Escuela Profesional de
Ciencia de la Computación

BDI Fase 3

# Base de Datos I

Dr. Edward Hinojosa C.
Dr. Edgar Sarmiento C.

Universidad Nacional de San Agustín de Arequipa

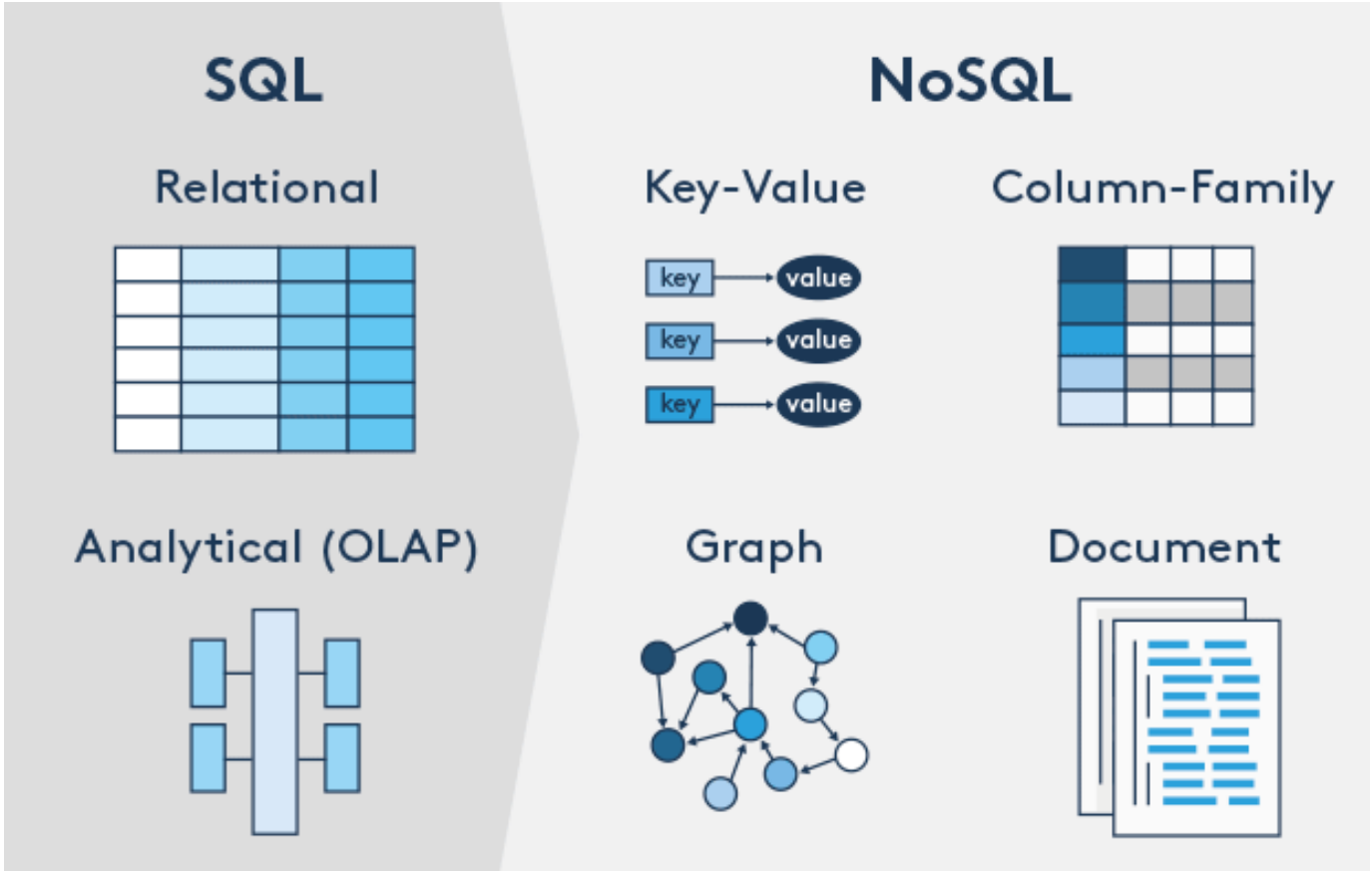
2020/Semestre Par

# Índice



ehinojosa@unsa.edu.pe

# Tipos de Bases de Datos NoSQL




ehinojosa@unsa.edu.pe

# BD NoSQL Orientados a Documentos

- Las Bases de Datos Orientadas a Documentos (BDOD) utilizan el concepto de datos y documentos **autocontenidos y autodescriptivos**.
- Esto implica que el documento en sí define cómo debe presentarse y cuál es el significado de los datos almacenados en su estructura.

ehinojosa@unsa.edu.pe

# BD NoSQL Orientados a Documentos

- Una BDOD tiene la característica de contener:
  - Toda la información importante en un solo documento.
  - Estar libre de esquemas.
  - Contar con identificadores universales únicos (UUID), posibilitando la consulta de documentos a través de métodos avanzados de agrupamiento y filtrado (MapReduce).
  - Permitir redundancia e inconsistencia.

ehinojosa@unsa.edu.pe

# BD NoSQL Orientados a Documentos

## Document Database




ehinojosa@unsa.edu.pe

# MongoDB

- MongoDB es una base de datos orientada a documentos que comenzó en 2007 pero se completó en el 2009.


ehinojosa@unsa.edu.pe

# MongoDB

- MongoDB tiene la característica de ser open source licenciada por la GNU AGPL (Affero General Public License) hsata versión 3.0.
- Actualmente MongoDB continua siendo de código abierto, aunque, para las ediciones empresariales, debemos pagar la licencia.
- Se considera de alto rendimento.
- No contiene esquemas.
- Escrita en C++.
- Es Multiplataforma. MongoDB tiene binarios para varias plataformas como Windows, Mac OS X, Linux y Solaris.
- Utiliza el lenguaje JSON.
- Más características en `http://mongodb.com/`

ehinojosa@unsa.edu.pe

# MongoDB

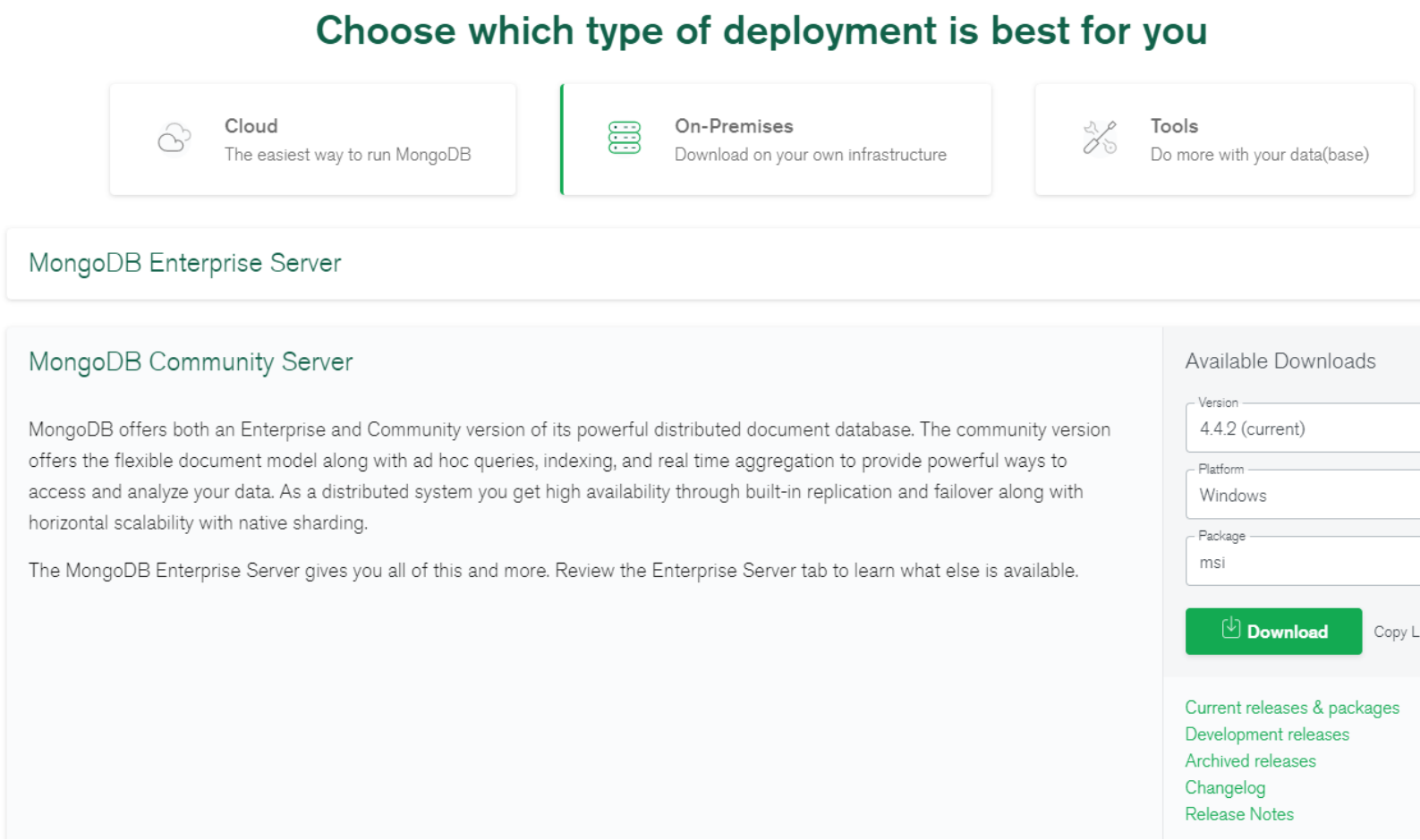
- Entre las empresas que ya utilizan MongoDB se encuentran:
  - Disney
  - SAP
  - The New York Times
  - The Guardian
  - Viber
  - ...

ehinojosa@unsa.edu.pe

# MongoDB

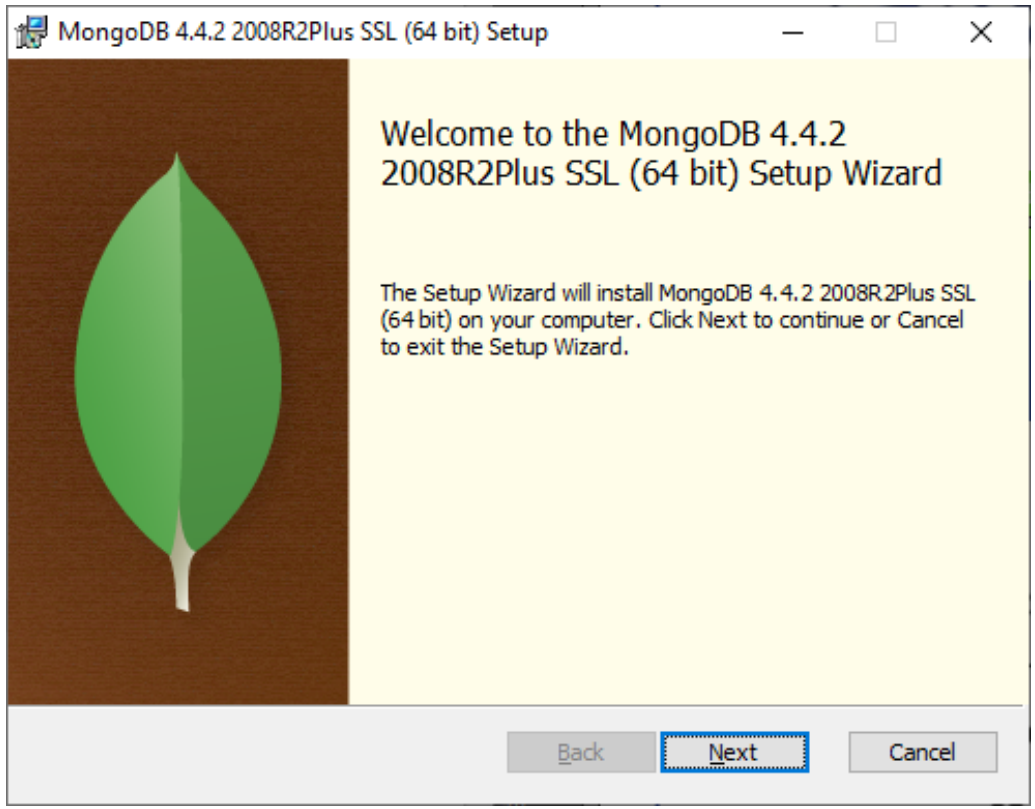

ehinojosa@unsa.edu.pe

# Instalando MongoDB




ehinojosa@unsa.edu.pe

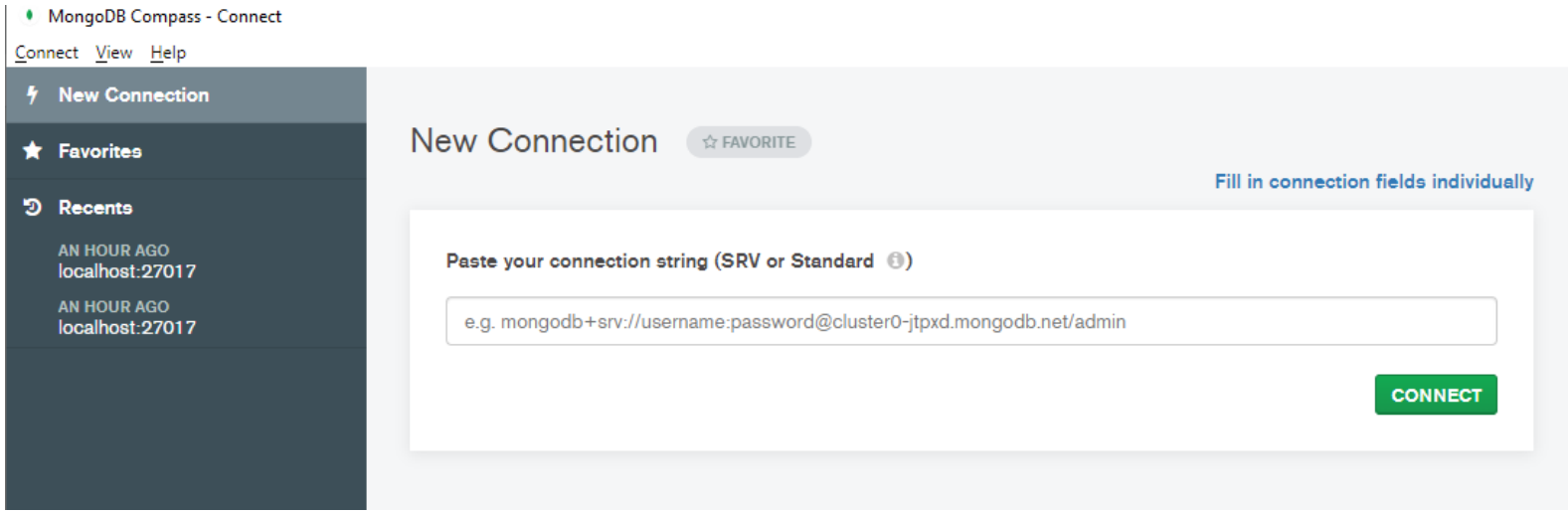
# Instalando MongoDB




ehinojosa@unsa.edu.pe

# Instalando MongoDB

- Utilizaremos MongoDB Compass. La interfaz gráfica de usuario de MongoDB.




ehinojosa@unsa.edu.pe

# Ejemplo - Blog

- Cada blog posee un enlace único que se utiliza para ser accesado.
- Cada blog posee un autor y fecha de creación.
- Cada blog posee un título y un contenido.
- Cada blog posee categorías y un tag.
- Un blog puede tener varios comentarios. De cada comentario se almacena el nombre de quien realizó el comentario, fecha y comentario.

`https://www.consumidor.ftc.gov/blog/2020/04/cheques-por-el-coronavirus-aplanemos-la-curva-de-las-esta`

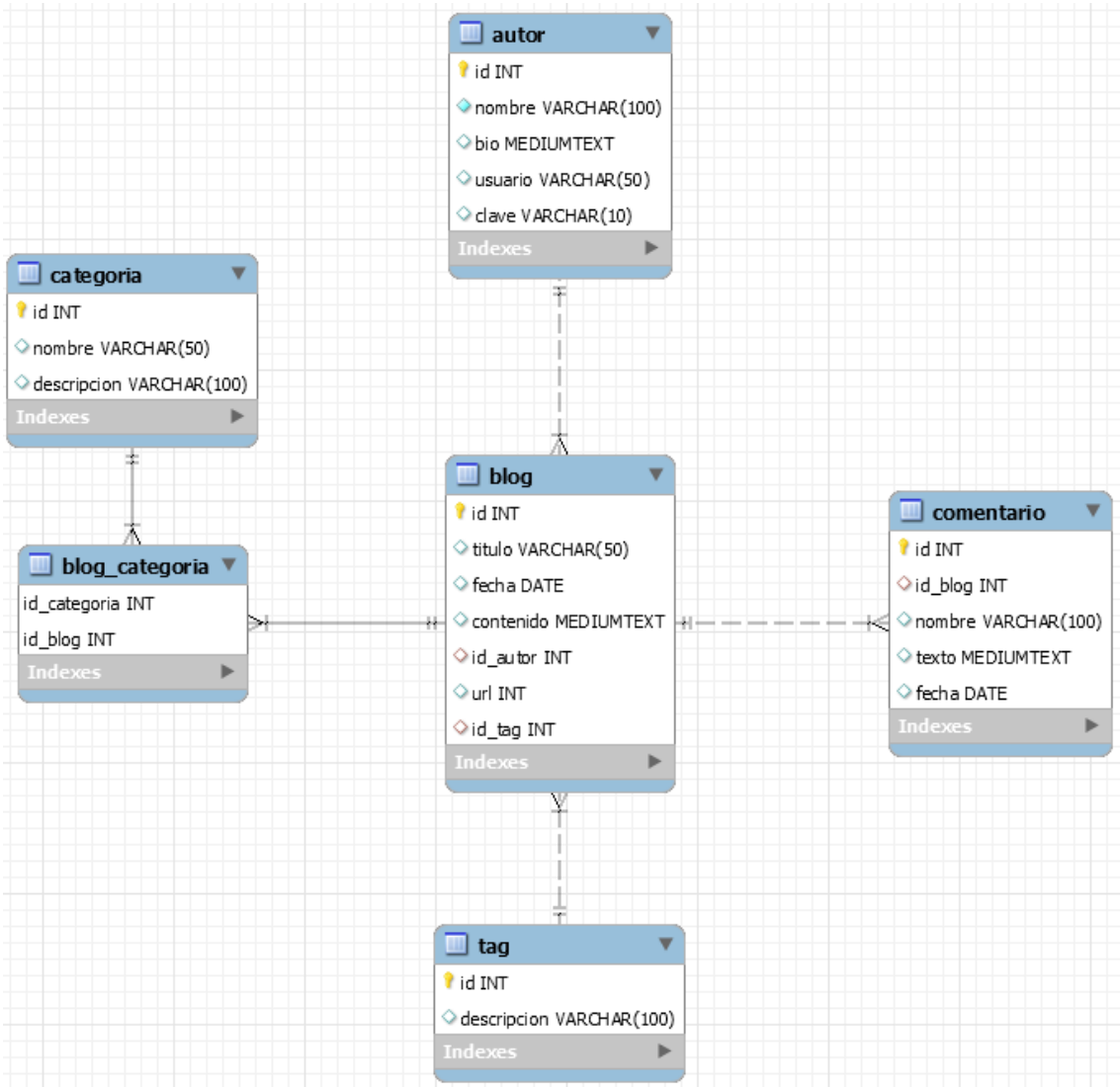
ehinojosa@unsa.edu.pe

# Blog - Modelo Relacional

- Blog con un FK con Autor.
- Autor
- Categoria
- BlogCategoria: Tabla Intermedia, FK con Categoría y Blog
- Tag: FK con Artículo
- Comentarios: FK con Blog

ehinojosa@unsa.edu.pe

# Blog - Modelo Relacional




ehinojosa@unsa.edu.pe

# Blog - Modelo Relacional

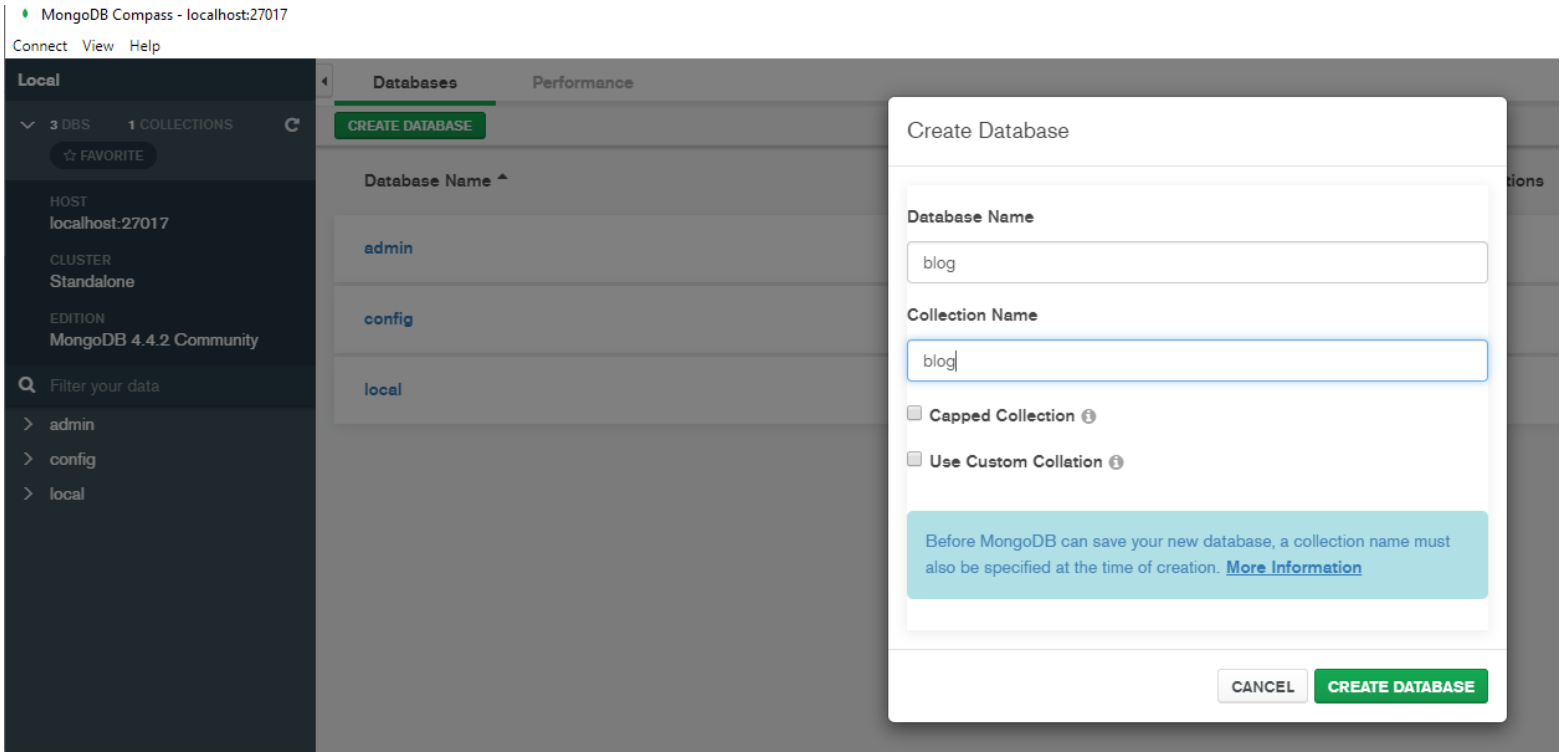
- Para poder mostrar el blog debo realizar la siguiente consulta (se debe utilizar varios JOINS):

```
SELECT blog.*, autor.*
....
INNER JOIN autor ....
WHERE URL='XX'
SELECT categoria.*
....
INNER JOIN blog_categoria ....
SELECT categoria.*
.....
SELECT comentario.*
.....
```

ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

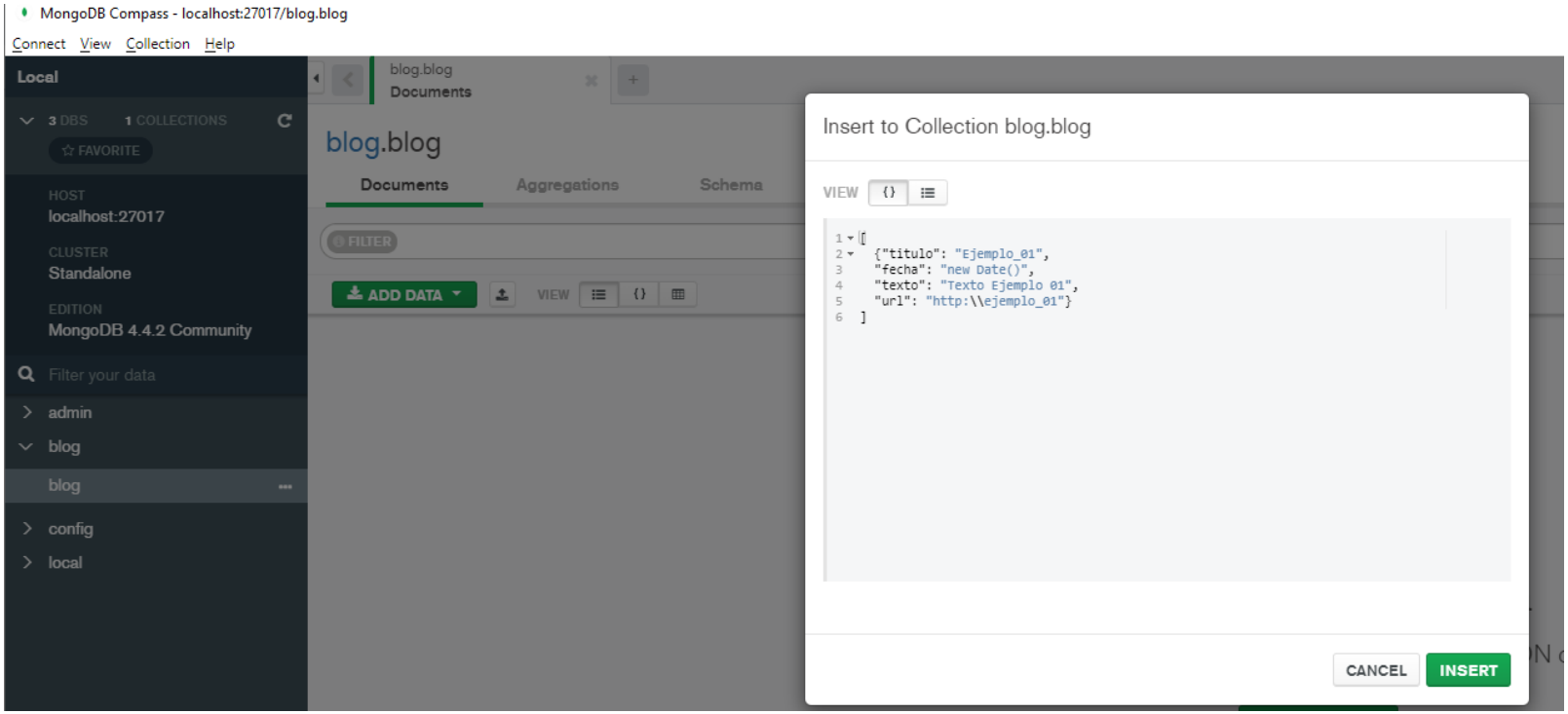
- Creamos la base de datos y la primera colección de documentos:




ehinojosa@unsa.edu.pe

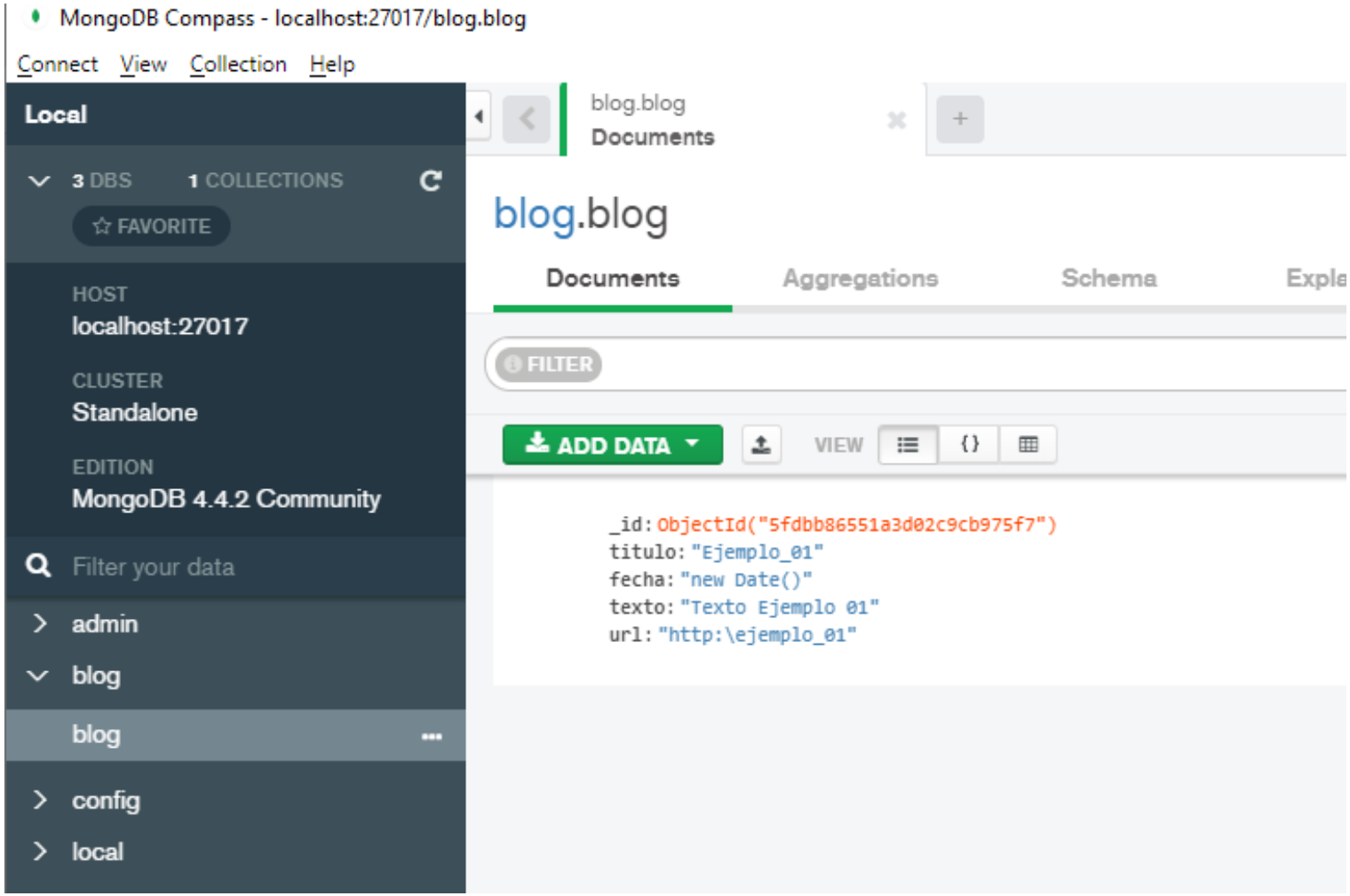
# Blog - Modelo no Relacional

- Insertamos un nuevo documento (en este caso un blog):




ehinojosa@unsa.edu.pe

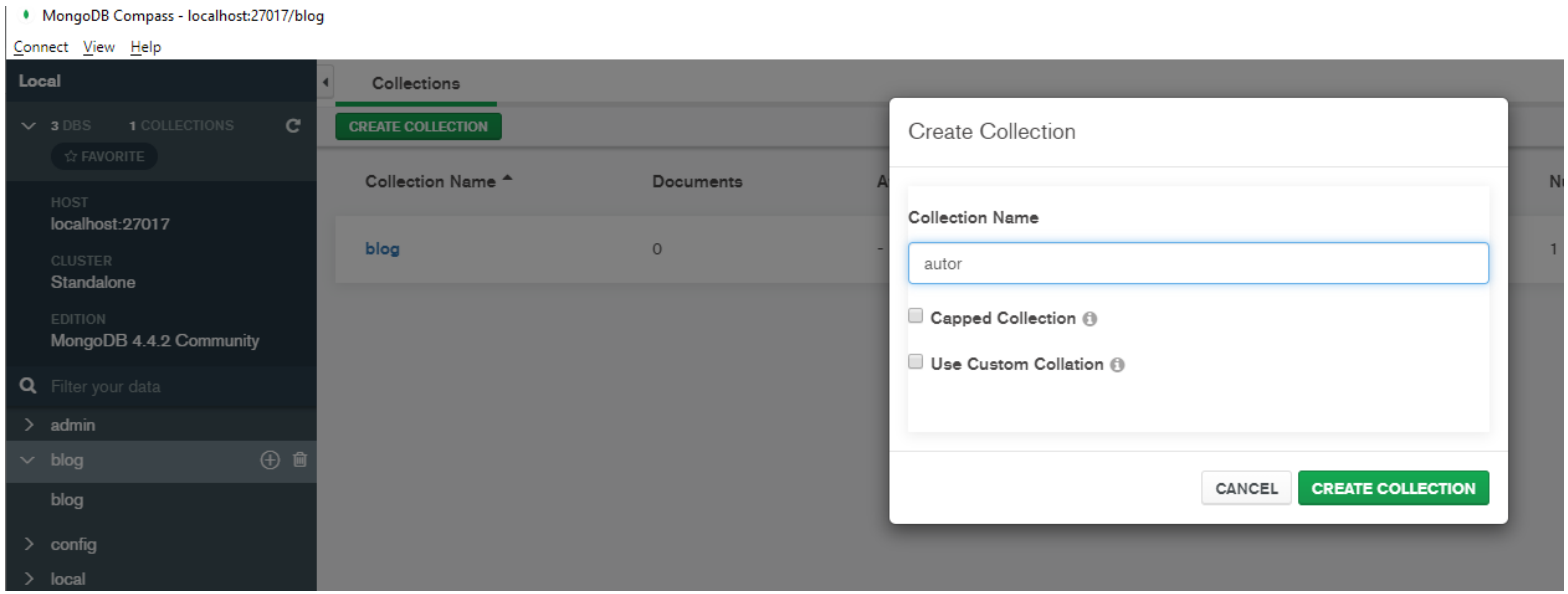
# Blog - Modelo no Relacional




ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

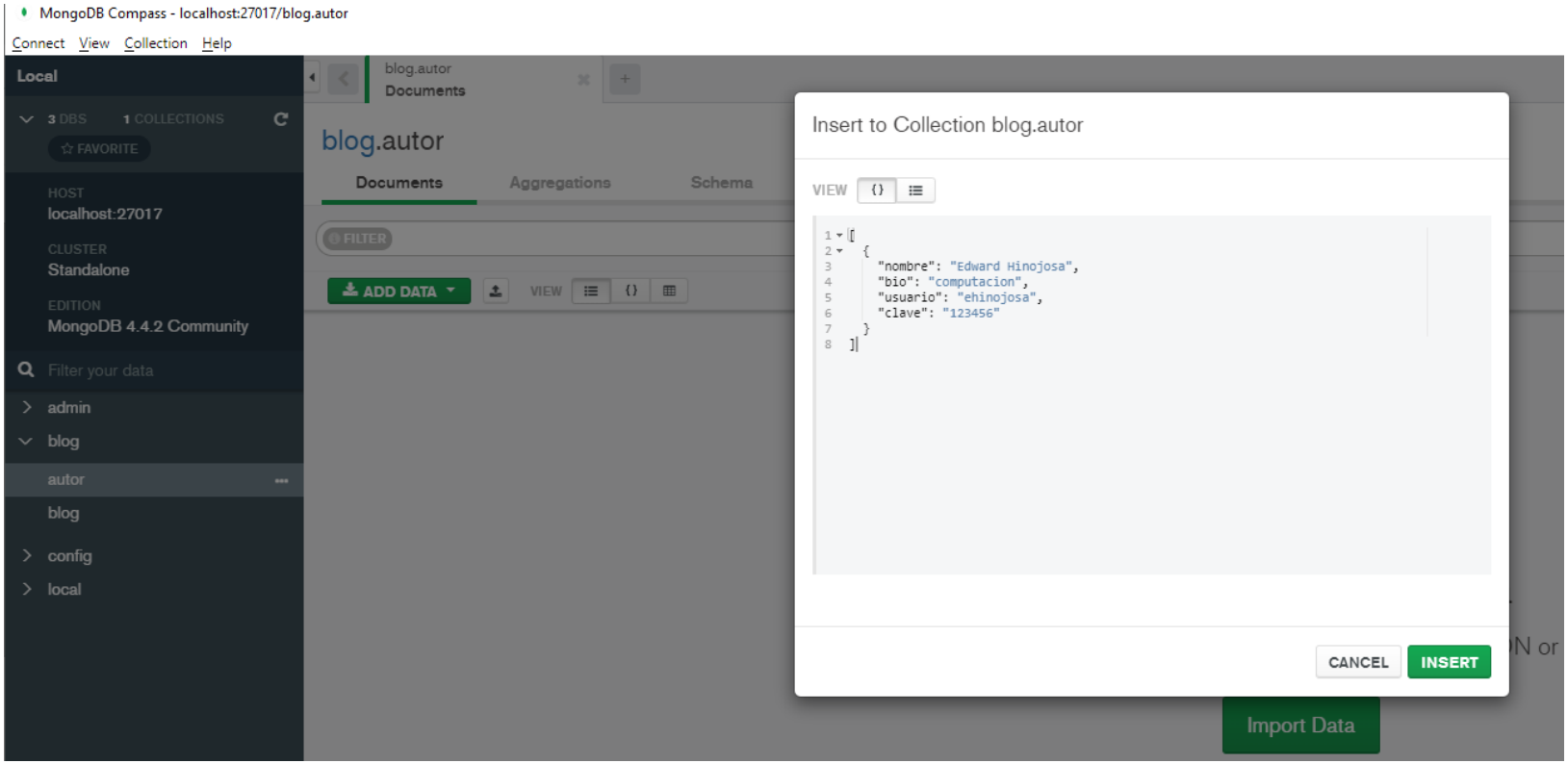
- Creamos la colección autor:




ehinojosa@unsa.edu.pe

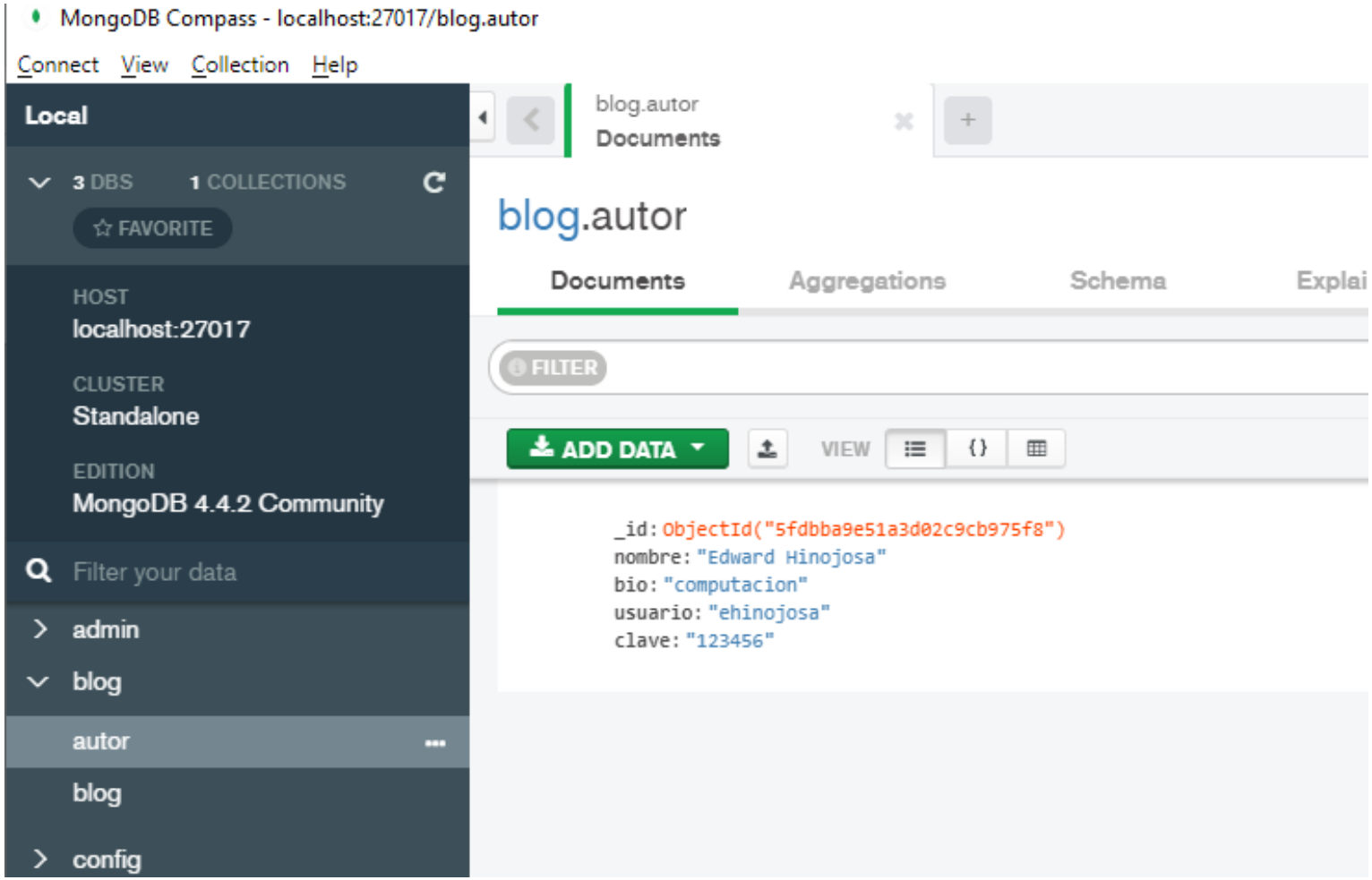
# Blog - Modelo no Relacional

- Insertamos un nuevo autor:




ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional




ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

- Si quisieramos asignar el anterior autor al blog anterior pensariamos en modificar el documento de blog adicionando solo el código generado anteriormente.
- En este caso no solo debemos adicionar le código, sino también lo que deseamos que aparezca en el blog (solo el nombre, no se necesita el usuario y contraseña). Recordemos, cada documento debe ser autocontenido y autodescriptivo.

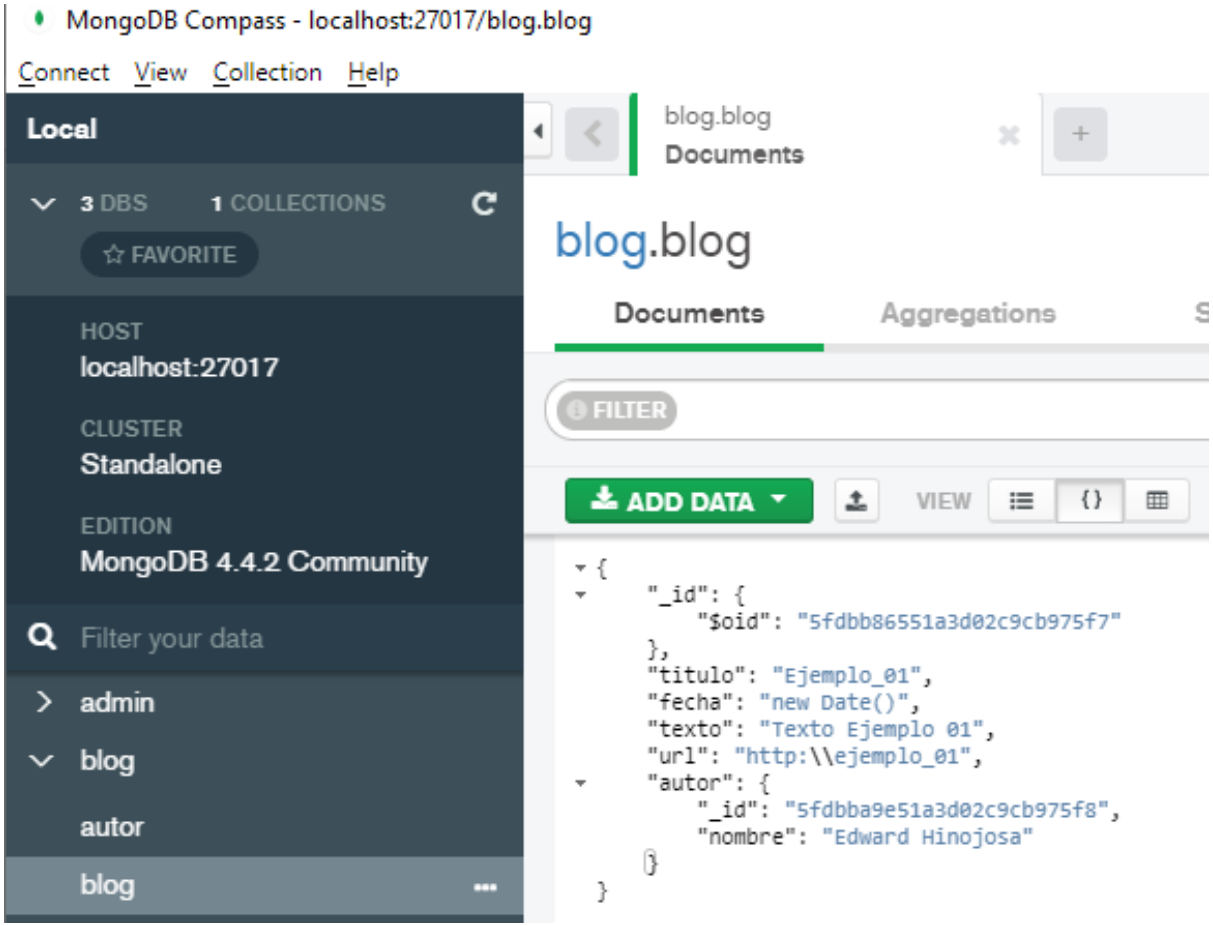
ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

- Con ello en una consulta para mostrar el blog, no necesitamos realizar un join. Debemos pensar que la consulta sea lo más simple posible.
- Desventaja: En caso el nombre cambie, debemos cambiarlo en todos los documentos.

ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional




ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

- En el caso del tag del blog lo podemos adicionar como un nuevo campo.

```
{
    "_id": {
        "$oid": "5fdbb86551a3d02c9cb975f7"
    },
    "titulo": "Ejemplo_01",
    "fecha": "new Date()",
    "texto": "Texto Ejemplo 01",
    "url": "http:\\ejemplo_01",
    "autor": {
        "_id": "5fdbba9e51a3d02c9cb975f8",
        "nombre": "Edward Hinojosa"
    },
    "tag": "tag 01"
}
```

ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

- En el caso de categorías podemos adicionar un array de campos.
- Con ello evitamos la tabla intermedia.

```
{
    "_id": {
        "$oid": "5fdbb86551a3d02c9cb975f7"
    },
    "titulo": "Ejemplo_01",
    "fecha": "new Date()",
    "texto": "Texto Ejemplo 01",
    "url": "http:\\ejemplo_01",
    "autor": {
        "_id": "5fdbba9e51a3d02c9cb975f8",
        "nombre": "Edward Hinojosa"
    },
    "tag": "tag 01",
    "categorias": [
        {"nombre": "categoria 01"},
        {"nombre": "categoria 02"},
        {"nombre": "categoria 03"}
      ]
}
```

ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

- Por último, los comentarios se pueden adicionar como las categorías, considerando más campos para cada array.


```
{
    "_id": {
        "$oid": "5fdbb86551a3d02c9cb975f7"
    },
    "titulo": "Ejemplo_01",
    "fecha": "new Date()",
    "texto": "Texto Ejemplo 01",
    "url": "http:\\ejemplo_01",
    "autor": {
        "_id": "5fdbba9e51a3d02c9cb975f8",
        "nombre": "Edward Hinojosa"
    },
    "tag": "tag 01",
    "categorias": [
        {"nombre": "categoria 01"},
        {"nombre": "categoria 02"},
        {"nombre": "categoria 03"}
      ],
      "comentarios": [
        {
          "nombre": "Edgar Sarmiento",
          "texto": "Buen blog",
          "fecha": "2018-11-06T:15:24:14.047Z"
        },
        {
          "nombre": "Juan Carlos Gutierrez",
          "texto": "Interesante blog",
          "fecha": "2018-11-10T:18:50:17.024Z"
        }
      ]
}
```


ehinojosa@unsa.edu.pe

# Blog - Modelo no Relacional

- Con ello podemos mostrar todo el blog con una consulta simple. Ese es el objetivo principal.
- Se pueden realizar operaciones de inserción, actualización y eliminación (no las veremos) mediante comandos.
- Las operaciones mencionadas son abtraidas de los programadores.

ehinojosa@unsa.edu.pe

# ¡GRACIAS!

ehinojosa@unsa.edu.pe