

## PROGRAMA de Sistemas Distribuidos

**Carreras:** Tecnicatura en Programación Informática / Licenciatura en Informática

**Asignatura:** Sistemas Distribuidos

**Núcleo al que pertenece:** Avanzado

**Profesores:** Federico Repond y Esteban Dimitroff

**Asignaturas Correlativas:** Programación Concurrente – Laboratorio de Sistemas Operativos y Redes

### Objetivos:

Aprendizaje de conceptos básicos y algoritmos relacionados con las problemáticas típicas de los sistemas distribuidos modernos.

Experimentación de primera mano los desafíos de un sistema complejo como un sistema distribuido..

La modalidad del curso será la de taller y durante el recorrido del curso se estudiarán los diversos aspectos de los sistemas distribuidos a medida que resuelven los problemas que se van presentando y experimentan en el laboratorio. Una vez finalizado el trabajo se pide a quienes cursan que escriban un reporte para ser discutido en clase sobre los hallazgos, problemas que se presentaron y posibles extensiones.

Como actividad extra-áulica se deben completar los trabajos que se asignan en clase y el informe correspondiente para discusión de resultados en forma colectiva la siguiente clase.

### Contenidos mínimos:

- Introducción a los sistemas de procesamiento distribuido y su terminología.
- Comunicación en sistemas distribuidos, pasaje de mensaje y llamadas a procedimiento remoto (rpc)
- Tiempo, Sincronización y Coordinación Distribuida.
- Memoria compartida distribuida, asignación de tareas y balance de cargas (Algoritmos básicos)
- Manejo de archivos distribuidos

Carga horaria semanal: 4 hs

Programa analítico:

Unidad 1

Introducción a los sistemas distribuidos. Introducción a Erlang. Primeros pasos en Erlang.

Unidad 2

Modelo de cliente/servidor, estructura de un server, protocolo y sockets.

Unidad 3

Servicios de resolución de nombres.

Unidad 4

Ruteo, protocolo link-state.

Unidad 5

Detección de fallas en un sistema distribuido.

Unidad 6

Tiempo lógico en un sistema distribuido, noción de orden.

Unidad 7

Sistema de membresía, multicast atómico y estado coordinado.

Unidad 8

Exclusión mutua distribuida

Unidad 9

Control de concurrencia distribuido.

## Bibliografía obligatoria:

- George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair. *Distributed Systems: Concepts and Design*. (5th ed.). Addison-Wesley Publishing Company, USA. 2011
- Joe Armstrong. *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf. 2007

## Bibliografía de consulta:

- Sukumar Ghosh. *Distributed Systems: An Algorithmic Approach. Second Edition* (2nd ed.). Chapman & Hall/CRC. 2014
- Wan Fokkink. *Distributed Algorithms: An Intuitive Approach*. The MIT Press. 2013
- Fred Hebert. *Stuff go Bad: Erlang in Anger* – (<http://www.erlang-in-anger.com>)
- Fred Hebert. *Learn You Some Erlang for Great Good!* – (<http://learnyousomeerlang.com>)

## Organización de las clases:

### Trabajo práctico 1

El objetivo de esta práctica es familiarizarse con el lenguaje y las herramientas de desarrollo utilizadas con la materia. Construcciones básicas del lenguaje, procesos, pasaje de mensajes y un ejemplo simple de programación distribuida con un ejercicio de ping-pong.

### Trabajo Práctico 2

Rudy: Un pequeño servidor web. Se propone la construcción de un pequeño servidor web que permita entender la estructura de un sistema cliente servidor y el manejo de un protocolo de comunicación simple.

### Trabajo Práctico 3

Namy: Un name server distribuido. Se trabajan los conceptos de resolución de nombres y caching en estructura de árbol. Balanceo de cargas.

### Trabajo Práctico 4

Routy: Un pequeño protocolo de ruteo. Construcción de un protocolo de ruteo link-state, describir cómo se mantiene una vista consistente y reflejar los problemas relacionados a fallos

de red. Sistema de archivos distribuido.

### Trabajo Práctico 5

Detector: Un detector de fallas. Se introduce la detección de fallos en sistemas distribuidos, los sistemas de monitoreo y supervisión y posibles esquemas para la recuperación de fallos

### Trabajo Práctico 6

Loggy: Un time logger lógico. Sincronización de procesos. Tiempo lógico. Relojes de Lamport y Vector clocks.

### Trabajo Práctico 7

Groupy: Un servicio de membresía de grupo. Se trabaja la comunicación y coordinación de grupo, multicast atómico, esquema líder-subordinado, elección de Líder y consenso.

### Trabajo Práctico 8

Muty: Un servicio de exclusión mutua distribuido. Se vuelve sobre multicast, sincronización en red asincrónica y se introduce el manejo de deadlocks y el concepto de fairness en la asignación de recursos a los procesos.

### Trabajo Práctico 9

Opty: Un servidor de transacciones. Control de concurrencia optimista con validación backwards. Manejo de memoria compartida.

## Modalidad de evaluación:

Los mecanismos de evaluación en modalidades libre y presencial de esta asignatura están reglamentados según los siguientes artículos del Régimen de estudios de la UNQ (Res. CS 201/18)

En la modalidad de libre, se evaluarán los contenidos de la asignatura con un examen escrito, un examen oral e instancias de evaluación similares a las realizadas en la modalidad presencial.

## CRONOGRAMA TENTATIVO

Semana	Tema/unidad	Actividad*				Evaluación
		Teórico	Práctico			
			Res Prob.	Lab.	Otros Especificar	
1	Introducción a los sistemas distribuidos. Introducción a Erlang.	x				
2	Primeros pasos en Erlang.			x		
3	Modelo de cliente/servidor, estructura de un server, protocolo y sockets. Rudy: un pequeño web server.	x		x		
4	Entrega y discusión de resultados de Rudy.			x		x
5	Servicios de resolución de nombres. Namy: un name server distribuido.	x		x		
6	Entrega y discusión de resultados de Namy.			x		x
7	Ruteo, protocolo link-state. Routy: un pequeño protocolo de ruteo.	x		x		
8	Entrega y discusión de Routy.			x		x
9	Detección de fallas en un sistema	x		x		

	distribuido. Detector: un detector de fallas.					
10	Entrega y discusión de resultados de Detector.			x		x
11	Tiempo lógico en un sistema distribuido, noción de orden. Loggy: un time logger lógico.	x		x		
12	Entrega y discusión de Loggy.			x		x
13	Sistema de membresía, multicast atómico y estado coordinado. Groupy: un servicio de membresía de grupo.	x		x		
14	Entrega y discusión de Groupy.			x		x
15	Control de concurrency distribuido. Opty: control de concurrency optimista.	x		x		
16	Entrega y discusión de Opty.			x		x
17	Balance y cierre de curso.			x		

**\*indique con una cruz la modalidad**