

Algunas aplicaciones webs necesitan poder extraer documentos que contengan determinadas palabras o palabras que comienzan por determinados textos. Este tipo de consultas se puede realizar muy fácilmente mediante la búsqueda de texto. El objeto de la presente lección.

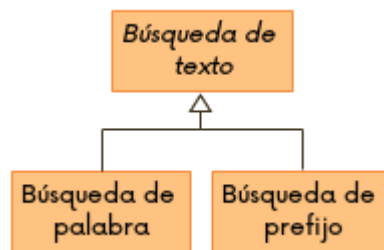
Comenzamos introduciendo el concepto de búsqueda de texto. A continuación, se muestra cómo crear los índices `fulltext`, indispensables para realizar búsquedas de texto. Y finalmente, la función `AQL fulltext()` con la que realizar las búsquedas.

Al finalizar la lección, el estudiante sabrá:

- Qué es la búsqueda de texto.
- Qué tipos de búsqueda se puede realizar en `ArangoDB`.
- Cómo crear un índice `fulltext`.
- Cómo usar la función predefinida `fulltext()`.

Introducción

La **búsqueda de texto** (*full-text search*) es una técnica para comprobar campos de texto de manera más avanzada. En `ArangoDB`, se puede realizar dos tipos de búsqueda: la de palabras o la de prefijos.



Una **búsqueda de palabra** (*word search*) es aquella que comprueba si una determinada palabra se encuentra en el valor de un campo. Mientras que una **búsqueda de prefijo** (*prefix search*) es la que comprueba si existe alguna palabra en el valor de un campo que comience por un determinado texto.

La ventaja de la búsqueda de texto es que es capaz de extraer palabras a partir de texto con símbolos de puntuación como, por ejemplo, la coma, el punto, los paréntesis, etc. Al hacerlo esta técnica, ya no tenemos que realizar complejas expresiones regulares con las que hacerlo nosotros.

En `ArangoDB`, el campo a consultar puede ser de tipo texto, un *array* de textos o bien un objeto cuyas propiedades son de tipo texto. Por ejemplo, supongamos que disponemos de una colección cuyos documentos representan páginas webs. Ahora, supongamos que deseamos extraer aquellas que tienen dos palabras determinadas, sin importar su posición. Esta comprobación se puede hacer muy fácilmente mediante búsqueda de texto.

Para poder realizar una búsqueda de texto sobre una colección, es necesario que disponga de un índice de tipo `fullindex` sobre el campo en el que realizar la búsqueda. Esta búsqueda se realizará mediante la función `AQL fulltext()`. Veamos ambas cosas separadamente.

Índices de búsqueda de texto

Para poder realizar búsquedas de texto, ya sea de palabra o prefijo, sobre un campo de documento, es necesario disponer de un **índice de búsqueda de texto** (*full-text index*). Este tipo de índice se crea, al igual que los demás, mediante el método `ensureIndex()` de la colección en cuestión. Pero debe proporcionar como tipo el valor `fulltext`:

```
ensureIndex({
```

```

    type: "fulltext",
    fields: ["campo"],
    minLength: mínimo
  })

```

Los índices de búsqueda son simples y parciales.

Como son simples, sólo se puede indicar un campo de indexación. Sólo se puede realizar búsquedas sobre un campo indexado mediante un índice de tipo `fulltext`.

Son también parciales, así pues no todos los documentos se indexarán. Por un lado, cualquier documento que no disponga del campo de indexación, no se indexará. Y por otra parte, todo documento que no disponga de tantas palabras a indexar como se indique en la propiedad `minLength` tampoco se indexará.

El objeto del índice no es otro que mantener una tabla con todas las palabras que aparecen en el campo de indexación de los documentos. Para cada palabra, registra qué documentos la tienen. Lo que ayuda a encontrar fácilmente qué documentos tienen una determinada palabra o comienzan por un determinado prefijo

He aquí un ejemplo ilustrativo de creación de este tipo de índice mediante el `shell` de **ArangoDB**:

```

db.bands.ensureIndex({
  type: "fulltext",
  fields: ["description"],
  minLength: 5
});

```

Recordemos que los índices también se pueden crear mediante la interfaz web, tal como vimos en la lección de índices.

Función predefinida `fulltext()`

Una vez definido un índice sobre el campo de búsqueda, ya podemos utilizar la función **AQL `fulltext()`** para extraer aquellos documentos de la colección que contienen una determinada palabra o prefijo:

```

fulltext(coll, field, query) : object[]
fulltext(coll, field, query, limit) : object[]

```

Parámetro	Tipo de datos	Descripción
<code>coll</code>	ArangoCollection	Colección donde buscar.
<code>field</code>	string	Campo donde realizar la búsqueda.
<code>query</code>	string	Elementos a buscar.
<code>limit</code>	number	Número máximo de documentos a devolver.

La función devuelve un `array` con aquellos documentos de la colección para los que se cumple la condición de búsqueda `query`. Este parámetro, de tipo texto, contiene los elementos de búsqueda. Un **elemento de búsqueda** (*search token*) representa una palabra o prefijo a buscar. Puede indicarse varios elementos, separándolos por comas. Por un lado, si se desea comprobar si el campo contiene una palabra tal cual, el elemento es la palabra en cuestión:

palabra

Si lo que se desea es comprobar si el campo contiene alguna palabra que comienza por un determinado texto, el elemento se indica siguiendo la siguiente sintaxis:

prefix:texto

Por ejemplo, si deseamos buscar todas las bandas que tienen en su nombre la palabra `the` y alguna palabra que comienza por `a`, podemos realizar una consulta como la siguiente:

```

FOR b IN fulltext(bands, "name", "the,prefix:a")
RETURN b

```

Si se desea indicar opcionalidad, uno u otro, se puede preceder el elemento de búsqueda por `|`. Así pues, el siguiente ejemplo extrae los documentos cuyo nombre tiene la palabra `the` y/o tiene alguna palabra que comienza por `a`:

```

FOR b IN fulltext(bands, "name", "the,|prefix:a")
RETURN b

```

También es posible indicar que un texto no aparezca, para ello se precede el elemento en cuestión por

un guión (-). El siguiente ejemplo extrae aquellos documentos que tienen la palabra `the` y ninguna comenzada por `a`:

```
FOR b IN fulltext(bands, "name", "the,-prefix:a")  
RETURN b
```

Cuando no se precede el elemento de ningún símbolo, se asume que debe aparecer. En resumen, tenemos:

- `uno,dos` equivale a uno y dos.
- `uno,|dos` equivale a uno y/o dos.
- `uno,-dos` equivale a uno y *no* dos.

Aunque en los ejemplos hemos indicado uno o dos elementos, puede indicar tantos como necesite.