**Comandos MongoDB:**

* MongoDB é case sensitive, armazena documentos no formato JSON, facilmente escalável para múltiplos processadores, não tem esquema.
* show dbs : mostra os bancos instalados
* use nomedobanco : cria um novo banco de dados e muda pra ele
* show collections : mostra os documentos
* db.createCollection(nomedacolecao) : cria documento novo
* db.nomedacolecao.drop() : deleta documento
* db.nomedacolecao.insert({chave: valor, ...}) : insere registro
* db.nomedacolecao.save({chave: valor, ...}) : insere se nao existe ou atualiza existente
* db.nomedacolecao.find({filtros}) : lista todos os registros de acordo com o filtro
* db.nomedacolecao.find().pretty() : lista com melhor leitura
* db.nomedacolecao.findOne({filtros}) : busca um registro a partir do filtro
* db.nomedacolecao.find({filtros}).skip(elemento) : pula até o elemento indicado
* db.nomedacolecao.find({filtros}).limit(quantidade) : limita a quantidade retornada
* db.nomedacolecao.find({filtros}).skip(elemento).limit(quantidade) : PAGINACAO
* db.nomedacolecao.aggregate() : extrai o sumário
  + $project : extrai do documento somente as informações que interessam
  + $group : agrupamento
* db.nomedacolecao.update({filtro}, {$set : {campos a serem atualizados}})
  + para atualizar multiplos registros basta adicionar , {multi: true})
* db.nomedacolecao.remove({filtro}) : remove todos elementos de acordo com o filtro
* db.dropDatabase() : deleta o banco

**Node JS:**

* Backend do javascript
* Sublime pode executar javascript criando um build novo apontando para o node
* “Cada arquivo representa um modulo”
* module.exports = {atributoexterno: origem} : Se quiser expor o modulo
  + se o externo foi igual ao interno, no ecma 2015 pode escrever apenas o nome da funcao origem
* this também torna o objeto exposto
* npm init : Para iniar um projeto node, com dependencias
  + npm i library --save : para salvar a dependencia no package.json
* template string : `string ${param}` : Ecma 2015 , permite expressoes embutidas
* node possui uma variavel chamada **process** , possibilitando passar argumentos para a mesma, por exemplo com process.argv
  + process.stdout.write() : escreve no console de saida
  + process.stdin.on(‘variavel’, funcao) : recebe e faz um callback function
* fs: Módulo FileSystem nativo do node, para trabalhar com arquivos.
  + fs.readdirSync(\_\_dirname) : lista nome dos arquivos que estao no diretorio atual.
* http : modulo http nativo do node