

PROYECTO CARGA DE DATOS APACHE COUCHDB

ALTAMIRANO TACO JOSÉ LUIS
jose.altamirano@epn.edu.ec

TIPÁN VILLEGAS JENNY PATRICIA
jenny.tipan@epn.edu.ec

ESCUELA POLITÉCNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS

I. INTRODUCCIÓN

CouchDB es un sistema distribuido de base de datos basado en nodos. Un numero variable de nodos CouchDB (servidores y clientes offline) puede tener copias réplicas independiente de la misma base de datos, lo que permite que las aplicaciones puedan tener interactividad completa con la base de datos. [1]

Todo lo que se guarda en CouchDB son documentos JSON y teniendo como objetivo encontrar altos volúmenes de registros de información confiables, se establece las fuentes adecuadas para cumplir con este objetivo; las fuentes que se utilizarán son Twitter, CSV y SQL Server.

II. OBJETIVO DEL PROYECTO

- Diseñar un Data Warehouse en el cual confluyan varias fuentes de información estructurada, no estructurada y semi estructurada.

III. ARQUITECTURA DE LA BASE DE DATOS DE DESARROLLO

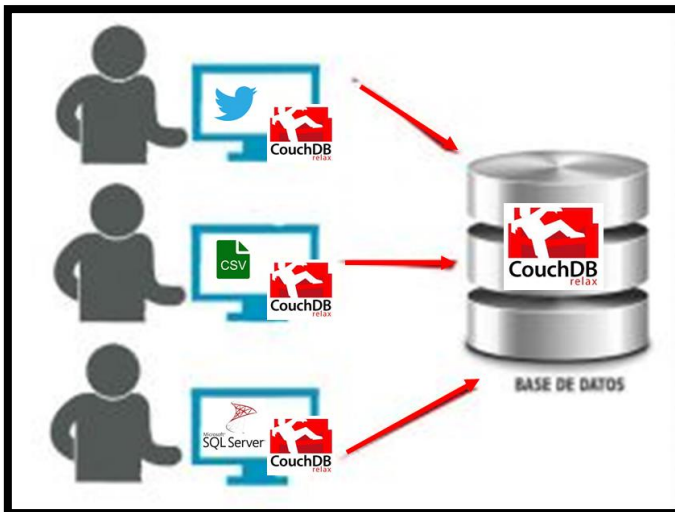


Ilustración 1 Arquitectura de la base de datos

IV. DESARROLLO

PREPARAR Y EXPLORAR DATOS DESDE TWITTER

Utilizar un script en Python para cargar datos de Twitter, para ello se realiza las siguientes actividades:

1. Importar las librerías correspondientes al uso de Twitter. Ingresamos al cmd e ingresamos a la dirección en la que se encuentra ubicada la carpeta Script de Python y ejecutamos el comando **pip install tweepy**.

```
C:\Users\jenti>C:\Users\jenti\AppData\Local\Programs\Python\Python36\Scripts\pip install tweepy
Collecting tweepy
  Downloading https://files.pythonhosted.org/packages/05/f1/2e8c7b202dd04117a378ac0c55cc7dafa80280ebd7/tweepy-3.6.0-py2.py3-none-any.whl
Collecting PySocks>=1.5.7 (from tweepy)
  Downloading https://files.pythonhosted.org/packages/53/12/6bf1d764f128636cef7408e8156b7235b150ea3165/PySocks-1.6.8.tar.gz (283kB)
100% |#####| 286kB 220kB/s
```

Ilustración 2 Importar librería tweepy

2. Una vez que se haya instalado las librerías realizamos el script correspondiente a la carga de datos desde Twitter a CouchDB.

```
import couchdb # Libreria de CouchDB (requiere ser
from tweepy import Stream # tweepy es la libreria q
from tweepy import OAuthHandler
from tweepy.streaming import StreamListener
import json # Libreria para manejar archivos JSON
```

Ilustración 3 Librerías importadas

3. Para la ejecución de este script hemos seleccionado las siguientes claves:

```
ckey = "m4Fq2Pr4yHn1YLLg6nmPYxXyz"
csecret = "0WC0ZLD9sT4aMiBY5xDrRjFIQqT3KbU8oSaNEsFEkKPHZCAsE4"
atoken = "999027411613356032-NvGF9YveYjVjQq4sf61x5IbFDe0KBej"
asecret = "ZHTEn2rxBoKLIkFa57Ksm3Hs4jYtwimhgYcsq8TtAXVXv"
```

Ilustración 4 Credenciales de cuenta privada

- Si la base de datos CouchDB tiene nombre de usuario y contraseña es importante ponerlo en el código, también hay que crear una base de datos y poner las palabras clave para la búsqueda de información.

```
# Setear la URL del servidor de couchDB
server = couchdb.Server('http://admin:1234@localhost:5984/')
try:
    # Si no existe la Base de datos la crea
    db = server.create('primera fuente')
except:
    # Caso contrario solo conectarse a la base existente
    db = server['primera fuente']

# Aqui se define el bounding box con los limites geograficos donde recolectar
twitterStream.filter(track="salud", "medicinas")
# twitterStream.filter(locations=[-78.586922,-0.395161,-78.274155,0.021973])
```

Ilustración 5 Información relevante en el script

- Ejecutar el script y esperar que la información empiece a cargarse en la base de datos.

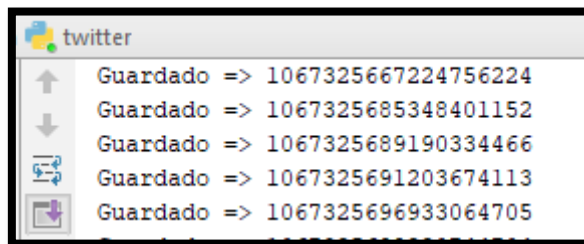


Ilustración 6 Almacenamiento de datos en la base de datos

- Verificamos que la base de datos se haya creado en CouchDB y que los datos se estén cargando.

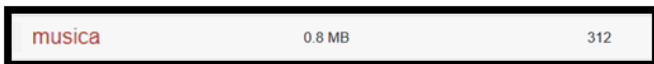




Ilustración 7 Documentos en CouchDB

INSTALACION DE NODE.JS

Node.js es un entorno JavaScript que nos permite ejecutar en el servidor, de manera asíncrona, con una arquitectura orientada a eventos y basado en el motor V8 de Google. Es una plataforma que avanza muy rápidamente y cada vez está más presente en el mercado.

El motor V8 compila JavaScript en código máquina nativo en vez de interpretarlo en el navegador, consiguiendo así una velocidad mucho más alta. Node.js es de código abierto y puede ejecutarse en Mac OS X, Windows y Linux. [2]

- Descargamos Node.js de la página oficial <https://nodejs.org/es/> y seleccionamos la opción **Recomendado para la mayoría**.



Ilustración 8 Herramienta Node.js

- Se despliega una nueva ventana dar clic en **Guardar archivo** y esperar unos minutos a que se descargue la aplicación.

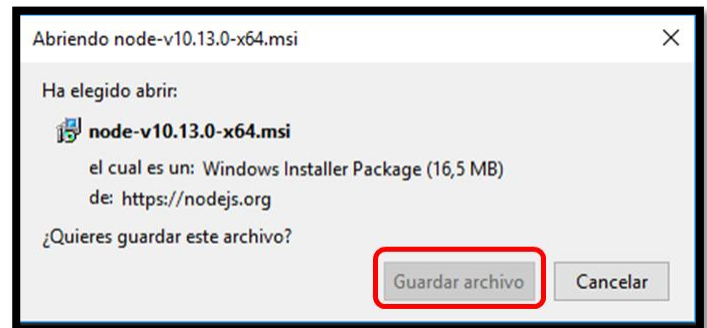


Ilustración 9 Descarga de Node.js

- Ingresar a la carpeta descargas y ejecutar el archivo descargado.



Ilustración 10 Archivo descargado

9. Una vez que se ejecuta el archivo, se despliega una nueva ventana dar clic en **Next** para iniciar con la instalación.

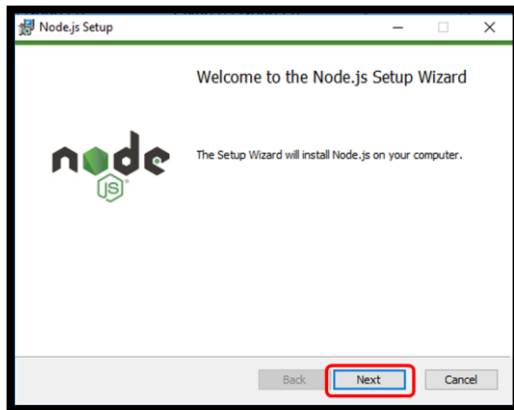


Ilustración 11 Ventana de bienvenida a la instalación de la aplicación

10. En la siguiente ventana **Aceptar los términos de licencia** y dar clic en **Next**.

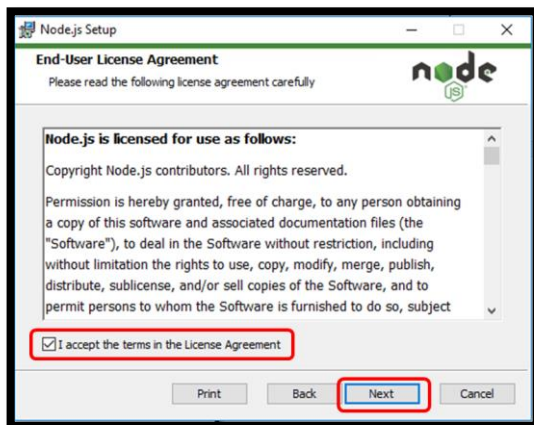


Ilustración 12 Ventana de Aceptación de términos de licencia

11. En la siguiente ventana verificar la ruta de instalación y dar clic en **Next**.

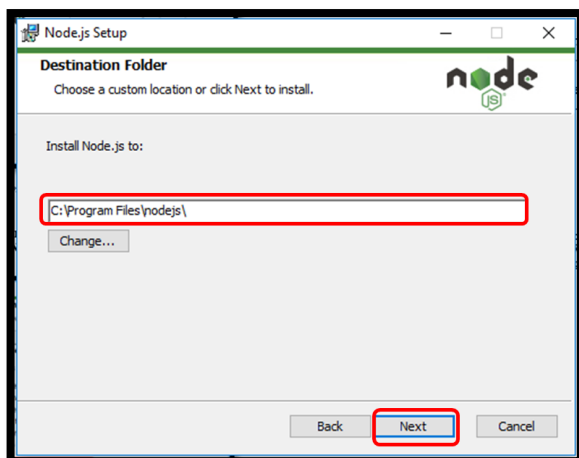


Ilustración 13 Ventana de selección de ruta de instalación

12. En la siguiente ventana nos indica que archivos se van a instalar dar clic en **Next**.

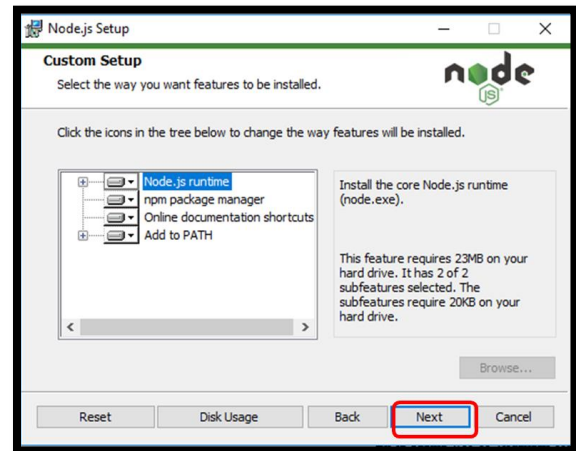


Ilustración 14 Ventana de bienvenida a la instalación de la aplicación

13. En la siguiente venta indica otras herramientas dar clic en **Next**.

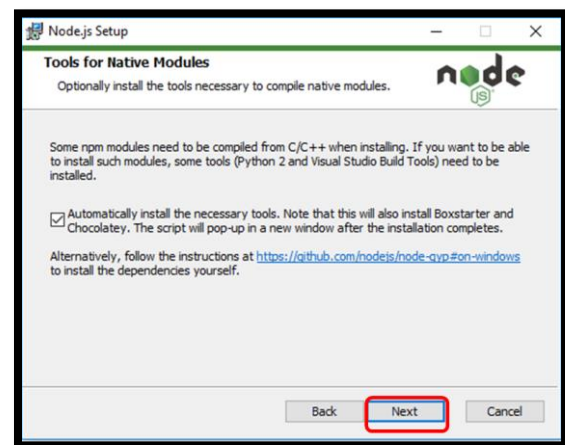


Ilustración 15 Ventana de otras herramientas

14. En la siguiente venta dar clic en **Install** para iniciar la instalación.

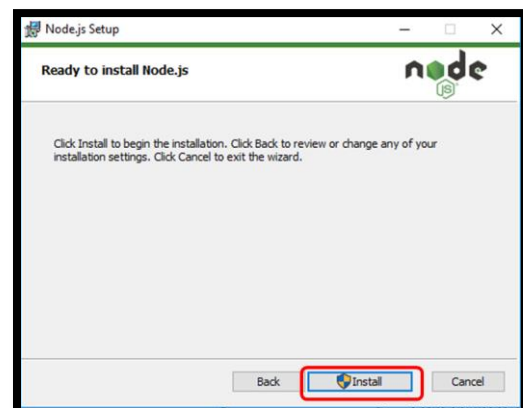


Ilustración 16 Ventana de instalación

- Esperar unos minutos hasta que el programa se instale.

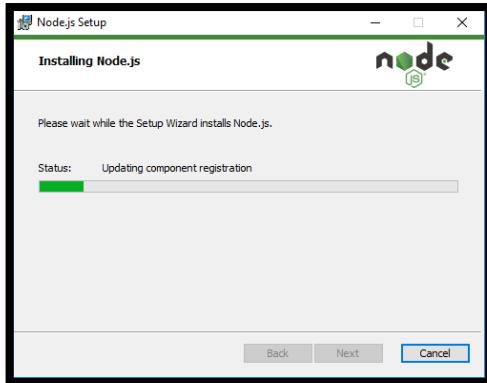


Ilustración 17 Ventana de instalación

- Una vez finalizada la instalación dar clic en **Finish**.

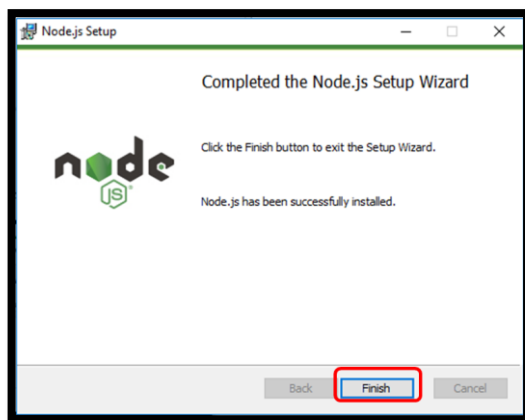


Ilustración 18 Instalación completa

- Ingresamos a cmd y ejecutamos el siguiente comando **npm install -g couchimport** para importar la librería CouchDB.

```
C:\Users\jenti> npm install -g couchimport
C:\Users\jenti\AppData\Roaming\npm\couchexport -> C:\Users\jenti\AppData\Roaming\npm\couchexport
C:\Users\jenti\AppData\Roaming\npm\couchimport -> C:\Users\jenti\AppData\Roaming\npm\couchimport
+ couchimport@1.1.2
added 121 packages from 150 contributors in 161.028s
```

Ilustración 19 Importar CouchDB en Node.js

PREPARAR Y EXPLORAR DATOS SQL

- Descargar una base de datos de tipo SQL Server.

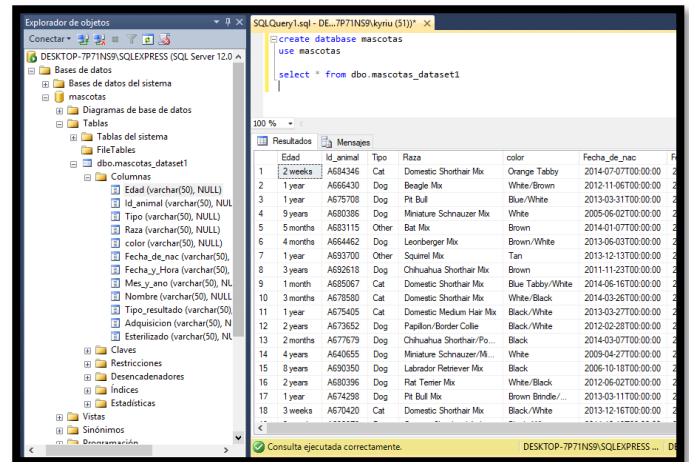


Ilustración 20 Base de datos tipo SQL Server

- Esta base de datos es necesario exportarla a formato CSV. Para ello dar clic derecho sobre la base de datos, seleccionar la opción **Tareas** y dar clic en **Exportar base de datos**.
- Se despliega la ventana de **Asistencia para importación y exportación de SQL Server**. Dar clic en **Siguiente**.

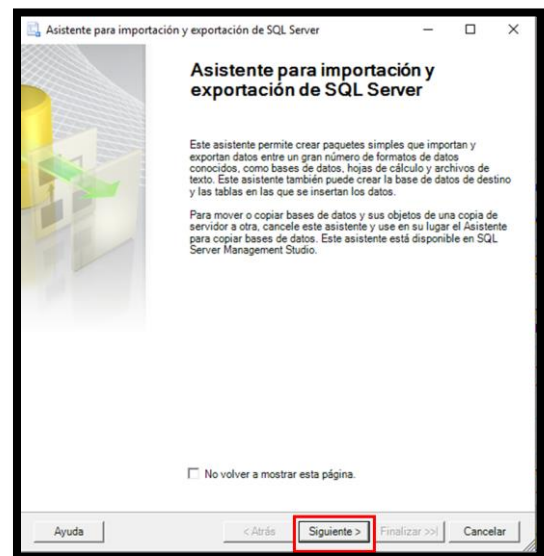


Ilustración 21 Asistente para importación y exportación de SQL Server

- Seleccionar el **Origen de los datos** en este caso es **SQL Server**. Verificar el **Nombre del servidor** y seleccionar la base de datos. Dar clic en **Siguiente**.

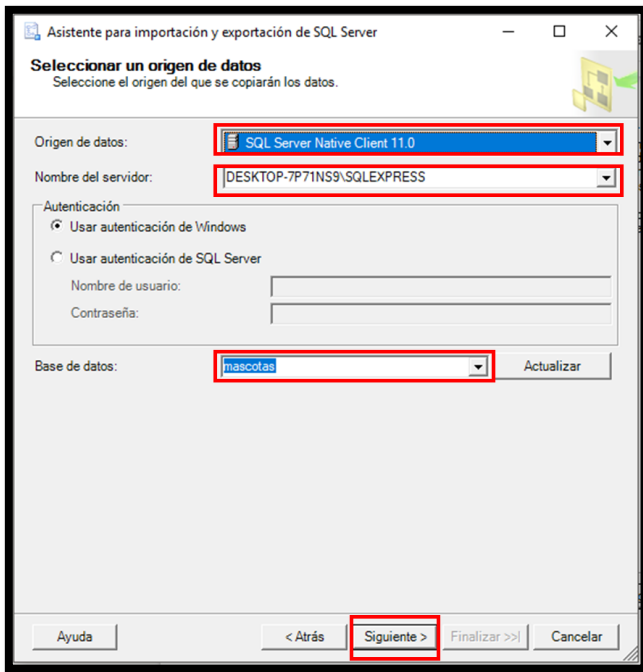


Ilustración 22 Ventana de selección de origen de datos

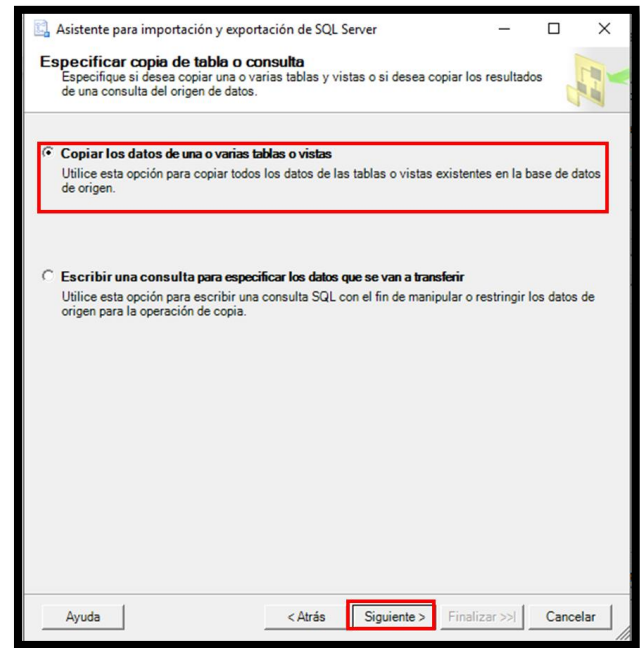


Ilustración 24 Ventana de especificación copia de tabla o consulta

5. En la ventana de **Selección de destino** seleccionar **Destino de archivo plano**, seleccionar el **Nombre de archivo mascotas_dataset1.csv** y la carpeta en la cual se encuentra ubicado y dar clic en **Siguiente**.

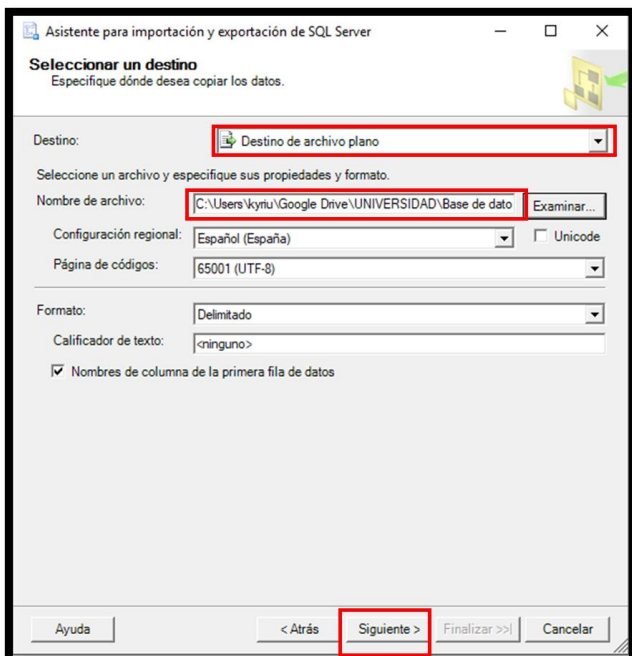


Ilustración 23 Ventana de selección de un destino

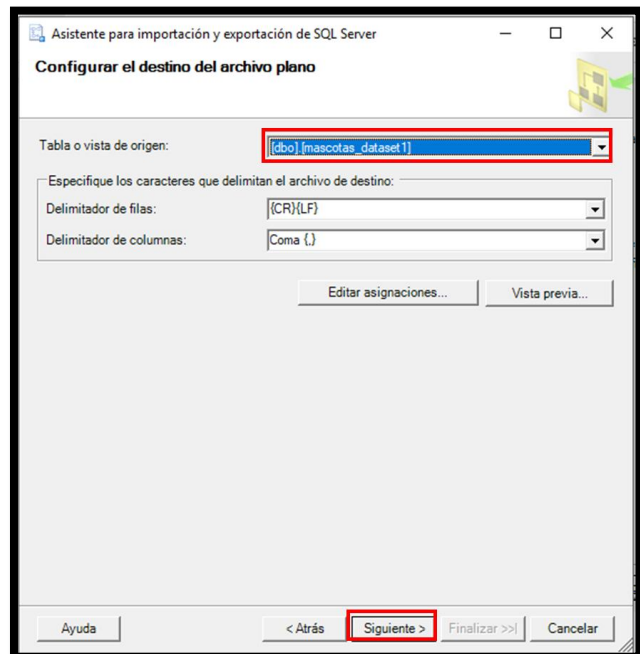


Ilustración 25 Ventana de configuración de destino de archivo plano

6. En la ventana de **Especificar copia de tabla o de consulta**, seleccionar **Copiar los datos de una o varias tablas o vistas**. Dar clic en **Siguiente**.

7. En la tabla de **Configurar el destino del archivo plano**, seleccionar la tabla o vista de origen y dar clic en **Siguiente**.

8. En la ventana de **Ejecutar paquete** dar clic en **Siguiente**.

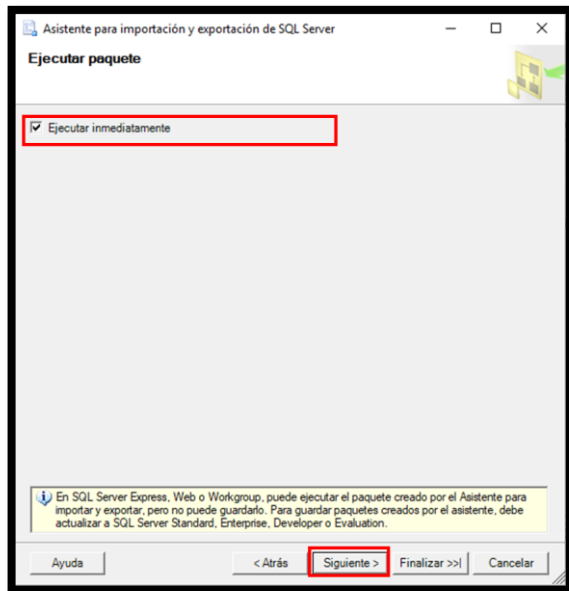


Ilustración 26 Ventana de Ejecutar paquete

9. En la ventana de **Finalización del asistente** dar clic en **Finalizar**.

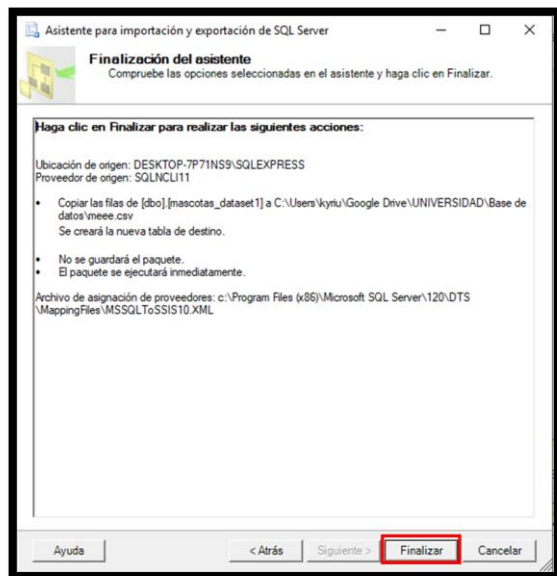


Ilustración 27 Ventana de Finalización del asistente

10. Se puede observar en la siguiente ventana que el archivo ha sido exportado exitosamente.

Acción	Estado	Mensaje
✓ Inicializando la tarea Flujo de datos	Correcto	
✓ Inicializando conexiones	Correcto	
✓ Configurando comando SQL	Correcto	
✓ Configurando la conexión de origen	Correcto	
✓ Configurando conexión de destino	Correcto	

Ilustración 28 Finalización de exportación de base de datos

11. Se puede observar el archivo exportado.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Edad	Id_animal	Tipo	Raza	color	Fecha_de_nu	Fecha_y_Hor	Mes_y_ano	Nombre	Tipo_resulta	Adquisicion	Esterilizado	
2	2 weeks	A684346	Cat	Domestic Sh Orange Tabby		2014-07-07T14:04:00	2014-07-22T16:04:00		Partner	Transfer	Intact Male		
3	1 year	A66430	Dog	Beagle Mix	White/Brown	2013-11-06T13:11:07T	2013-11-07T14:04:00		Partner	Transfer	Spayed Female		
4	1 year	A675708	Dog	Pit Bull	Blue/White	2013-03-31T14:06:03T	2014-06-03T14:06:03T		Johnny	Adoption	Neutered Male		
5	9 years	A680386	Dog	Miniature Sc White		2005-06-02T14:06:15T	2014-06-15T14:06:15T		Monday	Partner	Transfer	Neutered Male	
6	5 months	A683115	Other	Bat Mix	Brown	2014-01-07T14:07:07T	2014-07-07T14:07:07T			Rabies Risk	Euthanasia	Unknown	
7	4 months	A664462	Dog	Leonberger	Brown/White	2013-06-03T13:10:07T	2013-10-07T14:04:00		Edgar	Partner	Transfer	Intact Male	
8	1 year	A693700	Other	Squirrel Mix	Tan	2013-12-13T14:12:13T	2014-12-13T12:20:00			Suffering	Euthanasia	Unknown	
9	3 years	A692618	Dog	Chihuahua	S Brown	2011-11-23T14:12:08T	2014-12-08T14:12:08T		Ella	Partner	Transfer	Spayed Female	
10	1 month	A685067	Cat	Domestic Sh Blue Tabby		2014-06-16T14:08:14T	2014-08-14T14:08:14T		Lucy	Adoption	Intact Female		
11	3 months	A678380	Cat	Domestic Sh White/Black		2014-03-26T14:06:29T	2014-06-29T14:06:29T		Frida	Offsite	Adoption	Spayed Female	
12	1 year	A675405	Cat	Domestic M/Black/White		2013-03-27T14:03:28T	2014-03-28T14:03:28T		Stella Luna	Partner	Return to Ov	Spayed Female	
13	2 years	A673652	Dog	Papillon/Bor	Black/White	2012-02-28T14:03:28T	2014-03-28T14:03:28T		Fancy	Partner	Transfer	Neutered Male	
14	2 months	A677679	Dog	Chihuahua	S Black	2014-03-07T14:05:26T	2014-05-26T14:05:26T		Kash	Foster	Adoption	Neutered Male	
15	4 years	A640655	Dog	Miniature Sc White		2009-04-27T14:04:25T	2014-04-25T14:04:25T		Sandy	Partner	Return to Ov	Spayed Female	
16	8 years	A690350	Dog	Labrador Ret	Black	2006-10-18T14:10:26T	2014-10-26T14:10:26T		Shy	Partner	Return to Ov	Neutered Male	
17	2 years	A680396	Dog	Rat Terrier	W White/Black	2012-06-02T14:06:15T	2014-06-15T14:06:15T		Truman	Partner	Transfer	Neutered Male	
18	1 year	A674298	Dog	Pit Bull Mix	Brown Brindle	2013-03-11T14:04:10T	2014-04-10T14:04:10T		Newt	Partner	Transfer	Neutered Male	
19	3 weeks	A670420	Cat	Domestic Sh Black/White		2013-12-16T14:01:09T	2014-01-09T13:29:00			Partner	Transfer	Intact Male	
20	2 months	A692378	Dog	German Shep	Black/White	2014-10-19T14:12:21T	2014-12-21T14:12:21T		Bonnie	Foster	Adoption	Spayed Female	
21	2 months	A684460	Cat	Domestic Sh Brown Tabby		2014-06-02T14:08:13T	2014-08-13T14:08:13T		Elsa	Partner	Transfer	Spayed Female	
22	8 months	A673952	Cat	Domestic Sh Brown Tabby		2013-07-05T14:03:06T	2014-03-06T14:29:00			SCRIP	Adoption	Unknown	
23	8 months	A686662	Cat	Domestic Sh Black Tabby		2014-03-20T14:08:33T	2014-08-33T14:08:33T		Chloe	Adoption	Neutered Male		

Ilustración 29 Archivo CSV exportado

12. Ingresar a CouchDB y creamos la base de datos **mascotas**



Ilustración 30 Creación de base de datos mascotas

13. Ya tenemos instalado la aplicación Node.js por lo tanto las siguientes acciones se ejecutarán correctamente. Ingresar al cmd y ejecutar el comando `type C:\Users\kyriu\Desktop\mascotas_dataset1.csv | couchimport --url http://admin:1234@localhost:5984 --db mascotas --delimiter ;`. Este comando permite transformar el archivo **mascotas_dataset1.csv** a un archivo **json** y guardar cada registro a la base de datos **mascotas** de CouchDB.

```
C:\Users\kyriu>type C:\Users\kyriu\Desktop\mascotas_dataset1.csv | couchimport --url http://admin:1234@localhost:5984 --db mascotas --delimiter ;
couchimport
-----
url      : "http://admin:1234@localhost:5984"
database: "mascotas"
delimiter: ";"
buffer   : 500
parallelism: 1
type     : "text"
-----
couchimport Written ok:500 - failed: 0 - (500) %ms
couchimport { documents: 500, failed: 0, total: 500, totalfailed: 0 } %ms
couchimport Written ok:1000 - failed: 0 - (1000) %ms
couchimport { documents: 1000, failed: 0, total: 1000, totalfailed: 0 } %ms
couchimport Written ok:1499 - failed: 0 - (1499) %ms
couchimport { documents: 1499, failed: 0, total: 1499, totalfailed: 0 } %ms
couchimport writecomplete { total: 1499, totalfailed: 0 } %ms
couchimport Import complete %ms
```

Ilustración 31 Importar CSV a CouchDB a la base de datos mascotas

14. Verificamos que los documentos se hayan almacenado en la base de datos **mascotas**.

Nombre	tamaño	# de docs
mascotas	0.7 MB	1499

Ilustración 32 Carga de datos en la base mascotas

PREPARAR Y EXPLORAR DATOS CSV

1. Descargar un archivo CSV de algún tema de interés de la página <https://www.kaggle.com/datasets>. En este caso descargamos el archivo **MotorcycleData.csv**.



Ilustración 33 Descargar archivo CSV

1. Ingresar a CouchDB y creamos la base de datos **motos**

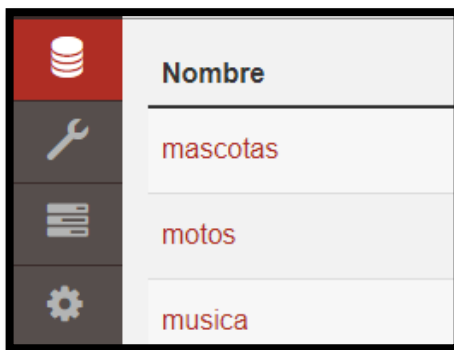


Ilustración 34 Creación de base de datos motos

2. Ya tenemos instalado la aplicación Node.js por lo tanto las siguientes acciones se ejecutarán correctamente. Ingresar al cmd y ejecutar el comando `type C:\Users\kyriu\Desktop\MotorcycleData.csv | couchimport --url http://admin:1234@localhost:5984 --db motos --delimiter ;`. Este comando permite transformar el archivo **MotorcycleData.csv** a un archivo **json** y guardar cada registro a la base de datos **motos** de CouchDB.

```

C:\Users\kyriu>type C:\Users\kyriu\Desktop\motorcycle_data1.csv | couchimport --url http://admin:1234@localhost:5984
--db motos --delimiter ;
couchimport
-----
url           : "http://admin:1234@localhost:5984"
database      : "motos"
delimiter     : ";"
buffer        : 500
parallelism   : 1
type          : "text"
-----
couchimport Written ok:500 - failed: 0 - (500) 100%
couchimport { documents: 500, failed: 0, total: 500, totalfailed: 0 } 100%
couchimport Written ok:500 - failed: 0 - (1000) 100%
couchimport { documents: 500, failed: 0, total: 1000, totalfailed: 0 } 100%
couchimport Written ok:499 - failed: 0 - (1499) 100%
couchimport { documents: 499, failed: 0, total: 1499, totalfailed: 0 } 100%
couchimport writecomplete { total: 1499, totalfailed: 0 } 100%
couchimport Import complete 100%

```

Ilustración 35 Importa datos CSV a CouchDB base de datos motos

3. Verificamos que los datos se hayan almacenado correctamente en la base de datos **motos**.

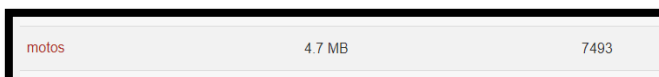


Ilustración 36 Carga de datos en la base motos

4. Se puede observar que tenemos tres bases de datos cargadas con documentos de diferentes fuentes.

Nombre	tamaño	# de docs
mascotas	0.7 MB	1499
motos	4.7 MB	7493
musica	11.8 MB	4187

Ilustración 37 Bases de datos en CouchDB

BASE DE DATOS UNIFICADA

1. Crear la base de datos **unificada** en CouchDB.

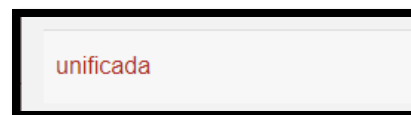


Ilustración 38 Creación de base de datos unificada

2. Se replica cada una de las tres bases de datos antes mencionados con la base de datos **unificada**. De forma que todos los datos de estas bases se encontraran almacenados en conjunto en la base de datos **unificada**.

Fuente	Objetivo
http://localhost:5984/motos	http://localhost:5984/unificada
http://localhost:5984/musica	http://localhost:5984/unificada
http://localhost:5984/mascotas	http://localhost:5984/unificada

Ilustración 39 Réplicas de las bases de datos

3. Se puede observar que la base de datos **unificada** tiene todos los documentos de las otras bases de datos almacenados correctamente.

Name	Size	# of Docs
_replicator	4.9 KB	4
mascotas	0.7 MB	1499
motos	4.7 MB	7493
musica	11.8 MB	4187
unificada	9.5 MB	8329

Ilustración 40 Base de datos unificada con todos los datos cargados

VISUALIZACIÓN DE LOS DATOS

Primero realizamos las siguientes configuraciones en la base de datos CouchDB.

1. En la pestaña de **Configuración** agregar una nueva opción.



Ilustración 41 Configuración de la opción cors

2. En la opción **httpd** cambiar a **true** la opción **enable_cors**.



Ilustración 42 Configuración de la opción enable_cors

3. Una vez hecha las configuraciones en CouchDB, se realiza el diseño de las páginas web para mostrar los datos de las bases de datos restantes.
4. En cada una de las páginas se indica la url, las credenciales de inicio a la base de datos y el nombre de la base de datos de la cual se desea mostrar los datos:
5. En este caso se indica la página para mostrar los datos de la base de datos **música**:
'http://admin:1234@localhost:5984/musica/_all_docs/?limit=200&include_docs=true'

```
var jsonData = $.ajax({
  url: 'http://admin:1234@localhost:5984/musica/_all_docs/?limit=200&include_docs=true',
  //url: 'http://127.0.0.1:5984/test/_all_docs?include_docs=true&conflicts=true',
  data: { page: 1 },
  dataType: 'json',
}).done(function (results) {
```

Ilustración 43 URL de la base de datos motos

6. Para esta base de datos hemos tomado los siguientes datos para mostrarlos.

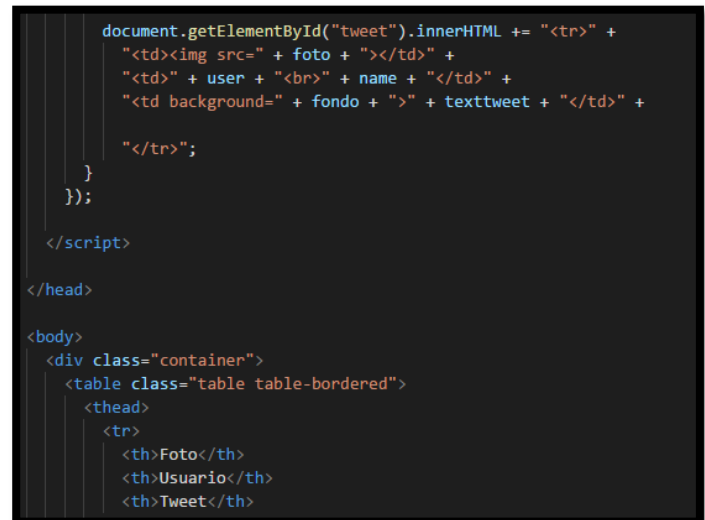


Ilustración 44 Datos que se visualizarán de la base de datos musica

7. Se puede observar los datos de la siguiente manera:

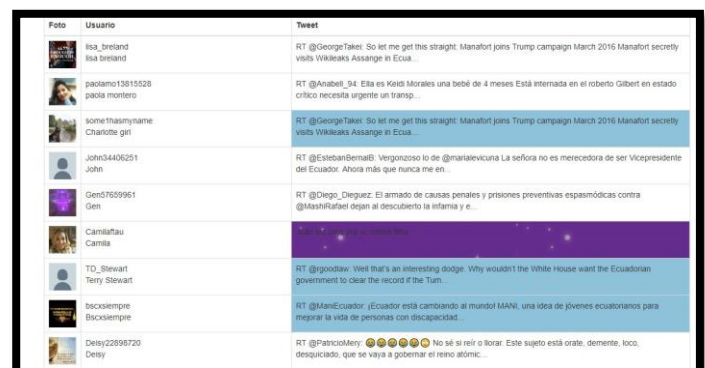


Ilustración 45 Visualización de la Página Música

8. Para la página web que indica los datos de la base de datos motos se utiliza:
'http://admin:1234@localhost:5984/motos/_all_docs/?limit=300&include_docs=true'
9. En esta base de datos se indica los siguientes datos:


```

<body>
  <h4><center>Base de datos Motos</center></h4>
  <div id="contenedor" style="display: inline-flex">
    <div id="chart_div" style="padding:0px 0px;"></div>
    <div id="chartdiv"></div>
  </div>
  <div class="container">
    <table class="table table-bordered" >
      <thead>
        <tr>
          <th>Marca</th>
          <th>Tipo</th>
          <th>Submodelo</th>
          <th>Precio</th>
        </tr>
      <tbody id="tweet">
      </tbody>
    </table>
  </div>
</body>
</html>

```

Ilustración 46 Datos que se mostraran en la página de la base de datos motos

10. Los datos se ven de la siguiente manera:



Ilustración 47 Visualización de la Página Motos

11. Para la página web que indica los datos de la base de datos motos se utiliza:
 'http://admin:1234@localhost:5984/mascotas/_all_docs/?limit=300&include_docs=true'

```

var jsonData = $.ajax({
  url: 'http://admin:1234@localhost:5984/mascotas/_all_docs/?limit=300&include_docs=true',
  data: {page: 1},
  dataType: "json",
}).done(function (results){

```

Ilustración 48 URL de la base de datos mascotas

12. Los datos que se visualizarán en esta página son:

```

<body>
  <h4><center>Base de datos Mascotas</center></h4>
  <div id="contenedor" style="display: inline-flex">
    <div id="chart_div" style="padding:0px 0px;"></div>
    <div id="columnchart_material" style="width: 600px; height: 400px;"></div>
  </div>
  <div class="container">
    <table class="table table-bordered" >
      <thead>
        <tr>
          <th>Nombre</th>
          <th>Tipo</th>
          <th>Raza</th>
          <th>Color</th>
        </tr>
      </thead>
    </table>
  </div>
</body>

```

Ilustración 49 Datos que se mostraran en la página de la base de datos mascotas

13. Los datos se verán de la siguiente manera:



Ilustración 50 Visualización de la Página Mascotas

V. CONCLUSIONES

- El nodo principal es la base de datos unificada que contiene en conjunto todos los datos de los tres nodos restantes cargados con datos de diferentes fuentes, lo que quiere decir que cada vez que se vaya a registrar un documento en cualquiera de los tres nodos este también se almacenará en la base de datos principal, en caso de usar una réplica Continua.
- Si se va a optar por una réplica OnTime, primero se cargará los datos correspondientes en cada uno de los nodos independientes, una vez que los nodos estén completamente cargados se procede a realizar la réplica a la base de datos principal puesto que se cargaran únicamente los datos que en ese momento este almacenados en cada nodo, no se actualizará la base de datos principal si se agrega nuevos documentos en los nodos.

- En la página web se visualizan solo los datos que se consideraron más relevantes, es importante conocer como están estructurados los documentos que contiene cada base de datos.

VI. RECOMENDACIONES

- Revisar más sobre carga de documentos en la base de datos desde el programa Python.
- Es importante importar todas las librerías respectivas a las fuentes utilizadas para cargar datos en CouchDB desde Python.
- Es importante revisar correctamente como están estructurados los documentos cargados en la base de datos, para presentarlos en la página web.

VII. REFERENCIAS

- [[En línea]. Available:
1 https://www.google.com/search?q=cargar+datos+en+couchdb&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjKwIO06fPeAhVDq1kKHWtBDucQ_AUIDygC&biw=650&bih=647#imgsrc=CR9YZgWq-3KuoM: [Último acceso: 27 Noviembre 2018].
2
3 [J. Cabana, 01 Agosto 2017. [En línea]. Available:
4 <https://www.drauta.com/que-es-nodejs-y-para-que-sirve>. [Último acceso:
5 27 Noviembre 2018].