



## Laboratorio Introducción a la Programación y computación 1

<b>CÓDIGO:</b>	0770	<b>CRÉDITOS:</b>	4
<b>ESCUELA:</b>	Ciencias y Sistemas	<b>AREA A LA QUE PERTENECE:</b>	Desarrollo de Software
<b>PRE REQUISITO:</b>	33 créditos y 0130 Matemática Básica 2	<b>POST REQUISITOS:</b>	0771 Introducción a la Programación y Computación 2, 0796 Lenguajes Formales y de Programación.
<b>CATEGORIA:</b>	Obligatorio	<b>SEMESTRE:</b>	1er. Semestre 2018
<b>CATEDRATICO(A):</b>	Ing. Nefalí Calderón	<b>AUXILIAR:</b>	Robin Salvatierra José Portillo
<b>EDIFICIO:</b>	T - 7	<b>SECCIÓN:</b>	E
<b>SALON DEL CURSO:</b>	101	<b>SALON DEL LABORATORIO:</b>	404
<b>HORAS POR SEMANA DEL CURSO</b>	4	<b>HORAS POR SEMANA DEL LABORATORIO</b>	2
<b>DÍAS QUE SE IMPARTE EL CURSO</b>	Martes y Jueves	<b>DÍAS QUE SE IMPARTE EL LABORATORIO</b>	Miércoles
<b>HORARIO DEL CURSO</b>	7:10 AM -- 8:50 AM	<b>HORARIO DEL LABORATORIO</b>	7:10 AM -- 8:50 AM

### DESCRIPCIÓN DEL LABORATORIO:

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.

**OBJETIVOS:****General**

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

**Específico**

- Integrar al estudiante a la tecnología de la computación.
- Conocer las diferentes metodologías de programación.
- Organizar soluciones utilizando un lenguaje de programación.
- Adquirir la habilidad de hacer algoritmos.
- Aprender a elaborar diseños de clases preliminares en UML.
- Analizar los problemas con metodología orientada a objetos.
- Conocer el lenguaje Java como el primer lenguaje de programación para computadoras.

**METODOLOGÍA:**

- Clases presenciales con contenido audiovisual, clases virtuales, lecturas, etc.
- Capacitación personalizada al estudiantado.
- Elaboración de Actividades.
  - Tareas
  - Prácticas.
  - Proyectos.
  - Evaluaciones.
  - Evaluación Final.

**REQUISITOS:**

- El laboratorio se aprueba con 61 puntos.
- Para la calificación de las actividades se tomara en cuenta la presentación, calidad y contenido de ellas.

**EVALUACIÓN:**

La nota de promoción para aprobar el **laboratorio** es de 61 puntos de 100.  
La ponderación para cada actividad de evaluación es la siguiente:

● Tareas	06
● Exámenes Cortos	06
● Examen Final	09
● Practica presencial	09
● 2 Prácticas (10 c/u)	20
● Proyecto 1	20
● Proyecto 2	<u>30</u>
● <b>Total</b>	<b>100</b>

## **CONTENIDO**

**Módulo introducción a la computación.**

**Módulo introducción a HTML.**

**Módulo de importancia de diseño para el desarrollo del software.**

### **1. Fundamentos de Programación**

- 1.1 Introducción a Algoritmos.
- 1.2 Diagramas de Flujo.
- 1.3 Introducción a la programación.

### **2. Programación Estructurada**

- 2.1 Tipos de Variables.
- 2.2 Estructuras de Datos.
- 2.3 Estructuras de Control.
- 2.4 Funciones y Procedimientos.

### **3. Paradigma de Objetos y UML**

- 3.1 Introducción al Paradigma Orientado a Objetos.
- 3.2 Introducción a UML.
- 3.3 Definiciones.
- 3.4 Simbología.

### **4. Introducción a Java**

- 4.1 Que es Java.
- 4.2 Versiones y ambiente de Java(JDK, IDE, máquina virtual, etc.).
- 4.3 Características de Java.
- 4.4 Tipos de programas en Java.
- 4.5 Componentes del lenguaje Java(variables, constantes, tipos de datos etc.).
- 4.6 Estructuras de control.
- 4.7 Arreglos.

### **5. Introducción a AWT y SWING**

- 5.1 Frames.
- 5.2 Botones.
- 5.3 Textbox, etc..
- 5.4 Ejemplos.

### **6. Programación Orientada a Objetos en Java(POO)**

- 6.1 Conceptos de la programación orientada a objetos.
- 6.2 Creación de clases y objetos.
- 6.3 Manejo de memoria.
- 6.4 Casteo de datos.
- 6.5 Constructores.
- 6.6 Herencia.
- 6.7 Polimorfismo.

## **7. Clases abstractas, interfaces y paquetes**

- 7.1 Control de acceso a clases y métodos.
- 7.2 Clases abstractas y métodos.
- 7.3 Interfaces.
- 7.4 Paquetes.

## **8. Manejo de Excepciones**

- 8.1 Comprensión de excepciones y el manejo de estas.
- 8.2 Sentencia Try Catch.
- 8.3 Try anidados.

## **9. Manejo de hilos en Java**

- 9.1 Creación de hilos.
- 9.2 Multi-hilo.
- 9.3 Animación empleando hilos.

## **10. Archivos**

- 10.1 Introducción a flujo de datos.
- 10.2 Flujo de entrada y salida de datos.
- 10.3 Abrir, Cerrar y Modificar Archivos.
- 10.4 Buffer de lectura y escritura.

## **11. Listas y Colas**

- 11.1 Pilas usando listas.
- 11.2 Colas usando listas.
- 11.3 Listas n-encadenadas.

### **PUNTOS IMPORTANTES A CONSIDERAR:**

- Para tener derecho a nota de laboratorio se debe cumplir con el 80% de asistencia a clase de laboratorio a excepción de presentar carta de trabajo membretada.
- Toda entrega tarde de actividades se sancionará con el 20% de la nota de la actividad a excepción de la primera vez que se falle.
- El único medio de entrega de actividades será la carpeta compartida de dropbox de cada estudiante.

### **BIBLIOGRAFIA:**

- 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, P. 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.