

Universidad Mayor de San Andrés  
Facultad de Ciencias Puras y Naturales

# Informe sobre CouchDB

DAT251 - Base de Datos III

## Docente:

Lic. Celia Elena Tarquino Peralta

## Estudiante:

- Gabriel Muñoz Marcelo Callisaya

## Dupla:

23

## Fecha de entrega:

7 de mayo del 2025

La Paz - Bolivia



# Análisis de CouchDB: Fundamentos y Aplicaciones

## 1. Marco Teórico

### 1.1. Concepto

CouchDB es una base de datos NoSQL orientada a documentos, desarrollada bajo el auspicio de la Apache Software Foundation. Creada en 2005 por Damien Katz, su diseño responde a la necesidad de sistemas distribuidos que prioricen la disponibilidad y la tolerancia a particiones. En 2008, se convirtió en un proyecto de Apache, consolidándose como una solución robusta para entornos distribuidos. CouchDB almacena datos en documentos JSON, identificados por un campo `_id` único, lo que permite estructuras flexibles sin un esquema rígido. Su arquitectura se basa en una API RESTful, que utiliza solicitudes HTTP para operaciones CRUD (Crear, Leer, Actualizar, Eliminar). El lema «relax» refleja su enfoque en la simplicidad, escalabilidad y facilidad de uso.

La evolución de CouchDB incluye mejoras en replicación, consultas avanzadas (Mango Queries, introducidas en 2015) y soporte para entornos móviles. Su diseño distribuido lo distingue de bases de datos relacionales y lo alinea con otras soluciones NoSQL como MongoDB y Cassandra, aunque con un enfoque único en la sincronización bidireccional.

### 1.2. Utilidad

CouchDB es ideal para aplicaciones que requieren alta disponibilidad y operación en entornos con conectividad intermitente, como aplicaciones móviles, sistemas de gestión de contenido (CMS) y plataformas web escalables. Según el teorema CAP (Consistencia, Disponibilidad, Tolerancia a Particiones), CouchDB prioriza la disponibilidad y la tolerancia a particiones sobre la consistencia inmediata, adoptando un modelo de consistencia eventual. Esto permite que los nodos funcionen de forma independiente y sincronicen datos cuando sea posible, lo que es crucial para sistemas distribuidos globalmente.

Por ejemplo, CouchDB es utilizado en aplicaciones como IBM Cloudant y en proyectos de sincronización offline, donde los usuarios pueden trabajar sin conexión y sincronizar datos al reconectarse. Su capacidad de replicación bidireccional lo hace adecuado para entornos donde los datos deben estar disponibles en múltiples ubicaciones sin un punto central de fallo.

### 1.3. Herramientas

CouchDB ofrece dos interfaces principales:

- **Interfaz de línea de comandos (CLI):** Utiliza comandos `curl` para enviar solicitudes HTTP a la API RESTful, permitiendo operaciones como creación de documentos y consultas.
- **Fauxton:** Una interfaz gráfica web, accesible en [http://localhost:5984/\\_utils/](http://localhost:5984/_utils/), que simplifica la gestión de bases de datos, documentos y consultas.

Los datos se organizan en documentos JSON, almacenados en bases de datos. Cada documento es un objeto autónomo con un `_id` único y campos arbitrarios.

CouchDB soporta **Mango Queries**, un sistema de consultas declarativas similar a MongoDB, que utiliza selectores JSON para buscar documentos. Las consultas se optimizan mediante índices, que deben crearse manualmente para mejorar el rendimiento.

**Funcionamiento de las consultas:** Las Mango Queries operan sobre la API `_find`, evaluando condiciones definidas en un selector (por ejemplo, `{"precio": {"$gte": 10}}`). La respuesta incluye solo los campos especificados, reduciendo la carga de datos. La eficiencia depende de índices predefinidos; sin ellos, CouchDB realiza búsquedas completas, lo que puede ser lento en bases de datos grandes. Este enfoque combina flexibilidad con control, aunque requiere planificación para consultas complejas.

#### 1.4. Problemática a abordar

CouchDB aborda la necesidad de bases de datos distribuidas en entornos donde la conectividad es inestable o los datos deben replicarse globalmente. Su arquitectura permite nodos independientes que sincronizan datos mediante protocolos de replicación, resolviendo problemas de latencia y disponibilidad. Sin embargo, presenta desafíos:

- **Ventajas:**
  - Alta disponibilidad y tolerancia a particiones.
  - Replicación bidireccional para sincronización offline.
  - API RESTful accesible desde cualquier lenguaje de programación.
  - Tolerancia a fallos mediante diseño distribuido.
- **Desventajas:**
  - Consistencia eventual, que puede generar conflictos en datos.
  - Consultas complejas menos intuitivas que en bases de datos relacionales.
  - Rendimiento limitado en operaciones de escritura intensiva debido a la arquitectura de almacenamiento.

CouchDB es óptimo para aplicaciones como CMS, aplicaciones móviles con sincronización offline y sistemas IoT, pero no para escenarios que requieren transacciones ACID estrictas o joins complejos, donde bases de datos relacionales o MongoDB pueden ser más adecuadas.

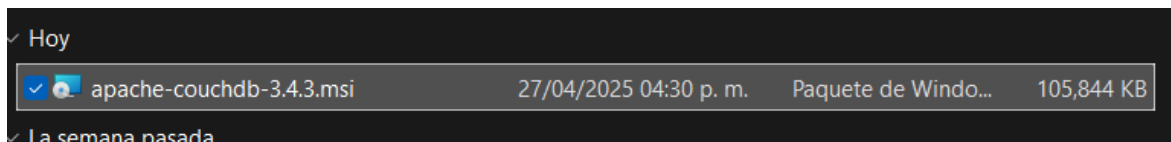
## 2. Marco Aplicativo

El marco aplicativo detalla la implementación práctica de CouchDB, incluyendo instalación, configuración, gestión de datos y consultas. Cada proceso se presenta como una solución práctica, con ejemplos ilustrativos y resultados verificables.

### 2.1. Instalación

CouchDB 3.4.3 se instaló en Windows usando el archivo `apache-couchdb-3.4.3.msi`. Los pasos clave fueron:

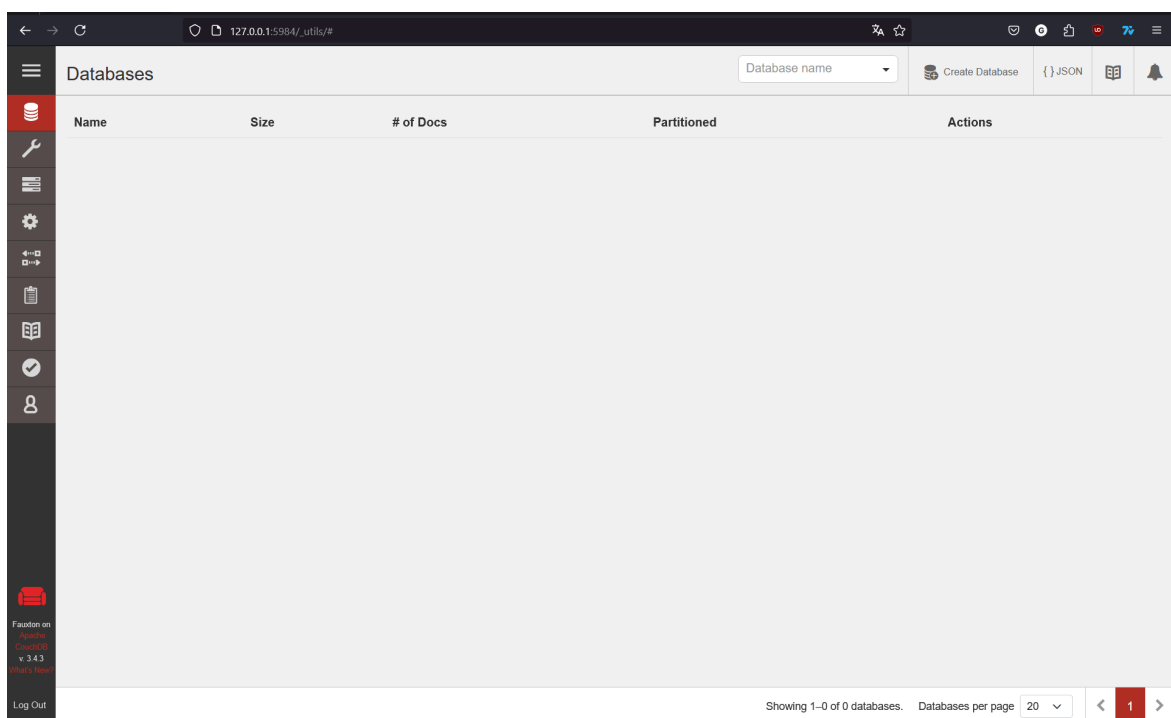
1. Ejecutar el instalador y aceptar los términos.



2. Configurar una cuenta de administrador (usuario: `admin`, contraseña: `123`).

A screenshot of a 'Create Administrator Account' dialog box. The title bar says 'Create Administrator Account'. The main text reads: 'Create CouchDB Admin User' and 'CouchDB 3.x+ requires an admin user to start. Existing credentials will not be overwritten.' Below this, there are two input fields: 'Account name (Example: admin):' with the value 'admin' entered, and 'Password:' with three dots indicating a masked password. A 'Validate Credentials' button is below the password field. At the bottom, there are 'Back', 'Next', and 'Cancel' buttons.

3. Completar la instalación y verificar el acceso a Fauxton en `http://localhost:5984/_utils/`.



**Resultado:** CouchDB se ejecutó correctamente, con Fauxton mostrando una interfaz vacía lista para crear bases de datos.

## 2.2. Configuración de puerto

Si el puerto predeterminado (5984) está ocupado, se puede cambiar:

1. Detener el servicio:

```
net stop CouchDB
```

2. Modificar `C:\Program Files\Apache CouchDB\etc\couchdb\local.ini`, ajustando `port = 9548` en `[chttpd]`.

```
[chttpd]
port = 9548 ;o el puerto que quieras
bind_address = 127.0.0.1
```

3. Reiniciar el servicio:

```
net start CouchDB
```

**Resultado:** Fauxton fue accesible en `http://localhost:9548/_utils/`, demostrando flexibilidad en la configuración.

## 2.3. Creación de base de datos y documentos

La creación de una base de datos y documentos en CouchDB es se realizó utilizando Fauxton y la CLI, demostrando la interoperabilidad de las interfaces.

**Creación de base de datos:** En Fauxton, se creó una base de datos llamada `libreria` accediendo a `http://localhost:5984/_utils/`, iniciando sesión con las credenciales de administrador, y seleccionando «Create Database» en la interfaz gráfica. Fauxton valida el nombre de la base de datos (sin caracteres especiales) y la crea mediante una solicitud HTTP PUT interna al endpoint `/db`. Este proceso es intuitivo, ya que la interfaz guía al usuario a través de un formulario simple.

Database name

Create Database {} JSON

## Create Database

Database name

libreria

Partitioning

☒ Non-partitioned - recommended for most workloads

☐ Partitioned

> Which should I choose?

Cancel Create

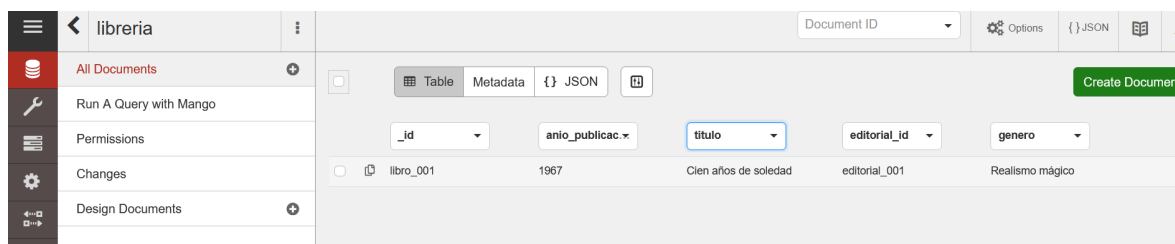
Showing 1\_0 of 0 databases

**Inserción en Fauxton:** Se insertó un documento manualmente en la base de datos `libreria` utilizando el botón «Create Document». Fauxton proporciona un editor JSON donde se ingresa el documento, que debe cumplir con el formato JSON válido y contener un campo `_id` único. El documento insertado fue:

```
{
  "_id": "libro_001",
  "titulo": "Cien años de soledad",
  "autor_id": "autor_001",
  "editorial_id": "editorial_001",
  "genero": "Realismo mágico",
  "anio_publicacion": 1967,
  "precio": 19.99
}
```

- **Sintaxis y propósito:**

- ▶ **\_id**: Identificador único del documento, obligatorio para todas las operaciones CRUD.
- ▶ **titulo, autor\_id, editorial\_id, genero, anio\_publicacion, precio**: Campos personalizados que definen los atributos del libro, flexibles según las necesidades de la aplicación.
- **Funcionamiento**: Al guardar, Fauxton envía una solicitud HTTP PUT al endpoint `/libreria/libro_001`, almacenando el documento en la base de datos. La interfaz valida el JSON y muestra el documento creado en la vista «Table».



**Inserción vía CLI:** Un segundo documento se creó utilizando un comando `curl` que interactúa directamente con la API RESTful:

```
curl -X PUT http://admin:123@127.0.0.1:5984/libreria/libro_002 -H "Content-Type: application/json" -d '{"_id": "libro_002", "titulo": "El amor en los tiempos del cólera", "autor_id": "autor_001", "editorial_id": "editorial_002", "genero": "Romance", "anio_publicacion": 1985, "precio": 15.50}'
```

- **Desglose de la sintaxis:**
  - ▶ `curl`: Herramienta de línea de comandos para enviar solicitudes HTTP.
  - ▶ `-X PUT`: Especifica el método HTTP PUT, usado para crear o actualizar un documento en el endpoint especificado.
  - ▶ `http://admin:123@127.0.0.1:5984/libreria/libro_002`: URL del endpoint, donde:
    - `admin:123`: Credenciales de autenticación (usuario:contraseña).
    - `127.0.0.1:5984`: Dirección y puerto del servidor CouchDB.
    - `libreria`: Nombre de la base de datos.
    - `libro_002`: ID del documento a crear.
  - ▶ `-H "Content-Type: application/json"`: Header que indica que el cuerpo de la solicitud es JSON.
  - ▶ `-d "{...}"`: Cuerpo de la solicitud, un objeto JSON con los mismos campos que el documento de Fauxton.
- **Funcionamiento**: El servidor valida el JSON, asigna un campo `_rev` (versión del documento) y responde con un mensaje `{"ok":true, "id":"libro_002", "rev":"1-..."}`, indicando éxito. La CLI permite automatización y scripting, a diferencia de la interfaz gráfica.

|                          | Table                              | Metadata                                   | { } JSON                              |                                     |                                     |
|--------------------------|------------------------------------|--|---------------------------------------|-------------------------------------|-------------------------------------|
|                          | <input type="text" value="_id"/>   | <input type="text" value="anio_publicac"/> | <input type="text" value="autor_id"/> | <input type="text" value="titulo"/> | <input type="text" value="genero"/> |
| <input type="checkbox"/> | <input type="checkbox"/> libro_001 | 1967                                       | autor_001                             | Cien años de soledad                | Realismo mágico                     |
| <input type="checkbox"/> | <input type="checkbox"/> libro_002 | 1985                                       | autor_001                             | El amor en los tiempos del c...     | Romance                             |

**Resultado:** Ambos documentos fueron visibles en Fauxton, confirmando que las operaciones en CLI y Fauxton son equivalentes, ya que ambas interactúan con la misma API RESTful subyacente.

## 2.4. Importación masiva

La importación masiva permite insertar múltiples documentos en una sola solicitud, optimizando el rendimiento. Se importaron documentos (libros, autores, editoriales) desde un archivo `libreria.json` usando el endpoint `_bulk_docs`:

```
curl -X POST -H "Content-Type: application/json" --data @libreria.json http://admin:123@127.0.0.1:5984/libreria/_bulk_docs
```

- **Desglose de la sintaxis:**

- ▶ `-X POST`: Método HTTP POST, usado para enviar datos al endpoint `_bulk_docs`.
- ▶ `-H "Content-Type: application/json"`: Indica que el cuerpo es JSON.
- ▶ `--data @libreria.json`: Lee el contenido del archivo `libreria.json`, que debe contener un objeto JSON con una clave `"docs"` y un arreglo de documentos.
- ▶ `http://admin:123@127.0.0.1:5984/libreria/_bulk_docs`: Endpoint que procesa múltiples documentos en una transacción atómica.

- **Funcionamiento:** CouchDB valida cada documento en el arreglo, asigna un `_rev` a los documentos creados y responde con un arreglo de resultados (por ejemplo, `[{"ok":true, "id":"autor_001", "rev":"1-...", ...}]`). Esto reduce la sobrecarga de múltiples solicitudes individuales.



| <div> <div></div> <div>Table Metadata {} JSON</div> <div></div> </div> |               |                          |           |                 |
|--|---------------|--------------------------|-----------|-----------------|
|  | _id           | nombre                   | tipo      | anio_publicac.▼ |
| <input type="checkbox"/>   | autor_001     | Gabriel García Márquez   | autor     |                 |
| <input type="checkbox"/>   | autor_002     | George Orwell            | autor     |                 |
| <input type="checkbox"/>   | autor_003     | Miguel de Cervantes      | autor     |                 |
| <input type="checkbox"/>   | autor_004     | Jane Austen              | autor     |                 |
| <input type="checkbox"/>   | autor_005     | Fiódor Dostoyevski       | autor     |                 |
| <input type="checkbox"/>   | autor_006     | Antoine de Saint-Exupéry | autor     |                 |
| <input type="checkbox"/>   | autor_007     | Franz Kafka              | autor     |                 |
| <input type="checkbox"/>   | autor_008     | Emily Brontë             | autor     |                 |
| <input type="checkbox"/>   | autor_009     | J.R.R. Tolkien           | autor     |                 |
| <input type="checkbox"/>   | autor_010     | Julio Cortázar           | autor     |                 |
| <input type="checkbox"/>   | autor_011     | Ray Bradbury             | autor     |                 |
| <input type="checkbox"/>   | autor_012     | Isabel Allende           | autor     |                 |
| <input type="checkbox"/>   | editorial_001 | Sudamericana             | editorial |                 |
| <input type="checkbox"/>   | editorial_002 | Alfaguara                | editorial |                 |
| <input type="checkbox"/>   | editorial_003 | Penguin Books            | editorial |                 |
| <input type="checkbox"/>   | editorial_004 | Espasa                   | editorial |                 |

**Verificación:** La importación se confirmó con:

```
curl -X GET http://admin:123@127.0.0.1:5984/libreria/_all_docs
```

```
C:\Users\gabri\Downloads\guiaCouchDB>curl -X GET http://admin:123@127.0.0.1:5984/libreria/_all_docs
{"total_rows":35,"offset":0,"rows":[
{"id":"autor_001","key":"autor_001","value":{"rev":"1-6272f97b70b68c23c988db40cb2ddb57"}},
{"id":"autor_002","key":"autor_002","value":{"rev":"1-f4036228540bee96b57d515dba4778eb"}},
{"id":"autor_003","key":"autor_003","value":{"rev":"1-6be48be7c2363448648a1ec0d36b4f7e"}},
{"id":"autor_004","key":"autor_004","value":{"rev":"1-57ec0ec085b3439e4c865dbfc85b5ab5"}},
{"id":"autor_005","key":"autor_005","value":{"rev":"1-ce4466ad283abdbfbb662c6760dbf98a"}},
{"id":"autor_006","key":"autor_006","value":{"rev":"1-7f1489f5f136058b9e52307d6701b3de"}},
{"id":"autor_007","key":"autor_007","value":{"rev":"1-e1396e5efffe084ce9ea48f239918a2c"}},
{"id":"autor_008","key":"autor_008","value":{"rev":"1-03e2ebb36c025cbbcc0a170654c92195"}},
{"id":"autor_009","key":"autor_009","value":{"rev":"1-cc663b90dd7588ab13ea18ba0365b613"}},
{"id":"autor_010","key":"autor_010","value":{"rev":"1-bc07a1cd9ee591d05127e52e535e28f1"}},
{"id":"autor_011","key":"autor_011","value":{"rev":"1-5812e5bad9ffbc13ffc6248c3e30c37a"}},
{"id":"autor_012","key":"autor_012","value":{"rev":"1-1a11d1a4e47bb637d4371d7cddffae29"}},
{"id":"editorial_001","key":"editorial_001","value":{"rev":"1-c566480c5ef36b16657e249112b80e8d"}},
{"id":"editorial_002","key":"editorial_002","value":{"rev":"1-d1275284af2e11254501c2bae97307a2"}},
{"id":"editorial_003","key":"editorial_003","value":{"rev":"1-9b0be850f40b8539ca0cfa7d73362b1f"}},
{"id":"editorial_004","key":"editorial_004","value":{"rev":"1-c356447dc52a3bd705d225a31c7a37bc"}},
{"id":"editorial_005","key":"editorial_005","value":{"rev":"1-1bbdf73612ab41797a6bd30b2de367fe"}},
{"id":"editorial_006","key":"editorial_006","value":{"rev":"1-90c03c091a4e10ecdbf37288cbf56926"}},
{"id":"editorial_007","key":"editorial_007","value":{"rev":"1-f7751204f36f82f4d1e6a066cdf27cd4"}},
{"id":"editorial_008","key":"editorial_008","value":{"rev":"1-75a2ba232c0e5276dfb25e952ec842e5"}},
{"id":"editorial_009","key":"editorial_009","value":{"rev":"1-6fe20ef20084cb5d1fe07ab3a0666120"}},
{"id":"editorial_010","key":"editorial_010","value":{"rev":"1-d79c6d5b2eab09814c50dfbc08aa4637"}},
{"id":"libro_001","key":"libro_001","value":{"rev":"1-f58c485467589a70f1a3662ac076e2be"}},
{"id":"libro_002","key":"libro_002","value":{"rev":"1-4e6abdc4d6f993f5ba8e6da1be068fe1"}},
{"id":"libro_003","key":"libro_003","value":{"rev":"1-7e8e518d59b546b4f853d105f6323e32"}},
{"id":"libro_004","key":"libro_004","value":{"rev":"1-e26a2f33c5b1c9cd702b67f8d7c44dc7"}},
{"id":"libro_005","key":"libro_005","value":{"rev":"1-5f6e586f528aeb2c8e3e54085c289657"}},
{"id":"libro_006","key":"libro_006","value":{"rev":"1-ff85a1a753e5360c7941480f0706896d"}},
{"id":"libro_007","key":"libro_007","value":{"rev":"1-ded134daca20b34bc6dac413a9ad6e1a"}},
{"id":"libro_008","key":"libro_008","value":{"rev":"1-d9c6c371937e8044913d118511ee849d"}},
{"id":"libro_009","key":"libro_009","value":{"rev":"1-6c3e8187980784df1f46f095b51a0ca7"}},
{"id":"libro_010","key":"libro_010","value":{"rev":"1-25cd0d9f705d13f94083370e94183ffb"}},
{"id":"libro_011","key":"libro_011","value":{"rev":"1-2c38159ffc09fb8e12976df869179f3"}},
{"id":"libro_012","key":"libro_012","value":{"rev":"1-9936bf7c1d899f1b651b7e6a5c21a7ea"}},
{"id":"libro_013","key":"libro_013","value":{"rev":"1-bbfd301064d51179700c76f3ca385d5b"}}
]}

C:\Users\gabri\Downloads\guiaCouchDB>
```

- **Desglose:** Este comando envía una solicitud GET al endpoint `_all_docs`, que retorna un objeto JSON con una lista de todos los documentos en la base de datos, incluyendo sus `_id` y `_rev`. La respuesta confirma que los documentos de `libreria.json` se añadieron correctamente.

**Resultado:** La operación `_bulk_docs` permitió la inserción eficiente de múltiples documentos, escalable para grandes volúmenes de datos.

## 2.5. Consultas

CouchDB utiliza **Mango Queries** para realizar búsquedas declarativas, similares a MongoDB, ejecutadas a través de la CLI o Fauxton. Se implementaron dos consultas para demostrar su funcionalidad.

**Consulta CLI:** Obtener el nombre y nacionalidad del autor con `_id` «autor\_001»:

```
curl -X POST http://admin:123@127.0.0.1:5984/libreria/_find -H "Content-Type: application/json" -d '{"selector": {"_id": "autor_001"}, "fields": ["nombre", "nacionalidad"]}'
```

- **Desglose de la sintaxis:**

- ▶ `-X POST` y `-H "Content-Type: application/json"`: Indican una solicitud POST con un cuerpo JSON.
- ▶ `http://admin:123@127.0.0.1:5984/libreria/_find`: Endpoint para Mango Queries.
- ▶ `-d "{...}"`: Cuerpo JSON con dos claves:

- **"selector"**: Define las condiciones de búsqueda. Aquí, `{"_id": "autor_001"}` busca documentos con `_id` exacto.
- **"fields"**: Lista de campos a retornar (`["nombre", "nacionalidad"]`), optimizando la respuesta.
- **Funcionamiento**: CouchDB evalúa el selector contra los documentos de la base de datos. Si un índice existe para `_id`, la consulta es eficiente; de lo contrario, realiza una búsqueda completa. La respuesta es un objeto JSON con una clave `"docs"` que contiene los documentos coincidentes, por ejemplo:

```
{"docs": [{"nombre": "Gabriel García Márquez", "nacionalidad": "Colombiana"}]}
```

```
C:\Users\gabri\Downloads\guíaCouchDB>curl -X POST http://admin:123@127.0.0.1:5984/libreria/_find ^
¿Más? -H "Content-Type: application/json" -d ^
¿Más? {"^
¿Más? "selector": {^
¿Más? "_id": "autor_001"^
¿Más? },^
¿Más? "fields": ["nombre", "nacionalidad"]^
¿Más? }"
{"docs": [
{"nombre": "Gabriel García Márquez", "nacionalidad": "Colombiana"}
],
"bookmark": "g1AAAABCeJzLYWBgYMpgSmHgKy5JLCrJTq2MT8lPzkzJBypzJpaW5BfFGxgYghRwwBSgSmUBAGRFEv0",
"warning": "No matching index found, create an index to optimize query time."}

C:\Users\gabri\Downloads\guíaCouchDB>
```

**Consulta Fauxton (Mango Query):** Obtener título, año de publicación, autor y precio de libros con precio entre 10 y 17:

```
{
  "selector": {
    "precio": {
      "$gte": 10,
      "$lte": 17
    }
  },
  "fields": ["titulo", "anio_publicacion", "autor_id", "precio"]
}
```

- **Desglose de la sintaxis:**
  - ▶ **"selector"**: Condiciones de búsqueda. El objeto `"precio"` usa operadores:
    - `$gte`: Mayor o igual a 10.
    - `$lte`: Menor o igual a 17.
  - ▶ **"fields"**: Campos a incluir en la respuesta, reduciendo el tamaño del resultado.
- **Funcionamiento**: En Fauxton, el usuario ingresa esta consulta en el «Mango Query Editor». Fauxton traduce el JSON a una solicitud HTTP POST al endpoint `_find`, enviándola al servidor. CouchDB evalúa la condición, filtra los documentos y retorna solo los campos especificados. La interfaz muestra los resultados en una tabla, facilitando la visualización.

The screenshot shows the Mango Query interface. On the left, a query is written in a text editor:

```

1 {
2   "selector": {
3     "precio": {
4       "$gte": 10,
5       "$lte": 17
6     }
7   },
8   "fields": [
9     "titulo",
10    "anio_publicacion",
11    "autor_id",
12    "precio"
13  ]
14 }

```

Below the editor are buttons for "Run Query", "Explain", and "Manage Indexes". On the right, the results are displayed in a table view. The table has columns for "anio\_publicacion", "autor\_id", "precio", and "titulo".

| anio_publicacion | autor_id  | precio | titulo                            |
|------------------|-----------|--------|-----------------------------------|
| 1985             | autor_001 | 15.5   | El amor en los tiempos del cólera |
| 1949             | autor_002 | 12.99  | 1984                              |
| 1813             | autor_004 | 14.75  | Orgullo y prejuicio               |
| 1943             | autor_006 | 10.99  | El principito                     |
| 1915             | autor_007 | 11.5   | La metamorfosis                   |
| 1847             | autor_008 | 13.8   | Cumbres borrascosas               |
| 1953             | autor_011 | 13.25  | Fahrenheit 451                    |
| 1982             | autor_012 | 16.99  | La casa de los espíritus          |

- **Por qué funciona:** Las Mango Queries son eficientes cuando se usan índices predefinidos para los campos consultados (por ejemplo, `precio`). Sin índices, CouchDB escanea todos los documentos, lo que puede ser lento. La sintaxis declarativa permite consultas complejas (por ejemplo, `$or`, `$and`) sin necesidad de vistas MapReduce, simplificando el desarrollo.

**Resultado:** Ambas consultas retornaron resultados precisos, demostrando la flexibilidad de Mango Queries. La CLI es ideal para automatización, mientras que Fauxton ofrece una experiencia visual para usuarios no técnicos.

### 3. Conclusiones

CouchDB se consolida como una base de datos NoSQL robusta para entornos distribuidos, gracias a su arquitectura orientada a documentos y su enfoque en disponibilidad y tolerancia a particiones. Su API RESTful y herramientas como Fauxton y CLI facilitan la gestión de datos, mientras que las Mango Queries ofrecen un equilibrio entre simplicidad y potencia para búsquedas. Sin embargo, su modelo de consistencia eventual plantea desafíos en aplicaciones que requieren datos siempre coherentes, como sistemas financieros. Comparado con MongoDB, CouchDB destaca en sincronización offline, pero puede ser menos eficiente en consultas complejas. Frente a Cassandra, ofrece una interfaz más accesible, aunque con menor escalabilidad en escritura.

La implementación práctica reveló la facilidad de instalación y uso de CouchDB, aunque la configuración de índices para consultas optimizadas requiere planificación. Su diseño es ideal para aplicaciones modernas como IoT, CMS y aplicaciones móviles, pero su adopción depende de la comprensión de sus limitaciones. Este análisis subraya la importancia de seleccionar bases de datos según las necesidades específicas del proyecto, equilibrando factores como disponibilidad, consistencia y rendimiento.

#### **4. Bibliografía**

- Apache CouchDB. (s.f.). Documentación oficial. Recuperado de <http://docs.couchdb.org/>
- Apache Software Foundation. (s.f.). CouchDB. Recuperado de <https://couchdb.apache.org/>
- Documentación de Fauxton. Recuperado de <https://docs.couchdb.org/en/stable/fauxton/>
- Instalador CouchDB 3.4.3. Recuperado de <https://couchdb.apache.org/#download>
- IBM. (2019). What Is CouchDB? Recuperado de <https://www.ibm.com/topics/couchdb>
- Cloudant. (s.f.). CouchDB and Cloudant Documentation. Recuperado de <https://cloud.ibm.com/docs/Cloudant?topic=Cloudant-couchdb>
- CouchDB Blog. (s.f.). Time to Relax! Recuperado de <https://blog.couchdb.org/>
- GitHub - Apache CouchDB. (s.f.). Repositorio oficial de CouchDB. Recuperado de <https://github.com/apache/couchdb>