**UNIDAD 2: CouchDB**

1. [UNIDAD 2: Introducción](https://learn.nextu.com/mod/page/view.php?id=10287&pid=P_WEB_DATABASE)

**Introducción**

**CouchDB**

CouchDB es un manejador de base de datos no relacionales (NoSQL) orientado a documentos JSON. Este manejador posee unas características que lo hacen diferente a los demás manejadores en la misma categoría, pero quizás la más relevante es que usa un API REST para que con el empleo de simples peticiones del protocolo HTTP permita manejar las bases de datos.

Esta unidad comienza con una descripción detallada de CouchDB, que incluye sus características generales y específicas, y la filosofía de API REST que soporta la forma de interactuar con las bases de datos. Luego de esta introducción, podrás conocer y practicar con el sistema a través del uso de un cliente Web, llamado curl y una Aplicación Web denominada Fauxton.

Tendrás entonces la oportunidad, no solo de manejar una base de datos CouchDB, sino también de realizar consultas básicas y avanzadas, usando el lenguaje Mango Query incluido en las nuevas versiones de CouchDB y las funciones “map” y “reduce” a través del lenguaje JavaScript. Adicionalmente, podrás crear funciones de validación para chequear que tus documentos tengan la estructura y contenido correcto. Finalmente, te familiarizarás con los procesos de seguridad destinados a la creación de usuarios y su permisología, y las operaciones de respaldo y recuperación a través del uso de peticiones y formatos Json.

**Objetivos de aprendizaje**

1. Crear bases de datos NoSQL orientadas a documentos JSON usando CouchDB a través de peticiones HTTP y API REST.
2. Aplicar validaciones a los documentos de una base de datos para que se creen y actualicen de la forma que se requiere.
3. Implementar consultas de mediana complejidad sobre las bases de datos CouchDB que podrán ser estructuradas en vistas que se pueden almacenar.
4. Aplicar las funciones de seguridad de CouchDB para crear usuarios y sus correspondientes permisos en el sistema con la finalidad de validar los usuarios que pueden acceder los diferentes recursos.
5. Aplicar las llamadas y correspondientes formatos para la recuperación y respaldo de datos en CouchDB.
6. Lección 1: Iniciandonos en CouchDB
   1. [Introducción a CouchDB](https://learn.nextu.com/mod/lesson/view.php?id=10288&pid=P_WEB_DATABASE)

Es una base de datos NoSQL orientada a documentos. Trabaja con documentos JSON

Características:

* Programado en Erlang: lenguaje usado para hacer aplicaciones distribuidas
* Licencia es OpenSource.
* Dirigida por el grupo Apache
* Se basa en la replicación (puede tener varios nodos y los mismo se pueden sincronizar)
* Es escalable debido a su arquitectura distribuida
* Lo anterior hace más eficiente el manejo del tráfico

Características resltantes:

* Tiene un API REST 🡪 Lo que significa que la manera de interactuar con la base de datos es HTTP y los fundamentos de la filosofía REST.
* La ventaja es que no requiere de librerías adicionales ya que todo se realiza a través de HTTP.
* Facilita el desarrollo Mobile First para datos, couchDB puede instalarse en sistemas operativos Android o IOS
* La característica de replicación se hace muy útil cuando los móviles no están conectados a la red y luego se conectan para sincronizar los datos con el server.
* Posee un lenguaje que se llama mango para efectuar queries a la DB.
* Maneja las versiones de los documentos.

Herramientas:

* Clientes HTTP: curl://
* Cliente Web llamado Postman el cual se instala en el navegador para hacer peticiones HTTP a otros sistemas.
* couchDB viene con una herramienta llamada Fauxton (es openSource) es una aplicación web que de manera gráfica permite interactuar con la DB.

Filosofía REST:

En la web lo que se tienen son recursos, ejemplo:

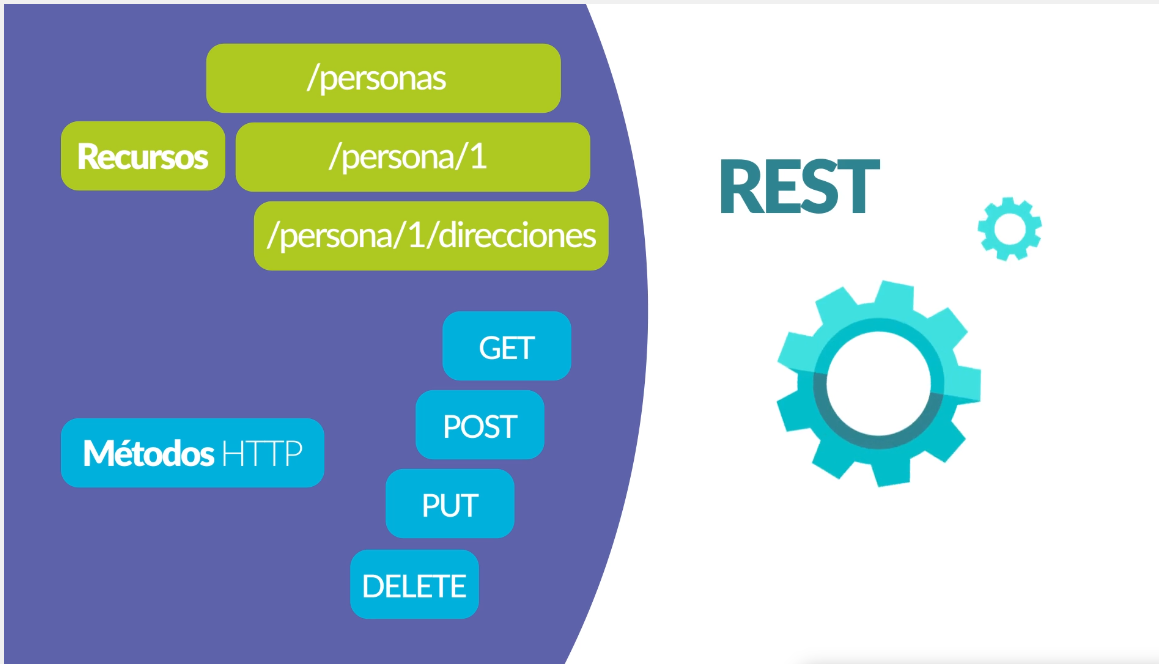
/personas 🡪 es un recurso

/personas/1 🡪 es otro recurso

/personas/1/direccion 🡪 es otro recurso

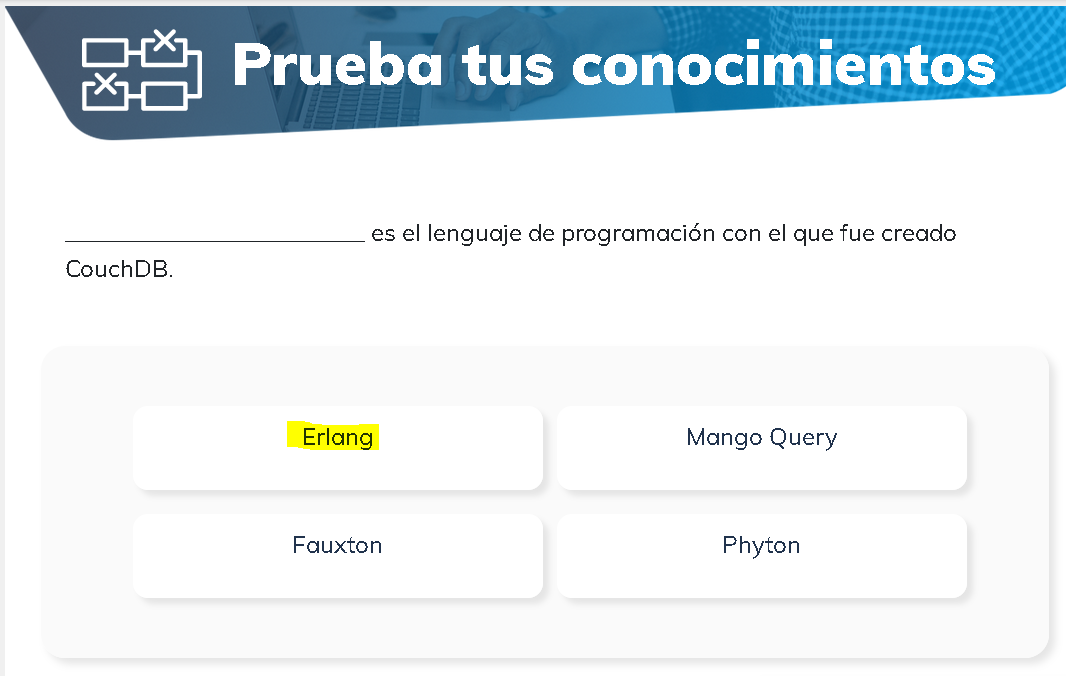
En HTTP se definen los métodos o verbos, y los más comunes son:

GET, POST, PUT y DELETE



Metodo GET se usa para obtener mientras que el método POST se usa para crear. El método PUT se usa para modificar

* 1. Actividad interactiva 1

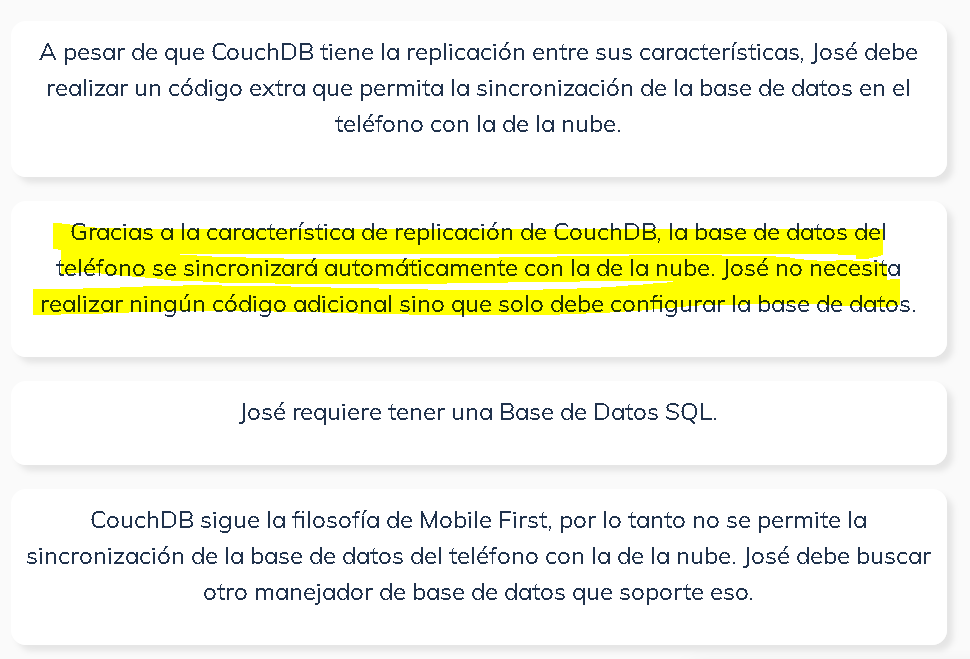


* 1. Actividad interactiva 2

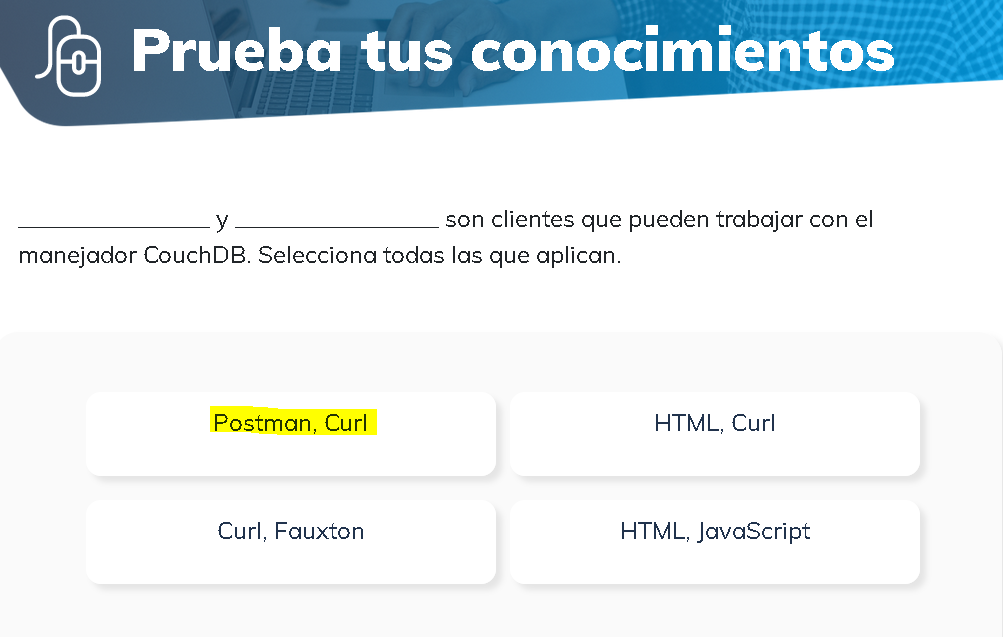
**Prueba tus conocimientos**

José está desarrollando una aplicación para teléfonos móviles, cuya base de datos NoSQL debe sincronizarse con una base de datos en la nube. Él está usando CouchDB como manejador de base de datos y necesita que le ayudes con este proceso de sincronización.

¿Qué requiere José para realizar esta operación?



* 1. Actividad interactiva 3



* 1. Taller 1
  2. Creando y almacenando datos

Curl debe estar instalado en el sistema

O utilizar Fauxton de la siguiente manera:

127.0.0.1/5984/\_utils

Ejemplo de uso con curl:

Curl -X GET <http://127.0.0.1:5984/>

Curl -X GET <http://127.0.0.1:5984/_all_dbs>

Curl -X PUT <http://127.0.0.1:5984/demo>

Curl -X PUT <http://127.0.0.1:5984/demo/_all_docs>

* 1. Taller 2
  2. Ejercicio práctico
  3. ¿Sabías qué?

**Cápsula de conocimiento**

**Resumen de Comandos: Iniciando en CouchDB**

La interacción con CouchDB se realiza a través del protocolo HTTP utilizando una API REST. Aquí tienes una guía rápida de los comandos iniciales para interactuar con la base de datos cuando usas el cliente HTTP curl.

* **Ver status del servidor:**

curl -X GET http://<ip-servidor>:<puerto>/

* **Listar las bases de datos:**

curl -X GET http://<ip-servidor>:<puerto>/\_all\_dbs

* **Crear base de datos:**

curl -X PUT http://<ip-servidor>:<puerto>/<base-de-datos>

* **Listar documentos:**

curl -X GET http://<ip-servidor>:<puerto>/<base-de-datos>/\_all\_docs

* **Listar documentos completos:**

curl -X GET http://<ip-servidor>:<puerto>/<base-de-datos>/\_all\_docs?include\_docs=true

* **Crear documento:**

curl -X PUT http://<ip-servidor>:<puerto>/<base-de-datos>/<id> -d '{<json>}'

Esperamos que esta guía te ayude a familiarizarte con los comandos de CouchDB para manejar la base de datos vía curl. Aunque estas nos son las únicas opciones en cada uno de los comandos sí son las más elementales.

1. Lección 2: Consultas
   1. [Operaciones Básicas](https://learn.nextu.com/mod/lesson/view.php?id=10289&pid=P_WEB_DATABASE)

curl -X PUT http://127.0.0.01:5984/registro

curl -X GET http://127.0.0.01:5984/\_all\_docs

curl -X GET http://127.0.0.01:5984/\_uuids

curl -X GET http://127.0.0.01:5984/\_uuids

curl -X GET http://127.0.0.01:5984/\_uuids?count=5

curl -X PUT http://127.0.0.01:5984/registro/<id> -d '{"nombre": "luis", "edad": 20, "tipo": "usuario"}'

curl -X GET http://127.0.0.01:5984/\_all\_docs

curl -X GET http://127.0.0.01:5984/\_all\_docs?include\_docs=true

curl -X PUT http://127.0.0.01:5984/registro/<id> -d '{"nombre": "luis", "edad": 10, "tipo": "usuario": \_rev: <id-rev>}

curl -X GET http://127.0.0.01:5984/registro/<id>

curl -X DELETE http://127.0.0.01:5984/registro/<id>?rev=<id-rev>

* 1. Taller 1
  2. Consultas con find

Mango Query

Función \_find:

* Find requiere de un JSON.

Selector

Limit

Skip

Sort

Fields

Ejemplo de selector:

{

“Selector”: {

“edad”: 15

}

}

Ejemplo de campos a mostrar:

{

“selector”: {

“edad”: {

“$gt”: 5 🡪 Greater than

}

},

“fields”: [ 🡪 los campos a mostrar

“nombre”,

“tipo”

]

}

Ejemplo de ordenamiento:

{

“selector”: {

“edad”: {

“$gt”: 5 🡪 Greater than

}

},

“sort”: [{“nombre”: “asc”}] 🡪 Para ordenar por nombre ascendente

}

Estos se pueden realizar por medio de la interfaz gráfica Fauxton pero también se puede por medio de peticiones con el formato:

/base\_de\_datos/\_find

* 1. Taller 2
  2. Ejercicio práctico
  3. ¿Sabías qué?

1. Lección 3: Vistas y Validaciones
2. Lección 4: Aplicando Seguridad y Respaldando Datos
3. UNIDAD 2: Prueba