

Una vez introducidos los conceptos teóricos y agnósticos de los índices, se presenta detenidamente el soporte de índices proporcionado por **ArangoDB**.

La lección comienza presentando qué soporte proporciona **ArangoDB** al respecto de los índices. Y a continuación, se muestra cómo crear, listar y suprimir índices mediante el *shell* y la interfaz web.

Al finalizar la lección, el estudiante sabrá:

- Cómo crear, listar y suprimir índices mediante el *shell* y la interfaz web de **ArangoDB**.
- Qué índices soporta **ArangoDB**.

Introducción

ArangoDB soporta los índices como sigue:

- Soporta índices principales y secundarios.
- Soporta varios métodos de indexación: índices de huella, índices por saltos, persistentes y otros tipos que quedan fuera del ámbito de esta lección y/o curso.
- Soporta claves de indexación, pero no expresiones.
- Soporta índices simples y compuestos.
- Soporta índices parciales y completos.
- Permite indexar por campos, subcampos y *arrays*.
- Soporta índices únicos.

Creación de índices secundarios

Recordemos que cada índice se asocia a una única colección, la cual puede contener tantos como sea necesario. Por otra parte, recordemos que toda colección se crea con un índice principal, aquel cuyo elemento de indexación es el campo `_key` de los documentos. Nosotros como usuarios podemos crear tantos índices secundarios, aquellos que no son el principal, como sea necesario. Y podemos hacerlo mediante el *shell* y la interfaz web de **ArangoDB**.

Creación de índices secundarios mediante *arangosh*

Para crear un índice mediante el *shell*, se puede utilizar el método `ensureIndex()` de los objetos de tipo colección:

```
ensureIndex(props)
```

Parámetro	Tipo de datos	Descripción
-----------	---------------	-------------

<code>props</code>	object	Propiedades de definición del índice: <ul style="list-style-type: none"> • <code>type</code> (string). Tipo de índice: <code>hash</code>, <code>skiplist</code>, <code>persistent</code>, <code>fulltext</code>, <code>geo1</code> o <code>geo2</code>. • <code>fields</code> (string[]). Campos de indexación. • <code>sparse</code> (boolean). ¿Índice parcial? • <code>unique</code> (boolean). ¿Índice único?
--------------------	--------	---

He aquí un ejemplo ilustrativo para una colección `bands`, usando el campo `name` como clave de indexación y asegurando que nunca habrá dos o más documentos con el mismo valor de indexación:

```
db.bands.ensureIndex({
  type: "persistent",
```

```

    fields: ["name"],
    unique: true
  })

```

ArangoDB soporta varios métodos de indexación: **hash**, **skiplist**, **persistent**, **fulltext**, **geo1** y **geo2**. Atendiendo a la consultas en las que se utilizará, hay que utilizar uno u otro. Los principales son **hash**, **persistent** y **skiplist**. Recordemos que el motor sólo usa índices de huella, **hash**, en sus consultas cuando la comparación de los campos de indexación se hace mediante el operador de igualdad (**==**) y además se consulta *todos* los campos que forman la clave de indexación. El método por saltos, **skiplist**, se usa con consultas que comparan los campos de indexación mediante comparación o rango y además se pueden utilizar con operaciones de ordenación.

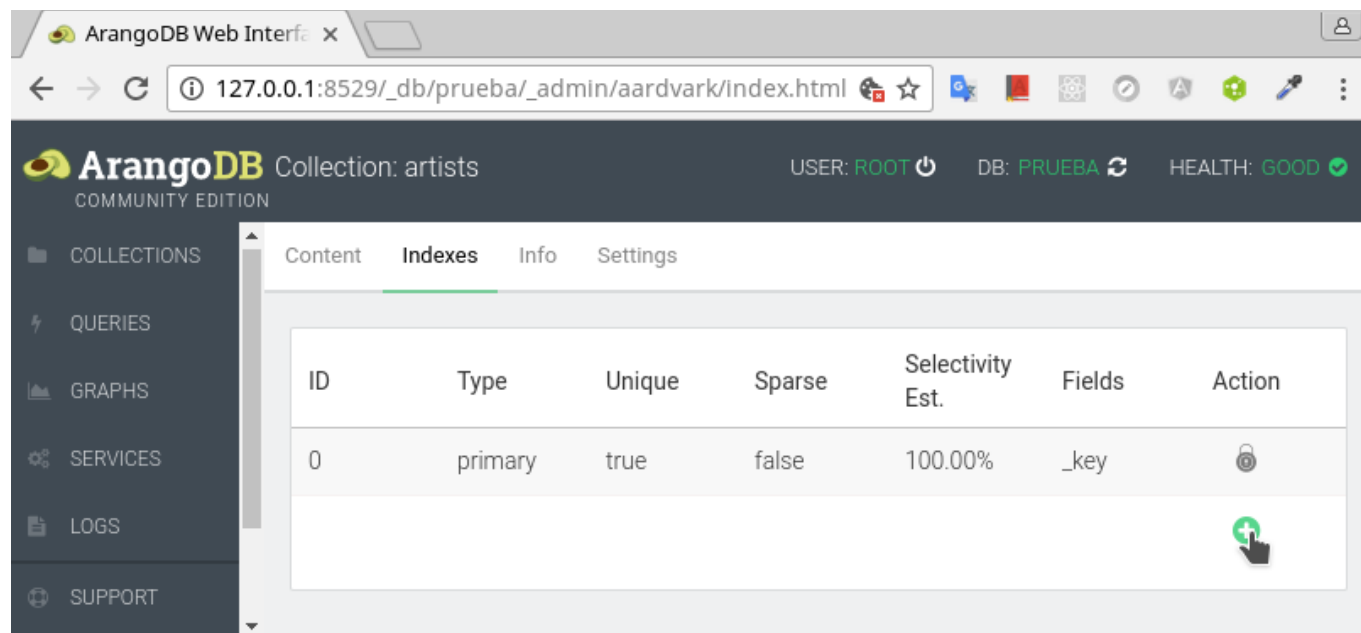
Un **índice persistente** (*persistent index*) es aquel que se almacena en disco. Aquellos que se almacenan en memoria se conocen como **índices volátiles** (*volatile indexes*). En ArangoDB, los índices persistentes se crean mediante **type** a **persistent**.

Los índices volátiles son mejores, son más rápidos, pero presentan los siguientes inconvenientes:

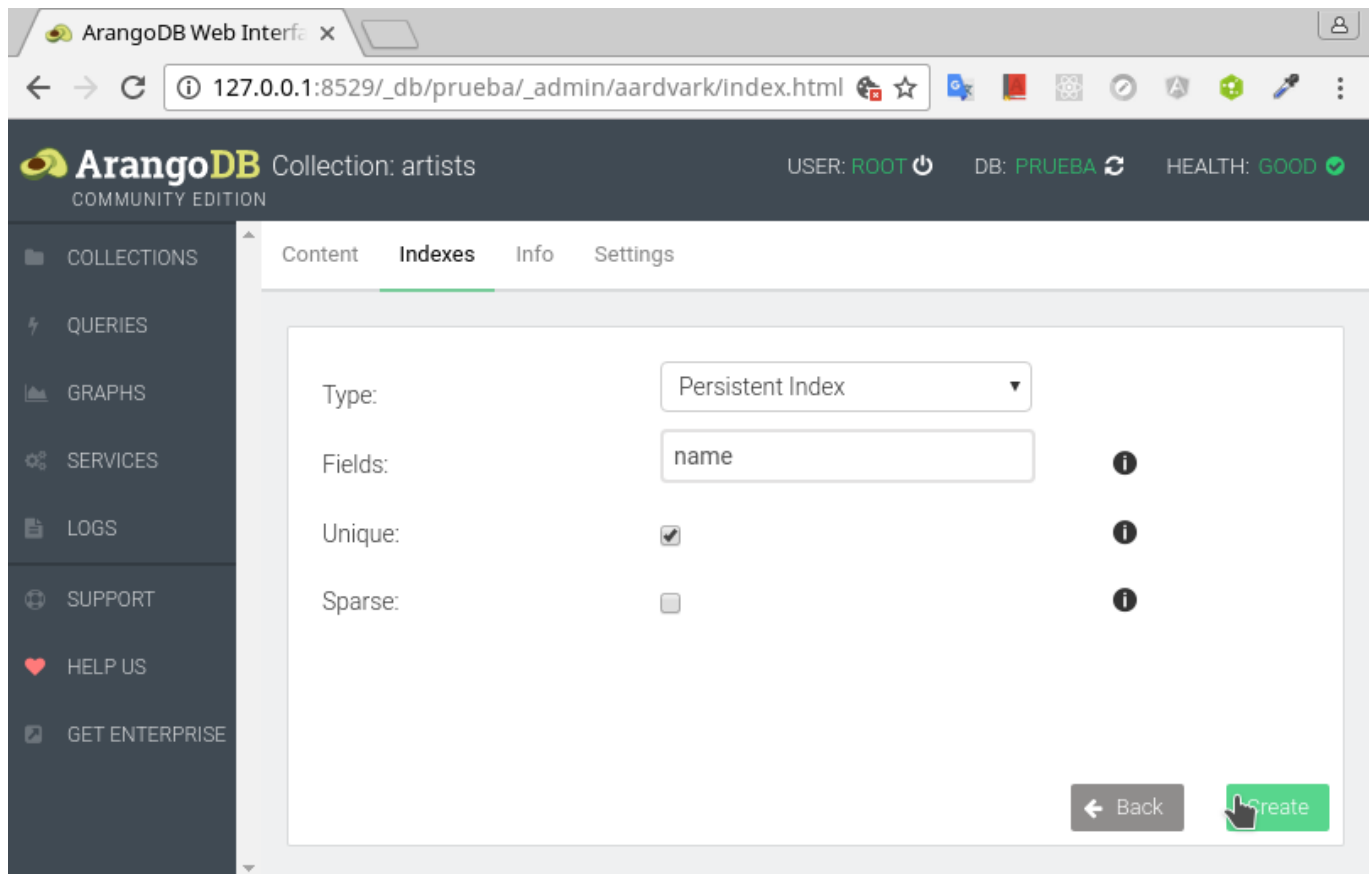
- Como se mantienen íntegramente en memoria, consumen memoria tanto si se usan como si no se llegan a usar nunca. Lo que resta memoria en el búfer para almacenar documentos de datos.
- Como se mantienen íntegramente en memoria, cuando se detiene la instancia, se borra su contenido, por lo que hay que regenerarlos en cada reinicio de la instancia. Así pues, el tiempo de reinicio será mayor a medida que tengamos muchos índices volátiles y/o muchas entradas en estos índices.

Creación de índices secundarios mediante interfaz web

Para crear un índice secundario mediante la interfaz web de ArangoDB, hay que ir a **COLLECTIONS**, a continuación, hacer clic en la colección a la que crear el índice e ir a **Indexes**:



Finalmente, hacer clic en **+**, especificar la información del índice y hacer clic en **Create**:



Claves de indexación

En **Arango**, un campo de indexación puede ser uno de los campos de los documentos o alguna de las propiedades de campos de tipo objeto. En el primer caso, se indica simplemente el nombre del **campo**, mientras que en el segundo se indica el del campo y el de su propiedad usando la siguiente sintaxis **campo.propiedad**. También es posible indexar propiedades de elementos de *array*, para lo que hay que usar la sintaxis **campo[*].propiedad**. Y si lo que deseamos es indexar los elementos de campos de tipo *array*, usaremos la sintaxis **campo[*]**.

He aquí un ejemplo ilustrativo:

```
db.providers.ensureIndex({
  type: "persistent",
  fields: ["address.town", "name"]
})
```

Listado de índices

Para conocer los índices asociados a una colección, se puede usar el *shell* y la interfaz web.

Listado de índices mediante arangosh

En el caso del *shell*, usar el método **getIndexes()** de la colección:

getIndexes() : object[]

Ejemplo ilustrativo:

```
127.0.0.1:8529@prueba> db.artists.getIndexes()
[
  {
    "id" : "artists/0",
    "type" : "primary",
    "fields" : [
      "_key"
    ],
    "selectivityEstimate" : 1,
    "unique" : true,
```

```

    "sparse" : false
  },
  {
    "id" : "artists/462256",
    "type" : "persistent",
    "fields" : [
      "name"
    ],
    "unique" : true,
    "sparse" : false
  }
]
127.0.0.1:8529@prueba>

```

Listado de índices mediante interfaz web

Para conocer los índices de una colección mediante la interfaz web, ir a [COLLECTIONS](#), hacer clic en la colección en cuestión y finalmente ir a [Indexes](#).

Supresión de índices secundarios

Mediante la supresión de un índice secundario, se borra su contenido y dejará de ser utilizable por el motor de bases de datos en las consultas. Esta supresión se puede hacer mediante el *shell* y la interfaz web.

Supresión de índices secundarios mediante arangosh

Para suprimir un índice secundario mediante el *shell*, utilizar el método `dropIndex()` de su objeto colección:

```
dropIndex(ix)
```

Parámetro	Tipo de datos	Descripción
-----------	---------------	-------------

<code>ix</code>	number	Identificador numérico asignado por ArangoDB en su creación.
-----------------	--------	--

Ejemplo:

```

127.0.0.1:8529@prueba> db.artists.getIndexes()
[
  {
    "id" : "artists/0",
    "type" : "primary",
    "fields" : [
      "_key"
    ],
    "selectivityEstimate" : 1,
    "unique" : true,
    "sparse" : false
  },
  {
    "id" : "artists/462256",
    "type" : "persistent",
    "fields" : [
      "name"
    ],
    "unique" : true,
    "sparse" : false
  }
]
127.0.0.1:8529@prueba> db.artists.dropIndex(462256)
true
127.0.0.1:8529@prueba>

```

Supresión de índices secundarios mediante interfaz web

Para suprimir un índice secundario mediante la interfaz web de [ArangoDB](#), ir a [COLLECTIONS](#), seleccionar la colección asociada al índice, ir a [Indexes](#) y finalmente hacer clic en [-](#) del índice a suprimir:

ArangoDB Web Interface x

127.0.0.1:8529/_db/prueba/_admin/aardvark/index.html

ArangoDB COMMUNITY EDITION Collection: artists USER: ROOT DB: PRUEBA HEALTH: GOOD

COLLECTIONS
QUERIES
GRAPHS
SERVICES
LOGS
SUPPORT
HELP US

Content **Indexes** Info Settings

ID	Type	Unique	Sparse	Selectivity Est.	Fields	Action
0	primary	true	false	100.00%	_key	
462256	persistent	true	false	n/a	name	