

Faça o que eu fiz na aula

Vamos separar as informações do autor em uma nova coleção, para isso, vamos utilizar a agregação:

```
db.artigos.aggregate([
  {
    $group: {
      _id: "$autor.email",
      nome: {
        $first: "$autor.nome"
      },
      email: {
        $first: "$autor.email"
      },
      descricao: {
        $first: "$autor.descricao"
      },
      senha: {
        $first: "$autor.senha"
      }
    }
  },
  {
    $project: {
      _id: 0
    }
  },
  {
    $out: "infoAutores"
  }
])
```

Agora precisamos remover o email e a senha do documento que está na coleção de artigos, e precisamos também adicionar o objectId do documento que contém as informações do autor, vamos fazer isso com o comando que deve ser executado na linha de comando do MongoDB:

```
db.artigos.find({}).forEach(function(artigo) {
  var infoAutor = db.infoAutores.findOne({ email: artigo.autor.email });

  artigo.autor.id_info = infoAutor._id;

  delete artigo.autor.email;
  delete artigo.autor.senha;

  db.artigos.save(artigo)
})
```

Se você não se lembra como executar este comando, abra a linha de comando do seu sistema operacional e digite o seguinte comando:

```
mongo
```

Assim entraremos na linha de comando do MongoDB, mas ainda falta informar para ele qual é o banco de dados que estamos utilizando, que no nosso caso chama `blog`.

```
use blog
```

E agora podemos copiar e colar o comando acima.

Se precisarmos fazer uma atualização em uma dessas informações, como o nome do autor, temos que replicar o `update` também, crie um arquivo chamado `atualizando-nome-autor.mongo`:

```
db.infoAutores.update({
  nome: "Gabriel Ferreira"
}, {
  $set: {
    nome: "Gabs Ferreira"
  }
})

db.artigos.update({
  "autor.nome": "Gabriel Ferreira"
}, {
  $set: {
    "autor.nome": "Gabs Ferreira"
  }
}, { multi:true })
```