Pautas para seleccionar la estructura repetitiva más adecuada. Anidamiento de estructuras de control.

## **Unidad 6: Lenguaje de Diseño de Algoritmos: estructuración de datos y modularización.**

Definición de estructura de datos. Tipo de datos estructurados versus tipos de datos simples. Concepto de arreglo lineal. Índice y componentes. Operaciones sobre arreglos lineales: asignación, recuperación y recorrido. Modularización de los problemas. Definición de subalgoritmos. Ambiente de un subalgoritmo. Funciones y procedimientos. Parámetros actuales y formales. Tipo de pasaje de parámetros. Invocación de subalgoritmos.

### **La asignatura se desarrolla mediante una combinación de estrategias didácticas orientadas al aprendizaje activo del estudiante:**

- Clases teóricas expositivas para la introducción de conceptos fundamentales.
- Resolución guiada de problemas en aula.
- Aprendizaje basado en problemas (ABP), promoviendo el análisis, diseño y evaluación de soluciones.
- Uso de herramientas de apoyo (PSeInt) para la experimentación y validación de algoritmos.
- Trabajo colaborativo en actividades prácticas.
- Uso del aula virtual como complemento para el acceso a materiales, entregas y retroalimentación.

## **VII - Plan de Trabajos Prácticos**

Las clases prácticas consistirán principalmente de la resolución de prácticos de aula, para que el estudiante pueda poner en práctica los conocimientos adquiridos y de un trabajo en laboratorio donde podrán comprobar, con la computadora, si los algoritmos definidos cumplen con las restricciones del lenguaje y resuelven el problema planteado.

Competencias que se esperan desarrollar al terminar el curso:

- Resolver problemas.
- Análisis y síntesis.
- Integrar la teoría con la práctica.
- Generar nuevas ideas (creatividad).
- Desarrollar y/o fortalecer el Razonamiento crítico.
- Trabajar en equipo.
- Abstraer, concreción, concisión, razonar, reconocer patrones, generalizar, ser preciso.
- Comparar y verificar el valor de las evidencias al ejecutar las soluciones propuestas.

Práctico 1: Arquitectura de las computadoras.

Objetivos específicos:

- Poner en contacto al estudiante con la terminología específica, de tal manera que identifique los principales componentes de una computadora y sus funciones.
- Dejar en evidencia al estudiante su influencia en el proceso de resolución computacional, como parte del sistema informático.

El práctico incluye ejercicios de reconocimiento de los principales componentes (monitor, teclado, impresora, etc.) y su clasificación según diferentes criterios (hardware y software, función que cumplen los periféricos de entrada, salida o de entrada/salida, memoria volátil y no volátil, etc.)

Práctico 2: Introducción a la lógica Proposicional.

Objetivos específicos:

- Introducir al estudiante en los conceptos básicos de la lógica proposicional, especialmente en su componente sintáctico.
- Lograr que el estudiante identifique proposiciones y variables proposicionales, y utilice correctamente los conectivos lógicos para construir expresiones simbólicas.
- Desarrollar la capacidad de interpretar enunciados del lenguaje natural y expresarlos mediante fórmulas de lógica proposicional.

El práctico consiste en ejercicios de interpretación de enunciados en lenguaje natural y su posterior formalización en lenguaje simbólico utilizando lógica proposicional.

Práctico 3: Resolución de Problemas y Algoritmos.

Objetivos específicos:

- Lograr que el estudiante sea capaz de interpretar claramente los objetivos del problema y poder resolverlo, es decir,

identificar las posibles restricciones o condiciones que deben ser consideradas en la resolución.

-Lograr que el estudiante sea capaz de aplicar una adecuada metodología de trabajo para la resolución de los problemas, introduciendo diferentes estrategias para resolverlos.

-Lograr que el estudiante sea capaz de crear modelos y abstracciones de problemas de la vida cotidiana. Dado un problema que pueda identificar los datos de entrada y los datos de salida como así también las restricciones del mismo.

-Lograr que el estudiante sea capaz de descomponer problemas complejos en tareas más sencillas con el objeto de resolver el problema (refinamientos sucesivos).

El práctico abarca ejercicios que plantean diferentes problemas computacionales y no computacionales presentes en el mundo real, los cuales deben ser resueltos aplicando la metodología propuesta de resolución de problemas: analizar el problema, diseñar una solución, ejecutar el plan, mirar hacia atrás. La construcción de la representación gráfica de la solución planteada.

#### Práctico 4: Lenguaje de Diseño de Algoritmos PSeInt.

Objetivos específicos:

-Desarrollar en el estudiante la capacidad de elaborar soluciones generales a diferentes clases de problemas.

-Desarrollar en el estudiante la habilidad en el buen manejo de una herramienta que ayude a complementar el proceso de resolución, en forma independiente de restricciones específicas y que facilite el entendimiento de conceptos fundamentales en el proceso de programar (variables, atributos de la variable, etc.).

-Lograr que el estudiante sea capaz de construir la representación gráfica de soluciones planteadas.

La práctica involucra actividad en el aula y de laboratorio. La práctica áulica consistirá en: codificar en Lenguaje de Diseño ejercicios puntuales de prácticos anteriores, desarrollar algoritmos de problemas nuevos para reforzar la metodología de trabajo planteada en el práctico anterior, representación gráfica de los algoritmos, detección de errores en algoritmos propuestos por el docente y su posterior corrección.

La práctica de laboratorio consistirá en introducir al estudiante en el manejo de un ambiente determinado que le permitirá, a partir de este momento, probar en una computadora si los algoritmos definidos cumplen con las restricciones del lenguaje y resuelven el problema planteado. Además, la realización de los diagramas de flujo de los algoritmos definidos en ejercicios anteriores. En el laboratorio se trabajará con herramientas que permite realizar los diagramas de flujo correspondientes a ciertos algoritmos.

#### Práctico 5: Lenguaje de Diseño de Algoritmos PSeInt: estructuración de datos.

Objetivos específicos:

-Introducir a los estudiantes en el concepto de manipulación colectiva de datos: estructuras de datos, sus características, ventajas de su uso e implementación de arreglos en Lenguaje de Diseño.

La práctica involucra actividad en el aula y de laboratorio. La práctica áulica consistirá en desarrollar nuevos algoritmos de problemas y su codificación en Lenguaje de Diseño, la detección de errores en algoritmos propuestos.

#### Práctico 6: Lenguaje de Diseño de Algoritmos PSeInt: modularización.

Objetivos específicos:

-Determinar la necesidad de implementar el concepto de modularización en la resolución de problemas planteados.

-Reconocer la utilidad de subalgoritmos en lenguaje de diseño.

-Realizar ejecuciones teniendo en cuenta el ambiente o ámbito de un subalgoritmo.

-Diferenciar parámetros actuales y formales.

-Implementar correctamente el lenguaje de diseño, las invocaciones a subalgoritmos.

La práctica áulica consistirá en desarrollar nuevos algoritmos de problemas y su codificación en Lenguaje de Diseño, detección de errores en algoritmos propuestos y su posterior corrección. Para la resolución de los ejercicios se deberá aplicar la metodología de trabajo planteada a partir del práctico 3 y la realización de los diagramas de flujos respectivos.

Ejes transversales.

Los ejes transversales se abordan a lo largo de la asignatura a partir de las actividades prácticas de aula y laboratorio, en coherencia con los objetivos formativos de la carrera.

Eje: Identificación, formulación y resolución de problemas de ingeniería en computación

•Cómo se aborda: Se trabaja de manera progresiva a partir de la resolución de problemas en los prácticos, especialmente en los prácticos 3 al 6, donde los estudiantes analizan, modelizan, descomponen y diseñan soluciones algorítmicas.

•Cómo se evalúa: Mediante la resolución de ejercicios, la correcta identificación de datos de entrada/salida, la coherencia de las soluciones propuestas y la implementación en lenguaje de diseño.

Eje: Utilización de técnicas y herramientas de aplicación en la ingeniería en computación

- Cómo se aborda: A través del uso del lenguaje de diseño PSeInt y herramientas para la representación de algoritmos, tanto en actividades de aula como de laboratorio.
- Cómo se evalúa: Mediante la correcta utilización de la herramienta, la sintaxis empleada en los algoritmos y la capacidad de verificar su funcionamiento.

Eje: Fundamentos para el desempeño en equipos de trabajo

- Cómo se aborda: En actividades prácticas que promueven el intercambio de ideas, la discusión de soluciones y la resolución colaborativa de problemas en el aula.
- Cómo se evalúa: A partir de la participación en clase, la interacción con pares y la actitud frente al trabajo colaborativo.

Eje: Fundamentos para una comunicación efectiva

- Cómo se aborda: Mediante la expresión de soluciones en lenguaje natural y simbólico, la escritura de algoritmos y la explicación de los procedimientos utilizados.
- Cómo se evalúa: A través de la claridad en la resolución de ejercicios, el uso adecuado del lenguaje técnico y la correcta interpretación de consignas.

Eje: Fundamentos para la actuación profesional ética y responsable

- Cómo se aborda: Promoviendo el cumplimiento de consignas, la realización individual de actividades y el respeto por las producciones propias y ajenas.
- Cómo se evalúa: Mediante la entrega en tiempo y forma de las actividades y el seguimiento del desempeño individual.

Eje: Fundamentos para el aprendizaje continuo

- Cómo se aborda: Fomentando la resolución autónoma de ejercicios, la consulta de material teórico y la práctica constante fuera del aula.
- Cómo se evalúa: A través del progreso en el desempeño del estudiante y la capacidad de incorporar correcciones en sucesivas actividades.

Eje: Impacto social de la ingeniería en computación

- Cómo se aborda: A partir de la resolución de problemas contextualizados que permiten reconocer el rol de la computación en la vida cotidiana.
- Cómo se evalúa: Mediante la comprensión del problema planteado y la pertinencia de la solución propuesta.

## VIII - Régimen de Aprobación

La evaluación contempla:

- Evaluación diagnóstica inicial.
- Evaluación formativa continua mediante actividades prácticas.
- Evaluación sumativa mediante parciales y evaluación integradora.
- Autoevaluación y retroalimentación en actividades del aula virtual.

Cada evaluación parcial cuenta con dos instancias de recuperación, conforme al régimen académico vigente.

Régimen de Promoción.

- Asistencia al 80% o más de las clases.
  - Aprobar las dos evaluaciones parciales con un 80% o más. Cada una de las evaluaciones parciales tiene dos recuperaciones. Para el régimen de promoción se deben aprobar ambas evaluaciones parciales en primera instancia o en la primera recuperación.
  - Aprobar, en su primera instancia o en alguna de sus dos recuperaciones, todos los parcialitos con nota 6 o más.
  - Aprobar una evaluación final integradora de promoción con un 80% o más.
- La nota final resultará de la nota obtenida en la evaluación final y no podrá ser menor a 7.

Régimen de Regularización.

- Asistencia al 70% o más de las clases.
- Aprobar las dos evaluaciones parciales con un 70% o más. Cada una de las evaluaciones parciales tiene dos recuperaciones.

- Aprobar, en su primera instancia o en alguna de sus dos recuperaciones, todos los parcialitos con nota 6 o más.
- Los estudiantes que hayan cumplido con los requisitos anteriormente citados podrán presentarse a rendir examen final oral o escrito en cualquiera de los turnos establecidos por la reglamentación de facultad.

Régimen de Condición Libre.

Dada la necesidad de un constante seguimiento de la/el estudiante en clase, la materia no se puede rendir en calidad de libre.

## IX - Bibliografía Básica

- [1] Material de estudio del curso. <http://www.servicios.dirinfo.unsl.edu.ar>. 2026.
- [2] "Algoritmos y Estructuras de Datos", Wirth, Niklaus – Editorial Pearson Educación / Prentice Hall; 1ra Edición. ISBN: 9789688801130, 1987 (Reimpresión 2011).
- [3] "Cómo plantear y resolver problemas", G. Polya, Editorial Trillas, ISBN-10 968-24-00643, ISBN-13 978-9682400643, 1981. Disponible en <https://dn721602.ca.archive.org/0/items/polya-g.-como-plantear-y-resol-ver-problemas/POLYA%2CG.%20COMO%20PLANTEAR%20Y%20RESOL%20VER%20PROBLEMAS%2C.pdf> (última visita 20/3/2024).
- [4] "Fundamentos de Algoritmia", Brassard, Gilles y Bratley, Paul, Prentice Hall, 1a. edición, 2000, ISBN: 84-89660-00-X, Ubicación en Biblioteca: 004.021.B823f.
- [5] "Introducción a la Computación", Andrés Gómez de Silva Garza, Ignacio de Jesús Ania Briseño - Editorial Cengage Learning, ISBN-13: 978-970-686-768-1, 2008, Ubicación en Biblioteca: 004.I61. (Nº inventario 85884).
- [6] "Introducción a la Informática", Prieto Espinosa, Alberto, Lloris Ruiz, Antonio – Torres Cantero, Juan Carlos. McGraw-Hill Interamericana de España; 4ta Edición. ISBN: 8448132173, 2006. Disponible en <https://dn721704.ca.archive.org/0/items/alberto-prieto-introduccion-a-la-informatica-4ta-ed./Alberto%20Prieto%20-%20Introduccion%20a%20la%20informatica%204ta%20Ed..pdf> (última visita 20/3/2026).
- [7] "Lógica Computacional", Paniagua Arís, Enrique, Sánchez Gonzalez, Juan Luis, Rubio, Fernando Martín. Thomson; ISBN: 8497321820, 2003.
- [8] PSeInt sitio web. <http://pseint.sourceforge.net> (última visita 18/3/2026).

## X - Bibliografía Complementaria

- [1] "Algorithmic Problem Solving", R. Backhouse, Wiley, ISBN: 978-0-470-68453-5, 2011.
- [2] "Computational Thinking For The Modern Problem Solver" D. Riley y K. Hunt, CRC Press - ISBN: 978-1-4665-8777-9, 2014.
- [3] "Introduction to Mathematical Logic, Third Edition", Elliott Mendelson - Van Nostrand Reinhold Company - ISBN-10:0534066240, ISBN-13: 978-0534066246, 1987.
- [4] "Organización y Arquitectura de Computadores - Diseño para optimizar prestaciones" – William Stallings - Prentice Hall - 5ta Edición - ISBN: 84-205-2993-1, 2004.
- [5] "Peter Norton's New Inside the PC", Peter Norton, Ed. Sams, ISBN 0672322897, 2002.
- [6] "Problem Solving & Computer Programming", P. Grogono y S. Nelson, Addison-Wesley Publishing Company – ISBN 0-201-02460-8, 1982.
- [7] "Puzzle-based Learning: Introduction to critical thinking, mathematics, and problem solving", Z. Michalewicz y M. Michalewicz, (Paperback). Hybrid Publishers; 1ra Edición (21 de Mayo, 2008).
- [8] "Structured Programming", O.-J. Dahl, E. W. Dijkstra, C. A. R. Hoare, Academic Press, London, ISBN 0-12-200550-3, 1972.

## XI - Resumen de Objetivos

Desarrollar en el estudiante las capacidades de:

- Resolver problemas.
- Diseñar e implementar algoritmos de solución a problemas en forma computacional.
- Usar diferentes herramientas para el diseño y codificación de las soluciones.
- Evaluar y analizar la viabilidad de las soluciones planteadas.

## XII - Resumen del Programa

Unidad 1: Arquitectura de las Computadoras.

Unidad 2: Introducción a la Lógica Proposicional.  
Unidad 3: Resolución de Problemas.  
Unidad 4: Algoritmos.  
Unidad 5: Lenguaje de Diseño de Algoritmos: introducción.  
Unidad 6: Lenguaje de Diseño de Algoritmos: estructuras de datos y modularización.

### **XIII - Imprevistos**

Cualquier imprevisto de carácter académico será analizado por la Comisión de Carrera.

### **XIV - Otros**

El seguimiento continuo de los estudiantes se lleva a cabo mediante encuentros áulicos presenciales y a través del aula virtual en el Campus Virtual de la UNSL. En ambos espacios, los estudiantes entregan ejercicios prácticos resueltos para recibir retroalimentación de los mismos. Además, en el aula virtual pueden acceder a teorías, trabajos prácticos, avisos importantes, material de estudio y el cronograma, así como realizar entregas de tareas, hacer consultas o dejar comentarios.

Mail de Contacto: rpyaunsl@gmail.com

| <b>ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA</b> |                             |
|------------------------------------------------|-----------------------------|
|                                                | <b>Profesor Responsable</b> |
| Firma:                                         |                             |
| Aclaración:                                    |                             |
| Fecha:                                         |                             |