

Agregaciones

Las agregaciones sirven para realizar una secuencia de operaciones sobre los documentos de una colección que toman como entrada. La forma en la que se definen las operaciones que se realizarán sobre esta colección de documentos es mediante *pipelines* de agregaciones. Un pipeline de agregaciones consiste en un array de fases o etapas (*stages*) de agregación. Cada etapa de agregación está a su vez definida por un documento que describe una operación de agregación. Las etapas de agregación se ejecutan en el orden en que están descritas, de forma que la salida de una etapa de agregación es la entrada que recibe la siguiente.

Un ejemplo de *pipeline* de agregaciones sería el siguiente:

```
[
  // Stage 1: Seleccionamos los listings que tengan un precio menor o igual a
  100 y un mínimo de noches menor o igual a 3.
  {
    $match: { $and: [ { price: { $lte: 100 } }, { minimun_nights: { $lte: 3 } } ] }
  },
  // Stage 2: Agrupamos los listings por barrio y contamos el número de listings
  válidos.
  {
    $group: { _id: 'neighbourhood', valid_listings: { $count: {} } }
  }
]
```

Una agregación que transforme un campo de texto a numérico podría ser la siguiente:

Como podemos ver este *pipeline* tiene dos etapas de agregación. En la primera filtramos (`$match`) los documentos de entrada y nos quedamos con los que tengan un precio menor o igual a 100 y un mínimo de noches menor o igual a 3. En la segunda etapa de agregación agrupamos los documentos de entrada por barrio y contamos el número de listings.

En esta operación de agregación se creará (`$addField`) un nuevo campo (o se sobrescribirá en este caso), `price`, aplicando una serie de operaciones al valor previo del mismo (`$150.00`). La primera operación será `$substring` que obtendrá las posiciones que van de la 1 a la `$strLenCP: '$price'` (longitud total de la cadena `$price` original: 7). Esto dará como resultado una cadena en la que se suprime el primer símbolo (que siempre es \$). A continuación se aplica el operador `$toInt` a la cadena resultante (`150.00`). De maneral que finalmente guardaremos el valor numérico 150.00 en el campo `price`.

Lo que estamos haciendo es convertir la cadena de texto "\$150.00" al valor numérico decimal 150.00.

Resultado

El resultado de una agregación será un cursor con los documentos salida de la última etapa del pipeline de agregación. Este cursor desaparecerá una vez consumido si no se incluye una fase de `$out`, que permite salvar los resultados en una nueva colección.

Una agregación **no modifica los documentos originales**.

Las operaciones se realizan todas en memoria por lo que hay [límites de memoria](#) establecidos para ciertas operaciones de agregación (como `$group`).

Update y pipelines de agregación

La operación de modificación de colecciones, `update`, soporta que se le indiquen las modificaciones de realizar mediante un pipeline de agregación. Este pipeline estará limitado a las siguientes fases:

- [\\$addFields](#): Permite añadir nuevos campos a los documentos.
- [\\$set](#): Es un alias de `$addFields`.
- [\\$project](#): Permite añadir **y eliminar** campos de una colección.
- [\\$unset](#): Es un alias de `$project`.
- [\\$replaceRoot](#): Permite reemplazar el documento de entrada por otro documento.
- [\\$replaceWith](#): Es la versión nueva de `$replaceRoot` con otra sintaxis.