



ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS



ANÁLISIS DE DATOS

(TDSD353)

ASIGNATURA:

Análisis de datos

PROFESOR:

Ing. Juan Pablo Zaldumbide Proaño.

PERÍODO ACADÉMICO:

2021-B

DEBER 1

MongoDB



Nombre del estudiante: Farinango Eduardo

INSTRUCCIONES

- 1.- Generar un script que ingrese "n" ciudadanos en MongoDB, conformados por
{
 "nombres_completos": "<conformado por dos elementos al azar de la lista nombres>",
 "apellidos_completos": "<conformado por dos elementos al azar de la lista apellidos>",
 "edad": "<valor al azar entero, mayor de edad y menor que 80 años>",
 "cedula": "<string al azar formado de 10 dígitos>"
}
2.- Crear una función que muestre los elementos ingresados en la base de datos.
Debe ejecutar el ejercicio localmente y en la nube.
Entregables: Capturas de pantalla con los pasos seguidos y código fuente (.ipynb o .py) en un repositorio de github

Desarrollo:

Para podernos conectar a Atlas MongoDB debemos primeramente instalar en driver respectivo ue creara la conexión.

```
Anaconda Prompt (anaconda3)

(base) D:\AD\Jupyter>pip install pymongo
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)
Requirement already satisfied: pymongo in c:\users\eduardo\anaconda3\lib\site-packages (4.0.1)
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -ymongo (c:\users\eduardo\anaconda3\lib\site-packages)

(base) D:\AD\Jupyter>
```

Una vez instalado en Jupyter importamos MongoClient de nuestra librería pymongo.

```
: #Farinango Eduardo
import couchdb

import random
import string
import numpy as np
import pymongo
from pymongo import MongoClient
```

Por el lado del servidor en la nube creamos una nueva conexión utilizando MongoDB Compass localmente utilizando las credenciales ingresadas al momento de la instalación.

MongoDB Compass - cluster0.juqfh.mongodb.net

Connect View Help

Local

3 DBS 1 COLLECTIONS

☆ FAVORITE

HOSTS

- cluster0-shard-00-00.juqfh...
- cluster0-shard-00-01.juqfh...
- cluster0-shard-00-02.juqfh...

CLUSTER

Replica Set (atlas-csmle-s...

3 Nodes

EDITION

MongoDB 4.4.11 Enterprise

Filter your data

- admin
- deber
- local

+ MONGOSH

Databases Performance

Create database View

admin

Storage size: 0 B Collections: 0

deber

Storage size: 20.48 kB Collections: 1

local

Storage size: 0 B Collections: 0

cloud.mongodb.com/v2/61dd08f5db876f19c4486324#clusters/detail/Cluster0

Aplicaciones Área personal Microsoft Teams Descarga de lección... CD - OneDrive Correo: EDUARDO... Correo: FARINANG... My Apps física EPN Estadística Otros marcados

Eduardo's Org - 202... Access Manager Billing All Clusters Get Help Eduardo

Project 0 Atlas Realm Charts

Cluster0

VERSION 4.4.11 REGION AWS N. Virginia (us-east-1) CLUSTER TIER M0 Sandbox (General)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line To

SANDBOX NODES REPLICA SET

CONNECT CONFIGURATION

REGION N. Virginia (us-east-1)

- cluster0-shard-00-00.juqfh... SECONDARY
- cluster0-shard-00-01.juqfh... SECONDARY
- cluster0-shard-00-02.juqfh... PRIMARY

This is a Shared Tier Cluster

If you need a database that's better for high-performance production applications, upgrade to a dedicated cluster.

Upgrade

Logical Size 20.3 KB

512.0 MB max

0.0 B

Last 30 Days

Operations R: W: 100.0/s

0

Last 6 Hours

Connections 500 max

0

Last 6 Hours

Una vez creado el cluster0 deberemos crear la conexión de Python a Atlas MongoDB, este nos brinda una opción para crear un cliente y el código necesario con los parámetros aceptados.

Connect to Cluster0

✓ Setup connection security

✓ Choose a connection method

Connect

1 Select your driver and version

DRIVER

Python

VERSION

3.12 or later

2 Add your connection string into your application code

☐ Include full driver code example

```
mongodb+srv://admin:<password>@cluster0.juqfh.mongodb.net/myFirstDatabase?retryWrites=true&w=majority
```

Replace **<password>** with the password for the **admin** user. Replace **myFirstDatabase** with the name of the database that connections will use by default. Ensure any option params are [URL encoded](#).

Having trouble connecting? [View our troubleshooting documentation](#)

Creamos nuestro cliente

```

]: try:
    client = pymongo.MongoClient('mongodb+srv://admin:cas@cluster0.juqfh.mongodb.net/myFirstDatabase?retryWrites=true&w=
    client.server_info()
    print("Conexion exitosa")
    client.close
except pymongo.errors.ServerSelectionTimeoutError as errorTiempo:
    print("Conexión rechazada")

```

Conexion exitosa

```

]: dbm = client['deberDatos']
   col = dbm['personasDatos']

```

Creamos nuestro script que nos permita generar las n personas con sus datos randomicos.

```

]:
cantidad = int(input("Ingrese la cantidad de personas"))

#cantidad = 20
tamanio = 10
age = np.arange(18,80)
edad =age.tolist()

for x in range(cantidad):
    aux = (''.join(random.choice(string.digits) for _ in range(tamanio)))
    cedula.append(aux)

for i in range (cantidad):
    col.insert_one({"Nombres":[random.choice(nombre),random.choice(nombre)],
    "Apellidos":[random.choice(apellido),random.choice(apellido)],
    "Cedula":[random.choice(cedula)],
    "Edad":[random.choice(edad)]})

for i in range (cantidad):
    doc={"Nombres":[random.choice(nombre),random.choice(nombre)],
    "Apellidos":[random.choice(apellido),random.choice(apellido)],
    "Cedula":[random.choice(cedula)],
    "Edad":[random.choice(edad)]}
    db.save(doc)

```

Ingrese la cantidad de personas 10

Procedemos a revisar del lado del servidor .

Y del lado de Compass.

Para poder realizar el ejercicio en CouchDB iniciamos con nuestras credenciales, creamos la base de datos a la que vamos a apuntar, utilizando los drivers necesarios y realizando la conexión con el cliente procedemos a ingresar los datos de las personas.

```
couch=couchdb.Server('http://admin:cas@127.0.0.1:5984')
```

```
db=couch['datos_personas']
```

```
cantidad = int(input("Ingrese la cantidad de personas"))

#cantidad = 20
tamanio = 10
age = np.arange(18,80)
edad =age.tolist()

for x in range(cantidad):
    aux = (''.join(random.choice(string.digits) for _ in range(tamanio)))
    cedula.append(aux)

for i in range (cantidad):
    doc={"Nombres":[random.choice(nombre),random.choice(nombre)],
        "Apellidos":[random.choice(apellido),random.choice(apellido)],
        "Cedula":[random.choice(cedula)],
        "Edad":[random.choice(edad)]}
    db.save(doc)
```

←

datos_personas

⋮

Document ID

Options

{ } JSON

All Documents

Run A Query with Mango

Permissions

Changes

Design Documents

Table

Metadata

{ } JSON

Create Document

	Apellidos	Cedula	Edad	Nombres	_id
<input type="checkbox"/>	["Cifuentes", "Escobar"]	["4005603024"]	[65]	["Segundo", "Juan"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Vargas", "Males"]	["9891979618"]	[61]	["Eduardo", "Tomas"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Vargas", "Mendez"]	["3085879573"]	[77]	["Luis", "Eduardo"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Mendez", "Mendez"]	["6743876565"]	[64]	["Segundo", "Samuel"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Mendez", "Males"]	["8626078075"]	[30]	["Gadiel", "Gadiel"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Andrade", "Quito"]	["3976293232"]	[60]	["Gadiel", "Juan"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Mendez", "Escobar"]	["4841077697"]	[58]	["Rene", "Gadiel"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Loreto", "Escobar"]	["7682567298"]	[66]	["Gadiel", "Rene"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Escobar", "Zambrano"]	["6133388731"]	[72]	["Samuel", "Juan"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Vargas", "Vargas"]	["1249571839"]	[55]	["Juan", "Eduardo"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Zambrano", "Vargas"]	["8029638718"]	[54]	["Rene", "Eduardo"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Males", "Cifuentes"]	["8008316228"]	[28]	["Juan", "Eduardo"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Escobar", "Andrade"]	["1249571839"]	[65]	["Luis", "Gadiel"]	e854690abd3da3e239f14ba05...
<input type="checkbox"/>	["Quito", "Andrade"]	["5644416738"]	[60]	["Rene", "Gadiel"]	e854690abd3da3e239f14ba05...

Showing 5 of 6 columns. ☐ Show all columns.

Showing document 1 - 20. Documents per page: 100