

			Mostra os documentos de uma coleção do maior para o menor Db. <code>nome da coleção</code> .find() .sort ({ <code>nome da chave</code> : 1}) Exemplo Mostra os documentos na ordem: iniciando do que possui o maior nome para o menor nome Db. <code>nome da coleção</code> .find() .sort ({ <code>nome</code> : 1})
			Mostra os documentos de uma coleção do menor para o maior Db. <code>nome da coleção</code> .find() .sort ({ <code>nome da chave</code> : -1}) Mostra os documentos na ordem: iniciando do que possui o menor nome para o maior nome Db. <code>nome da coleção</code> .find() .sort ({ <code>nome</code> : -1})

Adicionar, apagar e sobrescrever dados de um documento	Para utilizar os comandos deve: - abrir o prompt - digitar a pasta e o arquivo do mongo C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe		
	adicionar dado a um documento de uma coleção	Db. <code>nome da coleção</code> .updateOne({ <code>nome do dado da coleção</code> }, { \$set: { <code>novos dados</code> } })	Adicionar uma nova informação a um dado que está em uma determinada coleção de um banco de dados Db. <code>nome da coleção</code> .updateOne({ name: "fabio" }, { \$set: { telefone: "12345556" } })
	Adicionar um novo dado a todos os documentos de uma coleção	Db. <code>nome da coleção</code> .updateMany({ }, { \$set: { <code>novos dados</code> } })	Adicionar uma nova informação a todos os documentos de uma coleção de um banco de dados Db. <code>nome da coleção</code> .updateMany({ }, { \$set: { classe: "turma B" } })
	Apagar um dado que tem em todos os documentos de uma coleção	Db. <code>nome da coleção</code> .updateMany({ }, { \$unset: { <code>dado a ser apagado</code> } })	Apagar uma informação que todos os documentos tem Db. <code>nome da coleção</code> .updateMany({ }, { \$unset: { classe: "turma B" } })
	sobrescrever um dado de um documento de uma coleção	Db. <code>nome da coleção</code> .update({ <code>nome do documento a ser modificado</code> }, { <code>dado a ser colocado</code> })	Alterar todo um dado de um documento (tirando todos os dados anteriores e colocando apenas os novos dados definidos) Db. <code>nome da coleção</code> .update({ name: "fabio" }, { name: "igor", password: "das1345" })
	Verificar se tem um documento se tiver vai ser sobrescrito seu dados. Caso não tenha via ser criado esse documento com esses dados	Db. <code>nome da coleção</code> .update({ <code>nome do documento a ser modificado caso tenha</code> }, { <code>dado a ser colocado</code> }, { upsert: true })	Alterar todo um dado de um documento caso esse documento exista. Caso ele não exista deve ser criado documento com os dados definidos Db. <code>nome da coleção</code> .update({ name: "fabio" }, { name: "igor", password: "das1345" }, { upsert: true })
	Renomear um dado que existe em varios documento de uma coleção	Db. <code>nome da coleção</code> .updateMany({ }, { \$rename: { <code>nome do dado</code> : " <code>novo nome que sera colocado</code> " } })	Renomear um dado [password] que tem em varios documentos Db. <code>nome da coleção</code> .updateMany({ }, { \$rename: { password: "senha" } })
	Incrementar um dado de documento de uma coleção	Db. <code>nome da coleção</code> .updateOne({ <code>nome do documento</code> }, { \$inc: { <code>nome do dado</code> : <code>valor do incremento</code> } })	Incrementar um dado de um documento onde sera atribuido uma soma de +1 para cada visualização para algum perfil Db. <code>nome da coleção</code> .updateOne({ name: "fabio" }, { \$inc: { visualização: 1 } })