



**Universidad Andrés Bello**

**Facultad de Ingeniería  
Ingeniería en Automatización y Robótica**

**Fundamentos de Programación**

## **I.- IDENTIFICACIÓN**

Nombre	:	Fundamentos de Programación
Código	:	AUT1302
Tipo de Actividad	:	T-P
Modalidad	:	Presencial
Créditos Unab	:	4
Créditos SCT	:	1
Requisitos	:	Sin requisitos - Ingreso

## **II.- DESCRIPCIÓN DEL CURSO**

Comprender los conceptos claves del lenguaje de programación Java. Analizar problemas usando análisis orientado a objetos. Desarrollo y testeo de programas desarrollados con tecnología Java.

## **III.- OBJETIVOS O RESULTADOS DE APRENDIZAJE**

### **A) Sentido y ubicación de la asignatura en el plan de estudios**

Se encuentra en el eje Control y Automatización de Procesos Industriales, en el que el alumno una vez aprobado el eje, se encontrará en condiciones de Comprender los conceptos claves del lenguaje de programación Java, analizar problemas usando análisis orientado a objetos. Desarrollo y testeo de programas desarrollados con tecnología Java, trabajar en equipos multidisciplinarios, liderar grupos de trabajo y manejar conflictos, comprender y usar el modelamiento de proyectos por UML

## **B) Aprendizajes esperados**

El estudiante desarrollará aprendizajes que le permitirán:

1. Identificar los elementos del entorno de desarrollo, las tecnologías que se utilizan el lenguaje de programación orientada a objetos
2. Identificar y usar las etapas en el ciclo de vida de desarrollo de un producto, analizar un problema de la vida real y modelar utilizando la metodología de Orientación a objetos.
3. Comprender la metodología de la programación orientada a objetos, aplicar dicha metodología sobre la resolución de problemas.

## **IV.- CONTENIDOS**

### **A) Unidad 1: Conceptos básicos de programación**

- 1.1. Análisis del problema.
- 1.2. Diseño de Algoritmo.
- 1.3. Codificación o Compilación.
- 1.4. Prueba de Depuración.

### **B) Unidad 2: Estructura básica de un programa**

### **C) Unidad 3: Tipos de Variables**

- 3.1. Primitivas.
- 3.2. Variables referenciadas.
- 3.3. Constantes.

### **D) Unidad 4: Operadores**

- 4.1. Matemáticos (Jerarquía).
- 4.2. Lógicos.

### **E) Unidad 5: Estructuras de control**

- 5.1. If - else.
- 5.2. Switch – case.
- 5.3 For.
- 5.4. While.
- 5.5. Do-while

### **F) Unidad 6: Métodos y Funciones**

### **F) Unidad 7: Strings y Matrices**

- 6.1. Strings
- 6.2 Vectores y Matrices

### **G) Unidad 8: Escritura y lectura de archivos**

## **H) Unidad 8: Puertos y Sockets**

### **8.1 Socket TCP/IP**

### **8.2 Puertos Serie**

## **H) Unidad 9: Programación orientada a objeto**

### **9.1. Creación de una clase.**

### **9.2. Creación de métodos.**

### **9.3. Constructores.**

### **9.4. Herencia.**

### **9.5. Instanciar y usar objetos.**

## **I) Unidad 10: Introducción a otros ambientes de programación**

### **10.1 LabVIEW**

### **10.2 C++**

## **IV.- METODOLOGÍAS**

El modulo se centra en el aprendizaje mediante la permanente actividad del estudiante. Como procedimiento didáctico se sugiere que el alumno aplique cada una de los contenidos de forma inmediata para que aprecie su funcionamiento.

El modulo enfatiza la aplicación de todo lo que se enseña, para hacer más agradable el estudio de las materias es recomendable que los ejemplos se realicen de acuerdo a la realidad del alumno, dando ejemplos acorde a su carrera ajustada a la complejidad de la materia.

Los conocimientos de los módulos son acumulativas para el alumno adquiera los conocimientos más rápidamente es útil que el profesor haga mención de la materia de los módulos anteriores para que el alumno vaya repasando la materia.

Las unidad 1 es importante que el alumno tenga en claro cómo van las instrucciones en especial de control y ciclo, es recomendable que el profesor muestre como trabajan esas instrucciones mediante figuras.

Las horas que no son de laboratorio es recomendable que se realicen ejercicios en la pizarra donde se les demuestre a los estudiantes como se aplican las instrucciones o la metodología para realizar los problemas en los laboratorios siempre con ejercicios para aplicar la materia.

Para las unidades de la 2 hasta la 6 es recomendable que se realicen un ejercicio que aumentará su complejidad en la medida en que usen más instrucciones que son vistas en clases así el alumno se dará cuenta como aumenta la complejidad y el funcionamiento debido a que tiene más instrucciones para realizarla, se recomienda que el alumno aplique los conocimientos aprendidos en ejercicios de aplicación que tengan el mismo estilo que los ejemplos vistos en clases.

## **V.- MODALIDAD DE EVALUACIÓN**

### **A) Criterios de evaluación**

1. Para la evaluación de las competencias cognitivas se sugiere:
  - Preguntas de desarrollo
  - Asociación
  - Completación
  - Preguntas de respuesta breve (cerrada)
  - Opción única
  - Opción múltiple
  - Ordenamiento
  - Términos pareados
2. Para la evaluación de las competencias procedimentales se sugiere:
  - Observación sistemática
  - Pautas o listas de cotejo
  - Escalas de valoración o apreciación
  - Pruebas de realización de tareas prácticas
  - Escala de Likert
  - Diario de clases
  - Resolución de casos
  - Presentaciones orales
  - Confección de informes
3. Para la evaluación de las competencias actitudinales se sugiere:
  - Trabajo en equipo
  - Participación en clases

### **B) Ponderación de notas**

La nota de presentación a examen se calculará como sigue:

Solemne I	=	30%
Solemne II	=	35%
Promedio de Controles	=	15%
Promedio de Trabajos Prácticos	=	20%

Las evaluaciones anteriores Constituyen la nota de presentación a examen. El examen tendrá una ponderación de un 30 % de la nota final. Se eximirán del examen aquellos estudiantes que cumplan con los requisitos establecidos por la Universidad.

Nota presentación a examen	= 70%
Nota examen	= 30%

### **C) Dimensión de evaluación**

En relación al estándar 12 del proceso de evidencia de la Middle States Commission on Higher Education, esta asignatura se evalúa según la dimensión: MANEJO DE RECURSOS DE LA INFORMACIÓN Y TECNOLOGÍA

## **VI.- BIBLIOGRAFÍA**

### **A) Bibliografía Básica**

- Java – Como Programar. Séptima Edición. Deitel y Deitel ©2008 Pearson Educación.
- UML Para Programadores Java. Primera Edición. Rober C. Martin ©2003 Pearson Prentice Hall.
- UML y Patrones. Segunda Edición. Craig Larman ©2003 Prentice Hall.

### **B) Bibliografía Complementaria**

- Aprendiendo UML en 24 horas. Joseph Schmuller ©2003 Prentice Hall.

### **C) Recursos de Internet**

- <http://www.uml.org/>
- <http://www.ingenierosoftware.com/analisisydiseno/uml.php>