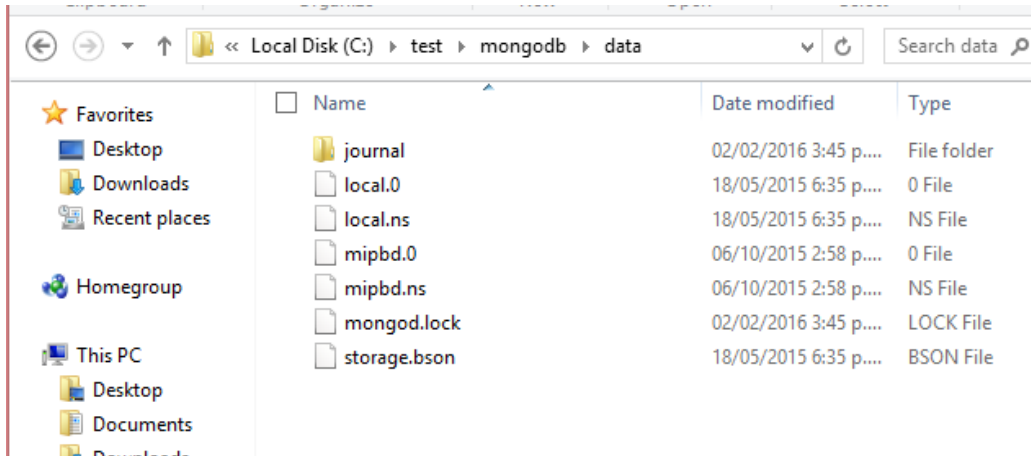
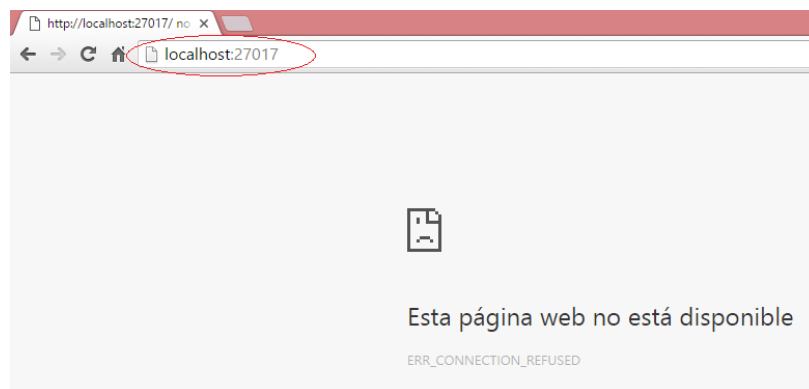


La instalación de MongoDB es muy sencilla, usted debe crear una carpeta en el **c:** llamada **C:\mongodb** y debe extraer todo aquí de modo que la carpeta **bin** esté en **C:\mongodb\bin**.

MongoDB necesita un directorio donde guardar los datos por defecto, este directorio hay que crearlo y su ruta es, **C:\test\mongodb\data**



Este es el directorio que se usara como repositorio de las colecciones de la base de datos a implementar.



Para verificar que el servicio se encuentre disponible, realice un test en su browser al puerto **27017**.

Localhost:27017

Se debe subir el servicio de mongo relacionando al sitio de almacenamiento.

Ahora como no está disponible nos re direccionamos a la carpeta:

C:\mongodb\bin

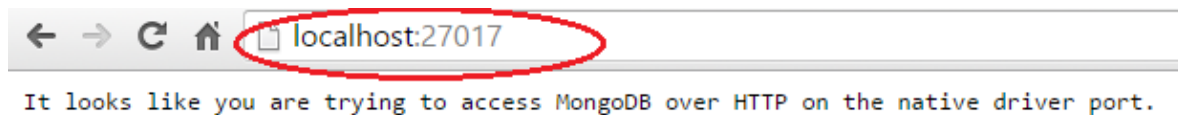
Donde usamos el comando:

mongod.exe --dbpath c:\test\mongodb\data



```
Administrator: Command Prompt - mongod.exe --dbpath c:\test\mongodb\data
C:\mongodb\bin>mongod.exe --dbpath c:\test\mongodb\data
2016-02-16T14:14:07.431-0500 I JOURNAL [initandlisten] journal dir=c:\test\mongodb\data\journal
2016-02-16T14:14:07.436-0500 I JOURNAL [initandlisten] recover: no journal files present, no recovery needed
2016-02-16T14:14:07.521-0500 I JOURNAL [durability] Durability thread started
2016-02-16T14:14:07.523-0500 I JOURNAL [journal writer] Journal writer thread started
2016-02-16T14:14:07.732-0500 I CONTROL [initandlisten] MongoDB starting : pid=2644 port=27017 dbpath=c:\test\mongodb\data 6
2016-02-16T14:14:07.732-0500 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2016-02-16T14:14:07.733-0500 I CONTROL [initandlisten] db version v3.0.3
2016-02-16T14:14:07.733-0500 I CONTROL [initandlisten] git version: b40106b36eed1b4407eb1ad1af6bc60593c6105
2016-02-16T14:14:07.734-0500 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1m-fips 19 Mar 2015
2016-02-16T14:14:07.734-0500 I CONTROL [initandlisten] build info: windows sys.getwindowsversion(major=6, minor=1, build=76
VERSION=1.49
2016-02-16T14:14:07.735-0500 I CONTROL [initandlisten] allocator: system
2016-02-16T14:14:07.735-0500 I CONTROL [initandlisten] options: { storage: { dbPath: "c:\test\mongodb\data" } }
2016-02-16T14:14:07.802-0500 I NETWORK [initandlisten] waiting for connections on port 27017
```

Ejecuta de nuevo el test y vera que el servicio está escuchando por el puerto **27017**.




MongoDB

Los datos en *MongoDB* tienen un esquema flexible, a diferencia de las bases de datos SQL, en el que debe determinar y declarar una tabla de esquema antes de realizar la inserción de datos, por lo contrario las colecciones en *MongoDB* no hacen cumplir la estructura del documento, esta flexibilidad facilita la asignación de documentos a una entidad o un objeto, cada documento puede coincidir con los campos de datos de la entidad representada, incluso si los datos tienen una variación sustancial. En la práctica, sin embargo, los documentos en una colección comparten una estructura similar.

Recuerde:

Que el desafío clave en el modelado de datos es equilibrar las necesidades de la aplicación, las características de rendimiento del motor de base de datos y los patrones de recuperación de datos. En el diseño de modelos de datos, siempre tenga en cuenta el uso de las aplicaciones de los datos (es decir, consultas, actualizaciones, y el procesamiento de los datos), así como la estructura inherente de los propios datos.

	<p>Servicio Nacional de Aprendizaje SENA CEAI - Centro de Electricidad y Automatización Industrial – Regional Valle Instructor : JAIR ROA VELÁSQUEZ</p>	<p>Fecha: Febrero de 2016 Página 3 de 5</p>
---	---	---

Estructura del documento

La decisión clave en el diseño de modelos de datos para aplicaciones de MongoDB gira en torno a la estructura de los documentos y cómo la aplicación representa relaciones entre los datos. Hay dos herramientas que permiten a las aplicaciones representar estas relaciones: las referencias y documentos incrustados.

Referencias

Referencias almacenan de relaciones entre los datos mediante la inclusión de enlaces o referencias de un documento a otro para tener acceso a los datos relacionados. En términos generales, estos son los modelos de datos normalizados.

Valores asociados

Documentos embebidos o Incrustados que capturan las relaciones entre documentos de datos mediante el almacenamiento de data relacionada en una sola estructura del documento.

MongoDB hace posible incrustar estructuras de documento en un campo o matriz dentro de un documento. Todo esto des normalizado y los modelos de datos permiten que las aplicaciones puedan recuperar y manipular los datos relacionados en una sola operación de base de datos, evitando los Join y obsequiando mayor rendimiento al proceso en sus tiempos de ejecución.

Gráfica 1

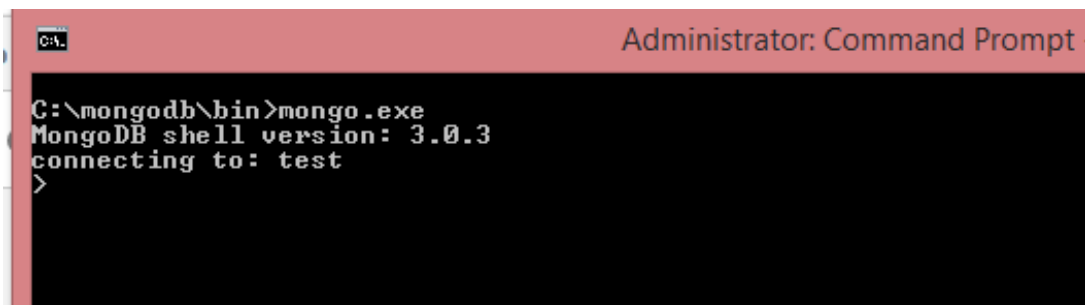
```
{
  "_id": "ObjectId1",
  "empleado": "rodrigo Garcia Perez",
  "contacto": {
    "telefono": "3206877636",
    "email": "rodrigo.garcia@siesa.com"
  },
  "acceso": {
    "nivel": "5",
    "grupo": "desarrollo"
  }
}
```

} Subdocumento Embebido

} Subdocumento Embebido

Ahora en una ventana aparte de cmd usamos el comando:

Mongo.exe



```
C:\mongodb\bin>mongo.exe
MongoDB shell version: 3.0.3
connecting to: test
>
```



```
connecting to: test
> use empleados
switched to db empleados
>
```

Nuestra colección se llama empleados.

```
C:\mongodb\bin>mongo.exe
MongoDB shell version: 3.0.3
connecting to: test
> use empleados
switched to db empleados
> a = {nombre : "OSCAR RENE"}
< "nombre" : "OSCAR RENE" >
> b = {edad : 30}
< "edad" : 30 >
>
```

Se inicia la creación de un documento con sus respectivos campos y datos.

```
C:\mongodb\bin>mongo.exe
MongoDB shell version: 3.0.3
connecting to: test
> use empleados
switched to db empleados
> a = {nombre : "OSCAR RENE"}
< "nombre" : "OSCAR RENE" >
> b = {edad : 30}
< "edad" : 30 >
> show dbs
local 0.078GB
mipbd 0.078GB
>
```

Aun no se ve reflejada nuestra colección empleados


Una vez insertamos los datos en el documento, nuestra colección se verá reflejada.

```
C:\mongodb\bin>mongo.exe
MongoDB shell version: 3.0.3
connecting to: test
> use empleados
switched to db empleados
> a = {nombre : "OSCAR RENE"}
< "nombre" : "OSCAR RENE" >
> b = {edad : 30}
< "edad" : 30 >
> show dbs
local 0.078GB
mipbd 0.078GB
> db.empleados.insert(a)
WriteResult<{ "nInserted" : 1 }>
> db.empleados.insert(b)
WriteResult<{ "nInserted" : 1 }>
> show dbs
empleados 0.078GB
local 0.078GB
mipbd 0.078GB
>
```

Si desea ver las colecciones que posee el comando es:

Show collections

```
> show collections
empleados
system.indexes
>
```

	<p>Servicio Nacional de Aprendizaje SENA CEAI - Centro de Electricidad y Automatización Industrial – Regional Valle Instructor : JAIR ROA VELÁSQUEZ</p>	<p>Fecha: Febrero de 2016 Página 5 de 5</p>
---	---	---

Aquí podemos observar que tenemos dos documentos y cada uno identificado con su propio OID.

```
> db.empleados.find()
{ "_id" : ObjectId("56c386db7d6e3e29735cfbc7"), "nombre" : "OSCAR RENE" }
{ "_id" : ObjectId("56c386e47d6e3e29735cfbc8"), "edad" : 30 }
```

Ahora insertemos el documento del objeto de la [Gráfica 1](#).

```
> db.empleados.find()
{ "_id" : ObjectId("56c386db7d6e3e29735cfbc7"), "nombre" : "OSCAR RENE" }
{ "_id" : ObjectId("56c386e47d6e3e29735cfbc8"), "edad" : 30 }
> documento = {id : 'obj1', empleado : "Rodrigo Garcia Perez", contacto : {telefono : "3206877636", email : "rodrigo.garcia@siesa.com"}, acceso : { nivel : "5", grupo : "desarrollo" }}
> db.empleados.insert(documento)
WriteResult({ "nInserted" : 1 })
```

Se observa la diferencia que existe entre la primera inserción y la segunda ya que en la primera tenemos datos aislados conformando documento propios (no está bien), en la segunda inserción ya podemos ver un documento real con sus subdocumentos embebidos.

```
> db.empleados.insert(documento)
WriteResult({ "nInserted" : 1 })
> db.empleados.find()
{ "_id" : ObjectId("56c386db7d6e3e29735cfbc7"), "nombre" : "OSCAR RENE" }
{ "_id" : ObjectId("56c386e47d6e3e29735cfbc8"), "edad" : 30 }
{ "_id" : ObjectId("56c38a3b7d6e3e29735cfbc9"), "id" : "obj1", "empleado" : "Rodrigo Garcia Perez", "contacto" : { "telefono" : "3206877636", "email" : "rodrigo.garcia@siesa.com"}, "acceso" : { "nivel" : "5", "grupo" : "desarrollo" } }
```

Actividad

Usted debe de crear una colección de datos llamada **wiki** donde grabara documentos (mínimo 20) que hagan referencia a los criterios de auto aprendizaje del proceso de capacitación de buenas prácticas, recuerde la socialización que se realizó con el ejemplo de la impresora LX-300.