



NOMBRE DEL CURSO: Introducción a la Programación Y Computación 1

CODIGO:	770	CREDITOS:	4
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Ciencias de la Computación
PRE REQUISITO:	✓ 103 – Matemática Básica 2 ✓ 34 créditos	POST REQUISITO:	✓ 0771- Introducción a la Programación y Computación 2 ✓ 0796 - Lenguajes Formales y de Programación ✓ 0962 - Matemática para Computación 2
CATEGORIA:	Obligatorio	SEMESTRE:	Primer Semestre 2015
CATEDRÁTICO (A):	Ing. Luis Espino	AUXILIAR:	✓ Juan Pablo Gutierrez ✓ Pablo Musus
EDIFICIO:	T-7	SECCIÓN:	A
SALON DEL CURSO:	201	SALON DEL LABORATORIO:	111 – T3
HORAS POR SEMANA DEL CURSO:	4	HORAS POR SEMANA DEL LABORATORIO:	2
DÍAS QUE SE IMPARTE EL CURSO:	Martes y Jueves	DIAS QUE SE IMPARTE EL LABORATORIO:	Jueves
HORARIO DEL CURSO:	07:10 A.M. – 08:50 A.M.	HORARIO DEL LABORATORIO:	09:10AM – 10:50AM

DESCRIPCIÓN DEL CURSO:

El curso es el acercamiento inicial del estudiante de la carrera de sistemas, a la programación mediante el uso de disciplinas y metodologías especializadas. El curso se fundamenta en el concepto de algoritmo para la resolución de problemas de programación, enfatizando el uso del paradigma orientado a objetos. Se introducen conceptos básicos de UML como guía para el diseño de sistemas orientados a objetos. Se acerca al estudiante al conocimiento de los principales algoritmos de búsquedas y ordenamientos. Se cubre una parte importante de las estructuras de datos, los tipos de datos abstractos. Asimismo, el estudiante conocerá el lenguaje Java como el lenguaje oficial de programación del curso.

OBJETIVO GENERAL:

Lograr que el estudiante adquiera la habilidad de programar y los conocimientos básicos de la programación utilizando el paradigma orientado a objetos.

Objetivos Específicos:

Conocimientos

- Integrar al estudiante a la tecnología de la computación.

- Conocer las diferentes metodologías de programación.
- Organizar soluciones utilizando un lenguaje de programación.
- Conocer el lenguaje Java como el primer lenguaje de programación para computadoras.

Habilidades

- Adquirir la habilidad de hacer algoritmos.
- Aprender a elaborar diseños de clases preliminares en UML.
- Analizar los problemas con metodología orientada a objetos.

Competencias

- Que el estudiante pueda realizar una aplicación con los conceptos de programación orientada a objetos y el lenguaje java.

METODOLOGIA:

Clase presencial impartida un día a la semana con duración de 2 horas (contenido audiovisual, ejemplos, etc.).

Elaboración de investigaciones y tareas.

Evaluaciones complementarias (Cortos y hojas de trabajo).

Elaboración de prácticas y proyectos de programación.

EVALUACIÓN DEL RENDIMIENTO ACADEMICO: El laboratorio se aprueba con una nota mayor o igual a 61 puntos y tiene un equivalente en la zona de 30 puntos.

Procedimiento	Ponderación
Tareas, Investigaciones, Cortos, Hojas de trabajo	10 puntos
Repositorio	5 puntos
3 Prácticas	25 puntos
Proyecto Único	50 puntos
Examen Final	10 puntos

Nota de Promoción	100 puntos

CONTENIDO PROGRAMATICO Y CALENDARIZACIÓN:

- 1 Fundamentos de Programación
 - 1.1 Algoritmos
 - 1.2 Diagrama de Flujo
 - 1.3 Lenguajes de Programación
 - 1.4 Ciclo de vida del software

- 1.2 Introducción al control de versiones
 - 1.2.1 Git

- 1.2.2 Conceptos básicos:
 - 1.2.2.1 Clone
 - 1.2.2.2 Merge
 - 1.2.2.3 Branch
 - 1.2.2.4 Fork
 - 1.2.2.5 Pull/Push
- 1.2.3 GitHub

- 2 Programación Modular
 - 2.1 Variables
 - 2.2 Operadores aritméticos
 - 2.3 Operadores relacionales y lógicos
 - 2.4 Estructuras de Control Condicionales
 - 2.4.1 IF
 - 2.4.2 IF ELSE
 - 2.4.3 IF ELSEIF
 - 2.4.4 IF ELSEIF ELSE
 - 2.4.5 SWITCH
 - 2.5 Estructuras Cíclicas
 - 2.5.1 FOR
 - 2.5.2 WHILE
 - 2.5.3 DO-WHILE / REPEAT-UNTIL
 - 2.6 Procedimiento/Función
 - 2.6.1 Parametros
 - 2.6.2 Entorno de Variables
 - 2.6.3 Valor Retorno
 - 2.7 Recursividad

- 3 Programación orientada a objetos
 - 3.1 Abstracción
 - 3.2 Clases y objetos
 - 3.3 Miembros de una clase
 - 3.3.1 Atributos
 - 3.3.1 Metodos
 - 3.3.1 Constructores y destructores
 - 3.4 Relación entre clases y objetos
 - 3.4.1 Asociación
 - 3.4.1 Agregación y composición
 - 3.4.1 Herencia
 - 3.4 Principio de encapsulamiento
 - 3.5 Polimorfismo
 - 3.6 Visibilidad

- 4 UML
 - 4.1 Introducción a UML
 - 4.2 Definiciones
 - 4.3 Simbología

- 5 Programación en JAVA

5.1	Introducción a JAVA
5.2	Definiciones del Lenguaje
5.3	Herramientas de Desarrollo (JDK, JRE, JVM)
5.4	Entorno de Desarrollo Integrado (IDE): NetBeans
5.5	Ejemplos
5.5.1	Clases en JAVA
5.5.2	Atributos y Metodos
5.5.3	Variables Globales y Locales
5.5.4	Estructuras de Control Condicionales
5.5.5	Estructuras Cíclicas
5.6	Estructuras algorítmicas
5.6.1	Arreglos (una, bidimensionales, n-dimensionales)
5.6.2	Busqueda de datos en arreglos
5.6.3	Ordenamiento de datos en arreglos
5.6.4	Pila
5.6.5	Cola
6	Interfaces Gráficas de Usuarios (GUI)
6.1	Paquetes AWT y SWING
6.2	Frames, Componentes: Botones, Cuadros de Texto, Etiquetas, etc.
6.3	Eventos y Excepciones
6.3.1	Excepciones: Try & Catch
6.3.2	Eventos: Action, Listener, etc.
6.4	Archivos de Texto y Archivos Binarios
6.5	Abrir, cerrar y modificar archivos
7	Tipos de datos abstractos (TDA)
7.1	Tipos de apuntadores (estáticos y dinámicos)
7.2	Tipos de TDA
7.3	Representación de listas: simples, doblemente enlazadas, circulares
7.4	Representación de Pilas y Colas
9	HTML (HyperText Markup Language --- Lenguaje de marcado de HiperTexto)
9.1	Introducción, Definición, ejemplos

BIBLIOGRAFÍA:

- JOYANES, L. y ZAHONERO, I. "Programación en Java 2 (algoritmos, estructura de datos y programación orientada a objetos)". España, McGraw-Hill / Interamericana de España, S. A. 2002, PP 725
- BUDD, Timothy. "Introducción a la programación orientada a objetos", EUA, Addison-Wesley, Iberoamericana, S. A. 1994, PP. 409
- JOYANES, L. "Programación en Turbo Pascal Versiones 5.5, 6.0, y 7.0", (2da Edición), México, McGraw-Hill / Interamericana de España, S. A. 1995, PP. 914
- Manuales de Referencia de Java, <<http://www.sun.com/java>>

- Documentacion Git <http://git-scm.com/doc>
- Documentacion GitHub <https://help.github.com/>