Ministerio de Cultura y Educación **(Programa del año 2018)**
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area III: Servicios

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
FUNDAMENTOS DE LA INFORMATICA	TEC.REDES COMP.	12/13	2018	1° cuatrimestre
FUNDAMENTOS DE LA INFORMATICA	TCO.UNIV.EN WEB	08/13	2018	1° cuatrimestre
FUNDAMENTOS DE LA INFORMATICA	TEC.UNIV.GEOINF	09/13	2018	1° cuatrimestre
FUNDAMENTOS DE LA INFORMATICA	TEC.UNIV.ELECT.	15/13 -CD	2018	1° cuatrimestre
FUNDAMENTOS DE LA INFORMATICA	TEC.REDES COMP.	12/15	2018	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
FERNANDEZ, JACQUELINE MYRIAM	Prof. Responsable	P.Adj Exc	40 Hs
VIANO, HUGO JOSE	Responsable de Práctico	JTP Exc	40 Hs
GRANERO, JAVIER ALEJANDRO	Auxiliar de Práctico	A.2da Simp	10 Hs
HERRERA, JESUS WALTER	Auxiliar de Práctico	A.2da Simp	10 Hs
LIENDO, CLAUDIA ALEJANDRA	Auxiliar de Práctico	A.1ra Semi	20 Hs
ROSAS, MARIA VERONICA	Auxiliar de Práctico	A.1ra Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
6 Hs	2 Hs	4 Hs	Hs	6 Hs

Tipificación	Periodo
C - Teoria con prácticas de aula	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
12/03/2018	23/06/2018	15	90

IV - Fundamentación

La resolución de problemas mediante la programación es un proceso complejo en el cual intervienen técnicas con diferente grado de formalismo. Se requiere de un proceso inicial que promueva en el alumno actividades vinculadas a la resolución de problemas de diferentes características cuya solución, en principio, pueda ser expresada de una manera flexible, apelando a metodologías tales como el Aprendizaje Basado en Resolución de Problemas. Se pretende introducir al alumno con la problemática de analizar y resolver problemas de carácter general y la transformación de los mismos para, posteriormente, poder ser resueltos por una computadora. Al mismo tiempo que se introduce al alumno en relación a las componentes fundamentales de una computadora y de cómo éstas interactúan entre sí, se plantea la necesidad de definir soluciones siguiendo un enfoque lógico y algorítmico que permitirá dar al alumno una formación sólida en el área de programación de computadoras.

V - Objetivos / Resultados de Aprendizaje

Se pretende enfrentar al alumno con la problemática de analizar y resolver problemas de carácter general y la transformación de los mismos para poder ser resueltos posteriormente por una computadora; al mismo tiempo que se lo interioriza de los componentes fundamentales de la misma y de cómo éstos interactúan entre sí.
Las principales expectativas de logro incluyen el desarrollar en el alumno la capacidad de:
• Resolver problemas de tipo general trabajando los 4 pilares fundamentales del pensamiento computacional: descomposición, abstracción, reconocimiento de patrones y algoritmos.
• Diseñar e Implementar algoritmos de solución a dichos problemas en forma computacional, utilizando un lenguaje de diseño de algoritmos.
• Usar diferentes herramientas para el análisis, diseño y codificación de las soluciones.
• Integrar conceptualmente los componentes fundamentales de una máquina con el proceso de resolución computacional.

VI - Contenidos

Bolilla I: Arquitectura de las Computadoras.
Hardware. Las partes de una computadora. Componentes Internos Principales: Unidad Central de Proceso, unidad de Memoria. Componentes Externos: Dispositivos Periféricos; Clasificación de los periféricos: Periféricos de Entrada, Periféricos de Salida y Periféricos de Entrada/Salida. Memoria Auxiliar. Procesadores y MicroProcesadores.
Software. La información (datos) dentro de la computadora. Software de base. Software de aplicación. El Sistema Operativo. Tipos de sistemas operativos

Bolilla II: Introducción a la Lógica Proposicional.
Proposiciones. Funciones proposicionales. Variables proposicionales. Funtores de verdad (a) Conectivos: Conjunción, Disjunción (Inclusiva y Exclusiva), Condicional, Bicondicional. b) Noconectivos: Negación. Interpretación y Resolución de problemas lógicos. Algebra de Boole. Circuitos Lógicos: Conceptos básicos.

Bolilla III: Resolución de Problemas.
Estrategia de resolución. 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.

Bolilla IV: Algoritmos.
Conceptos de Algoritmo y Programa. Características de los Algoritmos. Lenguajes algorítmicos gráficos y no gráficos. Lenguaje algorítmico no gráfico: Lenguaje del problema. Acciones y Estructuras de control de las acciones: Secuencia, Condicional, Repetición o iteración. Datos. Tipos de datos. Concepto de variable. Lenguaje algorítmico gráfico: Diagrama de flujo.

Bolilla V: Diseño y Desarrollo de Algoritmos.
Lenguaje de Diseño. Datos: tipos de datos primitivos y manipulación. Sintaxis de las acciones. Estructuras de control de las acciones: Secuencia, Condicional, Repetición o iteración en Lenguaje de Diseño. Estructuración de los datos: concepto de Arreglo. Modularización de los problemas: concepto de Subalgoritmo, declaración, invocación, clases de parámetros. Pasaje de parámetros.

Bolilla VI: Lenguaje de Programación "C".
Acciones y descripción de datos. Datos: declaración. Constantes, variables, tipos de datos primitivos: entero, flotante y caracter. Sentencia de asignación, Estructuras de control de las acciones: Secuencia, Condicional, Repetición o iteración en C. Particularidades de almacenamiento de Datos. Arreglos. Características. Declaración. Usos. Modularización: Concepto de Función, declaración, invocación, parámetros. Pasaje de parámetros. Uso de las funciones.

VII - Plan de Trabajos Prácticos

Práctico Nº 1: Familiarización con el equipo.

Objetivos específicos: Poner en contacto al alumno con la terminología específica, que identifique los principales componentes de una computadora y sus funciones, y determine su influencia en la resolución de problemas de índole

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, memoria volátil y no volátil, entre otros).

Práctico 2: Lógica Proposicional.

Objetivos específicos: Introducir al alumno en el mundo de la Lógica a partir de su primer componente, el sintáctico, mediante el cual se podrán construir enunciados acerca de su entorno (real o imaginario).
El práctico propone una combinación de ejercicios de interpretación de texto y su posterior representación en forma simbólica, la interpretación en especificaciones en forma simbólica y su expresión en forma verbal, el uso de reglas de equivalencias, construcción de circuitos digitales a partir de expresiones simbólicas y definición de formas simbólicas a partir de determinados circuitos digitales.

Práctico Nº 3: Resolución de Problemas. (Primera Parte)
Objetivos específicos: La Computadora es el medio y no es el fin para la resolución de problemas. La mayor dificultad radica en el propio proceso de resolver el problema (determinar un algoritmo) más que en escribir un programa en un lenguaje de programación. Se desea introducir al alumno en el análisis de problemas computacionales, en el razonamiento y en la definición de la correspondiente solución algorítmica.
Esta primera parte estará destinada a trabajar con el alumno el cómo razonar la posible solución de un problema determinado. El alumno deberá encontrar algoritmos que resuelvan problemas de orden general, partiendo de problemas cotidianos y simples (por ejemplo cambiar la rueda de un auto) hasta llegar a problemas computacionales más complejos.
Refinamiento Sucesivo. (Segunda Parte)
Objetivos específicos: Afianzar en el alumno la capacidad de análisis de problemas para la determinación de una solución algorítmica. Para ello se desea profundizar en el proceso de análisis de problemas computacionales, la determinación de patrones, la generalización, la construcción de abstracciones, el razonamiento y posterior definición del correspondiente algoritmo a través de la aplicación de la Técnica de Refinamiento Sucesivo.
Esta segunda parte estará destinada a trabajar con el alumno el cómo razonar una posible solución a un problema determinado. El alumno deberá encontrar algoritmos que resuelvan problemas de orden general.

Práctico Nº 4: Lenguaje Algorítmico No Gráfico: Lenguaje TIMBA (Primera Parte)
Objetivos específicos: Profundizar en el desarrollo de la capacidad de análisis de problemas y en el razonamiento de la posible solución algorítmica a los mismos determinando las estructuras de control necesarias a partir de las características del proceso definido.
La práctica involucra actividad en el aula(obligatoria) y de laboratorio (opcional). La práctica de aula está destinada a determinar el algoritmo y su posterior codificación en Lenguaje de Timba de ejercicios puntuales. La práctica de laboratorio consistirá en introducir al alumno 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 TIMBA y resuelven el problema planteado.

Lenguaje Algorítmico Gráfico:Diagramas de Flujo (Segunda Parte)
Objetivos específicos: Introducir al alumno en la definición de la representación gráfica de los algoritmos.
La práctica consistirá en la realización de los diagramas de flujo de los algoritmos definidos en ejercicios anteriores.

Práctico Nº 5: Lenguaje de Diseño: Introducción.

Objetivos específicos: Profundizar en el alumno la capacidad de elaborar soluciones generales a diferentes problemas computacionales e introducirlo en el manejo de una herramienta que ayude a complementar el proceso de resolución, independientemente de restricciones específicas (un procesador) y que facilite la comprensión de conceptos fundamentales en el proceso de programar como por ejemplo el uso de variables, atributos de la variable, estructuras de control, entre otros.
La práctica involucra actividad en el aula (obligatoria) y de laboratorio (opcional). La práctica áulica consistirá en: desarrollar algoritmos de problemas nuevos para reforzar la aplicación de la metodología de trabajo planteada desde el práctico 3, codificar ejercicios puntuales de prácticos anteriores, de la representación gráfica de los algoritmos, la detección de errores en algoritmos propuestos por el docente y su posterior corrección. Para la resolución de los ejercicios se continuará aplicando la metodología del Refinamiento Sucesivo y la realización de los diagramas de flujos respectivos.

La práctica de laboratorio consistirá en poner en contacto al alumno con el manejo de un ambiente determinado que le permitirá probar en una computadora si los algoritmos definidos cumplen con las restricciones del lenguaje y si resuelven el problema planteado.

Práctico Nº 6: Lenguaje de Diseño: Estructuración de datos.

Objetivos específicos: Introducir el concepto de manipulación colectiva de datos: estructuras de datos, sus características, ventajas de su uso y su implementación en Lenguaje de Diseño.
La práctica involucra actividad en el aula (obligatoria) y de laboratorio (opcional). La práctica áulica consistirá en desarrollar nuevos algoritmos de problemas donde se requiere la definición de estructuras de datos y su codificación en Lenguaje de Diseño, la 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.
La práctica de laboratorio consistirá en probar, en computadora, si los algoritmos desarrollados en la práctica áulica cumplen con las limitaciones del lenguaje, las restricciones propias de la estructura de datos arreglo y resuelven el problema planteado.

Práctico Nº 7: Lenguaje de Diseño: Modularización.

Objetivos específicos: Introducir el concepto de modularización, concepto de parámetros, tipos de parámetros, ventajas de su uso y su implementación en Lenguaje de Diseño: Subalgoritmos.
La práctica involucra actividad en el aula. 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.

Práctico Nº 8: Lenguaje C: Introducción.

Objetivos específicos: Introducir al alumno con las características del Lenguaje C, tipos de datos, estructuras de control. Aplicación de los conceptos aprendidos en Lenguaje de Diseño al Lenguaje C.
La práctica involucra actividad en el aula. La práctica áulica consistirá en: codificar en Lenguaje C ejercicios puntuales del práctico anterior, desarrollar algoritmos de problemas nuevos para reforzar la metodología de trabajo planteada y el manejo del lenguaje, representación gráfica de los algoritmos, detección de errores en algoritmos propuestos por el docente y su posterior corrección.

Práctico Nº 9: Lenguaje C: Arreglos.

Objetivos específicos: Profundizar en el concepto de manipulación colectiva de datos: estructuras de datos, sus características, ventajas de su uso. Definir su implementación en Lenguaje C. Estudio comparativo entre Lenguaje de Diseño y Lenguaje C.
La práctica involucra actividad en el aula. La práctica áulica consistirá en desarrollar nuevos algoritmos de problemas donde se requiere la definición de esta estructura de datos y su codificación en Lenguaje C, la detección de errores en algoritmos propuestos y su posterior corrección.

Práctico Nº 10: Lenguaje C: Funciones.

Objetivos específicos: Profundizar en el concepto de modularización, pasaje de parámetros en Lenguaje C, ventajas de su uso y su implementación. Estudio comparativo entre Lenguaje de Diseño y Lenguaje C.
La práctica involucra actividad en el aula. La práctica áulica consistirá en: codificar en Lenguaje C ejercicios puntuales donde es clara la necesidad de determinar funciones, ejercicios de traducción al lenguaje C de programas codificados en Lenguaje de Diseño desarrollados en el práctico de Subalgoritmos en Lenguaje de Diseño), desarrollar algoritmos de problemas nuevos para reforzar el concepto de modularización y el manejo del lenguaje, detección de errores en algoritmos propuestos por el docente y su posterior corrección.

VIII - Regimen de Aprobación

La materia se divide en 3 ejes temáticos principales, a saber: Lógica, Resolución de Problemas y Resolución de Problemas en Lenguaje C, los cuales se evalúan en forma independiente. La asistencia, los cuestionarios y el trabajo en las clases prácticas también serán considerados como elementos de evaluación. Las evaluaciones parciales serán de tipo teórico-prácticas. Se tomaran 2 evaluaciones parciales. Por cada parcial se tomarán 2 recuperaciones.

Régimen de Promoción

-Asistencia al 80% de todas las actividades a realizar.

-Aprobar el 80% de los cuestionarios que se toman al comienzo de cada tema.

-Aprobar las evaluaciones parciales o su correspondiente primer recuperación, con un mínimo del 80%.

-Aprobar una evaluación adicional teórica en carácter de General, a fin de cuatrimestre, la cual se debe aprobar con un mínimo del 80%.

- Si un alumno que ha aprobado un parcial desea rendir nuevamente la correspondiente primer recuperación para optar por la promoción, se considerará la última nota obtenida.

- La nota final surgirá de los resultados obtenidos en todas las evaluaciones, la cual será 7 o más.

Régimen de Regularización

-Asistencia al 70% de de todas las actividades a realizar.

-Aprobar el 70% de los cuestionarios que se toman al comienzo de cada tema.

-Aprobar las evaluaciones parciales, o sus respectivas recuperaciones, con un mínimo del 70%.

-Los alumnos que hayan cumplimentado los requisitos anteriormente citados podrán presentarse a rendir examen final en cualquiera de los turnos establecidos por la reglamentación de facultad.

Régimen de Alumnos Libres

Dada la necesidad de un constante seguimiento del alumno en clase y de tratarse de una materia netamente práctica, no está habilitada la posibilidad de rendirla 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] “Lógica simbólica y elementos de metodología de la ciencia”, Gianella de Salama, Alicia; Roulet, Margarita - Publicación Buenos Aires: El Ateneo , 1996.

[4] "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).

[5] “Organización y Arquitectura de Computadores - Diseño para optimizar prestaciones” – William Stallings - Prentice Hall -5ta Edición - ISBN: 84-205-993-1, 2004.

[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.“Lógica Computacional”, Paniagua Arís, Enrique, Sánches Gonzalez, Juan Luis, Rubio, Fernando Martín. Thomson; ISBN: 8497321820, 2003.

[7] “Algoritmos + Estructuras de Datos = Programas”, N. Wirth, Ed. Dossat - ISBN: 8421901729, 1999.

[8] “Peter Norton's New Inside the PC”, Peter Norton, Ed. Sams, ISBN 0672322897, 2002.

[9] PSeInt (http://pseint.sourceforge.net/)

[10] "The C programming language", Kernighan, Brian W.; Ritchie, Dennis M.; Segunda Edición; ISBN: 0131103628, 1988.

[11] "Practical C programming" Oualline, Steve, Editorial O'Reilly Media, ISBN:1565923065, Ubicación: Depósito,1997.

[12] "Cómo plantear y resolver problemas" Polya, G.; XXI Reimpresión; Traducción de How to Solve it?; Editorial Trillas; ISBN:968-24-0064-3; 1997.

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] “Algorithmic Problem Solving”, Roland Backhouse – Ed. Willey – ISBN 978-0-470-68453-5, 2011

[3] “Introducción a la Informática”, Prieto Espinosa, Alberto, LLoris Ruiz, Antonio Torres Cantero, Juan Carlos. McGraw-Hill Interamericana de España; 3ra Edición. ISBN: 8448132173, 2001.

[4] “Introduction to Mathematical Logic, Third Edition”, Elliott Mendelson - Van Nostrand Reinhold Company -

ISBN-10:0534066240, ISBN-13: 978-534066246, 1987.
[5] “Structured Programming”, O.-J. Dahl, E. W. Dijkstra, C. A. R. Hoare, Academic Press, London, ISBN 0-12-200550-3,1972.
[6] "Problem Solving & Computer Programming", Peter Grogono, Addison-Wesley Publishing Company, 1982, ISBN 0-201-02460-8.

XI - Resumen de Objetivos

Desarrollar en el alumno la capacidad de:
-Resolución de Problemas.
-Diseñar e Implementar algoritmos de solución a dichos problemas en forma computacional.
-Usar diferentes herramientas para el análisis, diseño y codificación de las soluciones.

XII - Resumen del Programa

Arquitectura de las Computadoras: componentes fundamentales. Introducción a la Lógica. Resolución de Problemas: Refinamiento Sucesivo. Diseño de Algoritmos. Estructuras de control. Lenguaje de Diseño. Lenguaje de Programación "C"

XIII - Imprevistos

En caso que el desarrollo normal de las actividades planificadas se vean afectadas por reclamos laborales, se realizará una selección y recorte de contenidos priorizando aquellos que resulten esenciales al objetivo del curso, suspendiendo aquellos que no resultanse tan prioritarios o que se pueden retomar en cursos más avanzados.

XIV - Otros