

# PyMongo

## Programación desde Python - Bases de datos II

Alberto Díaz Álvarez (<alberto.díaz@upm.es>)

Departamento de Sistemas Informáticos

Escuela Técnica superior de Ingeniería de Sistemas Informáticos

License CC BY-NC-SA 4.0

# Acceso a una base de datos MongoDB

Importar la librería `pymongo` :

```
import pymongo
```

Conectar a la base de datos:

```
client = pymongo.MongoClient("mongodb://localhost:27017/")
```

Acceder a una base de datos:

```
db = client["people"]
```

Crear/acceder una colección

```
db = client["people"]
```

# Inserción de documentos

Insertar un documento:

```
user_data = {  
    "name": "Ana",  
    "address": "Calle Pez, 1"  
}  
x = users.insert_one(user_data)
```

Insertar varios documentos:

```
users_data = [  
    {"name": "Carlota", "address": "Calle Pez, 2"},  
    {"name": "Cui", "address": "Calle Pez, 3"},  
    {"name": "Irene", "address": "Calle Pez, 4"},  
    {"name": "Noa", "address": "Calle Pez, 5"},  
]  
x = users.insert_many(users_data)
```

# Consulta de documentos (I)

Consultar un único documento:

```
>>> x = users.find_one()
>>> print(x)
{'_id': ObjectId('7815696ecbf1c96e6894b772'), 'name': 'Noa', 'address': 'Calle Pez, 4'}
```

Consultar todos los documentos:

```
>>> for x in users.find():
>>>     print(x)
{'_id': ObjectId('3dad9cbf9baaa0360c0f2ba3'), 'name': 'Ana', 'address': 'Calle Pez, 1'}
{'_id': ObjectId('411cd305d84659e7479e5700'), 'name': 'Carlota', 'address': 'Calle Pez, 2'}
{'_id': ObjectId('753b799baa21c39e13ab6f11'), 'name': 'Cui', 'address': 'Calle Pez, 3'}
{'_id': ObjectId('a742b7c762a62c9b9b8d7e3a'), 'name': 'Irene', 'address': 'Calle Pez, 4'}
{'_id': ObjectId('74c6eb7c4384de00190f4a4a'), 'name': 'Noa', 'address': 'Calle Pez, 5'}
```

# Consulta de documentos (II)

Consultar un campo concreto:

```
>>> for x in users.find({}, {"_id": 0, "name": 1, "address": 1}):
>>>     print(x)
{'name': 'Ana', 'address': 'Calle Pez, 1'}
{'name': 'Carlota', 'address': 'Calle Pez, 2'}
{'name': 'Cui', 'address': 'Calle Pez, 3'}
{'name': 'Irene', 'address': 'Calle Pez, 4'}
{'name': 'Noa', 'address': 'Calle Pez, 5'}
```

Consultar un documento concreto:

```
>>> query = { "address": "Calle Pez, 4" }
>>> user = users.find(query)
>>> for x in user:
>>>     print(x)
{'_id': ObjectId('a742b7c762a62c9b9b8d7e3a'), 'name': 'Irene', 'address': 'Calle Pez, 4'}
```

# Consulta de documentos (III)

Consultar documentos con un filtro:

```
>>> query = { "address": { "$gt": "Calle Pez, 3" } }
>>> user = users.find(query)
>>> for x in user:
>>>     print(x)
{'_id': ObjectId('a742b7c762a62c9b9b8d7e3a'), 'name': 'Irene', 'address': 'Calle Pez, 4'}
{'_id': ObjectId('74c6eb7c4384de00190f4a4a'), 'name': 'Noa', 'address': 'Calle Pez, 5'}
```

Consultar documentos con un filtro y un límite:

```
>>> result = users.find().limit(2)
>>> for x in result:
>>>     print(x)
{'_id': ObjectId('3dad9cbf9baaa0360c0f2ba3'), 'name': 'Ana', 'address': 'Calle Pez, 1'}
{'_id': ObjectId('411cd305d84659e7479e5700'), 'name': 'Carlota', 'address': 'Calle Pez, 2'}
```

# Consulta de documentos (y IV)

Consultar documentos con orden:

```
>>> result = users.find().sort("name")
>>> for x in result:
>>>     print(x)
{'_id': ObjectId('3dad9cbf9baaa0360c0f2ba3'), 'name': 'Ana', 'address': 'Calle Pez, 1'}
{'_id': ObjectId('411cd305d84659e7479e5700'), 'name': 'Carlota', 'address': 'Calle Pez, 2'}
{'_id': ObjectId('753b799baa21c39e13ab6f11'), 'name': 'Cui', 'address': 'Calle Pez, 3'}
{'_id': ObjectId('a742b7c762a62c9b9b8d7e3a'), 'name': 'Irene', 'address': 'Calle Pez, 4'}
{'_id': ObjectId('74c6eb7c4384de00190f4a4a'), 'name': 'Noa', 'address': 'Calle Pez, 5'}
```

Consultar documentos con orden inverso:

```
>>> result = users.find().sort("name", -1)
>>> for x in result:
>>>     print(x)
{'_id': ObjectId('74c6eb7c4384de00190f4a4a'), 'name': 'Noa', 'address': 'Calle Pez, 5'}
{'_id': ObjectId('a742b7c762a62c9b9b8d7e3a'), 'name': 'Irene', 'address': 'Calle Pez, 4'}
{'_id': ObjectId('753b799baa21c39e13ab6f11'), 'name': 'Cui', 'address': 'Calle Pez, 3'}
{'_id': ObjectId('411cd305d84659e7479e5700'), 'name': 'Carlota', 'address': 'Calle Pez, 2'}
{'_id': ObjectId('3dad9cbf9baaa0360c0f2ba3'), 'name': 'Ana', 'address': 'Calle Pez, 1'}
```

# Actualización de documentos

Actualizar un documento:

```
>>> query = { "address": "Calle Pez, 3" }
>>> values = { "$set": { "name": "Che" } }
>>> users.update_one(query, values)
>>> query = { "name": "Che" }
>>> for x in users.find(query):
>>>     print(x)
{'_id': ObjectId('753b799baa21c39e13ab6f11'), 'name': 'Che', 'address': 'Calle Pez, 3'}
```

Actualizar varios documentos:

```
>>> query = { "address": { "$regex": ".*3" } }
>>> values = { "$set": { "name": "Che" } }
>>> x = users.update_many(query, values)
>>> print(f"Updated: {x.modified_count}")
Updated 1
```



# Eliminación de documentos y colecciones

Eliminar un documento:

```
>>> query = { "address": "Calle Pez, 3" }  
>>> users.delete_one(myquery)
```

Eliminar varios documentos:

```
>>> query = { "address": {"$regex": "^S"} }  
>>> x = users.delete_many(query)  
>>> print(f"Updated: {x.modified_count}")  
Updated 1
```

Eliminar una colección:

```
users.drop()
```

# Creación/borrado de índices

Podemos crear índices para mejorar el rendimiento de las consultas

```
users.create_index("name", unique=True)
```

Y borrarlos cuando nos demos cuenta de que no nos valían para nada:

```
users.drop_index("name")
```

# *Pipelines* de agregación

Operaciones en serie a aplicar sobre documentos de una colección.

```
>>> pipeline = [  
>>>     { "$match": { "address": "Calle Pez, 2" } },  
>>>     { "$group": { "_id": "$name", "count": { "$sum": 1 } } }  
>>> ]  
>>> result = users.aggregate(pipeline)  
>>> for x in result:  
>>>     print(x)  
{'_id': 'Carlota', 'count': 1}
```

**Gracias**