

**Facultad de Ingeniería y Ciencias Agropecuarias**  
**Carrera Ingeniería en Sistemas de Computación e Informática**  
**IER710 – Aplicaciones Distribuidos**  
 Período 2017-2

**1. Identificación.-**

Número de sesiones: 48  
 Número total de hora de aprendizaje: 120  
 Créditos – malla actual: 3  
 Profesor: Jhon Benalcázar  
 Correo electrónico del docente (Udlanet): jr.benalcazar@udlanet.ec  
 Director: Marco Antonio Galarza Castillo  
 Campus: Queri  
 Pre-requisito: Co-requisito:  
 Paralelo:  
 Tipo de asignatura:

Optativa	
Obligatoria	X
Práctica	

Organización curricular:

Unidad 1: Formación Básica	
Unidad 2: Formación Profesional	X
Unidad 3: Titulación	

Campo de formación:

Campo				
Fundamentos teóricos	Praxis profesional	Epistemología y metodología de la investigación	Integración de saberes, contextos y cultura	Comunicación y lenguajes
	X			

**2. Descripción del curso.-**

El presente curso es de naturaleza teórico práctica, contempla el, análisis, diseño e implementación de un Sistema Distribuido, para lo cual se presenta al estudiante en las clases teóricas, conceptos de: Sistemas Operativos, Arquitectura de Computadoras y Redes, ayudándolos a manejar los conceptos como un todo, junto con el desarrollo de laboratorios que fortalecerán su entendimiento

**3. Objetivo del curso.-**

Impartir conocimientos técnicos acerca del diseño de sistemas distribuidos y sus características para lograr implementar un sistema distribuido sobre la base de su utilidad, funcionalidad y disponibilidad, desarrollando en el estudiante el pensamiento crítico, motivado por los problemas que resuelven estos sistemas y su popularidad actual

**4. Resultados de aprendizaje deseados al finalizar el curso:**

Resultados de aprendizaje (RdA)	RdA perfil de egreso de carrera	Nivel de desarrollo (carrera)
<b>1.</b> Compara conceptos, características y servicios de comunicación de sistemas distribuidos para su aplicación. <b>2.</b> Aplica conceptos de sistemas distribuidos en la implementación de un sistema que cumple los requerimientos de diseño de este tipo de sistemas	<b>4.</b> Gestiona tecnologías de computadoras, arquitecturas de software y tecnologías de redes de información.	<b>Inicial</b> ( ) <b>Medio</b> ( ) <b>Final</b> ( X )

## 5. Sistema de evaluación.-

De acuerdo al Modelo Educativo de la UDLA la evaluación busca evidenciar el logro de los resultados de aprendizaje (RdA) enunciados en cada carrera y asignatura, a través de mecanismos de evaluación (MdE). Por lo tanto la evaluación debe ser continua, formativa y sumativa. La UDLA estipula la siguiente distribución porcentual para los reportes de evaluaciones previstas en cada semestre de acuerdo al calendario académico:

### **Reporte de progreso 1** **35%**

a.- Creación de Tutoriales:

**Puntaje: 5% / 35%**

El estudiante deberá crear tutoriales evidenciando el entendimiento del desarrollo de las clases y de información adicional consultada en las fuentes bibliográficas los mismos que deberán ser publicados en el aula virtual y en redes sociales para su acceso.

b.- Informes, Consultas:

**Puntaje: 10% / 35%.**

Estos informes evidenciarán los resultados, de las investigaciones enviadas a ser desarrolladas por el estudiante.

c.- Examen Escrito:

**Puntaje: 20% / 35%**

El estudiante debe rendir un examen totalmente escrito acumulativo, mismo que consta de dos partes: 1) Conceptos de la materia 2) Solución de problemas aplicando conocimientos de clase

### **Reporte de progreso 2** **35%**

a.- Creación de Tutoriales:

**Puntaje: 5% / 35%**

El estudiante deberá crear tutoriales evidenciando el entendimiento del desarrollo de las clases y de información adicional consultada en las fuentes bibliográficas los mismos que deberán ser publicados en el aula virtual y en redes sociales para su acceso.

b.- Informes, Consultas:

**Puntaje: 10% / 35%.**

Estos informes evidenciarán los resultados, de las investigaciones enviadas a ser desarrolladas por el estudiante.

c.- Examen Escrito:

**Puntaje: 20% / 35%**

El estudiante debe rendir un examen totalmente escrito acumulativo (desde el primer día de clases), mismo que consta de dos partes: 1) Conceptos de la materia 2) Solución de problemas aplicando conocimientos de clase

**Evaluación final**

**30%**

a.- Trabajo Escrito:

**Puntaje: 10% / 30%**

El estudiante deberá formar grupos de trabajo y presentar un trabajo escrito del sistema distribuido a implementar, para lo cual deberá realizar el análisis, diseño de su sistema, haciendo una investigación empírica (sin implementar) de todos los conceptos relacionados con el tema seleccionado. Este trabajo deberá ser subido al aula virtual.

De este trabajo el grupo deberá realizar una presentación al curso y responder correctamente a las preguntas del docente.

Adjunto descripción y rúbrica del proyecto final del documento.

b.- Trabajo Práctico:

**Puntaje: 20% / 30%.**

El estudiante deberá formar grupos de trabajo e implementar y probar un sistema distribuido, para lo cual deberá realizar, para esta implementación deberá usar la información levantada en el trabajo escrito.

El grupo deberá demostrar que su implementación cumple los requerimientos de diseño de los sistemas distribuidos realizando una presentación al curso y responder correctamente a las preguntas del docente.

Adjunto descripción y rúbrica del proyecto final del documento.

Al finalizar el curso habrá un examen de recuperación para los estudiantes que, habiendo cumplido con más del 80% de asistencia presencial a clases, deseen reemplazar la nota de un examen anterior (ningún otro tipo de evaluación). Este examen debe integrar todos los conocimientos estudiados durante el periodo académico, por lo que será de alta exigencia y el estudiante necesitará prepararse con rigurosidad. La nota de este examen reemplazará a la del examen que sustituye. Recordar que para rendir el EXAMEN DE RECUPERACIÓN, es requisito que el estudiante haya asistido por lo menos al 80% del total de las sesiones programadas de la materia. No se podrá sustituir la nota de un examen previo en el que el estudiante haya sido sancionado por una falta grave, como copia o deshonestidad académica.

**6. Metodología del curso y de mecanismos de evaluación.**

La metodología que se implementará en el desarrollo de este curso se centra en el estudiante, mediante la constante lectura, laboratorios dirigidos y tareas autónomas que vincula la teoría y la práctica. Las metodologías y mecanismos de evaluación que se usarán se basan en los siguientes escenarios de aprendizaje:

1. Escenario de aprendizaje presencial.

Las metodologías que se aplicarán para el desarrollo del curso son:

- a) **Instrucción Directa:** A través de diapositivas y utilizando ambientes virtuales el docente contrasta los temas teóricos dictados en clase con escenarios prácticos.
- b) **Prácticas en Laboratorio:** Utilizando ambientes virtuales el docente propone casos que va desarrollando en clase para que el alumno contraste la teoría y escenarios prácticos.

2. Escenario de aprendizaje autónomo.

Las metodologías que se aplicarán para el desarrollo del curso son:

- a) **Lectura Previa:** Para cada unidad el docente facilitará fuentes de información que el estudiante deberá leer previa la clase. En clase los estudiantes desarrollarán cuestionarios en el aula virtual.
- b) **Tareas Individuales – Grupales:** El docente, propone a los alumnos un portafolio de actividades sobre ambientes virtuales que deben ser desarrollados en casa.
- c) **Investigación:** El estudiante seleccionara un sistema distribuido para investigar y deberá mediante búsqueda de información, análisis de material bibliográfico realizar el análisis del sistema, su diseño, implementación y pruebas, presentar su trabajo escrito y práctico y defenderlo en el aula.

7. Temas y subtemas del curso.-

RDA	Tema	Subtemas
Compara conceptos, características y servicios de comunicación de sistemas distribuidos para su aplicación.	Fundamentos de Sistemas Operativos	1.1 Introducción
		1.2 Sistemas
		1.3 Sistemas Operativos (Kernel)
		1.4 Microkernel, Mononucleo
	Fundamentos de Arquitectura de Computadores	2.1 Introducción
		2.2 Von Newmann
		2.3 Comunicación de procesos
	Fundamentos de redes	3.1 Modelo OSI
		3.2 Nodos de red
	Fundamentos de Sistemas Distribuidos	4.1 Introducción
	Análisis de Sistemas Distribuidos	5.1 Sistemas Centralizados vs Distribuidos
		5.2 Comunicación de procesos sistemas Distribuidos
		5.3 Transparencia
	Diseño de Sistemas Distribuidos	6.1 Requerimientos de Diseño
	Mensajes	7.1 Modelo Cliente / Servidor
		7.2 Primitivas de comunicación
	Sockets	8.1 Fundamentos
		8.2 Sockets, Puertos, IP
	RPC	9.1 Fundamentos
		9.2 Marshalling/Demarshalling
		9.3 PortMapper
	RMI	10.1 Fundamentos
		10.2 Java RMI
Aplica conceptos de sistemas distribuidos	Análisis, Diseño	11.1 Selección, análisis y documentación
		11.2 Presentación y Defensa del documento

en la implementación de un sistema que cumple los requerimientos de diseño de este tipo de sistemas	Implementación	12.1	Implementación, pruebas
		12.2	Presentación y Defensa del sistema implementado

8. Planificación secuencial del curso.-

# RDA	Tema	Subtemas		Actividad/ metodologí a/clase	Tarea/ trabajo autónomo	MdE/Producto/fecha de entrega			
Semana 1 y 2 ( 6 – 13 de Marzo)									
1	1.Fundament os de Sistemas Operativos	1.1	Introducción	(6.1) Instrucción Directa (6.2)Lectura s Previas, trabajos Individuales - grupales	(6.2) Sistema, Sistema Informático, partes. Datos, Información, Kernel, Sistema Operativo.  (6.2) Arquitectura de Computadores, flujo de información, Von NewMann  (6.2) (6.1) Comparación Sistemas: Paralelos, Centralizados, Concurrentes, Distribuidos  (6.2) (6.1)Dinámica Grupal MODELO OSI	Creación de Tutoriales Puntaje: 5% / 35% Fecha: 18 de Abril 2017  Informes Puntaje: 10%/35% Fecha: 14 Marzo 2017			
		1.2	Sistemas						
		1.3	Sistemas Operativos (Kernel)						
		1.4	MicroKernel, Mononucleo						
	2. Fundamento s de Arquitectura de Computador es	2.1	Introducción						
		2.2	Von NewMann						
3. Fundamento s de redes	3.1	Modelo OSI							
Semana 3 a 6 ( 20 de Marzo al 14 de Abril)									
1	4. Fundamento s de Sistemas Distribuidos	4.1	Diagramas de solución de sistemas distribuidos	(6.1) Instrucción Directa (6.2)Lectura s Previas, trabajos Individuales - grupales	(6.2) Nodos de Red (6.2) Diagrama de solución de Florícola (6.2) Comparación entre sistemas centralizados y distribuidos (6.2) Validación de dos sistemas como centralizado o distribuido	Creación de Tutoriales Puntaje: 5% / 35% Fecha: 18 de Abril 2017  Informes Puntaje: 10%/35% Fecha: 20 – 24 Mar 2017 27 – 31 Mar 2017 03 – 07 Abr 2017 10 – 14 Abr 2017			
		4.1	Introducción						
	5. Análisis de Sistemas Distribuidos	5.1	Sistemas Centralizados vs Distribuidos						
		5.2	Comunicación de procesos en sistemas Distribuidos						
		5.3	Transparencia						
	6. Diseño de Sistemas Distribuidos	6.1	Requerimientos de Diseño						
	Semana 7 Examen Progreso 1 ( 17 al 21 de Abril 2017)								
Semana 8 ( 24 al 28 de Abril)									
2	7. Análisis, Diseño sistema distribuido	7.1	Análisis, Diseño sistema distribuido	(6.2)Investi gación, trabajos grupales	(6.2) Documento de análisis, diseño e implementación de Sistema Distribuido (6.2) Presentación grupal y defensa del documento	Evaluación final – Trabajo escrito Puntaje: 10% / 30%			

**Semana 9 a 12 ( 1 de Mayo al 26 de Mayo)**

1	8. Mensajes	8.1	Modelo Cliente /servidor	(6.1) Instrucción Directa (6.2)Lectura s Previas, trabajos Individuales - grupales	(6.2)Investigación modelo Cliente –Servidor (6.1) Laboratorio	Creación de Tutoriales Puntaje: 5% / 35% Fecha: 29 May – 01 Jun 2017
		8.2	Primitivas de comunicación			
	9. Sockets	9.1	Fundamentos		(6.2)Investigación Sockets (6.1) Laboratorio	
		9.2	Sockets, Puertos, IP			
	10. RPC	10.1	Fundamentos		(6.2)Investigación RPC (6.1) Laboratorio	
		10.2	Marshalling/Demarshallin g			
		10.3	PortMapper			
	11. RMI	11.1	Fundamentos		(6.2)Investigación RMI (6.1) Laboratorio	
		11.2	Java RMI			

**Semana 13 Examen Progreso 2 ( 29 de Mayo al 2 de Junio 2017)**

**Semana 14 (5 al 9 de Junio del 2017)**

1	12. Transacciones Distribuidas	12.1	Fundamentos	(6.1) Instrucción Directa (6.2) Lecturas Previas, trabajos Individuales - grupales	(6.2) Dinámica grupal	
		12.2	Clasificación			
		12.3	Control de Concurrencia y Serialización			

**Semana 15, 16 ( 12 al 23 de Junio 2017 )**

2	13. Implementación y pruebas de sistema distribuido	13.1	Implementación y pruebas de sistema distribuido	(6.2) Investigación, trabajos grupales	(6.2) Implementación de Sistema Distribuido (6.2) Presentación grupal y defensa del proyecto final	Evaluación final – Trabajo Práctico Puntaje: 20% / 30%
---	---	------	---	--	---	---

**Semana 17 Semana de Recuperación de clases (26 al 30 de Junio 2017)**

**Semana 18 Continuación de presentación de proyectos finales ( 3 al 7 de Julio)**

**Semana 19 Examen de recuperación (10 al 14 de Julio)**

**6 Normas y procedimientos para el aula**

Sistemas Distribuidos, es una materia que compila como resultado los conocimientos adquiridos por los estudiantes en Sistemas Operativos, Arquitectura de Computadores y Redes, desarrollando en el estudiante competencias para generar soluciones distribuidas transparentes para el usuario y que cumplan requerimientos de escalabilidad, tolerancia a fallos, heterogeneidad, abiertos, controlando el acceso a recursos compartidos de manera consistente y confiable;

La participación del curso requiere de sus integrantes: responsabilidad, puntualidad, disciplina, respeto y ética, es por ello que se exige a los estudiantes la práctica diaria de estos valores y principios en todas las actividades relacionadas a la materia.

- Cualquier caso que esté en contra de la honestidad académica será reportado a las autoridades de la universidad.
- No se recibirán trabajos fuera de los plazos establecidos
- Se esperara 10 minutos después del inicio de clases para comenzar la misma, posterior a este tiempo el estudiante deberá asumir que tiene falta en la hora
- Es responsabilidad del estudiante el control de su asistencia
- Se exige al estudiante la presentación de trabajos escritos adecuados y acordes a los de un ingeniero, por lo cual se exigirá un alto nivel técnico de presentación
- No se admitirá por ningún motivo la copia de ejercicios, exámenes, proyectos, y todas las actividades de aprendizaje solicitadas por el docente; en el caso que se detectara cualquier caso que esté en contra de la honestidad académica se calificará con la mínima calificación (cero).
- No se recibirán trabajos fuera de la plataforma virtual.
- En el caso de inasistencia a una clase, es responsabilidad del estudiante igualarse en la implementación de los scripts que se realicen en el proyecto de clase.
- Se restringe durante el desarrollo de clases el uso de Internet, celulares, redes sociales y audífonos
- No se podrán ingresar alimentos al aula

## 7 Referencias bibliográficas.-

### Principal:

- Sukumar Gosh (2015), *Distributed Systems: An Algorithmic Approach*, CRC Press [ISBN 978-1-4665-5298-2](#)
- Anthony, Richard (2016), *Systems Programming: Designing and Developing Distributed Applications*, Elsevier [ISBN 978-0-12-800729-7](#)
- Tanenbaum, Andrew S (2014), *Distributed Systems: Principles and Paradigms (2th Edition Rev)*, Pearson Education [ISBN 13: 9781292025520](#)

### Secundarias:

- [Wan Fokkink](#) (2013), *Distributed Algorithms: An Intuitive Approach*, The MIT Press **ISBN-10:** 0262026775
- Coulouris, George et al (2011), *Distributed Systems: Concepts and Design (5th Edition)*, Addison-Wesley [ISBN 0-132-14301-1](#).

### Perfil del docente

Jhon Benalcázar

Ingeniero en Sistemas de la Politécnica Nacional, con un MBA, que permite tener un enfoque técnico administrativo de los diferentes problemas de ingeniería. Experiencia de 15 años en el mercado con varias certificaciones en productos de Hardware (Storage, Servidores) y Software (Windows, HP-UX, Sistemas Operativos de PC). Experiencia en la implementación de proyectos con la última tecnología del mercado, además de mantener altos niveles de disponibilidad de los servicios y gestión de



recursos humanos para cumplimiento de objetivos, 9 años de experiencia como docente en la UDLA, en varias materias: Certificación de Sistemas Operativos, certificación de Bases de Datos, Evaluación de Sistemas, Inteligencia Artificial, Sistemas Operativos, Arquitectura de computadores, Sistemas Distribuidos y Aplicaciones distribuidas.

Contacto: [Jhon.benalcazar@udla.edu.ec](mailto:Jhon.benalcazar@udla.edu.ec)

Twitter: @jbenalca

**PROYECTO FINAL**  
**ACI850 - SISTEMAS DISTRIBUIDOS**  
DOCENTE: JHON BENALCAZAR  
PERIODO: 2017-2

## 1. BALANCEADOR DE CARGA LINUX

Análisis, Desarrollo e implementación de un balanceador de carga LINUX.

### **REQUERIMIENTOS:**

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario
- El servicio debe ser personalizado no se puede tomar las páginas de Default de los Web Servers
- Deber permitir identificar el nodo que está atendiendo el servicio
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos y externos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación

## 2. BALANCEADOR DE CARGA WINDOWS

Análisis, Desarrollo e implementación de un balanceador de carga WINDOWS.

### **REQUERIMIENTOS:**

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario
- El servicio debe ser personalizado no se puede tomar las páginas de Default de los Web Servers
- Deber permitir identificar el nodo que está atendiendo el servicio
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos y externos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación

## 3. FILE SYSTEM DISTRIBUIDO

Análisis, Desarrollo e implementación de un FILE SYSTEM DISTRIBUIDO.

### **REQUERIMIENTOS:**

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario

- El servicio debe ser totalmente integrado con el usuario
- Debe tener al menos DOS nodos de replicación y un primario
- SI se usa WINDOWS, NO se debe usar el SYSTEM VOL del Active Directory para replicación
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación

## 4. CLUSTER DE IMPRESION

Análisis, Desarrollo e implementación de un CLUSTER DE IMPRESIÓN

### REQUERIMIENTOS:

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario
- El servicio debe ser totalmente integrado con el usuario
- Debe tener al menos DOS nodos de impresión
- Se requiere una impresora con tarjeta de red
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos y externos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación

## 5. CLUSTER ACTIVO PASIVO LINUX

Análisis, Desarrollo e implementación de un CLUSTER ACTIVO PASIVO LINUX

### REQUERIMIENTOS:

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario
- El servicio debe ser totalmente integrado con el usuario
- El servicio a clusterizar no puede ser un WEB SERVER
- El servicio de Alta disponibilidad lo debe proporcionar el Sistema Operativo
- Se requiere configurar un almacenamiento compartido, redes de servicio, Heartbeat
- El servicio debe estar integrado en el cluster
- Al presentar un incidente con el nodo activo, el nodo pasivo automáticamente debe tomar el control del servicio
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos y externos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación

## 6. CLUSTER ACTIVO PASIVO WINDOWS

Análisis, Desarrollo e implementación de un CLUSTER ACTIVO PASIVO WINDOWS

### REQUERIMIENTOS:

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario
- El servicio debe ser totalmente integrado con el usuario
- El servicio a clusterizar no puede ser un WEB SERVER
- El servicio de Alta disponibilidad lo debe proporcionar el Sistema Operativo
- Se requiere configurar un almacenamiento compartido, redes de servicio, Heartbeat
- El servicio debe estar integrado en el cluster
- Al presentar un incidente con el nodo activo, el nodo pasivo automáticamente debe tomar el control del servicio
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos y externos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación

## 7. RED P2P

Análisis, Desarrollo e implementación de un servicio PEER TO PEER

### REQUERIMIENTOS:

- Resolución de acceso por nombre
- Cumplir todos los requerimientos de los S.D.
- Ser transparente para el usuario
- El servicio debe ser totalmente integrado con el usuario
- Debe contar con al menos 4 clientes/servidores
- Al presentar un incidente con un nodo, cualquiera de los otros nodos automáticamente debe tomar el control del servicio
- Todos los nodos pueden ser seeder y leechers
- El servicio de descargar debe recuperarse desde el punto en que se perdió conexión con el nodo de descarga
- El servicio debe ser implementado tanto en el cliente como en el servidor
- Debe permitir descargas multiples y uploads multiples
- Realizar el análisis de factibilidad del S.D. planteándolo como reemplazo de un sistema centralizado
- Desarrollar la arquitectura de la solución con usuarios internos y externos
- Desarrollar la arquitectura de virtualización a utilizar.
- Desarrollar el marco teórico de todos los conceptos relacionados con el tema
- Desarrollar el manual de implementación
- Desarrollar el manual de Operación