



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

(Programa del año 2022)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS	ING. INFORM.	026/1 2- 08/15	2022	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
ARAGON, VICTORIA SOLEDAD	Prof. Responsable	P.Adj Exc	40 Hs
GATICA, CLAUDIA RUTH	Auxiliar de Práctico	A.1ra Semi	20 Hs
JOFRE PASINETTI, LUIS AGUSTIN	Auxiliar de Práctico	A.2da Simp	10 Hs
MAGUIRE, MARGARITA	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	4 Hs	1 Hs	7 Hs

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

Duración			
Desde	Hasta	Cantidad de Semanas	Cantidad de Horas
21/03/2022	24/06/2022	14	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

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; al mismo tiempo que se desarrolla la conceptualización sobre los componentes fundamentales de la misma, de cómo éstos interactúan entre sí y de las características a tener en cuenta en la ejecución de los programas.

Objetivos generales

La asignatura pretende:

- 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.

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.
- 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.

VI - Contenidos

Bolilla I: Arquitectura de las Computadoras

Las partes de una computadora. Las partes Internas: Unidad Central de Proceso, Unidad de Memoria (Memoria Principal). Las partes Externas: Dispositivos Periféricos, Clasificación de los periféricos. Memoria Auxiliar. Procesadores y MicroProcesadores. Representación de los datos dentro de la computadora. Software de Base y de Aplicación. Concepto de Sistema Operativo.

Bolilla II: 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.

Bolilla III: Introducción al Cálculo Proposicional y al Cálculo de Predicados

Proposiciones. Variables proposicionales. Futores de verdad (a) Conectivos: Conjunción-Disyunción, Condicional, Bicondicional. (b) No-conectivos: Negación. Interpretación y Resolución de problemas lógicos. Funciones proposicionales. Cuantificadores. Alfabeto. Vocabulario. Lenguaje: fórmulas atómicas y fórmulas bien formadas.

Variables Libres. Interpretación.

Bolilla IV: 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.

Bolilla V: Lenguaje de Diseño de Algoritmos – Estructuras de Control

Estructura de control secuencial: Concepto. Estructura de control condicional: Simple y Múltiple. Elección entre dos alternativas. Elección entre varias alternativas. 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 para visualizar los posibles flujos de ejecución de un algoritmo.

Bolilla VI: 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.

Bolilla VII: 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 14 semanas que dura el cuatrimestre. Para trabajar cada bolilla, las clases teóricas se dictarán de forma presencial. Las diapositivas de clase, los videos teóricos y los apuntes teóricos correspondientes se encontrarán en el aula virtual de la plataforma Moodle de la FCFMyN (Resolución de Problemas y Algoritmos 2022). Para cada bolilla 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 bolillas, 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 N° 1: La computadora: conceptualización.

Objetivos: Lograr que el estudiante conozca la terminología específica, que identifique los principales componentes de una computadora y sus funciones y determine su influencia en el proceso de resolución computacional.

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, etc.)

Práctico N° 2: Resolución de Problemas

Objetivos: 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.
- 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 N° 3: 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 N° 4: Introducción al Lenguaje de Diseño y a PSeInt- 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 N° 5: al Lenguaje de Diseño – Estructuras de Control Condicional

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.

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 N° 6: Lenguaje de Diseño – Estructuras de Control Repetición

Objetivos: 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 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 N° 7: 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 N° 8: Desarrollo e Implementación de Algoritmos. Conceptualización de Modularización. Implementación en 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 Resolución de Problemas y la realización de los diagramas de flujos respectivos.

PLAN DE TRABAJOS PRÁCTICOS DE LABORATORIO

El objetivo general de los prácticos de laboratorio es diseñar e implementar algoritmos que resuelvan un problema. Cada trabajo práctico del 4 al 7 estará acompañado de un práctico de laboratorio que consistirá en el uso del ambiente de PSeInt para probar en una computadora un algoritmo diseñado en el práctico de aula.

Los prácticos de laboratorio comprenden dos tipos de formación práctica:

- Actividades de análisis y diseño: El estudiante debe trabajar las etapas de análisis del problema y planificación de una solución (algoritmo).
- Actividades de formación experimental: El estudiante debe programar y probar el algoritmo diseñado.

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

VIII - Regimen de Aprobación

Régimen de Promoción

Los estudiantes deben:

- Asistir al 80% de las clases prácticas.
- Aprobar 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 parcial desea rendir la primera o segunda recuperación para optar por la promoción y mejorar la nota, 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.
- Aprobar 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.

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. <http://www.dirinfo.unsl.edu.ar>
- [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] “Organización y Arquitectura de Computadores - Diseño para optimizar prestaciones” – William Stallings - Prentice Hall -5ta Edición - ISBN: 84-205-2993-1, 2004.
- [7] “Introducción a la Informática”, Prieto Espinosa, Alberto, LLoris Ruiz, Antonio – Torres Cantero, Juan Carlos. McGraw-Hill Interamericana de España; 3ra Edición. ISBN: 8448132173, 2001.
- [8] [8] “Lógica Computacional”, Paniagua Arís, Enrique, Sánchez Gonzalez, Juan Luis, Rubio, Fernando Martín. Thomson; ISBN: 8497321820, 2003.
- [9] PSeInt (<http://pseint.sourceforge.net>)
- [10] “Algoritmos + Estructuras de Datos = Programas”, N. Wirth, Ed. Dossat - ISBN: 8421901729, 1999.
- [11] “Peter Norton's New Inside the PC”, Peter Norton, Ed. Sams, ISBN 0672322897, 2002.

X - Bibliografía 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, es decir, identificar 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

Bolilla I: Arquitectura de las Computadoras.

Bolilla II: Resolución de Problemas

Bolilla III: Introducción al Cálculo Proposicional y al Cálculo de Predicados.

Bolilla IV: Lenguaje de Diseño de Algoritmos – Tipos de Datos, Operaciones, Expresiones y Entrada y Salida de Datos

Bolilla V: Lenguaje de Diseño de Algoritmos – Estructuras de Control

Bolilla VI: Lenguaje de Diseño de Algoritmos – Estructuras de Datos

Bolilla VII: Lenguaje de Diseño de Algoritmos – Subalgoritmos

XIII - Imprevistos

Las vías de comunicación con los estudiantes son las siguientes:

- Correos electrónicos de los docentes:

Victoria Aragón vsaragon@email.unsl.edu.ar (Oficina: 4 – 1° piso – 2° Bloque)

- Campus Virtual: <https://www.evirtual.unsl.edu.ar/moodle/course/view.php?id=1138> (Resolución de Problemas y Algoritmos 2022) donde se pueden descargar las teorías, trabajos prácticos, avisos importantes, apuntes, cronograma, realizar entregas, hacer consultas o comentarios, etc.

La cantidad de horas dictadas en total en el cuatrimestre será de 90 horas.

XIV - Otros