



**PLAN DE ESTUDIOS (PE): Licenciatura en Ciencias de la Computación**

**AREA: Ciencias de la Computación**

**ASIGNATURA: Programación Distribuida**

**CÓDIGO: CCOS 256**

**CRÉDITOS: 6**

**FECHA: 18 – Mayo – 2017**





### 1. DATOS GENERALES

<b>Nivel Educativo:</b>	Licenciatura
<b>Nombre del Plan de Estudios:</b>	Licenciatura en Ciencias de la Computación
<b>Modalidad Académica:</b>	Presencial
<b>Nombre de la Asignatura:</b>	Programación Distribuida
<b>Ubicación:</b>	Nivel formativo
<b>Correlación:</b>	
<b>Asignaturas Precedentes:</b>	Programación Concurrente y Paralela
<b>Asignaturas Consecuentes:</b>	NA

### 2. CARGA HORARIA DEL ESTUDIANTE

Concepto	Horas por semana		Total de horas por periodo	Total de créditos por periodo
	Teoría	Práctica		
<b>Horas teoría y práctica (16 horas = 1 crédito)</b>	<b>3</b>	<b>2</b>	<b>90</b>	<b>6</b>

### 3. REVISIONES Y ACTUALIZACIONES

<b>Autores:</b>	Darnes Vilariño Ayala Enrique Colmenares Guillen Leticia Mendoza Alonso Mario Anzures García Maya Carrillo Ruiz Mireya Tovar Vidal Rafael de la Rosa Flores Yalu Galicia Hernández
<b>Fecha de diseño:</b>	1 de Junio de 2009
<b>Fecha de la última actualización:</b>	18 de mayo de 2017
<b>Fecha de aprobación por parte de la academia de área, departamento u otro.</b>	24 de mayo de 2017





Revisores:	Ana Patricia Cervantes Márquez Beatriz Beltrán Martínez Darnes Vilariño Ayala Hilda Castillo Zacatelco Leticia Mendoza Alonso María Luz Adolfina Sánchez Gálvez Mario Anzures García	Mario Rossainz López Meliza Contreras González Miguel Rodríguez Hernández Mireya Tovar Vidal Rafael De la Rosa Flores Yolanda Moyao Martínez
Sinopsis de la revisión y/o actualización:	Se actualizaron las unidades y se revisaron los contenidos de cada uno de ellos. Se actualizó la bibliografía. Se realizó una revisión de los objetivos y nombres de cada una de las unidades. Se realizó el cambio de formato y se pasó a competencias.	

#### 4. PERFIL DESEABLE DEL PROFESOR (A) PARA IMPARTIR LA ASIGNATURA:

Disciplina profesional:	Ciencias de la Computación
Nivel académico:	Maestría
Experiencia docente:	Mínima de 2 años
Experiencia profesional:	Mínima de 1 año

#### 5. PROPÓSITO:

El estudiante será capaz de aplicar los fundamentos teóricos de la programación distribuida en el desarrollo de aplicaciones centradas en ambientes distribuidos.

El alumno será capaz de:

- Identificar los aspectos fundamentales relacionados con la Programación Distribuida
- Conocer y aplicar tecnologías específicas basándose en middleware asíncrono.
- Conocer y aplicar tecnologías específicas basándose en middleware síncrono.

#### 6. COMPETENCIAS PROFESIONALES:

Modela y diseña soluciones computacionales con base en los fundamentos matemáticos, principios algorítmicos y teorías de la Ciencia de la Computación para resolver diversas problemáticas sociales y laborales.

Resuelve problemas complejos de computación mediante algoritmos y programas con la finalidad de eficientar cualquier sistema computacional.

En la materia se aplican fundamentos teóricos de la programación distribuida a fin de eficientar problemas que pueden ser complejos.





## 7. CONTENIDOS TEMÁTICOS

Unidad de Aprendizaje	Contenido Temático	Referencias
1. Introducción a los Sistemas Distribuidos.	1.1 Definición, ejemplo y aspectos de los Sistemas Distribuidos. 1.2 Fundamentos de Redes. 1.2.1 Características relevantes. 1.2.2 Protocolos de Red. 1.2.3 Modelos de sistemas distribuidos. 1.2.4 Arquitecturas de Software. 1.2.5 MiddleWare (MW). 1.2.5.1 Tareas del MW. 1.2.5.2 La estructura de la Plataforma MW. 1.2.5.3 Estandarización del MW. 1.2.5.4 Portabilidad e Interoperabilidad. 1.2.6 Interacción. 1.2.7 Fallas. 1.2.8 Seguridad.	Coulouris, G., Dollimore, J., Kindberg, T., Gordon B. (2011) Distributed Systems, Concepts and Design. USA: Addison-Wesley, Fifth Edition.  Anthony R. (2015). Systems Programming Designing and Developing Distributed Applications, Morgan Kaufmann.

Unidad de Aprendizaje	Contenido Temático	Referencias
2. Middleware Asíncronos.	2.1. Llamado a Procedimiento Remoto (RPC). 2.2. Invocación a Método Remoto (RMI). 2.3. Descomposición de Requisitos de Objetos (ORB). 2.3.1. Interfaz ORB. 2.3.2. Comunicación entre ORB's. 2.4. Caso de estudio: CORBA. 2.4.1. La perspectiva OMG. 2.4.2. Definición del Modelo de Componentes de CORBA. 2.4.3. El lenguaje IDL. 2.5. Caso estudio: COM/DCOM.	Coulouris, G., Dollimore, J., Kindberg, T., Gordon B. (2011) Distributed Systems, Concepts and Design. USA: Addison-Wesley, Fifth Edition.  Anthony R. (2015). Systems Programming Designing and Developing Distributed Applications, Morgan Kaufmann.





Unidad de Aprendizaje	Contenido Temático	Referencias
3. Middleware Síncronos.	3.1. Middleware síncronos.	Coulouris, G., Dollimore, J., Kindberg, T., Gordon B. (2011) Distributed Systems, Concepts and Design. USA: Addison-Wesley, Fifth Edition.  Anthony R. (2015). Systems Programming Designing and Developing Distributed Applications, Morgan Kaufmann.
	3.2. Modelo Orientado a Mensajes (MOM). 3.2.1. Sistema de Mensajes de Java (JMS).	
	3.3. Cómputo Orientado a Servicios (SOC). 3.3.1. Servicios Web. 3.3.1.1 SOA (Arquitectura Orientada a Servicios). 3.3.1.2. REST (Representación del Estado de Transferencia).	

## 8. ESTRATEGIAS, TÉCNICAS Y RECURSOS DIDÁCTICOS.

Estrategias y técnicas didácticas	Recursos didácticos
<p>Estrategias de aprendizaje:</p> <ul style="list-style-type: none"> <li>• Lectura y comprensión,</li> <li>• Reflexión,</li> <li>• Comparación,</li> <li>• Resumen.</li> </ul> <p>Estrategias de enseñanza:</p> <ul style="list-style-type: none"> <li>• ABP,</li> <li>• Aprendizaje activo,</li> <li>• Aprendizaje cooperativo,</li> <li>• Aprendizaje colaborativo,</li> <li>• Basado en el descubrimiento.</li> </ul> <p>Ambientes de aprendizaje:</p> <ul style="list-style-type: none"> <li>• Aula,</li> <li>• Laboratorio,</li> <li>• Simuladores.</li> </ul> <p>Actividades y experiencias de aprendizaje:</p> <ul style="list-style-type: none"> <li>• Visita a empresas.</li> </ul>	<p>Materiales:</p> <ul style="list-style-type: none"> <li>• Proyector,</li> <li>• TICs,</li> <li>• Plumón y pizarrón.</li> </ul>

## 9. EJES TRANSVERSALES

Eje (s) transversales	Contribución con la asignatura
Formación Humana y Social	Permite llevar a cabo la programación distribuida con una actitud ética y profesional para





	coadyuvar al bienestar social.
Desarrollo de Habilidades en el uso de las Tecnologías de la Información y la Comunicación	Establece los conceptos básicos para comprender la programación distribuida.
Desarrollo de Habilidades del Pensamiento Complejo	Ayuda a conceptualizar la programación distribuida desde una perspectiva crítica y fundamentada en análisis del problema.
Lengua Extranjera	Facilita la comprensión de la programación distribuida, ya que los textos referentes a la misma generalmente se encuentran escritos en inglés.
Innovación y Talento Universitario	Establece los elementos fundamentales para el desarrollo de proyectos innovadores, que tengan como punto central la programación distribuida.

## 10. CRITERIOS DE EVALUACIÓN

Criterios	Porcentaje
• Tareas	10 %
• Prácticas	15 %
• Proyecto final	20 %
• Pruebas objetivas	30 %
• Participación en clase	10 %
• Asistencia	5 %
• Entregas puntuales	5 %
• Presentación de trabajos	5 %
<b>Total</b>	<b>100%</b>

## 11. REQUISITOS DE ACREDITACIÓN

Estar inscrito como alumno en la Unidad Académica en la BUAP
Asistir como mínimo al 80% de las sesiones para tener derecho a exentar por evaluación continua y/o presentar el examen final en ordinario o extraordinario
La calificación mínima para considerar un curso acreditado será de 6
Cumplir con las actividades académicas y cargas de estudio asignadas que señale el PE

### Notas:

- La entrega del programa de asignatura con sus respectivas actas de aprobación, deberá realizarse en formato electrónico, vía oficio emitido por la Dirección o Secretaría Académica a la Dirección General de Educación Superior.
- La planeación didáctica deberá ser entregada a la coordinación de la licenciatura en los tiempos y formas acordados por la Unidad Académica.