



Laboratorio Introducción a la Programación y Computación 1

CÓDIGO:	0770	CRÉDITOS:	4
ESCUELA:	Ciencias y Sistemas	AREA A LA QUE PERTENECE:	Desarrollo de Software
PRE REQUISITO:	33 créditos y 0130 Matemática Básica 2	POST REQUISITOS:	0771 Introducción a la Programación y Computación 2, 0796 Lenguajes Formales y de Programación.
CATEGORIA:	Obligatorio	SEMESTRE:	2do. Semestre 2019
CATEDRATICO(A):	Ing. Marlon Francisco Orellana López	AUXILIAR:	Henry Edmar López Cardona, Mario Jeancarlo Morales Rivas
EDIFICIO:	T-3	SECCIÓN:	A
SALON DEL CURSO:	215	SALON DEL LABORATORIO:	Edificio T-3, Salón 311 (Miércoles) Edificio T-3, Salón 215 (Jueves)
HORAS POR SEMANA DEL CURSO	4	HORAS POR SEMANA DEL LABORATORIO	4
DÍAS QUE SE IMPARTE EL CURSO	Martes y Jueves	DÍAS QUE SE IMPARTE EL LABORATORIO	Miércoles y Jueves
HORARIO DEL CURSO	7:10 AM -- 8:50 AM	HORARIO DEL LABORATORIO	07:10 AM – 08:50 AM y 09:00 AM – 10:40 AM, respectivamente.

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:

- Es obligatorio aprobar ambos proyectos de laboratorio con una nota de 61 puntos.
- El laboratorio se aprueba con 61 puntos.
- Las actividades a realizar en el laboratorio (tareas, investigaciones, prácticas, proyectos, evaluaciones cortas, examen final, etc.) son dependientes. Consiste en que cada actividad será dependiente de la anterior utilizando como punto de partida la Actividad 0 la cual consiste en enviar un correo electrónico a los auxiliares del curso para registrarse en el laboratorio dando con ello el inicio de la ponderación de punteo.
- Para tener derecho a su nota final debe de entregar todas las actividades.
- La forma de entrega de las actividades será vía *Classroom*, cada estudiante deberá subir los archivos solicitados según la fecha y hora límite de entrega en base al enunciado de cada actividad.
- Para la calificación de las actividades se tomará 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 | 5 |
| ● Práctica presencial | 5 |
| ● Exámenes cortos | 6 |
| ● Hojas de trabajo | 9 |
| ● 2 Prácticas (7.5 c/u) | 15 |
| ● Proyecto 1 | 20 |
| ● Proyecto 2 | 30 |

• Examen final	<u>10</u>
Total	100

CONTENIDO

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 ¿Qué 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.

12. Cloud Computing

12.1 Introducción

12.2 Funcionamiento

12.3 Servicios de la nube

12.4 Aplicaciones

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 por medio del *Classroom*, si en dado caso de requiere otra herramienta se dará aviso.

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.