



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

(Programa del año 2014)

I - Oferta Académica

| Materia | Carrera | Plan | Año | Período |
|-------------------------------|--------------|-------|------|-----------------|
| INTRODUCCION A LA COMPUTACION | LIC.CS.COMP. | 32/12 | 2014 | 1° cuatrimestre |

II - Equipo Docente

| Docente | Función | Cargo | Dedicación |
|-----------------------------|-------------------------|------------|------------|
| ESQUIVEL, SUSANA CECILIA | Prof. Responsable | P.Tit. Exc | 40 Hs |
| LEGUIZAMON, MARIO GUILLERMO | Prof. Responsable | P.Adj Exc | 40 Hs |
| ARAGON, VICTORIA SOLEDAD | Responsable de Práctico | JTP Exc | 40 Hs |
| GATICA, CLAUDIA RUTH | Auxiliar de Práctico | A.1ra Semi | 20 Hs |

III - Características del Curso

| Credito Horario Semanal | | | | |
|-------------------------|----------|-------------------|---------------------------------------|-------|
| Teórico/Práctico | Teóricas | Prácticas de Aula | Práct. de lab/ camp/ Resid/ PIP, etc. | Total |
| 7 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 |
| 12/03/2014 | 19/06/2014 | 15 | 105 |

IV - Fundamentación

Esta materia, por ser la primera específica de la carrera, brinda los fundamentos básicos, requeridos por las posteriores materias de Programación, esencialmente apunta a que los estudiantes desarrollen estrategias de resolución de problemas usando estructuras algorítmicas para la expresión de las soluciones.

V - Objetivos / Resultados de Aprendizaje

Desarrollar en el alumno la capacidad de resolver distintos tipos de problemas expresando su solución en forma algorítmica. Desarrollar en el alumno la capacidad de diseñar algoritmos utilizando un lenguaje de diseño de algoritmos. Introducir al alumno en conceptos formales de verificación algoritmos.

VI - Contenidos

BOLILLA 1: LÓGICA PROPOSICIONAL

El lenguaje de las proposiciones: Alfabeto y sintaxis de las fórmulas bien formadas. Construcción de enunciados. Conjunto de significados. Equivalencia lógica.

BOLILLA 2: RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS

Problemas, modelos y abstracciones. Algunos conceptos sobre Aprendizaje Basado en Resolución de Acertijos.

Representación de problemas. Pasos en el proceso de resolución de problemas. Resolución de problemas y Computadoras: Formulación del problema, diseño de algoritmos, codificación y ejecución. Metodología de refinamiento por pasos sucesivos.

BOLILLA 3: LENGUAJE DE DISEÑO DE ALGORITMOS.

Formalización del concepto de algoritmo: formalización del ambiente de un problema. Transformación del ambiente. Objetos constantes y objetos variables. Tipos de datos primitivos. Expresiones. Asignación. Estructuras de control: secuencia, condicional, iteraciones.

BOLILLA 4: LENGUAJE DE DISEÑO DE ALGORITMOS Y ESTRUCTURAS DE DATOS

Definición de estructura de datos. Arreglos lineales. Operaciones sobre arreglos lineales.

BOLILLA 5: LENGUAJE DE DISEÑO DE ALGORITMOS Y SUBALGORITMOS

Definición de subalgoritmos. Ambiente de un subalgoritmo. Parámetros de un subalgoritmo. Invocación de subalgoritmos.

BOLILLA 6: LÓGICA DE PREDICADOS

Limitaciones de la Lógica Proposicional. Lenguaje de Predicados: alfabeto, términos y predicados. Objetos y conjuntos de objetos.

BOLILLA 7: INTRODUCCIÓN A LA VERIFICACION DE ALGORITMOS

Especificación de algoritmos. Pre/Post condiciones. Verificación de Algoritmos. Reglas básicas de verificación. Reglas específicas: asignación simple, composición secuencial, alternativa e iterativa.

VII - Plan de Trabajos Prácticos

El plan de trabajos prácticos de aula consiste en la realización ejercicios de tipo lápiz y papel. Además, si corresponde, se incluyen ejercicios de laboratorio, los cuales son resueltos utilizando PSeInt. Ésta es una herramienta para aprender la lógica de programación, orientada a estudiantes sin experiencia en dicha área.

La metodología seguida en todos los prácticos consiste en la realización de un porcentaje de prácticos bajo la supervisión de los ayudantes y el resto como tarea fuera de clase. Para el seguimiento de las actividades de los alumnos, se requiere la presentación de un conjunto de ejercicios de cada práctico los cuales son evaluados para realizar una devolución por parte de los ayudantes.

PRÁCTICO 1: LÓGICA PROPOSICIONAL

Descripción:

Esta guía está orientada a que los alumnos fijen, comprendan y apliquen los conceptos relacionados con la lógica proposicional. El lenguaje de las proposiciones: Alfabeto y sintaxis de las fórmulas bien formadas. Construcción de enunciados. Conjunto de significados. Equivalencia lógica.

Objetivos Específicos: Al finalizar práctico se espera que el alumno sea capaz de:

- a) Identificar el alfabeto y la sintaxis de las fórmulas bien formadas
- b) Poder construir formulas bien formadas a partir de enunciados expresados en lenguaje coloquial.
- c) Determinar el valor de verdad de formulas bien formadas utilizando las tablas de verdad, es decir el conjunto de significados de las fórmulas bien formadas.
- d) Determinar la equivalencia lógica de fórmulas bien formadas utilizando las tablas de verdad.

PRÁCTICO 2: RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS

Descripción:

Esta guía está orientada a que los alumnos fijen, comprendan y apliquen los conceptos relacionados con la resolución de problemas y el desarrollo de algoritmos.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de:

- a) Crear modelos y abstracciones de problemas de la vida cotidiana.
- b) Dado un problema identificar los datos de entrada y los datos de salida del mismo.
- c) Descomponer problemas complejos en tareas más sencillas con el objeto de resolver el problema original.
- d) Diseñar algoritmos que resuelvan problemas utilizando un conjunto finito de instrucciones y la posterior ejecución de dichas instrucciones.

PRÁCTICO 3: LENGUAJE DE DISEÑO DE ALGORITMOS – EXPRESIONES Y ASIGNACIONES.

Descripción:

Esta guía busca profundizar la escritura de expresiones del lenguaje de diseño, analizando sintaxis y significado. Se hace hincapié en la traducción de enunciados en lenguaje natural a expresiones aritméticas, lógicas y relacionales. Además de identificar los objetos variables y constantes de un problema.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de:

- a) Describir el ambiente de un algoritmo: variables de entrada, salida y auxiliares y constantes significativas.
- b) Identificar los objetos variables y constantes de un problema. Declaración de variables en el algoritmo.
- c) Identificar los tipos de datos primitivos que provee el lenguaje de diseño y las operaciones definidas para cada tipo de dato.
- d) Construir expresiones en lenguaje de diseño.
- e) Identificar el mecanismo de trabajo del operador de asignación. Asignaciones válidas e inválidas.

PRÁCTICO 4: LENGUAJE DE DISEÑO DE ALGORITMOS – ENTRADA Y SALIDA DE DATOS - SELECCION.

Descripción:

Esta guía busca profundizar estructura de control condicional simple y múltiple.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de comprender el mecanismo de trabajo de una estructura de control condicional:

- a) Ejecución de algoritmos con estructuras de control condicional.
- b) Desarrollo de algoritmos que involucran el uso de las estructuras de control condicional: simple y múltiple.

PRÁCTICO 5: LENGUAJE DE DISEÑO DE ALGORITMOS - ITERACION.

Descripción:

Esta guía busca profundizar estructura de control de iteración.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de comprender el mecanismo de trabajo de una estructura de control de iteración:

- a) Ejecución de algoritmos con estructuras de control de iteración.
- b) Desarrollo de algoritmos que involucran el uso de tres estructuras de control de iteración.

PRÁCTICO 6: LENGUAJE DE DISEÑO DE ALGORITMOS Y ESTRUCTURAS DE DATOS

Descripción:

Esta guía está orientada a que los alumnos fijen, comprendan y apliquen los conceptos relacionados con la estructura de datos: arreglo lineal.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de

- a) Identificar la necesidad de utilizar uno o más arreglos lineales para resolver un problema.
- b) Declaración de un arreglo en lenguaje de diseño.
- c) Conocer las operaciones válidas sobre las componentes de un arreglo: asignación de un valor y consulta del valor de un elemento del arreglo.

PRÁCTICO 7: LENGUAJE DE DISEÑO DE ALGORITMOS - SUBALGORITMOS

Descripción:

Esta guía está orientada a que los alumnos fijen, comprendan y apliquen los conceptos relacionados con la modularización de procesos.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de

- a) Definir subalgoritmos. Ambiente de un subalgoritmo. Parámetros de un subalgoritmo.
- b) Invocar subalgoritmos.

PRÁCTICO 8: LÓGICA DE PREDICADOS

Descripción:

Esta guía está orientada a que los alumnos fijen, comprendan y apliquen los conceptos relacionados con la lógica de predicados.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de

- a) Identificar las limitaciones de la Lógica Proposicional.
- b) Identificar el alfabeto, términos y predicados de la lógica de predicados. Objetos y conjuntos de objetos.

PRÁCTICO 9: INTRODUCCIÓN A LA VERIFICACION DE ALGORITMOS

Descripción:

Esta guía está orientada a que los alumnos fijen, comprendan y apliquen los conceptos relacionados con la verificación de algoritmos.

Objetivos Específicos: Al finalizar el práctico se espera que el alumno sea capaz de

- a) Desarrollar una especificación de algoritmos. Pre/Post condiciones.
- b) Desarrollar una verificación de Algoritmos.

VIII - Regimen de Aprobación

Los alumnos podrán promocionar o regularizar la materia, contando para ello con una asistencia, a los prácticos, de al menos el 80%, según los siguientes regímenes:

F.1. Régimen para alumnos promocionales

Para poder promocionar la materia los alumnos deberán:

1. Presentar el 80% de las preguntas requeridas por los docentes y
2. Aprobar el parcial o su respectiva recuperación con nota 7 o superior en cada uno.
3. Aquellos alumnos que han presentado, en tiempo y forma, el comprobante de trabajo, contarán con una recuperación extra integradora, la cual deberá ser aprobada con nota 7 o superior.

F.2. Régimen para alumnos regulares

Para poder regularizar la materia los alumnos deberán:

1. Presentar el 80% de las preguntas requeridas por los docentes y el práctico de laboratorio, y
2. Aprobar el parcial o su respectiva recuperación con nota menor que 7 pero superior o igual a 6.

3. Aquellos alumnos que han presentado, en tiempo y forma, el comprobante de trabajo, contarán con una recuperación extra integradora, la cual deberá ser aprobada con nota menor que 7 pero superior o igual a 6.

F.3. Régimen de alumnos libres

Los alumnos que no cumplen con los requisitos del régimen promocional o regular podrán rendir la materia como alumnos libres. Para ello deberán entregar un práctico de laboratorio, rendir un examen escrito sobre temas teóricos y prácticos, siendo obligatoria la aprobación de la parte práctica para considerar la parte teórica.

IX - Bibliografía Básica

- [1] - M.J. Abásolo, F. Guerrero y J. Perales López; Introducción a la Programación, Colección materiales didácticos, Universidad de las Islas Baleares, 2011.
- [2] - E.P. Arís, J. L. Sánchez González y F. M. Rubio, Lógica Computacional, Thomson Editores, España, 2003.
- [3] - G. Polya, Cómo plantear y resolver problemas, Editorial Trillas, México, 1970.
- [4] - S. Braustein y A. Gioia, Introducción a la Programación y a las Estructuras de Datos, Eudeba, Argentina, 1986.
- [5] - Z. Michalewicz y M. Michalewicz, Puzzle Based Learning: Introduction to critical thinking, mathematics, and problem solving; Hybrid Publishers, 2008.
- [6] - J.J. Garcia Molina, J. Fernandez Aleman, M.J. Majado Rosales y J. Montoya Dato Francisco, Una Introducción a la Programación - Un Enfoque Algorítmico, Editorial: PARANINFO, 2005.
- [7] - E. Paniagua Arís. J.L. Sánchez González y F. Martín Rubio, Lógica Computacional, Thomson Editores España - Paraninfo, 2003.
- [8] - A.G Hamilton, Lógica para Matemáticos, Paraninfo, Madrid, 1981.
- [9] - Apuntes de la Asignatura.
- [10] - PSeInt. Facultad de Ingeniería y Ciencias Hídricas de la Universidad Nacional del Litoral. <http://pseint.sourceforge.net/>

X - Bibliografía Complementaria

[1]

XI - Resumen de Objetivos

Desarrollar en el alumno la capacidad de resolver distintos tipos de problemas expresando su solución en forma algorítmica. Desarrollar en el alumno la capacidad de diseñar algoritmos utilizando un lenguaje de diseño de algoritmos. Introducir al alumno en conceptos formales de verificación algoritmos.

XII - Resumen del Programa

BOLILLA 1: LOGICA PROPOSICIONAL.

BOLILLA 2: RESOLUCIÓN DE PROBLEMAS Y ALGORITMOS. Interpretación de enunciados. Resolución de problemas tipo acertijo.

BOLILLA 3: LENGUAJE DE DISEÑO DE ALGORITMOS. Conceptos básicos. Expresiones. Asignación. Estructuras de control: secuencia, selección e iteración.

BOLILLA 4: LENGUAJE DE DISEÑO DE ALGORITMOS Y ESTRUCTURAS DE DATOS. Arreglos lineales.

BOLILLA 5: LENGUAJE DE DISEÑO DE ALGORITMOS Y SUBALGORITMOS. Modularización, parametrización, definición e invocación.

BOLILLA 6: LOGICA DE PREDICADOS.

BOLILLA 7: VERIFICACIÓN DE ALGORITMOS. Especificación y verificación de algoritmos.

XIII - Imprevistos

| |
|--|
| |
|--|

XIV - Otros

| |
|--|
| |
|--|