Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area III: Servicios

(Programa del año 2015)
(Programa en trámite de aprobación)
(Presentado el 02/05/2015 12:15:28)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
FUNDAMENTOS DE LA INFORMATICA	ING.ELECT.O.S.D	13/08	2015	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
FERNANDEZ, JACQUELINE MYRIAM	Prof. Responsable	P.Adj Exc	40 Hs
FUENTES, MONICA GRACIELA	Responsable de Práctico	JTP Exc	40 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**
Hs	2 Hs	4 Hs	1 Hs	7 Hs

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

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
16/03/2015	26/06/2015	15	105

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

Considerando que se pretende enfrentar al alumno con toda la problemática que involucra la resolución de problemas de carácter general y problemas computacionales, y posteriormente abordar su transformación para poder ser resueltos por una computadora; las principales expectativas de logro se basan fundamentalmente en desarrollar en el alumno la capacidad de:
- Resolver problemas de tipo general y computacionales,
- Diseñar e Implementar algoritmos de solución a problemas en forma computacional.
- 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. Lenguaje algoritmico que implementa las estructuras de control y acciones. 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. 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. Acciones: Sentencias de asignación, Selección y Repetición. 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

TP. 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).
TP. Nº 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 resulta de la 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.
TP. Nº 3: Lenguajes Algorítmicos: Lenguaje Timba y Diagrama de Flujo
Objetivos específicos: Introducir al alumno la capacidad en la tarea de análisis de problemas para la determinación de una

solución algorítmica. La inclusión de la herramienta Timba le permite al alumno descubrir las características de la programación estructurada a partir de la combinación de las diferentes estructuras de control.
La práctica involucra actividad en el aula y de laboratorio. La actividad en el aula consiste en determinar el algoritmo para cada uno de los ejercicios y su posterior codificación en Lenguaje Timba y Diagrama de flujo..
La práctica de laboratorio consistirá en poner en contacto al alumno con el manejo de un ambiente determinado que le permitirá, a partir de este momento, probar en una computadora si los programas codificados en Lenguaje Timba cumplen con las restricciones del lenguaje y resuelven el problema planteado.
TP. Nº 4: Resolución de Problemas algorítmicos: Lenguaje de Problemas
Objetivos específicos: 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 fortalecer en el alumno el desarrollo de las diferentes habilidades que se ponen en juego durante en el análisis de situaciones que involucran aspectos interpretativos, deductivos y perceptivos para determinar el algoritmo que lo resuelve.
Se desea introducir al alumno en el análisis de problemas computacionales, en el razonamiento y definición de la correspondiente solución algorítmica, la determinación de los datos, sus relaciones y de los resultados esperados. Para ello se profundizará en el proceso de resolución de problemas a través de la aplicación de la Técnica de Refinamiento Sucesivo.
El objetivo de esta práctica es afianzar en el alumno el cómo razonar una posible solución algorítmica a un problema determinado y su representación gráfica mediante diagramas de flujo.
TP. 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 y de laboratorio. La práctica áulica consistirá en: codificar en Lenguaje de Diseño ejercicios puntuales del práctico 4, desarrollar algoritmos de problemas nuevos para reforzar la aplicación de la metodología de trabajo planteada en el práctico 4 y 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, a partir de este momento, probar en una computadora si los algoritmos definidos cumplen con las restricciones del lenguaje y si resuelven el problema planteado.
TP. 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 y de laboratorio. 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 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 4 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.
TP. 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 4 y la realización de los diagramas de flujos respectivos.
TP. 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.

TP. 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.
TP. 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 7 (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 y el trabajo en las clases prácticas y de laboratorio 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 las clases prácticas/laboratorios.
-Aprobar las evaluaciones parciales o sus recuperaciones, 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%.
- La nota final surgirá de los resultados obtenidos en todas las evaluaciones, la cual debe ser 7 o más.
Régimen de Regularización
-Asistencia al 70% de las clases prácticas.
-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] “Introducción a la Informática”, Prieto Espinosa, Alberto, LLoris Ruiz, Antonio Torres Cantero, Juan Carlos. McGraw-Hill Interamericana de España; 3ra Edición. ISBN: 8448132173, 2001.
[3] “Introduction to Mathematical Logic, Third Edition”, Elliott Mendelson - Van Nostrand Reinhold Company - ISBN-10:0534066240, ISBN-13: 978-534066246, 1987.
[4] “Structured Programming”, O.-J. Dahl, E. W. Dijkstra, C. A. R. Hoare, Academic Press, London, ISBN 0-12-200550-3,1972.
[5] "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 y Lenguaje Timba. Diseño de Algoritmos. Lenguaje de Diseño. Lenguaje de Programación "C"

XIII - Imprevistos

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	