



CARRERA de TÉCNICO SUPERIOR en ANÁLISIS de SISTEMAS

Resolución 1148-MEGC-2007

DIAGRAMACIÓN LÓGICA

Curso: 1° Año

Profesor: Leandro E. Colombo Viña

Ciclo Lectivo: 2017

Régimen: Cuatrimestral. Primer Cuatrimestre.

Carga Horaria: 5Hs. Cat./Sem.

1. Fundamentación:

Esta instancia curricular brinda a los estudiantes herramientas de programación para utilizar en la etapa específica, desarrollando la capacidad de producir el código en el contexto de la tecnología asignada a un proyecto. Esto incluye la modificación (agregado, reemplazo o eliminación) de código ya escrito, sea para corregir errores observados en pruebas o, para cambiar funcionalidades o el comportamiento de productos con existencia previa.

2. Objetivos:

Que los estudiantes:

- ✓ Comprendan la estructura lógica para la resolución de problemas paso a paso.
- ✓ Adquieran el concepto de programa.
- ✓ Identifiquen componentes o partes que puedan ser potencialmente reutilizados en el futuro.
- ✓ Dominen las operaciones implicadas en la fase de programación.
- ✓ Respeten los estándares de buena práctica y normas internas estipuladas para el proyecto.

3. Promoción:

- ✓ Asistencia al 75% de las clases.
- ✓ Aprobación de los Trabajos Prácticos.
- ✓ Aprobación de 2 (dos) exámenes parciales.
- ✓ Aprobación de 1 (un) examen final.

4. Ejes de contenidos:

- Resolución de problemas y algoritmos. Técnicas de representación lógicas y gráficas.
- Diagramación lógica. Simbología y técnicas de construcción de diagramas. Estructura sintáctica de un programa. Estructuras de Control. Recursividad. Eventos. Excepciones. Concurrencia.
- Operaciones. Tipos de datos simples y compuestos. Diferencia entre variables y constantes.
- Arreglos – Introducción. Declaración de arreglos multidimensionales.
- Declaración de tipos definidos por el usuario.
- Estructuras de control.
- Funciones y procedimientos. Programación modular. Estructura sintáctica de un programa con procedimientos. Variables locales en procedimientos. Pasaje de parámetros por valor y por referencia.
- Registros, introducción, concepto y su utilización.
- Definición de clases, escritura de algoritmos, estructuración de datos necesarios.

5. Programa analítico de la asignatura:

La asignatura se divide en 3 (tres) unidades temáticas que nuclean los contenidos según la relación existente entre los mismos.

UNIDAD 1: INTRODUCCIÓN A LA PROGRAMACIÓN.

Definición de algoritmo. Resolución de problemas básicos. Concepto de variables y constantes. Programas secuenciales. Diagramas de Nassi-Schneiderman y Pseudocódigo.

UNIDAD 2: ALGORITMOS BÁSICOS.

Condicionales y estructuras cíclicas: MIENTRAS y PARA. Procedimientos típicos. Algoritmos de control. Resolución de problemas lógicos.

UNIDAD 3: EL CÓDIGO EN LENGUAJE PYTHON.

Sintaxis básica. Introducción a la programación estructurada. Programación modular. Operadores. Variables y Constantes: tipos de datos en Python. Traducción de pseudocódigo a código.

6. Bibliografía:

Autor	Título	Editorial	Observaciones
Allen Downey, Jeffrey Elkner & Chris Meyers	Aprenda a Pensar Como un Programador con Python	Green Tea Press. Año: 2002	ISBN: 0-9716775-0-6
Rosita Wachenchauzer & Colaboradores	Algoritmos y Programación I Aprendiendo a programar usando Python como herramienta	Apuntes de la Facultad de Ingeniería de la UBA.	ISBN: N/D
Raúl González Duque	Python Para Todos	http://mundogeek.net/tutorial-python/	ISBN: N/D
Comunidad Python Argentina	Tutorial de Python	http://docs.python.org.ar/tutorial/	Proyecto de PyAr
Paul Barry	Head First Python	O'Reilly. Año: 2011	ISBN: 978-1-449-38267-4
Eugenia Bahit	Curso: Python para Principiantes	http://library.originalhacker.org/biblioteca/articulo/autor/2	Safe Creative: 1207302042960
Donald E. Knuth	The Art of Computer Programming. Volume I.	Addison-Wesley. Año: 1997	ISBN: 0-201-89683-4

7. Planificación:

Clase	Unidad	Contenido	Actividad
1	I	Concepto de Algoritmo. Características y Representación. Técnicas para la formulación de algoritmos. Diagramas de flujo. Estructuras secuenciales.	Explicación de la metodología para la resolución de algoritmos. Técnicas de representación. Resolución de algoritmos secuenciales.
2	I	Tipos de datos. Expresiones y Operadores. Estructuras condicionales.	Descripción de los tipos de datos más utilizados en programación. Resolución de algoritmos condicionales.
3	I	Estructuras algorítmicas cíclicas. Tipos de ciclos: PARA, MIENTRAS y HACER-MIENTRAS.	Resolución de algoritmos cíclicos.
4	II	Introducción al Lenguaje de Programación Python.	Estructura de un programa en Python. Componentes básicos. Código fuente, intérprete y ejecución. Sintaxis básica.
5	II	Funciones. Concepto y definición. Uso de funciones en un programa. Resultados de las funciones.	Ejemplos prácticos de funciones en Python. Definición y ejecución.
6	-	Revisión de contenidos.	Evaluación escrita. 1º Parcial.
7	II	Condicionales en Python. Características y ejemplos de uso.	Resolución de algoritmos condicionales en Python.
8	II	Cadena de caracteres. Utilización.	Características principales de las cadenas de texto en los programas en Python.
9	II	Tuplas y Listas. Iteración.	Ejemplos de uso con tuplas y listas. Comparación entre tuplas y listas.
10	III	Diccionarios. Usos típicos.	Resolución de problemas utilizando diccionarios.
11	III	Funciones. Iteración vs. Recursividad. Características.	Concepto de recursividad.
12	III	Módulos y Paquetes. Definición y ejemplos de uso.	Ejemplos de uso de reutilización de código creando módulos propios y utilizando los módulos de la librería estándar.
13	-	Revisión de contenidos.	Recuperatorio de evaluación parcial escrita.
14	-	Revisión de contenidos.	Evaluación escrita. 2º Parcial.
15	-	Revisión de contenidos.	Recuperatorio de evaluación parcial escrita.
16	-	Revisión de contenidos.	Recuperatorio de evaluación parcial escrita.