

PROGRAMA DE ASIGNATURA													
<b>I.- DATOS GENERALES</b>													
Nombre de la Carrera o Programa: <b>Ingeniería Informática</b>													
Nombre de la Asignatura: <b>Algoritmos y Programación III</b>													
Departamento y/o cátedra: <b>Programación</b>													
Régimen: <b>Semestral</b>					Número de Unidades Crédito: <b>8</b>								
Ubicación en el plan de estudios: <b>Cuarto semestre</b>													
Tipo de asignatura:		Obligatoria	<input checked="" type="checkbox"/>	Electiva	<input type="checkbox"/>	N° horas semanales :		Teóricas	4	Prácticas/ Seminarios	0	Laboratorio	2
Prelaciones/Requisitos: <b>Algoritmos y Programación II</b>					Asignaturas a las que aporta: <b>Ingeniería de Software</b>								
Fecha de aprobación del Programa en el Consejo de Facultad: <b>Octubre 2016</b>													

II.- JUSTIFICACION
<p>La unidad curricular <b>Algoritmos y Programación III</b> contribuye a desarrollar un nivel más alto de abstracción, a través del conocimiento y práctica del paradigma de Programación Orientada a Objetos (POO) en todo lo relativo a la especificación, diseño, desarrollo y documentación de un sistema de información, empleando como técnica gráfica de documentación el Lenguaje Unificado de Modelado (UML). En las unidades curriculares anteriores relativas a Algoritmos y Programación, el estudiante desarrolla soluciones utilizando estructuras estáticas y dinámicas de información bajo el enfoque de programación estructurada. En Algoritmos y Programación III, el alumno se enfrenta por primera vez con el enfoque de programación orientada a objetos, una forma distinta de afrontar los problemas, que marca un antes y un después en su pensamiento algorítmico con un nivel de abstracción mayor. Con el enfoque de POO, el alumno adquiere la base para la habilidad de resolución de problemas similares a los que enfrentará como Ingeniero en Informática en un entorno profesional.</p>

III.- CONTRIBUCION DE LA ASIGNATURA AL DESARROLLO DE LAS COMPETENCIAS	
<b>Competencia General 1 (CG1):</b> Aprender a aprender con calidad	
<b>Unidad de Competencia 1 (CG1 – U1):</b> Abstrae, analiza y sintetiza información.	<b>Criterios de desempeño de la U1:</b> <ol style="list-style-type: none"> <li>1. Identifica elementos comunes en diferentes situaciones o contextos.</li> <li>2. Resume información de forma clara y ordenada.</li> <li>3. Integra los elementos de forma coherente.</li> </ol>

<b>Unidad de Competencia 2 (CG1 – U2):</b> Identifica, plantea y resuelve problemas.	<b>Criterios de desempeño de la U2:</b> 1. Analiza el problema y obtiene la información requerida para solucionarlo. 2. Formula opciones de solución que responden a su conocimiento, reflexión y experiencia previa. 3. Selecciona la opción de solución que resulta más pertinente, programa las acciones y las ejecuta.
<b>Unidad de Competencia 3 (CG1 – U3):</b> Trabaja con altos estándares de calidad.	<b>Criterios de desempeño de la U3:</b> 1. Actúa conforme a las normas y exigencias que denotan la calidad de su actuación.
<b>Competencia General 2 (CG2):</b> Aprender a trabajar con el otro	
<b>Unidad de Competencia 1 (CG2 – U1):</b> Participa y trabaja en equipo.	<b>Criterios de desempeño de la U1:</b> 1. Realiza las tareas establecidas por el equipo. 2. Cumple diversos roles dentro del equipo.
<b>Competencia Profesional Específica 1 (CPE1):</b> Gestiona Proyectos Informáticos	
<b>Unidad de Competencia 1 (CPE1 – U1):</b> Planifica y controla el desarrollo de un proyecto de tecnología de información	<b>Criterios de desempeño de la U1:</b> 1. Planifica el proyecto 2. Ajusta la planificación cuando sea necesario
<b>Unidad de Competencia 2 (CPE1 – U2):</b> Documenta técnica y funcionalmente un sistema informático.	<b>Criterios de desempeño de la U2:</b> 1. Elabora el manual de referencia técnica de un sistema informático
<b>Competencia Profesional Específica 2 (CPE2):</b> Desarrolla Software de aplicación	
<b>Unidad de Competencia 1 (CPE2 – U1):</b> Diseña e implementa algoritmos robustos y eficientes de forma modular.	<b>Criterios de desempeño de la U1:</b> 1. Descompone un problema en subproblemas más simples para resolverlo en forma modular. 2. Implementa la solución algorítmica. 3. Identifica y resuelve las situaciones excepcionales de un programa para lograr una solución robusta.
<b>Unidad de Competencia 1 (CPE2 – U2):</b> Diseña y ejecuta planes de pruebas de los sistemas informáticos.	<b>Criterios de desempeño de la U2:</b> 1. Define el plan de pruebas funcionales y técnicas 2. Aplica el plan de pruebas 3. Recomienda y/o aplica las mejoras identificadas en la ejecución del plan de pruebas.

IV.- UNIDADES TEMÁTICAS	
UNIDADES	TEMAS
1. Grafos	<p>1.1. Recapitulando sobre estructuras de datos y su importancia. Origen de los Grafos.</p> <p>1.2. Conceptos básicos. Caminos, cadenas, circuitos, ciclos. Representación de grafos.</p> <p>1.3. Alcance y conectividad. Clausura transitiva. Aplicaciones: Redes sociales y Caminos cortos</p> <p>1.4. Recorrido en Grafos. Aplicaciones</p> <p>1.5. Coloración en Grafos. Aplicaciones.</p>

<p><b>2.</b> Lenguaje Unificado de Modelado (UML)</p>	<p><b>3.1.</b> Orígenes, principios y notaciones de UML.  <b>3.2.</b> Diagrama de Casos de Uso.  <b>3.3.</b> Modelo de dominio y relaciones: asociación, agregación y generalización/especialización.  <b>3.4.</b> Diagrama de Clases.  <b>3.5.</b> Diagrama de Secuencia y Comunicación.  <b>3.6.</b> Aplicaciones y análisis de casos.</p>
<p><b>3.</b> Programación orientada a objetos (POO)</p>	<p><b>4.1.</b> Evolución de los lenguajes desde la programación funcional hasta la programación orientada a objetos.  <b>4.2.</b> Introducción a la Ingeniería de Software.  <b>4.3.</b> Fundamentos de la programación orientada a objetos: abstracción, encapsulamiento, ocultamiento de información, polimorfismo, herencia.  <b>4.4.</b> Definición de Clases y Objetos, Atributos, Métodos, pase de mensajes. Aplicaciones y ejemplos.  <b>4.5.</b> Principio de alta cohesión-bajo acoplamiento.</p>
<p><b>4.</b> Fundamentos del Lenguaje Java bajo el paradigma orientado a objetos</p>	<p><b>5.1.</b> Máquina Virtual de Java (JVM).  <b>5.2.</b> Entorno de ejecución Java (IDE).  <b>5.3.</b> Tareas de la JVM.  <b>5.4.</b> Cargador de clases.  <b>5.5.</b> Verificador de códigos de byte.  <b>5.6.</b> Convenciones de codificación.  <b>5.7.</b> Declaración de clases.  <b>5.8.</b> Declaración de atributos.  <b>5.9.</b> Declaración de métodos.  <b>5.10.</b> Acceso a miembros de un objeto.  <b>5.11.</b> Instrucción package.  <b>5.12.</b> Instrucción import.  <b>5.13.</b> Disposición de los directorios y los paquetes.  <b>5.14.</b> API de Java.  <b>5.15.</b> Tipos de referencia.  <b>5.16.</b> Construcción e inicialización de objetos.  <b>5.17.</b> Expresiones y control de flujo.  <b>5.18.</b> Instrucción for each.  <b>5.19.</b> Arrays.</p>
<p><b>5.</b> Conceptos Avanzados de Java bajo el paradigma orientado a objetos</p>	<p><b>6.1.</b> Constructores (declaración e implementación).  <b>6.2.</b> Referencia this.  <b>6.3.</b> Sobrecarga. Herencia. Sobreescritura. Polimorfismo.  <b>6.4.</b> Clases Abstractas e interfaces.  <b>6.5.</b> Componentes Swing.  <b>6.6.</b> Gestión de Eventos.  <b>6.7.</b> Cast  <b>6.8.</b> Manejo de Excepciones.  <b>6.9.</b> Colecciones (Set, List, Map y subcategorías derivadas).  <b>6.10.</b> Principios básicos de E/S. Flujo estándar.  <b>6.11.</b> Archivos. Serialización. Archivos XML.</p>

## V.- ESTRATEGIAS DE ENSEÑANZA Y DE APRENDIZAJE

Clases magistrales (exposiciones), preguntas generadoras y preguntas guías, resolución de ejercicios y problemas, participación en clase, talleres, enseñanza en pirámide, discusiones y método de proyectos.

## VI.- ESTRATEGIAS DE EVALUACIÓN

Técnica de la pregunta reflexiva, pruebas escritas, resolución de ejercicios y problemas (en aula y laboratorio), análisis de casos y proyectos por fases.

## VII.- REFERENCIAS BIBLIOGRÁFICAS

### Textos:

- ✓ Deitel & Deitel. **“Como programar en java”** 9na edición. Editorial Pearson. Año 2014
- ✓ Sintés, Anthony. **“Aprendiendo programación orientada a objetos en 21 lecciones avanzadas”**. Editorial Pearson.
- ✓ Arnold, Ken y Gosling James. **“El lenguaje de programación Java”**, 3e. Editorial Pearson. Año 2001.
- ✓ Joyanes, Luis y Zahonero, Ignacio. **“Estructuras de datos en Java”**. Editorial McGraw Hill. Año 2008.
- ✓ Eckel, Bruce. **“Piensa en Java”**. Editorial Pearson. Año 2007.
- ✓ Lewis, John y Chase, Joseph. **“Estructuras de datos con Java. Diseño de estructuras y algoritmos”**, Segunda edición. Editorial Pearson. Año 2006

### Página web:

### Guías y material de apoyo:

1. Envíos durante el semestre de material actualizado.
2. Guías y material de apoyo publicadas en la plataforma virtual CANVAS Módulo 7 (<https://m7.ucab.edu.ve/login>).