



Deber MongoDB

Materia: Análisis de datos

Integrantes:

- Auqui Manuel
- Chala Maria Jose
- Guambo Leoni
- Mendez Mayerli
- Ortiz Jorge

Dada una lista:

nombres=["Luis","Ana","Juan"...]

apellidos=["Zambrano", "Medina", "Arias"...]

cada lista tiene al menos 10 elementos.

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.

Pueden trabajar en los mismos grupos del trabajo en clase de CouchDB

Entregables: Capturas de pantalla con los pasos seguidos y código fuente (.ipynb o .py) en un repositorio de GitHub.

Capturas de pantalla

Código Fuente Localmente

```
jupyter MongoDBLocal Last Checkpoint: el sábado pasado a las 20:46 (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [6]: # Ejercicio en clases
# Auqui Manuel
# Chala Maria
# Guambo Leoni
# Mayerli Mendez
# Ortiz Jorge

import pymongo
from pymongo import MongoClient
client = MongoClient("localhost", 27017)
import random
dbs=client["ejercicio1"]
db=dbs.persona

In [7]: nombre=["Ricardo","Fernando","Luisa","Camila","Daniela","Vanessa","Karen","Cinthya","Stefany","Nicolas"]
apellido=["Auqui","Galarza","Mendez","Ortiz","Gallegos","Guambo","Naranjo","Chala","Cedeño","Flores"]
num=int(input("Ingrese el numero de personas: "))
for i in range(num):
    doc={"Nombre": [random.choice(nombre),random.choice(nombre)],
        "Apellido": [random.choice(apellido),random.choice(apellido)],
        "Edad": random.randint(18,80),
        "Cedula": random.randint(1700000000,1999999999)}
    db.insert_one(doc)

Ingrese el numero de personas: 4

In [19]: #Funcion
import pandas as pd
def mostrar_datos():
    for i in db.find():
        x=pd.Series(i)
        print(x)
mostrar_datos()

_id      61e3854542c23f5d2890f005
Nombre    [Stefany, Fernando]
Apellido    [Auqui, Cedeño]
Edad      69
Cedula    1719325192
dtype: object
_id      61e3854642c23f5d2890f006
Nombre    [Camila, Daniela]
Apellido    [Cedeño, Chala]
Edad      78
Cedula    1771442887
dtype: object
_id      61e3854642c23f5d2890f007
Nombre    [Vanessa, Nicolas]
Apellido    [Mendez, Mendez]
Edad      55
Cedula    1881267965
dtype: object
_id      61e3854642c23f5d2890f008
Nombre    [Karen, Fernando]
Apellido    [Naranjo, Auqui]
Edad      75
```

ESFOT
ESCUELA DE FORMACION
DE TECNOLOGOS



Código Fuente Nube

jupyter MongoDBNube Last Checkpoint: el sábado pasado a las 22:12 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel)

Run Code

```
In [12]: import pymongo
```

```
In [13]: pymongo.version
```

```
Out[13]: '4.0.1'
```

```
In [1]: pip install pymongo --user
```

Requirement already satisfied: pymongo in c:\users\cinthya\appdata\roaming\python\python310\site-packages (4.0.1)
Note: you may need to restart the kernel to use updated packages.

```
In [2]: pip install dnspython --user
```

Requirement already satisfied: dnspython in c:\users\cinthya\appdata\roaming\python\python310\site-packages (2.1.0)
Note: you may need to restart the kernel to use updated packages.

```
In [3]: import pymongo
```

```
my_client = pymongo.MongoClient(
    'mongodb+srv://admin:12345@cluster0.9gxx8.mongodb.net/myFirstDatabase?retryWrites=true&w=majority'
)

try:
    print("MongoDB version is %s" %
          my_client.server_info()['version'])
except pymongo.errors.OperationFailure as error:
    print(error)
    quit(1)
```

MongoDB version is 4.4.11

```
In [4]: my_database = my_client.test
```

```
In [5]: my_collection = my_database.ejercicio1
```

```
In [7]: import random
nombre=["Ricardo","Fernando","Luisa","Camila","Daniela","Vanessa","Karen","Cinthya","Stefany","Nicolas"]
apellido=["Auqui","Galarza","Mendez","Ortiz","Gallegos","Guambo","Naranjo","Chala","Cedeño","Flores"]
num=int(input("Ingrese el numero de personas: "))
for i in range(num):
    doc={"Nombre": [random.choice(nombre),random.choice(nombre)],
        "Apellido": [random.choice(apellido),random.choice(apellido)],
        "Edad": random.randint(18,80),
        "Cedula": random.randint(1700000000,1999999999)}
    my_collection.insert_one(doc)
```

Ingrese el numero de personas: 3

```
In [9]: #Funcion
import pandas as pd
def mostrar_datos():
    for i in my_collection.find():
        x=pd.Series(i)
        print(x)
mostrar_datos()
```

```
_id      61e62e6dcde5d68fabfca9
Nombre   [Ricardo, Daniela]
Apellido [Auqui, Gallegos]
Edad      31
Cedula    1856866842
dtype: object
_id      61e62e6dcde5d68fabfcaa
Nombre   [Nicolas, Camila]
Apellido [Naranjo, Gallegos]
Edad      30
Cedula    1881680490
dtype: object
_id      61e62e6dcde5d68fabfcab
Nombre   [Cinthya, Stefany]
Apellido [Naranjo, Chala]
Edad      26
Cedula    1942389587
dtype: object
```



Mongo DB (Atlas)

Project 0 Atlas Realm Charts

+ Create Database

NAMESPACES

test

ejercicio1

foods

test.ejercicio1

COLLECTION SIZE: 910B TOTAL DOCUMENTS: 7 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

FILTER { field: 'value' } OPTIONS Apply Reset

QUERY RESULTS 1-7 OF 7

```
{
  "_id": ObjectId("61e62e6dcde5d68fabfca9"),
  "Nombre": Array
    0: "Ricardo"
    1: "Daniela"
  "Apellido": Array
    0: "Auqui"
    1: "Gallegos"
  "Edad": 31
  "Cedula": 1856866842
}
```

```
{
  "_id": ObjectId("61e62e6dcde5d68fabfca0"),
  "Nombre": Array
    0: "Cinthya"
    1: "Stefany"
  "Apellido": Array
    0: "Naranjo"
    1: "Chala"
  "Edad": 26
  "Cedula": 1942389587
}
```

```
{
  "_id": ObjectId("61e62e6dcde5d68fabfcac"),
  "Nombre": Array
    0: "Stefany"
    1: "Camila"
  "Apellido": Array
    0: "Ortiz"
    1: "Flores"
  "Edad": 51
  "Cedula": 1786884750
}
```