Ministerio de Cultura y Educación **(Programa del año 2025)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area de Formacion Inicial en Informatica

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
RESOLUCIÓN DE PROBLEMAS Y	ING. INFORM.	026/12-08/15	2025	1° cuatrimestre
() RESOLUCION DE PROBLEMAS Y ALGORITMOS	TFA	04/14	2025	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ARAGON, VICTORIA SOLEDAD	Prof. Responsable	P.Adj Exc	40 Hs
GARIS, ANA GABRIELA	Prof. Colaborador	P.Adj Exc	40 Hs
CUELLO, CAROLINA YAMILE	Auxiliar de Práctico	A.1ra Semi	20 Hs
GARCIARENA UCELAY, MARIA JOSE	Auxiliar de Práctico	A.1ra Semi	20 Hs
GAUNA, DARIO EMILIO	Auxiliar de Práctico	A.1ra Simp	10 Hs
JOFRE PASINETTI, BETTINA ANDREA	Auxiliar de Práctico	A.2da Simp	10 Hs
PONCE, CECILIA ADRIANA	Auxiliar de Práctico	A.1ra Semi	20 Hs
RIERA BAUER, FABRIZIO JOSÉ	Auxiliar de Práctico	A.2da Simp	10 Hs
SORIA, DIEGO LAUTARO	Auxiliar de Práctico	A.2da 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	3 Hs	1 Hs	6 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
12/03/2025	24/06/2025	15	90

IV - Fundamentación

La formación de un profesional en Ingeniería en Informática requiere de un proceso inicial que promueva en el estudiante actividades intelectuales que favorezcan el desarrollo del pensamiento computacional y se vincule a la resolución de problemas de diferentes características cuya solución, en principio, pueda ser expresada de una manera flexible, apelando a enfoques metodológicos novedosos, tales como el Aprendizaje Basado en Resolución de Problemas. Al mismo tiempo, surge la necesidad de plantear soluciones siguiendo un enfoque lógico y algorítmico que permitirá dar al estudiante una formación sólida en el área de programación de computadoras que será la base fundamental para profundizar en las materias más avanzadas en el aprendizaje y utilización de lenguajes de programación.

V - Objetivos / Resultados de Aprendizaje

Objetivo general
Se pretende enfrentar al estudiante con la problemática de analizar y resolver problemas de carácter general y la transformación de los mismos para posteriormente ser resueltos por una computadora.

Objetivos específicos
Se espera que el estudiante sea capaz de:
- Desarrollar una adecuada metodología de trabajo para la resolución de los problemas introduciendo estrategias para resolver problemas, como por ejemplo, dividir un problema en subproblemas, obtener la solución a través de refinamientos sucesivos, entre otras.
- Generar la capacidad necesaria para saber interpretar claramente los objetivos del problema y poder resolverlo, es decir, identificar las posibles restricciones o condiciones que deben ser consideradas en la resolución del problema.
- Resolver distintos tipos de problemas expresando su solución en forma algorítmica.
- Diseñar un algoritmo en un lenguaje de diseño que modele la resolución del problema.
- Implementar un algoritmo en un lenguaje de programación.
- Elegir un conjunto adecuado y representativo de valores de los datos de entrada para realizar una correcta prueba de un algoritmo realizado.
- Introducir la notación lógica formal con el fin de expresar ideas o razonamientos de forma clara y precisa, fomentando la rigurosidad y formalidad.
- Reforzar y desarrollar competencias generales, como por ejemplo: comunicación en forma oral y escrita, aprendizaje autónomo, entre otras.
- Promover el uso de buenos hábitos de programación incentivando, desde el principio de la carrera, el ejercicio de la documentación, los comentarios y la indentación de los programas desarrollados.

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.

VI - Contenidos

Contenidos mínimos:
Problemas, modelos y abstracciones. Representación de problemas. Búsqueda de soluciones a problemas: Inferencia, analogía, similitud entre problemas, generalización particularización. Algoritmo: concepto, algoritmos computacionales. Resolución de problemas: métodos de resolución de problemas, descripción utilizando un lenguaje de diseño de algoritmos. Conceptos fundamentales de Lógica Proposicional y Cálculo de Predicados.
Laboratorio: Implementación y ejecución de programas cortos en un lenguaje de programación básico. Se pretende que los alumnos realicen prácticas iniciales de programación en computadora.

Programa:

Unidad 1: 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. Descomposición en Acciones Básicas. Técnica del Refinamiento Sucesivo. Concepto de algoritmo. Ambiente de un algoritmo. Datos de entrada y salida. Transformación del ambiente.

Unidad 2: Introducción al Cálculo Proposicional y al Cálculo de Predicados
Proposiciones. Variables proposicionales. Funtores de verdad (a) Conectivos: Conjunción-Disyunción, Condicional, Bicondicional. (b) No-conectivos: Negación. Interpretación y Resolución de problemas lógicos. Cuantificadores. Alfabeto. Vocabulario. Lenguaje: fórmulas atómicas y fórmulas bien formadas. Variables Libres.

Unidad 3: Lenguaje de Diseño de Algoritmos – Tipos de Datos, Operaciones, Expresiones y Entrada y Salida de Datos

Objetos del ambiente: constantes y variables. Tipos de datos primitivos. Expresiones. Operadores aritméticos, lógicos y relacionales. Precedencia y orden de evaluación. Funciones primitivas. Operación de asignación. Estructura general de un programa. Declaración de variables. Operaciones de entrada y salida de datos. Buenas prácticas de programación.

Unidad 4: Lenguaje de Diseño de Algoritmos – Estructuras de Control
Estructura de control secuencial: Concepto. Estructura de control condicional: Simple y Múltiple. Estructuras de control repetitiva: Conceptos. Estructuras con número de iteraciones predeterminado y no predeterminado. Pautas para seleccionar la estructura repetitiva más adecuada. Ciclos infinitos. Anidamiento de estructuras de control. Ejecución de un programa a través de tablas de ejecución. Utilización de diagramas de flujo.

Unidad 5: Lenguaje de Diseño de Algoritmos – Estructuras de Datos
Definición de estructura de datos. Tipo de datos estructurados versus tipos de datos simples. Arreglos lineales. Índice y componentes. Operaciones sobre arreglos lineales: asignación, recuperación y recorrido.

Unidad 6: Lenguaje de Diseño de Algoritmos – Subalgoritmos
Definición de subalgoritmos. Ambiente de un subalgoritmo. Funciones y procedimientos. Parámetros actuales y formales. Tipo de pasaje de parámetros. Invocación de subalgoritmos.

VII - Plan de Trabajos Prácticos

Metodología de trabajo

Los contenidos de la materia se dictarán durante las 15 semanas que dura el cuatrimestre. Para trabajar cada unidad, las clases teóricas se dictarán de forma presencial. Las diapositivas de clase, el material digital y los apuntes teóricos correspondientes se encontrarán en el reposito de la materia. Para cada unidad se presentan actividades, que permiten, tras su desarrollo y junto a las clases teóricas, alcanzar las pretensiones planteadas por los objetivos antes mencionados. Además, para algunas unidad, se designarán una o más de esas actividades, de carácter integrador, las cuales deben ser entregadas por cada estudiante en las clases presenciales para realizar evaluaciones formativas.

Práctico Nº 1: Resolución de Problemas
Objetivos: Lograr que el estudiante sea capaz de:
- Interpretar claramente los objetivos del problema y poder resolverlo identificando las posibles restricciones o condiciones que deben ser consideradas en la resolución.
- Aplicar una adecuada metodología de trabajo para la resolución de los problemas introduciendo diferentes estrategias para resolverlos.
- Crear modelos y abstracciones de problemas de la vida cotidiana. Dado un problema identificar los datos de entrada y los datos de salida como así también las restricciones del mismo.
- Descomponer problemas complejos en tareas más sencillas con el objeto de resolver el problema (refinamientos sucesivos).
El práctico abarca ejercicios que plantean diferentes problemas 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.

Práctico Nº 2: Introducción la Lógica Proposicional y Lógica de Predicados
Objetivos:
- Introducir al estudiante en el mundo de la Lógica proposicional a partir de su primer componente, el sintáctico.
- Lograr que el estudiante sea capaz de identificar las limitaciones de la Lógica Proposicional, identificar el alfabeto, términos y predicados de la Lógica de Predicados y la utilización de cuantificadores para construir fórmulas bien formadas que especifiquen las restricciones del problema.
El práctico resulta de la combinación de: ejercicios de interpretación de texto y su posterior especificación en forma

simbólica, en lenguaje de lógica proposicional y lógica de predicados.

Práctico Nº 3: Introducción al Lenguaje de Diseño de Algoritmos - Tipos de Datos, Operaciones, Expresiones y Entrada y Salida de Datos
Objetivos: Lograr que estudiante sea capaz de escribir expresiones del Lenguaje de Diseño, traducir enunciados en lenguaje natural a expresiones aritméticas, lógicas y relacionales y comprender la interacción con el usuario (Entrada/Salida).
La práctica involucra actividad en el aula que 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 de Resolución de Problemas.

Práctico Nº 4: al Lenguaje de Diseño – Estructuras de Control
Objetivos: Lograr que el estudiante sea capaz de comprender el funcionamiento de la estructura de control condicional simple y múltiple, y realizar pruebas para comprobar el funcionamiento de un algoritmo. Lograr que el estudiante sea capaz de comprender el funcionamiento de estructuras de control repetitivo, identificar los casos especiales en cada estructura (Para-Hasta, Mientras-Hacer, Repetir-HastaQue), e identificar cuándo es adecuado la utilización de cada estructura. Así mismo se espera que el estudiante sea capaz de realizar diagramas de flujo para visualizar los diferentes caminos de ejecución de un algoritmo.
La práctica involucra actividad en el aula que 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 de Resolución de Problemas.

Práctico Nº 5: Lenguaje de Diseño – Estructuras de Datos
Objetivos: Lograr que estudiante sea capaz de comprender y aplicar los conceptos relacionados con la estructura de datos arreglo lineal.
La práctica involucra actividad en el aula que consistirá en: codificar en Lenguaje de Diseño ejercicios puntuales extendidos de prácticos anteriores, desarrollar algoritmos de problemas nuevos para reforzar la metodología de trabajo planteada en el práctico de Resolución de Problemas.

Práctico Nº 6: Desarrollo e Implementación de Algoritmos. Conceptualización de Modularización. Implementación en el intérprete PSeInt.
Objetivos: Lograr que el estudiante sea capaz de comprender y aplicar los conceptos relacionados con la modularización de procesos.
La práctica áulica consistirá en desarrollar nuevos algoritmos de problemas y su codificación en Lenguaje de Diseño. Para la resolución de los ejercicios se deberá aplicar la metodología de trabajo planteada a partir del práctico Resolución de Problemas.

PLAN DE TRABAJO PRÁCTICO DE LABORATORIO

El objetivo general del práctico de laboratorio es diseñar e implementar un algoritmo que resuelva un problema (proyecto integrador), empleando el intérprete PSeInt para implementar el algoritmo diseñado en el lenguaje de diseño LDA.

El práctico de laboratorio comprende tres tipos de formación práctica:

- Actividades de análisis y diseño: El estudiante debe trabajar en grupo las etapas de análisis del problema y planificación de una solución (algoritmo).
- Actividades de formación experimental: El estudiante debe programar y probar, individualmente y en grupo, el algoritmo diseñado.
- Expresión escrita y oral: Los grupos deben presentar en forma oral y escrita un informe sobre las actividades previamente mencionadas.

Toda consulta teórica-práctica podrá realizarse en los horarios asignados para las clases presenciales.

Los ejes transversales se abordan a lo largo de la materia en las prácticas áulicas y práctica de laboratorio de acuerdo con:

Eje: Identificación, formulación y resolución de problemas de informática
Cómo se aborda: Se aborda a partir de la unidad temática 3 mediante el desarrollo de trabajos prácticos, guiadas por clases teóricas, las diapositivas de clase, los apuntes teóricos, prácticos de aula y consultas grupales e individuales. Además, se desarrolla un proyecto integrador a partir de la unidad 5.
Cómo se evalúa: Evaluación formativa: para cada unidad, se designa una o más actividades, de carácter integrador, las cuales deben ser desarrolladas individualmente y entregadas, se realiza una devolución personalizada de las entregas. El proyecto integrador se evalúa a través de una entrega parcial de un informe.
Evaluación sumativa: un examen práctico y entrega de código fuente, informe y presentación oral de un proyecto integrador.

Eje: Fundamentos para el desempeño en equipos de trabajo
Cómo se aborda: El desarrollo del proyecto integrador se realiza de forma grupal.
Cómo se evalúa: Los estudiantes deben describir en el informe escrito la dinámica grupal con la cual trabajaron, destacando ventajas y desventajas de las decisiones tomadas.

Eje: Utilización de técnicas y herramientas de aplicación en la informática
Cómo se aborda: Los estudiantes debe codificar el algoritmo del proyecto integrador empleando el lenguaje de programación PSeInt y un editor de texto a elección para la presentación del informe.
Cómo se evalúa: Inspección del código fuente en PSeInt y del formato del informe.

Eje: Fundamentos para la comunicación efectiva
Cómo se aborda: Los estudiantes debes presentar en forma oral y escrita un informe que releve las características y decisiones de diseño tomadas para la implementación del proyecto integrador junto con el rol que desempeñó cada integrante del grupo y la dinámica grupal con la cual trabajaron, entre otros. La mediación pedagógica realizada es la exposición de todos los criterios con los cuales se evaluará el informe.
Cómo se evalúa: El informe escrito se evalúa a través de una matriz de evaluación entregada a los estudiantes al inicio del cuatrimestre la cual considera a los siguientes criterios (con escala excelente, muy buena, buena y reajustar): Calidad de la Información, Organización de la información, Redacción, Diseño de casos de prueba y Trabajo en equipo. La presentación oral se evalúa a través de una matriz de evaluación entregada a los estudiantes al inicio del cuatrimestre la cual considera a los siguientes criterios (con escala excelente, muy buena, buena y reajustar): Contenido, Organización, Comunicación y claridad.

Eje: Fundamentos para la acción ética y responsable
Cómo se aborda: Pactando fechas de entrega al inicio del cuatrimestre, entregando un cronograma diario de las actividades a ser realizadas.
Cómo se evalúa: Entrega de códigos fuentes para comprobar plagios. Considerando la presentación en tiempo y forma de las entregas pautadas al inicio del cuatrimestre.

VIII - Regimen de Aprobación

Régimen de Promoción

Los estudiantes deben:

- Asistir al 80% de las clases prácticas.
- Entregar todas las actividades prácticas y/o de laboratorio, realizando las entregas en la fecha pautada.
- Aprobar un proyecto integrador grupal junto con su informe y defensa oral, o su respectiva recuperación, con una nota mayor o igual a 7.
- Aprobar el examen teórico-práctico integrador, o alguna de sus dos recuperaciones con una nota mayor o igual a 7.
- Si un estudiante que ha aprobado el examen teórico-práctico desea rendir la primera o segunda recuperación para optar por la promoción, se considerará la mejor nota obtenida.

La nota final resultará del promedio de todas las notas obtenidas en todas las evaluaciones realizadas y no podrá ser menor a 7.

Régimen de Regularización

- Asistir al 70% de las clases prácticas.
- Entregar todas las actividades prácticas y/o de laboratorio, realizando las entregas en la fecha pautada.
- Aprobar un proyecto integrador grupal junto con su informe y defensa oral, o su respectiva recuperación, con una nota mayor o igual a 7.
- Aprobar el examen teórico-práctico, o alguna de sus dos recuperaciones con una nota mayor que 6 pero menor que 7.
- 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 Alumnos Libres

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

IX - Bibliografía Básica

[1] Material de estudio del curso.
[2] “Fundamentos de Algoritmia”, Brassard, Gilles y Bratley, Paul, Prentice Hall, 1a. edición, 2000, ISBN: 84-89660-00-X, Ubicación en Biblioteca: 004.021.B823f
[3] "Cómo plantear y resolver problemas", G. Polya, Editorial Trillas, ISBN-10 968-24-00643, ISBN-13 978-9682400643, 2011.
[4] “Lógica simbólica y elementos de metodología de la ciencia”, Gianella de Salama, Alicia; Roulet, Margarita – Publicación Buenos Aires: El Ateneo, 1996.
[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; 3ra Edición. ISBN: 8448132173, 2001.
[7] “Lógica Computacional”, Paniagua Arís, Enrique, Sánches Gonzalez, Juan Luis, Rubio, Fernando Martín. Thomson; ISBN: 8497321820, 2003.
[8] http://pseint.sourceforge.net (última visita 18/3/2025)
[9] “Algoritmos + Estructuras de Datos = Programas”, N. Wirth, Ed. Dossat - ISBN: 8421901729, 1999.
[10] “Peter Norton's New Inside the PC”, Peter Norton, Ed. Sams, ISBN 0672322897, 2002.

X - Bibliografia Complementaria

[1] “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).
[2] "Problem Solving & Computer Programming", P. Grogono y S. Nelson, Addison-Wesley Publishing Company – ISBN 0-201-02460-8, 1982.
[3] "Algorithmic Problem Solving", R. Backhouse, Wiley, ISBN: 978-0-470-68453-5, 2011.
[4] "Computational Thinking For The Modern Problem Solver" D. Riley y K. Hunt, CRC Press - ISBN: 978-1-4665-8777-9, 2014.
[5] “Introduction to Mathematical Logic, Third Edition”, Elliott Mendelson - Van Nostrand Reinhold Company - ISBN-10:0534066240, ISBN-13: 978-0534066246, 1987.
[6] “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

- Fomentar el análisis crítico y evaluación de diferentes alternativas.
- Fomentar el espíritu de mejora continua y autoconocimiento.
- Fomentar la expresión escrita.
- Fomentar el trabajo grupal colaborativo.
- Desarrollar una adecuada metodología de trabajo para la resolución de los problemas introduciendo estrategias para resolver problemas, como por ejemplo, dividir un problema en subproblemas, obtener la solución a través de refinamientos sucesivos, entre otras.
- Generar la capacidad necesaria para saber interpretar claramente los objetivos del problema y poder resolverlo identificando las posibles restricciones o condiciones que deben ser consideradas en la resolución del problema.

- Desarrollar la capacidad de diseñar un algoritmo que modele la resolución del problema, de implementar el algoritmo en un lenguaje de programación y de elegir un conjunto adecuado y representativo de valores de los datos de entrada para realizar una correcta prueba del algoritmo realizado.
- Introducir la notación lógica formal con el fin de expresar ideas o razonamientos de forma clara y precisa, fomentando la rigurosidad y formalidad.
- Reforzar y desarrollar competencias generales, como por ejemplo: comunicación en forma oral y escrita, aprendizaje autónomo, entre otras.
- Promover el uso de buenos hábitos de programación incentivando, desde el principio de la carrera, el ejercicio de la documentación, los comentarios y la indentación de los programas desarrollados.

XII - Resumen del Programa

Unidad 1: Resolución de Problemas
Unidad 2: Introducción al Cálculo Proposicional y al Cálculo de Predicados.
Unidad 3: Lenguaje de Diseño de Algoritmos – Tipos de Datos, Operaciones, Expresiones y Entrada y Salida de Datos
Unidad 4: Lenguaje de Diseño de Algoritmos – Estructuras de Control
Unidad 5: Lenguaje de Diseño de Algoritmos – Estructuras de Datos
Unidad 6: Lenguaje de Diseño de Algoritmos – Subalgoritmos

XIII - Imprevistos

Las vías de comunicación con los estudiantes:

Victoria Aragón vsaragon@email.ensl.edu.ar (Oficina: 4 – 1° piso – 2° Bloque)
Ana Garis agaris@gmail.com (Oficina: 1 – 1° piso – 2° Bloque)

XIV - Otros