

Universidad Nacional de Quilmes, Departamento de Ciencia y Tecnología
Licenciatura en Desarrollo de Software

Sistemas Distribuidos

Introducción

Federico C. Repond
frepond@unq.edu.ar

Esteban Dimitroff Hódi
esteban.dimitroff@unq.edu.ar

March 17, 2016



- ▶ **Modalidad:** el curso tendrá la modalidad de taller.
- ▶ **Clases:** Jueves de 17 a 21.
- ▶ **Carga:** 8 a 12hs semanales aprox.
- ▶ **Evaluación:** entrega de trabajos cada 15 días.
 - ▶ las fechas de entrega se irán regulando según desempeño grupal.
 - ▶ no pueden deber trabajos ni entregar después del deadline pactado.
- ▶ **Equipos:** máximo 2 integrantes.
- ▶ **Página:** https://gitlab.com/sistemas_distribuidos/curso



- ▶ **Código:** se debe entregar el código fuente en repositorio git.
- ▶ **Documentación:** el código debe estar acompañado de un README.md con el informe del trabajo a ser discutido e instrucciones.

El curso constará seguirá mayormente la bibliografía de la materia (Distributed Systems: Concepts and Design). Leer por adelantado!

1. Erlang – programación concurrente y distribuida en Erlang.
2. Introducción – qué es un sistema distribuido y por que es diferente. (Cap. 1 y 2)
3. Redes y comunicación entre procesos. (Cap. 3 y 4)
4. Invocación remota – construcciones del lenguaje para programar sistemas distribuidos. (Cap. 5)
5. Comunicación indirecta – group communication/publish & subscribe/message queues. (Cap. 6)
6. Sistemas de archivos y name services – problema de sistemas de archivos distribuidos, performance y consistencia. (Cap. 12 y 13)

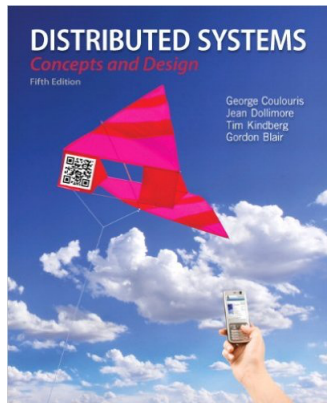


7. Time – algo simple que se vuelve complejo. (Cap. 14.1-4)
8. Estado global – podemos describir el estado de un sistema distribuido? (Cap. 14.5)
9. Coordinación y consenso. cómo consensuamos y cómo sabemos que estamos de acuerdo? (Cap. 15)
10. Transacciones – cómo hacemos que un conjunto de operaciones se comporten como una operación atómica? (Cap. 16)
11. Transacciones distribuidas – cómo lo hacemos ahora en presencia de múltiples servidores?. (Cap. 17)
12. Replicación – construyendo sistemas tolerantes a fallas. (Cap. 18)
13. Hashtables distribuidas – por qué hashear? (Cap. 10)

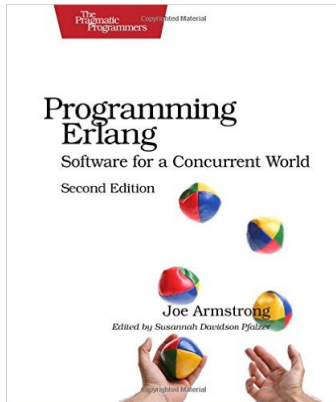


Durante el curso utilizaremos:

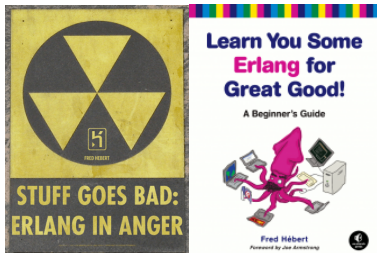
- ▶ **Erlang R18**
(<https://www.erlang-solutions.com/resources/download.html>)
- ▶ **Gitlab** (<http://gitlab.com>)
- ▶ **Editores/IDEs (con respectivos plugins):**
 - ▶ Emacs
 - ▶ Vim
 - ▶ Atom
 - ▶ Sublime Text
 - ▶ IntelliJ CE
 - ▶ Eclipse



- ▶ Distributed Computing – Concepts and Design
- ▶ 5th edition
- ▶ Coulouris et. al
- ▶ Addison Wesley



- ▶ Programming Erlang – Software for a Concurrent World
- ▶ 2nd edition
- ▶ Joe Armstrong
- ▶ The Pragmatic Programmers



- ▶ Stuff go Bad: Erlang in Anger (<http://www.erlang-in-anger.com>)
- ▶ Learn you some Erlang (<http://learnyousomeerlang.com>)
- ▶ Fred Hébert



Agradecimientos

Agradecemos a *Johan Montelius* del KTH ROYAL INSTITUTE OF TECHNOLOGY, por la guía e inspiración en el armado del curso y material facilitado para el mismo.