**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:

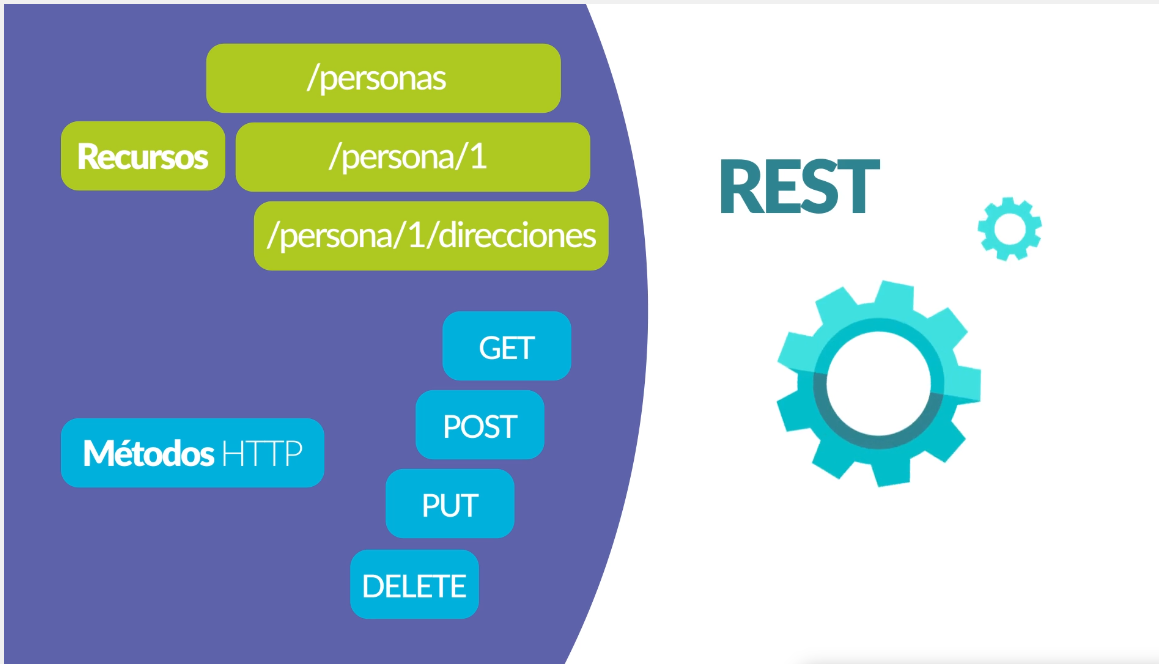
/personas 🡪 es un recurso

/personas/1 🡪 es otro recurso

/personas/1/direccion 🡪 es otro recurso

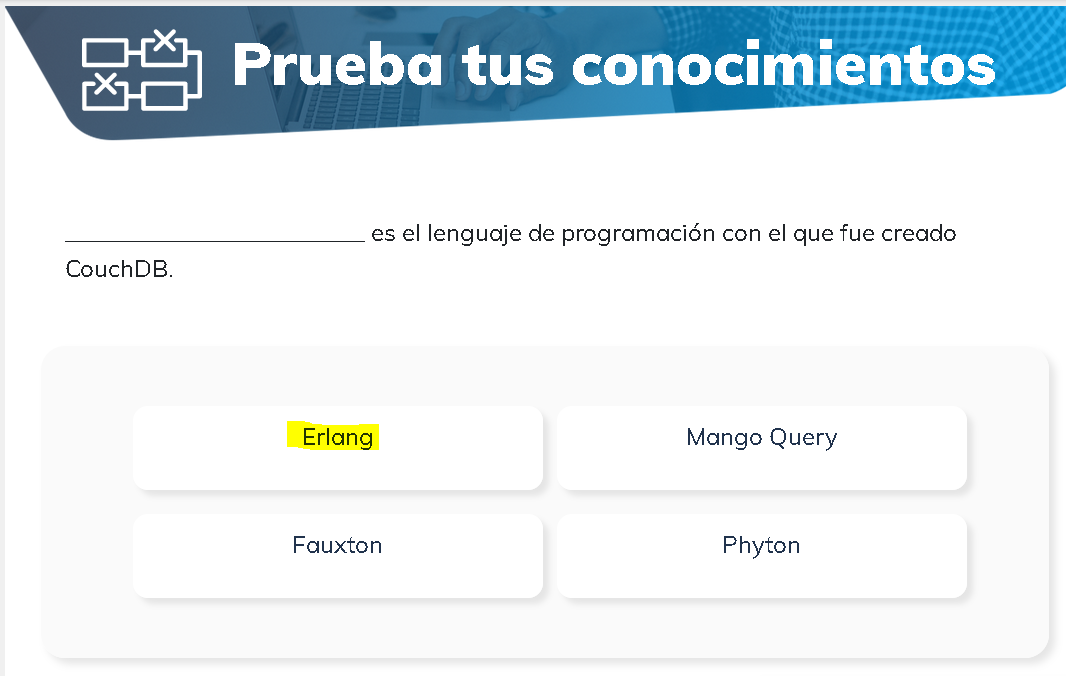
En HTTP se definen los métodos o verbos, y los más comunes son:

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

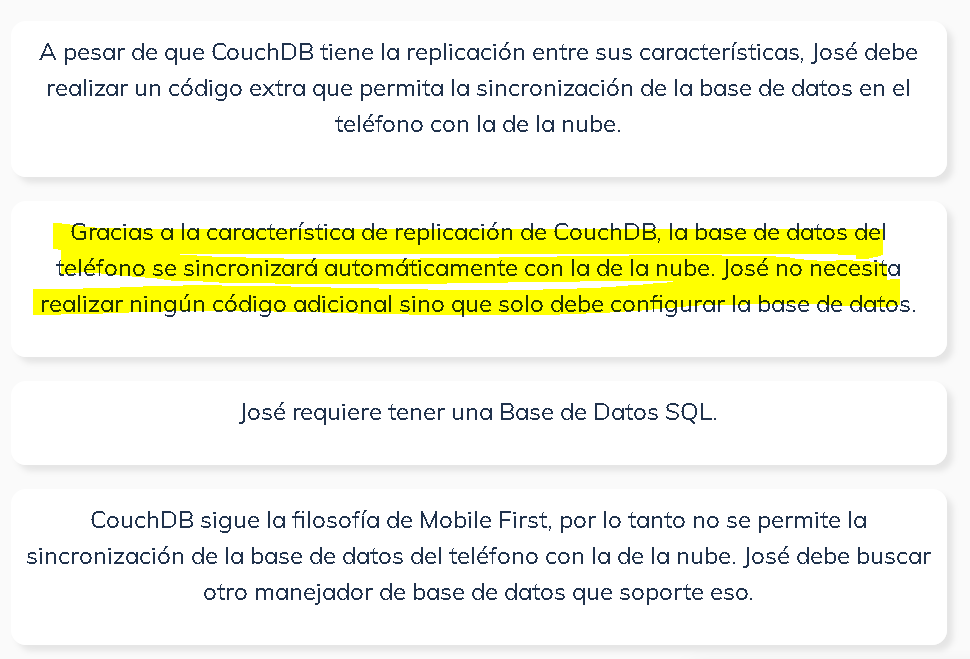


* 1. Actividad interactiva 2

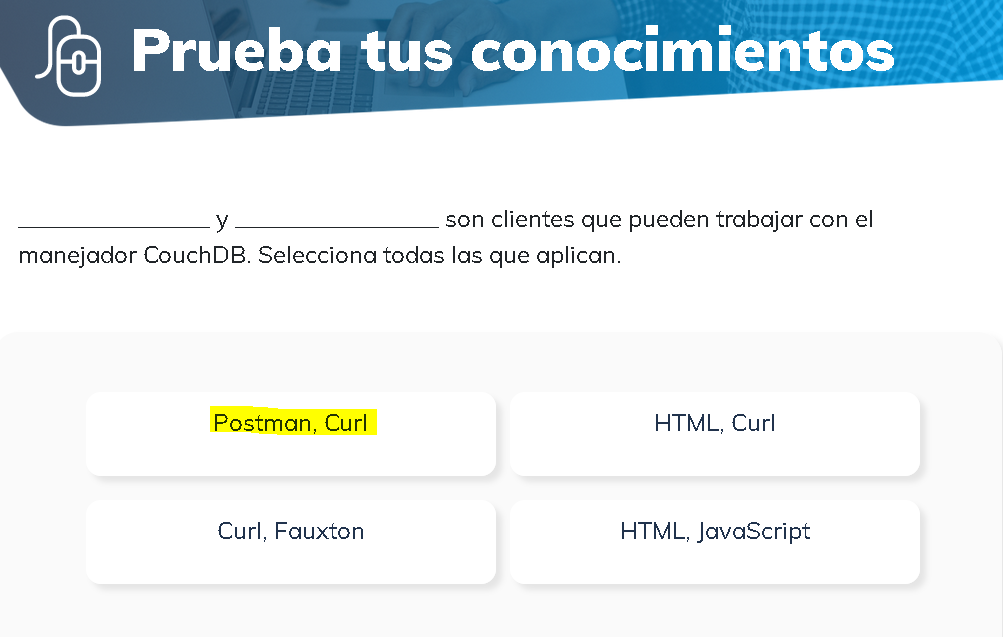
**Prueba tus conocimientos**

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?

**Cápsula de conocimiento**

**Resumen de Comandos: Ids y Mango**

Existe una amplia gama de comandos en CouchDB, sin embargo, seguidamente te ofrecemos una pequeña lista de aquellos relacionados a la obtención de IDs y un resumen de su sintaxis, para que puedas interactuar con el manejador cuando estés empleando curl. Adicionalmente, te presentamos un sumario de las opciones para ejecutar consultas vía el lenguaje Mango.

* 1. **Obtener Id:**

curl -X GET http://127.0.0.1:5984/\_uuids

* 1. **Obtener varios ids:**

curl -X GET http://127.0.0.1:5984/\_uuids?count=>cantidad<

* 1. **Crear documento con el id generado:**

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

* 1. **Mango Query:** Con Mango Query podemos hacer consultas de manera muy fácil, aquí tienes las opciones más usadas.
  + **selector: {>json<}**   
    Este recibe las condiciones que deben cumplir los documentos.
  + **limit: >número<**   
    Representa el número máximo de documentos en la consulta.
  + **sort: {>json<}**   
    Indica el orden de los documentos.
  + **fields: [>array<]**   
    Define los atributos que se quieren por documento.

Esperamos que esta guía te ayude a familiarizarte con los comandos de CouchDB y el lenguaje Mango.

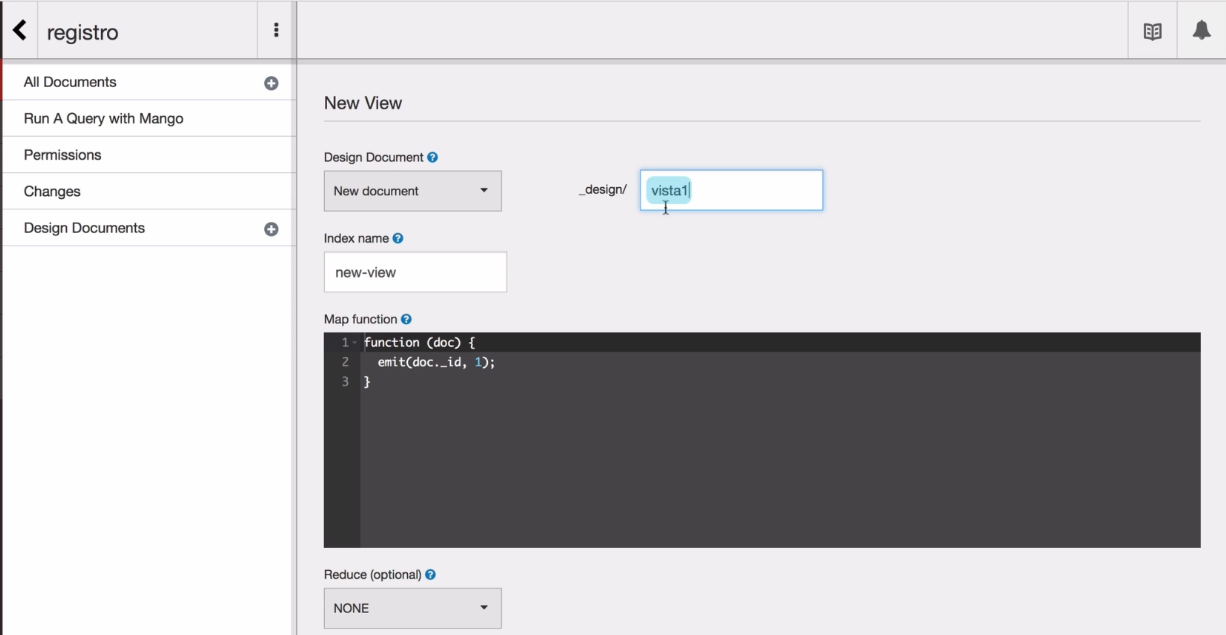
1. Lección 3: Vistas y Validaciones
   1. [Vistas](https://learn.nextu.com/mod/lesson/view.php?id=10290&pid=P_WEB_DATABASE)

Vistas 🡪 resultados personalizados

Fauxton:

Una vista nos permite hacer consultas específicas de una forma más personalizada. Y agregación de datos.

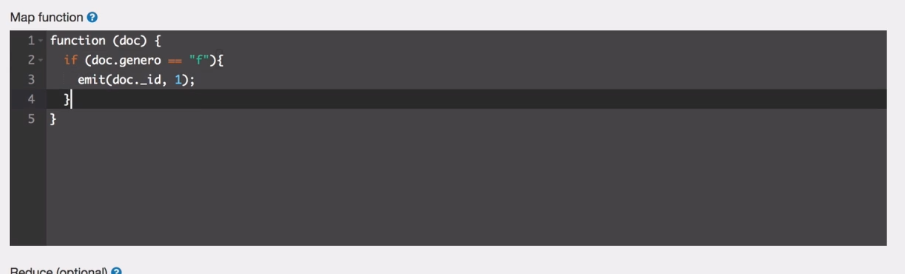
Se indica el nombre de la vista:



Función map

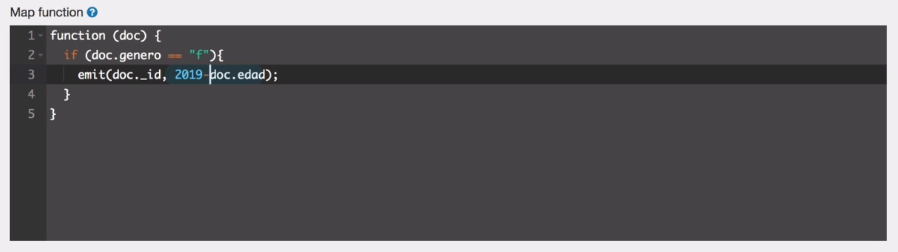
Emit 🡪va a emitir el documento con la clave-valor

Para modificar la vista con alguna condición ejemplo “genero f” de femenino



Las condiciones son más flexibles porque está escrita en javascript

Se pueden hacer modificaciones tal como



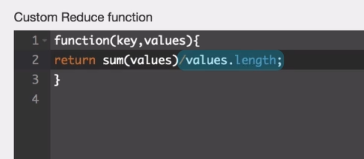
Que retorna algo como:



Por lo que es posible realizar múltiples transformaciones de acuerdo a los requerimientos

La función Reduce funciona como el group by para hacer count, sum o customizar una función

Por ejemplo, para calcular promedios:

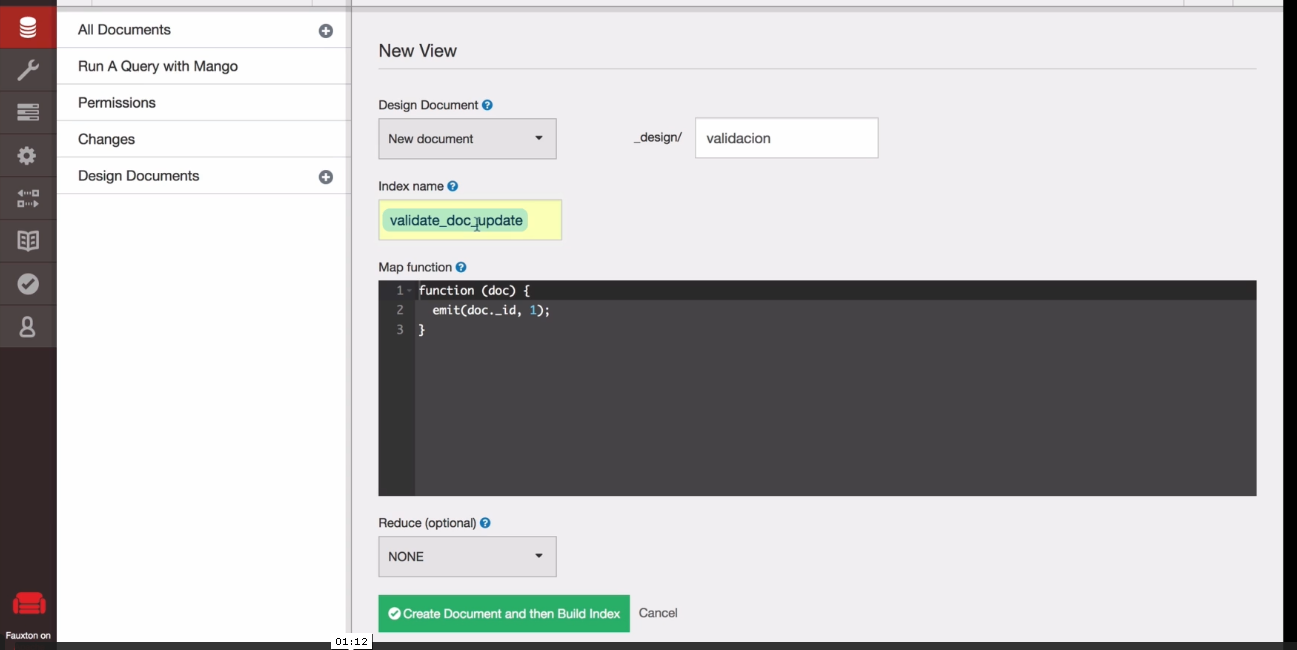


Las vistas quedan guardadas e incluso se le pueden pasar argumentos

* 1. Taller 1
  2. Validaciones de Documentos

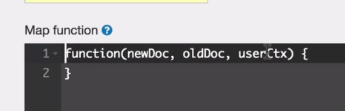
Una de las funcionalidades de CouchDb es que se puede crear validaciones para los documentos.

La función de validación se puede crear con Fauxton y debe estar en \_design/ seguido de un nombre, por ejemplo, validación.



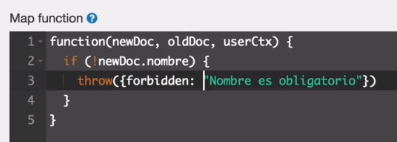
Se debe seleccionar validate\_doc\_update en Index name

La función debe tener nuevo documento, viejo documento y el contexto de usuario (por si se quisiera validar que el usuario puede modificar el documento).



Ya que utiliza JavaScript se pueden hacer validaciones muy flexibles. Y bastante complejas porque es un lenguaje de programación de propósito general.

Ejemplo:



Todas las validaciones son documentos

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

**Cápsula de conocimiento**

**Resumen de Comandos: Vistas y Validación**

CouchDB tiene una serie de comandos para definir vistas y realizar validaciones. A continuación, te proporcionamos una lista de dichos comandos y su sintaxis para interactuar con el manejador de base de datos.

**Vistas**

Para las vistas necesitas crear una función map que haga un emit (clave, valor) de lo que necesitas seleccionar.

1. **Función map:**

function (doc) {//tus reglas emit(<clave>,<valor>);}

1. **Función reduce:** La función reduce que es opcional usa el resultado de la función map para reducir todos los valores a un único valor.

function (keys, values, rereduce) { // tus reglas return <valor agregado>;}

1. **Reduce predefinidos**
   * \_sum: Suma de los valores
   * \_count: Cuenta los valores
   * \_stats: Da varios indicadores de los valores como min, max, etc.

**Validaciones**

Para validar debes crear un documento que tenga el atributo validate\_doc\_update y cuyo valor sea una función:

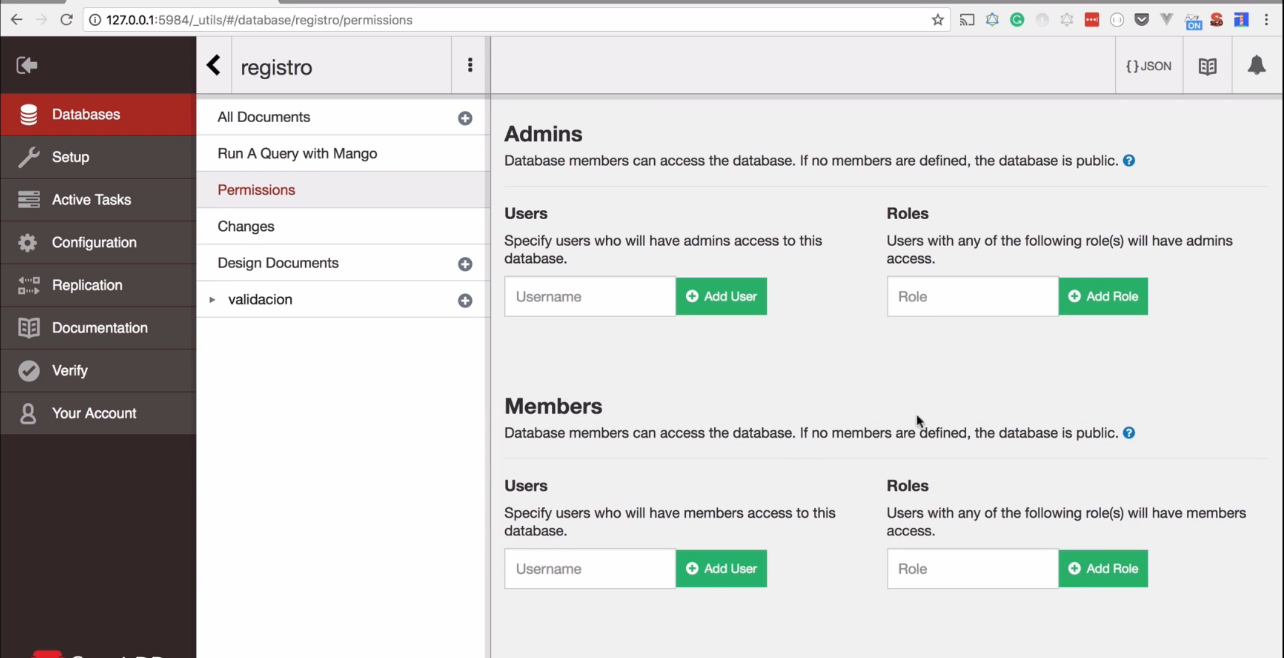
function(newDoc, oldDoc, userCtx) {//tus reglas y lanzar excepción cuando no sea válido throw({forbidden : '<mensaje>'});}

1. Lección 4: Aplicando Seguridad y Respaldando Datos
   1. [Usuarios y Permisos](https://learn.nextu.com/mod/lesson/view.php?id=10291&pid=P_WEB_DATABASE)

Quien puede acceder a los recursos:

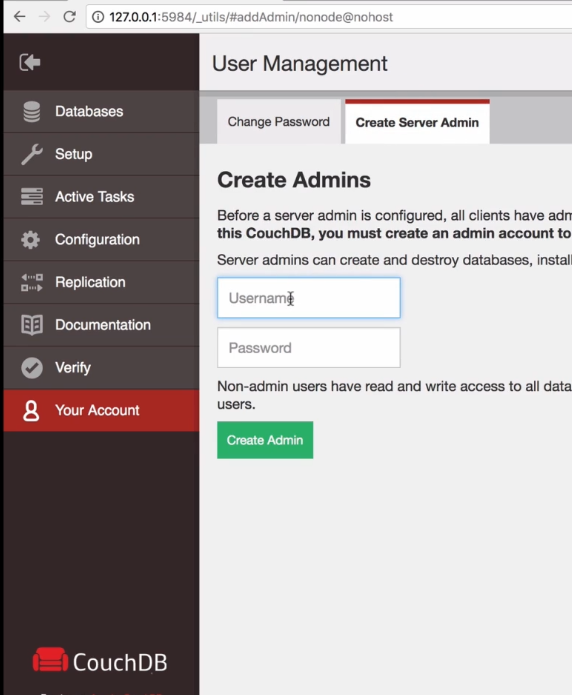
Si no hay usuarios definidos en registro, este puede ser accedida por cualquiera.

En la pestaña Databases 🡪 DB\_name 🡪 permisos



O en el link del candado.

Cuando no hay usuarios definidos se traduce en que la base de datos es pública.



En Fauxton es muy sencillo crear usuarios.

Cuando al menos un usuario es registrado en la base de datos ya no se puede consultar las bases de datos sin identificar el usuario. Por lo que el acceso de debe realizar de la siguiente manera. Ejemplo:

curl -X GET http://user\_name:password@127.0.0.1:5984/registro/\_all\_docs

* 1. Taller 1
  2. Respaldo y Recuperación

Funciones de respaldo y recuperación, couchDB no tiene una forma directa de realizar estas funciones

Se puede realizar usando comandos y ciertos formatos.

Se inserta registro en lote como si fuera una recuperación.

JSON 🡪 atributos “docs” y los documentos, para recuperar los datos se ejecuta el siguiente comando:

curl -X POST http://127.0.0.01:5984/registro/\_bulk\_docs -H "Content-Type: application/json" @dump.txt

Para ejecutar un respaldo se pueden utilizar herramientas de terceros, pero se puede realizar mediante comandos:

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

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

curl -X GET <http://127.0.0.01:5984/registro/_all_docs?include_docs=true> > respaldo.txt

* 1. Taller 2
  2. ¿Sabías qué?

**Cápsula de conocimiento**

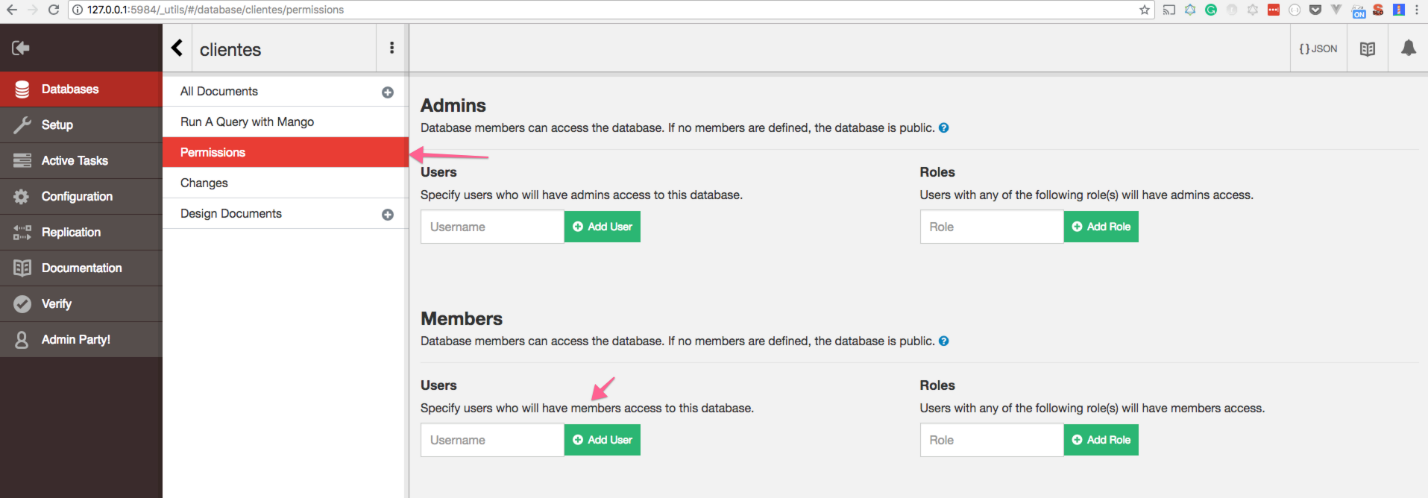
**Resumen de Comandos: Seguridad, Respaldo y Recuperación**

CouchDB proporciona comandos para el manejo de la seguridad, respaldo y recuperación. Seguidamente encontrarás una guía rápida de dichos comandos y su sintaxis.

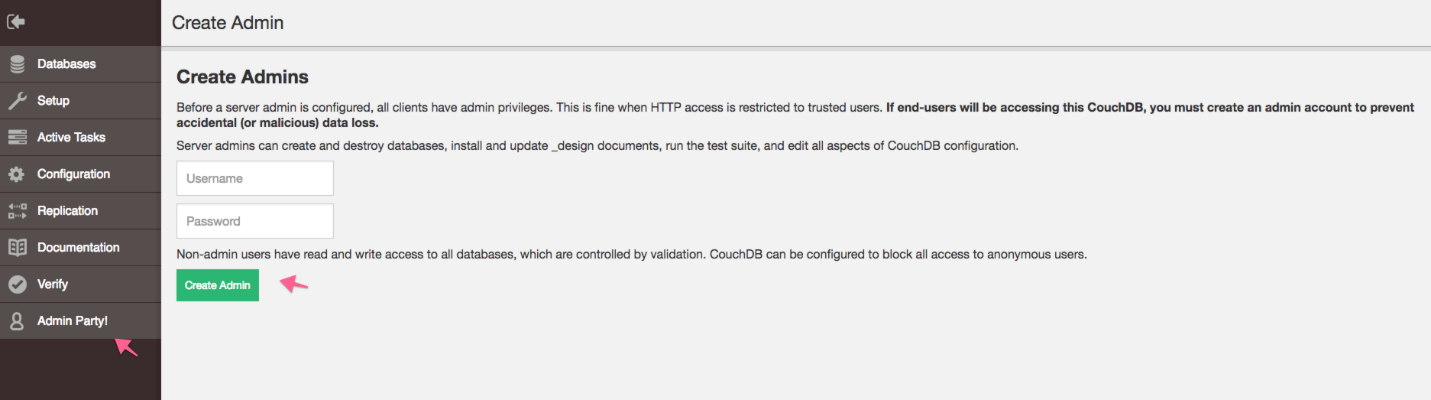
**Seguridad**

En cuanto a la seguridad usaremos Fauxton en donde debes centrarte en las interfaces de usuarios y permisos.

1. **Interfaz de usuarios:**



1. **Función reduce:**



**Respaldo**

En CouchDB no existe una herramienta directa para respaldar los datos, sin embargo, sí existe una estructura para lograr recuperar los datos. Esta estructura, es un “JSON” con el siguiente formato:

{ "docs": [ {<doc1>}, {<doc2>}, ... ]}

**Recuperación**

Para insertar documentos en lote con la estructura del JSON anterior, se debe utilizar el siguiente comando.

curl -d @ -H "Content-Type: application/json" -X POST http://127.0.0.1:5984/<base\_de\_datos>/\_bulk\_docs

1. UNIDAD 2: Prueba

# Bases de Datos No Relacionales > UNIDAD 2: Prueba

|  |  |
| --- | --- |
| **Comenzado en** | Monday, 31 de May de 2021, 15:05 |
| **Estado** | Terminados |
| **Finalizado en** | Monday, 31 de May de 2021, 15:11 |
| **Tiempo empleado** | 5 mins 28 segundos |
| **Puntos** | 9/10 |
| **Calificación** | **92** de un total de 100 |

Top of Form

### Pregunta 1

Parcialmente correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

En una función de validación ¿Para qué se puede utilizar el contexto?. Señales todas las que apliquen.

Seleccione una o más de una:

a. Para ver el usuario que la ejecuta y hacer reglas sobre eso

b. Para ver el documento viejo

c. Para ver el id

d. No tiene utilidad alguna

#### Retroalimentación

**El contexto tiene datos de la ejecución, tal  como el usuario**

Las respuestas correctas son: Para ver el usuario que la ejecuta y hacer reglas sobre eso, Para ver el documento viejo, Para ver el id

### Pregunta 2

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué hará la siguiente petición realizada usando curl, >curl -X Put http://127.0.0.1:5984/prueba?

Seleccione una:

a. Devolverá un error si la base de datos prueba existe, de lo contrario creará la base de datos prueba exitosamente.

b. Creará un documento con nombre prueba dentro de la base de datos que estemos usando.

c. Desplegará el contenido de la base de datos prueba.

d. Devolverá un error diciendo que el método GET no es aceptado en el manejador CouchDB.

#### Retroalimentación

**El método Put es la petición que interpreta CouchDB para crear una base de datos**

La respuesta correcta es: Devolverá un error si la base de datos prueba existe, de lo contrario creará la base de datos prueba exitosamente.

### Pregunta 3

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuáles son algunas de las opciones de Mango Query? Señales todas las que apliquen.

Seleccione una o más de una:

a. selector

b. sort

c. find

d. emit

#### Retroalimentación

**Algunas de las opciones de Mango Query son: el selector: {<json>} que recibe las condiciones que deben complir los documentos y  el sort: {<json>} que indica el orden de los documentos**

Las respuestas correctas son: selector, sort

### Pregunta 4

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuando se ejecuta una función de validación ? Indique todas las que aplique.

Seleccione una o más de una:

a. Crea una base de datos

b. Crea un documento

c. Actualiza un documento

d. Respalda la base de datos

#### Retroalimentación

**Una funcion de validacion se va a ejecutar cada vez que se cree o actualice un documento.**

Las respuestas correctas son: Crea un documento, Actualiza un documento

### Pregunta 5

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

La función reduce es equivalente a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ en MongoDB y a \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ en SQL

Seleccione una:

a. aggregate; GROUP BY

b. aplicar la condición ; WHERE

c. Proyecciones; SELECT

d. ORDER BY, find

#### Retroalimentación

**La función reduce, que reduce los registros a un valor,  es equivalente a las funciones de agregación (aggregate) en SQL, como GROUP BY y a los grupos en MongoDB.**

La respuesta correcta es: aggregate; GROUP BY

### Pregunta 6

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Qué hace a CouchDB resaltante a otros manejadores de base de datos No SQL?

Seleccione una:

a. Usa un API Rest para interactuar con la base de datos a través de HTTP y la filosofía REST.

b. Que puede trabajar con el lenguaje de programación Python

c. Que es un manejador de base de datos NoSQL orientado a documentos Json.

d. Que puede trabajar con base de datos móviles.

#### Retroalimentación

**CouchDB tiene un API REST, que significa que la manera de interactuar con la base de datos es a través de HTTP y los fundamentos de la filosofía REST. Así que la conexión es con  un cliente Web que ya existe y no necesitamos librerías adicionales para interactuar con el  manejador, sino por el contrario solo requerimos las librerías que ya existen para HTTP.**

La respuesta correcta es: Usa un API Rest para interactuar con la base de datos a través de HTTP y la filosofía REST.

### Pregunta 7

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ es uno de los clientes HTTP que se usa para interactuar con CouchDB.

Seleccione una:

a. Compass

b. Mango

c. curl

d. API Rest

#### Retroalimentación

**Una forma de interactuar con CouchDB es a través de un cliente HTTP, llamado curl, que es un comando que está disponible en Linux, asi que viene instalado en la mayoría de estos sistemas operativos, en Windows que debemos instalarlo y en MAC OS que algunas veces viene instalado.**

La respuesta correcta es: curl

### Pregunta 8

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuáles de las siguientes son las características generales de CouchDB?

Seleccione una o más de una:

a. Programado en Erlang

b. Usa una licencia Open Source

c. Es escalable

d. Eficiente manejo del tráfico

#### Retroalimentación

**Las características generales de CouchDB incluye el hecho de ser codificado en erlang, un lenguaje funcional; ser replicable permitiendo la sincronización de múltiples nodos; ser escalable permitiendo incluir más nodos y poder manejar eficientemente las peticiones a las bases de datos gracias a su escalabilidad**

Las respuestas correctas son: Programado en Erlang, Usa una licencia Open Source, Es escalable, Eficiente manejo del tráfico

### Pregunta 9

Correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

¿Cuáles son los parámetros que tiene una funcion de validacion?. Señales todas las que apliquen.

Seleccione una o más de una:

a. Nuevo Documento

b. Viejo Documento

c. Contexto de usuario

d. Revisión

#### Retroalimentación

**El nuevo documento representa los valores nuevos, el viejo documento representa el viejo documento, y el contexto de usuario indica el contexto de ejecución como tal.**

Las respuestas correctas son: Nuevo Documento, Viejo Documento, Contexto de usuario

### Pregunta 10

Parcialmente correcta

Puntúa 1 sobre 1

Señalar con bandera la pregunta

#### Texto de la pregunta

Relacione las definiciones  de CouchDB en la parte izquierda con sus correspondientes definiciones en la parte derecha.

|  |  |
| --- | --- |
| Vista | Respuesta 1 |

Bottom of Form

|  |
| --- |
|  |
| Fauxton | Respuesta 2 |

|  |
| --- |
|  |
| Mango | Respuesta 3 |

|  |
| --- |
|  |
| Javascript | Respuesta 4 |

|  |
| --- |
|  |

#### Retroalimentación

La respuesta correcta es: Vista → forma de estructurar los datos a través de la realización de consultas personalizadas y específicas que se pueden almacenar, Fauxton → Cliente WEB para interactuar con CouchDB, Mango → un lenguaje para realizar consultas, Javascript → un lenguaje usado para construir consultas en una función map