



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Programador Web Avanzado

Clase N° 5: Robo 3t

Profesor: Ing. Leandro Rodolfo Gil Carrano

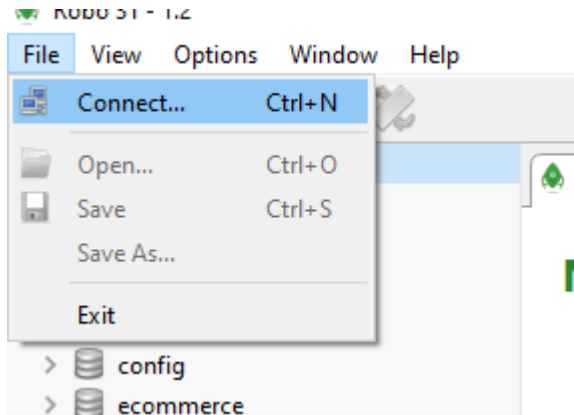
Email: leangilutn@gmail.com

Robo 3t

Node JS – Robo 3t

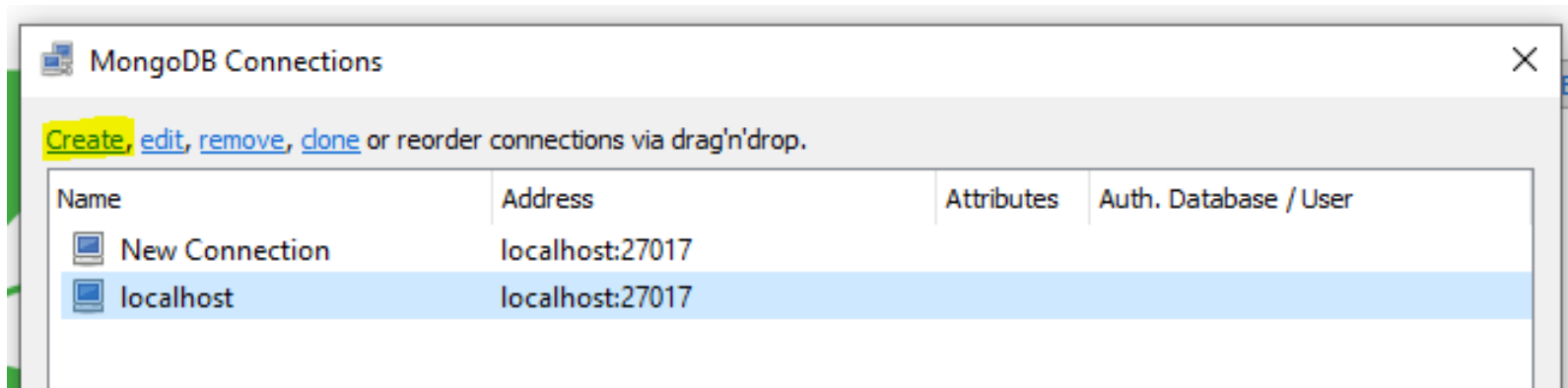
- Descargar desde <https://robomongo.org/download>
- Ubicarse (por cmd) en el directorio de Descargas y ejecutar:
sudo dpkg -i robomongo-0.8.5-x86_64.deb
- Abrir robomongo desde la terminal ejecutando:
robomongo

Node JS – Robo 3t



File -> Connect: agregan una conexión a una base de datos de mongo db

Node JS – Robo 3t



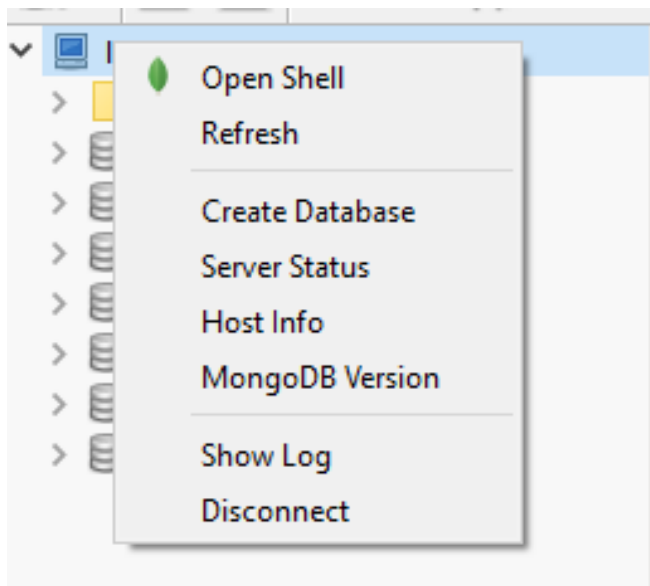
Creamos la nueva conexión



UTN.BA

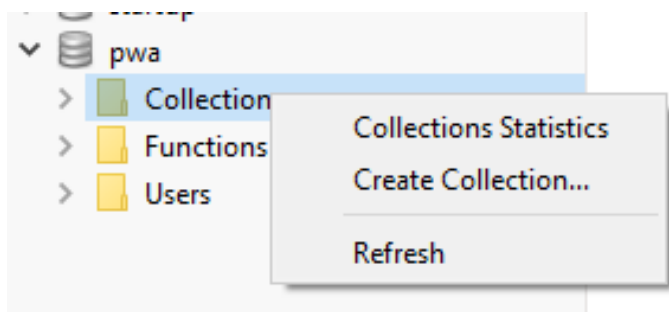
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Robo 3t



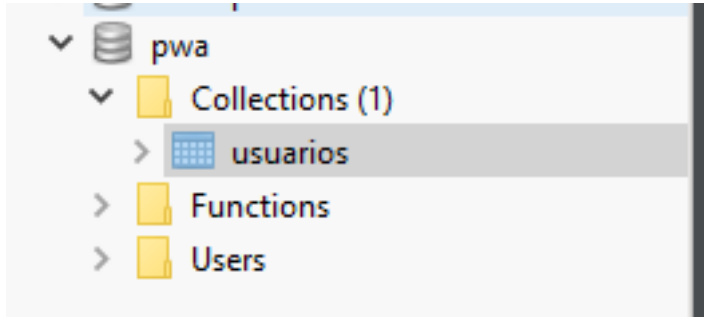
Click derecho en la conexión ->
Create database (creamos una nueva Base de datos)

Desplegamos la información de la base De datos.



Click derecho sobre “Collection” ->
Create Collection (creamos una colección)

Node JS – Robo 3t (leer documentos)



Doble click sobre la colección nos muestra todos los documentos de la misma.

Ejecuta comando:

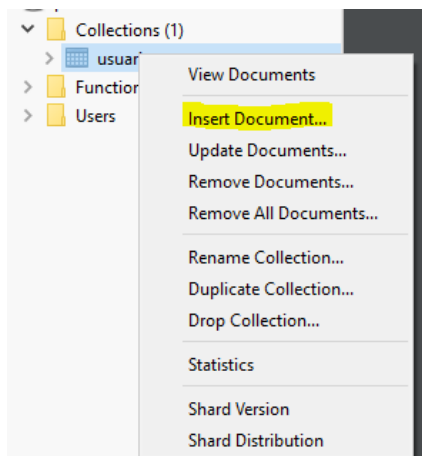
```
db.getCollection('usuarios').find({})
```



UTN.BA

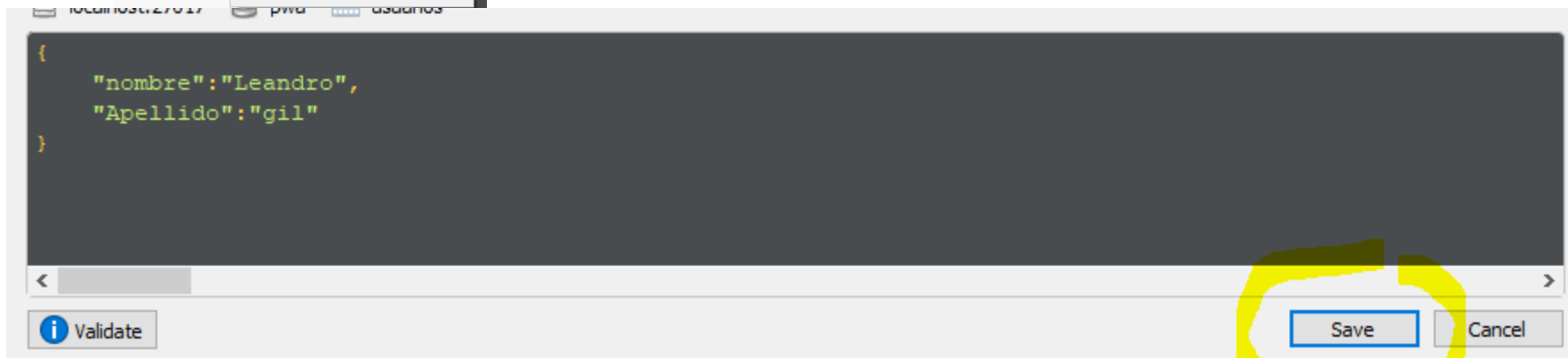
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Robo 3t (Insertar documentos)



Click derecho sobre la colección ->
Insert Document.

Permite insertar documentos a la
colección



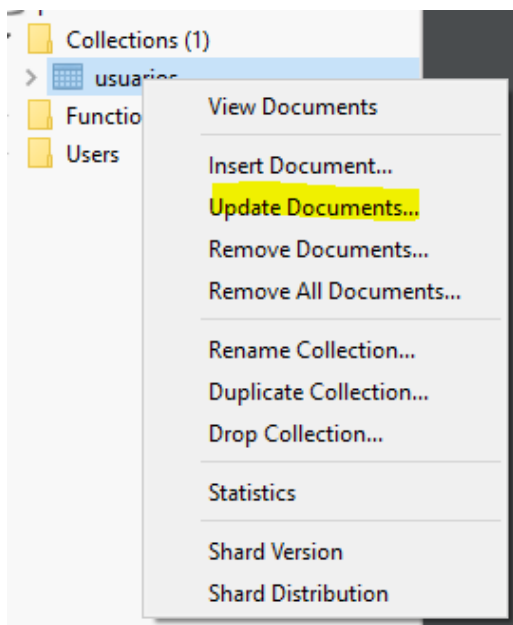
El documento a insertar es un json. Los documentos de una colección puede tener distinta estructura



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Robo 3t (Actualizar documentos)



Click derecho sobre la colección ->
Update Document.



UTN.BA

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

Node JS – Robo 3t (Actualizar documentos)

```
db.collection('usuarios').update(  
  // query  
  {  
    "key" : "value"  
  },  
  
  // update  
  {  
  },  
  
  // options  
  {  
    "multi" : false, // update only one document  
    "upsert" : false // insert a new document, if no existing document match the query  
  }  
);
```

El update se divide en tres partes:

- Query: Se especifican las condiciones para la actualización (Where)
- Update: Los datos a actualizar en el documento (set)
- Options:
 - Multi: True->Actualiza todos los documentos que cumplan con la condición.
False -> Solo actualiza 1 documento.
 - Upsert: Si el documento no existe se crea uno nuevo (on duplicate key)

Node JS – Robo 3t (Actualizar documentos)

Al actualizar un documento se actualiza completo, es decir no se actualizan solos los campos especificados en el “update”.

Para actualizar solo algunos campos se debe utilizar **\$set**:

```
db.getCollection('usuarios').update(  
  // query  
  {  
    "_id":ObjectId("5d77066530cd71ec5f5346c6")  
  },  
  
  // update  
  {  
    $set:{  
      "nombre":"Leandro R"  
    }  
  },  
  
  // options  
  {  
    "multi" : false, // update only one document  
    "upsert" : false // insert a new document, if no existing document match the query  
  }  
)
```

Node JS – Robo 3t (Eliminar documento)

```
db.getCollection('usuarios').remove({ '_id' : ObjectId("5d77066530cd71ec5f5346c6") });
```

0.001 sec.

Removed 1 record(s) in 2ms

También se puede eliminar por otra condición, al igual que find si en el json del primer parámetro especificamos varios elementos se establece un and:

```
{  
  "nombre": "Leandro",  
  "apellido": "Gil"  
}
```

Elimina todos los registros con nombre Leandro y apellido Gil

Node JS – Robo 3t

Video en youtube:

https://www.youtube.com/playlist?list=PLhA16iIUESxk1M_I4z0vLZNhi0WByosTK