

# COUCHDB

## Descripción General

Curso orientado a usuarios y desarrolladores de aplicaciones que requieren acceder a información almacenada en un modelo de documentos NoSQL.

## Objetivo

Que el estudiante aprenda y aplique las técnicas y metodologías de procesamiento de información almacenada en un modelo de documentos NoSQL

## Duración

20 horas

## Requisitos

Conocimientos básicos de Linux.

Part I. Introduction

1. Why CouchDB?

1.1. Relax

1.2. A Different Way to Model Your Data

1.3. A Better Fit for Common Applications

1.4. Self-Contained Data

1.5. Syntax and Semantics

1.6. Building Blocks for Larger Systems

1.7. CouchDB Replication

1.8. Local Data Is King

1.9. Wrapping Up

1. Eventual Consistency

1.1. Working with the Grain

1.2. The CAP Theorem

1.3. Local Consistency

1.4. The Key to Your Data

1.5. No Locking

1.6. Validation

1.7. Distributed Consistency

1.8. Incremental Replication

1.9. Case Study

1.10. Wrapping Up

1. Getting Started

1.1. All Systems Are Go!

1.2. Welcome to Futon

1.3. Your First Database and Document

1.4. Running a Query Using MapReduce

1.5. Triggering Replication

[www.cursoslinux.com.mx](http://www.cursoslinux.com.mx)

[ventas@plct.com.mx](mailto:ventas@plct.com.mx)

PLCT S.A. de C.V.

Tel.: 55 4522 7839/55 1800 7696/722447684

- 1.6. Wrapping Up
- 1. The Core API
  - 1.1. Server
  - 1.2. Databases
  - 1.3. Documents
  - 1.4. Revisions
  - 1.5. Documents in Detail
  - 1.6. Replication
  - 1.7. Wrapping Up

## Part II. Developing with CouchDB

- 1. Design Documents
  - 1.1. Document Modeling
  - 1.2. The Query Server
  - 1.3. Applications Are Documents
  - 1.4. A Basic Design Document
  - 1.5. Looking to the Future
- 1. Finding Your Data with Views
  - 1.1. What Is a View?
  - 1.2. Efficient Lookups

- 1.3. Find One
- 1.4. Find Many
- 1.5. Reversed Results
- 1.6. The View to Get Comments for Posts
- 1.7. Reduce/Rereduce
- 1.8. Lessons Learned
- 1.9. Wrapping Up
- 1. Validation Functions
  - 1.1. Document Validation Functions
  - 1.2. Validation's Context
  - 1.3. Writing One
  - 1.4. Type
  - 1.5. Required Fields
  - 1.6. Timestamps
  - 1.7. Authorship
  - 1.8. Wrapping Up
- 1. Show Functions
  - 1.1. The Show Function API
  - 1.2. Side Effect-Free
  - 1.3. Design Documents



- 1.4. Querying Show Functions
- 1.5. Design Document Resources
- 1.6. Query Parameters
- 1.7. Accept Headers
- 1.8. Etags
- 1.9. Functions and Templates
- 1.10. The ljson Macro
- 1.11. The lcode Macro
- 1.12. Learning Shows
- 1.13. Using Templates
- 1.14. Writing Templates
- 1. Transforming Views with List Functions
  - 1.1. Arguments to the List Function
  - 1.2. An Example List Function
  - 1.3. List Theory
  - 1.4. Querying Lists
  - 1.5. Lists, Etags, and Caching
- Part III. Example Application
  - 1. Standalone Applications
    - 1.1. Use the Correct Version
    - 1.2. Portable JavaScript
    - 1.3. Applications Are Documents
    - 1.4. Standalone
    - 1.5. In the Wild
    - 1.6. Wrapping Up
  - 1. Managing Design Documents
    - 1.1. Working with the Example Application
    - 1.2. Installing CouchApp
    - 1.3. Using CouchApp
    - 1.4. Download the Sofa Source Code
    - 1.5. CouchApp Clone
    - 1.6. ZIP and TAR Files
    - 1.7. Join the Sofa Development Community on GitHub
    - 1.8. The Sofa Source Tree
    - 1.9. Deploying Sofa
    - 1.10. Pushing Sofa to Your CouchDB
    - 1.11. Visit the Application
    - 1.12. Set Up Your Admin Account
    - 1.13. Deploying to a Secure CouchDB

[www.cursoslinux.com.mx](http://www.cursoslinux.com.mx)

[ventas@plct.com.mx](mailto:ventas@plct.com.mx)

PLCT S.A. de C.V.

Tel.: 55 4522 7839/55 1800 7696/722447684

1.14. Configuring CouchApp with .couchapprc

1. Storing Documents

1.1. JSON Document Format

1.2. Beyond \_id and \_rev: Your Document Data

1.3. The Edit Page

1.4. The HTML Scaffold

1.5. Saving a Document

1.6. Validation

1.7. Save Your First Post

1.8. Wrapping Up

1. Showing Documents in Custom Formats

1.1. Rendering Documents with Show Functions

1.2. The Post Page Template

1.3. Dynamic Dates

1. Viewing Lists of Blog Posts

1.1. Map of Recent Blog Posts

1.2. Rendering the View as HTML Using a List Function

1.3. Sofa's List Function

1.4. The Final Result

Part IV. Deploying CouchDB

1. Scaling Basics

1.1. Scaling Read Requests

1.2. Scaling Write Requests

1.3. Scaling Data

1.4. Basics First

1. Replication

1.1. The Magic

1.2. Simple Replication with the Admin Interface

1.3. Replication in Detail

1.4. Continuous Replication

1.5. That's It?

1. Conflict Management

1.1. The Split Brain

1.2. Conflict Resolution by Example

1.3. Working with Conflicts

1.4. Deterministic Revision IDs Wrapping Up

1. Load Balancing

1.1. Having a Backup

1. Clustering

- 1.1. Introducing CouchDB Lounge
- 1.2. Consistent Hashing
- 1.3. Redundant Storage
- 1.4. Redundant Proxies
- 1.5. View Merging
- 1.6. Growing the Cluster
- 1.7. Moving Partitions
- 1.8. Splitting Partitions

Part V. Reference

- 1. Change Notifications
  - 1.1. Polling for Changes
  - 1.2. Long Polling
  - 1.3. Continuous Changes
  - 1.4. Filters
  - 1.5. Wrapping Up
- 1. View Cookbook for SQL Jockeys
  - 1.1. Using Views
  - 1.2. Defining a View
  - 1.3. Querying a View

- 1.4. MapReduce Functions
- 1.5. Look Up by Key
- 1.6. Look Up by Prefix
- 1.7. Aggregate Functions
- 1.8. Get Unique Values
- 1.9. Enforcing Uniqueness
- 1. Security
  - 1.1. The Admin Party
  - 1.2. Creating New Admin Users
  - 1.3. Hashing Passwords
  - 1.4. Basic Authentication
  - 1.5. Update Validations Again
  - 1.6. Cookie Authentication
  - 1.7. Network Server Security
- 1. High Performance
  - 1.1. Good Benchmarks Are Non-Trivial
  - 1.2. High Performance CouchDB
  - 1.3. Hardware
  - 1.4. An Implementation Note
  - 1.5. Bulk Inserts and Mostly Monotonic DocIDs

1.6. Optimized Examples: Views and Replication

1.7. Bulk Document Inserts

1.8. Batch Mode

1.9. Single Document Inserts

1.10. Hovercraft

1.11. Trade-Offs

1.12. But...My Boss Wants Numbers!

1.13. A Call to Arms

1.12. Fast Paging (Do Use)

1.13. Jump to Page

## 1. Recipes

1.1. Banking

1.2. Accountants Don't Use Erasers

1.3. Wrapping Up

1.4. Ordering Lists

1.5. A List of Integers

1.6. A List of Floats

1.7. Pagination

1.8. Example Data

1.9. A View

1.10. Setup

1.11. Slow Paging (Do Not Use)