

tryber / **sd-010-a-mongodb-commerce** Publicgenerated from [betrybe/sd-0x-mongodb-commerce](#)

☆ 0 stars 🍴 8 forks

☆ Star

🔔 Stop ignoring ▼

Code

Issues

Pull requests 132

Actions

Projects

Wiki

Security

Insights

🔗 master ▼

...



vinicius-vasconcelos ...

on 20 Aug

[View code](#)

☰ README.md



Termos e acordos

Ao iniciar este projeto, você concorda com as diretrizes do Código de Ética e Conduta e do Manual da Pessoa Estudante da Trybe.

Boas vindas ao repositório do projeto de MongoDB Commerce!

Você já usa o GitHub diariamente para desenvolver os exercícios, certo? Agora, para desenvolver os projetos, você deverá seguir as instruções a seguir. Fique atento a cada passo, e se tiver qualquer dúvida, nos envie por *Slack*! #vqv 🚀

Aqui você vai encontrar os detalhes de como estruturar o desenvolvimento do seu projeto a partir deste repositório, utilizando uma branch específica e um *Pull Request* para colocar seus códigos.

Sumário

- [Habilidades](#)

- Entregáveis
 - O que deverá ser desenvolvido
 - Desenvolvimento
 - Data de entrega
- Instruções para entregar seu projeto
 - Antes de começar a desenvolver
 - Durante o desenvolvimento
- Como Desenvolver
 - Análise Estática
 - Instruções para restaurar o banco de dados commerce
 - Implementações técnicas
 - Linter
- Requisitos do projeto
 - 1 - Inclua o campo criadoPor em todos os documentos, colocando Ronald McDonald no valor desse campo
 - 2 - Inclua o campo valorUnitario em todos os documentos em que esse campo não existe e atribua a ele o valor 0.00, com o tipo NumberDecimal
 - 3 - Adicione o campo avaliacao em todos os documentos da coleção e efetue alterações nesse campo
 - 4 - Atribua a data corrente ao campo ultimaModificacao no sanduíche Big Mac
 - 5 - Adicione ketchup aos ingredientes para todos os sanduíches menos o McChicken, garantindo que não haja duplicidade nos ingredientes
 - 6 - Inclua bacon no final da lista de ingredientes dos sanduíches Big Mac e Quarteirão com Queijo
 - 7 - Remova o item cebola de todos os sanduíches
 - 8 - Remova o primeiro ingrediente do sanduíche Quarteirão com Queijo
 - 9 - Remova o último ingrediente do sanduíche Cheddar McMelt
 - 10 - Adicione a quantidade de vendas dos sanduíches por dia da semana
 - 11 - Insira os elementos combo e tasty no *array* tags de todos os sanduíches e aproveite para deixar os elementos em ordem alfabética ascendente
 - 12 - Ordene em todos os documentos os elementos do *array* valoresNutricionais pelo campo percentual de forma descendente
 - 13 - Adicione o elemento muito sódio ao final do *array* tags nos produtos em que o percentual de sódio seja maior ou igual a 40

- 14 - Adicione o elemento contém sódio ao final do *array* tags nos produtos em que o percentual de sódio seja maior do que 20 e menor do que 40
 - 15 - Conte quantos produtos contêm Mc no nome, sem considerar letras maiúsculas ou minúsculas
 - 16 - Conte quantos produtos têm 4 ingredientes
 - 17 - Conte quantos documentos contêm as palavras frango e hamburguer utilizando o operador `$text`
 - 18 - Conte quantos documentos contêm a expressão feito com utilizando o operador `$text`
 - 19 - Renomeie o campo `descricao` para `descricaoSite` em todos os documentos
 - 20 - Remova o campo `curtidas` do item Big Mac
 - 21 - Retorne o nome dos sanduíches em que o número de curtidas é maior que o número de sanduíches vendidos
 - 22 - Retorne o nome e a quantidade de vendas (vendidos) dos sanduíches em que o número de vendas é múltiplo de 5
- Depois de terminar o desenvolvimento (opcional)
 - Revisando um pull request
 - Avisos Finais

Habilidades

Neste projeto, verificamos se você é capaz de:

- Utilizar o método `updateOne()` e `updateMany()`
- Utilizar os operadores `$set`, `$mul`, `$inc`, `$min`, `$max` e `$currentDate`
- Renomear campos e remover campos
- Incorporar dados aos documentos através de arrays
- Utilizar os operadores `$pop`, `$pull` e `$push`
- Utilizar o operador `$addToSet`
- Utilizar os operadores `$each`, `$slice` e `$sort`
- Utilizar o operador `$all` para filtrar documentos
- Utilizar o operador `$elemMatch` para filtrar documentos

- Utilizar o operador `$size` para filtrar documentos pelo tamanho de arrays
- Utilizar o operador `$expr` para criar expressões de agregação
- Utilizar expressões regulares e o operador `$regex` para buscar documentos
- Utilizar o índice textual e o operador `$text`
- Utilizar o operador `$mod`

Entregáveis

O que deverá ser desenvolvido

Hoje você fará um projeto com o codinome *commerce*. Neste projeto, você praticará todos os conceitos de **MongoDB** já ensinados até aqui.

Para este projeto, escolhemos um dataset bem menor do que o **dataFlights**, mas isso não vai impedir que você aplique tudo o que viu até aqui, combinando os conhecimentos deste bloco e do anterior.

A ideia é trabalhar com o banco de dados *commerce*, que contém dados do cardápio do **McDonald's**, como ingredientes, valores nutricionais e dados fictícios de vendas. As instruções de como restaurar o banco podem ser lidas a seguir.

Desenvolvimento


Temos, neste projeto, uma série de desafios com diferentes níveis de complexidade. Cada desafio deve ser resolvido em seu arquivo próprio.

1. Leia a pergunta e crie no diretório *challenges* um arquivo chamado *desafioN.js*, em que N é o número do desafio.
2. O arquivo deve conter apenas o código MQL (*Mongo Query Language*) do desafio resolvido. **Não se esqueça de incluir o ponto e vírgula (";")** no final de suas queries, como no exemplo a seguir:

```
db.produtos.find();
```

 **Restrições** .

- **Não se deve usar aspas simples** para especificar campos e/ou valores. Quando for necessário usar aspas, **use somente aspas duplas**;
 - **Não se deve usar o comando** `use commerce` , haja visto que **os testes já se conectam automaticamente à base** `commerce` .
3. Faça isso até finalizar todos os desafios e depois siga as instruções de como entregar o projeto em [Instruções para entregar seu projeto](#).
4. Para entregar o seu projeto você deverá criar um *Pull Request* neste repositório. Este *Pull Request* deverá conter no diretório `challenges` os arquivos `desafio1.js` , `desafio2.js` e assim por diante até o `desafio22.js` , que conterão seu código `MQL` de cada desafio, respectivamente.

 **É importante que seus arquivos tenham exatamente estes nomes!** 

Qualquer dúvida, procure a monitoria. Lembre-se que você pode consultar nosso conteúdo sobre [Git & GitHub](#) sempre que precisar!

Data de Entrega

- Projeto individual.
- Será um dia de projeto.
- Data de entrega para avaliação final do projeto: 24/08/2021 - 14:00h .

Instruções para entregar seu projeto



Antes de começar a desenvolver:

1. Clone o repositório
 - `git clone git@github.com:tryber/sd-010-a-mongodb-commerce.git` .
 - Entre na pasta do repositório que você acabou de clonar:
 - `cd sd-010-a-mongodb-commerce`
2. Crie uma branch a partir da branch `master`
 - Verifique que você está na branch `master`
 - Exemplo: `git branch`

- Se não estiver, mude para a branch `master`
 - Exemplo: `git checkout master`
 - Agora crie uma branch à qual você vai submeter os `commits` do seu projeto
 - Você deve criar uma branch no seguinte formato: `nome-de-usuario-nome-do-projeto`
 - Exemplo: `git checkout -b seunome-mongodb-commerce`
3. Para cada exercício você deve criar um novo arquivo JS **dentro de uma pasta na raiz do seu projeto chamada `challenges`** seguindo a seguinte estrutura:
- `desafio1.js`, `desafio2.js`, ..., `desafioN.js`
4. Adicione as mudanças ao `stage` do Git e faça um `commit`
- Verifique que as mudanças ainda não estão no `stage`
 - Exemplo: `git status` (deve aparecer o arquivo que você alterou como `desafio1.js`)
 - Adicione o novo arquivo ao `stage` do Git
 - Exemplo:
 - `git add .` (adicionando arquivo de solução `challenges/desafio1.js` para desafio 1)
 - `git status` (deve aparecer listado o arquivo `challenges/desafio1.js` em verde)
 - Faça o `commit` inicial
 - Exemplo:
 - `git commit -m 'iniciando o projeto MongoDB Commerce'` (fazendo o primeiro `commit`)
 - `git status` (deve aparecer uma mensagem tipo *nothing to commit*)
 -
5. Adicione a sua branch com o novo `commit` ao repositório remoto
- Usando o exemplo anterior: `git push -u origin joaozinho-sd-010-a-mongodb-commerce`
6. Crie um novo `Pull Request (PR)`
- Vá até a página de *Pull Requests* do [repositório no GitHub](#)
 - Clique no botão verde *"New pull request"*
 - Clique na caixa de seleção *"Compare"* e escolha a sua branch **com atenção**
 - Clique no botão verde *"Create pull request"*
 - Adicione uma descrição para o *Pull Request* e clique no botão verde *"Create pull request"*
 - **Não se preocupe em preencher mais nada por enquanto!**

- Volte até a [página de Pull Requests do repositório](#) e confira que o seu *Pull Request* está criado

Durante o desenvolvimento

-  **LEMBRE-SE DE CRIAR TODOS OS ARQUIVOS DENTRO DA PASTA challenges** 
- Faça `commits` das alterações que você fizer no código regularmente
- Lembre-se de sempre após um (ou alguns) `commits` atualizar o repositório remoto
- Os comandos que você utilizará com mais frequência são:
 - i. `git status` (para verificar o que está em vermelho - fora do stage - e o que está em verde - no stage)
 - ii. `git add` (para adicionar arquivos ao stage do Git)
 - iii. `git commit` (para criar um commit com os arquivos que estão no stage do Git)
 - iv. `git push -u nome-da-branch` (para enviar o commit para o repositório remoto na primeira vez que fizer o `push` de uma nova branch)
 - v. `git push` (para enviar o commit para o repositório remoto após o passo anterior)

Como Desenvolver

Análise Estática

Usaremos o [ESLint](#) para fazer a análise estática do seu código.

Este projeto já vem com as dependências relacionadas ao *linter* configuradas no arquivos `package.json`.

Para poder rodar os `ESLint` em um projeto basta executar o comando `npm install` dentro do projeto e depois `npm run lint`. Se a análise do `ESLint` encontrar problemas no seu código, tais problemas serão mostrados no seu terminal. Se não houver problema no seu código, nada será impresso no seu terminal.

Devido ao fato de as configurações das regras do `ESLint` dos projetos de front e back serem diferentes, **é preciso executar o `ESLint` em cada projeto.**

Você pode também instalar o plugin do `ESLint` no `VSCode`, bastar ir em extensions e baixar o [plugin ESLint](#).

Instruções para restaurar o banco de dados commerce

1. Abra o terminal e conecte-se à sua instância local do **MongoDB**. Se você receber uma mensagem de erro com uma mensagem como **Connection refused**, tente reiniciar sua instância ([veja como fazer isso aqui](#)).
2. Agora que você tem certeza de que a sua instância está no ar e que você está conectado a ela, digite `exit`. Você voltará ao terminal para iniciar a importação dos dados.
3. Na raiz do diretório do projeto, execute o seguinte comando que fará a restauração da base de dados commerce :

```
DBNAME=commerce ./scripts/resetdb.sh assets/produtos
```

- A execução desse script criará um banco de dados chamado commerce e importará os dados para a coleção produtos .

⚠ Como tanto esse script quanto o script de execução local dos testes (mostrado na [seção seguinte](#)), **restauram a base de dados commerce**, se atente a salvar seu progresso nos arquivos de desafio! ⚠

Implementações técnicas

Para executar localmente os testes, é preciso escrever o seguinte no seu terminal, estando na raiz do diretório do projeto:

```
./scripts/evaluate.sh
```

Esse script passará por **todos os desafios** e imprimirá um relatório indicando se passou ou não para cada desafio. Como a execução do script **envolve restauração da base de dados commerce** de um teste para outro, recomenda-se esperar pela sua execução completa.

Para executar somente o teste de um desafio, execute o comando abaixo, substituindo N pelo número do desafio

```
./scripts/evaluate.sh desafioN
```

⚠ Como na avaliação o banco de dados commerce é restaurado de um teste para outro, **se atente a fazer uso do banco restaurado na hora de fazer um desafio**. ⚠

Linter

Para garantir a qualidade do código, vamos utilizar neste projeto o linter ESLint. Assim o código estará alinhado com as boas práticas de desenvolvimento, sendo mais legível e de fácil manutenção! Para rodar o *linter* localmente no projeto, execute o comando abaixo:

```
npm run lint
```

⚠ PULL REQUESTS COM ISSUES DE LINTER NÃO SERÃO AVALIADAS. ATENTE-SE PARA RESOLVÊ-LAS ANTES DE FINALIZAR O DESENVOLVIMENTO! ⚠

Aqui encontram-se os requisitos do projeto. Em cada requisito você encontrará uma imagem de um protótipo de como sua aplicação deve ficar. Estilo da página não será avaliado.

Requisitos do projeto

1 - Inclua o campo `criadoPor` em todos os documentos, colocando "Ronald McDonald" no valor desse campo.

Para isso, escreva no arquivo `desafio1.js` duas queries, **nesta ordem**:

1. Crie uma query que adicione o campo `criadoPor` em todos os documentos, colocando "Ronald McDonald" no valor desse campo.
2. Crie uma query que retorne o `nome` e `criadoPor` de todos os produtos.

2 - Inclua o campo `valorUnitario` em todos os documentos em que esse campo não existe e atribua a ele o valor "0.00", com o tipo `NumberDecimal`.

Para isso, escreva no arquivo `desafio2.js` duas queries, **nesta ordem**:

1. Crie uma query que adicione o campo `valorUnitario` em todos os documentos em que esse campo não existe e atribua a ele o valor "0.00", com o tipo `NumberDecimal`.
2. Crie uma query que retorne o `nome` e `valorUnitario` de todos os produtos.

3 - Adicione o campo `avaliacao` em todos os documentos da coleção e efetue alterações nesse campo.

Para isso, escreva no arquivo `desafio3.js` quatro queries, **nesta ordem**:

1. Crie uma query que inclua o campo `avaliacao` do tipo `NumberInt` e com o valor `0` em todos os documentos da coleção.
2. Crie uma query que incremente o valor do campo `avaliacao` em `5` em todos os sanduíches de carne do tipo `bovino`. Dica: utilize como filtro o campo `tags`.
3. Crie uma query que incremente o valor do campo `avaliacao` em `3` em todos os sanduíches de `ave`.
4. Crie uma query que retorne o `nome` e `avaliacao` de todos os sanduíches.

4 - Atribua a data corrente ao campo `ultimaModificacao` no sanduíche Big Mac.

Para isso, escreva no arquivo `desafio4.js` duas queries, **nesta ordem**:

1. Crie uma query que atribua a data corrente ao campo `ultimaModificacao` no sanduíche `Big Mac`. Para a data corrente faça uso do tipo `Date`.
2. Crie uma query que retorne o `nome` de todos os documentos em que o campo `ultimaModificacao` existe.

5 - Adicione ketchup aos ingredientes para todos os sanduíches menos o McChicken, garantindo que não haja duplicidade nos ingredientes.

Para isso, escreva no arquivo `desafio5.js` duas queries, **nesta ordem**:

1. Crie uma query que adicione `ketchup` aos `ingredientes` para todos os sanduíches menos o `McChicken`, garantindo que não haja duplicidade nos `ingredientes`.
2. Crie uma query que retorne o `nome` e `ingredientes` de todos os documentos.

6 - Inclua bacon no final da lista de ingredientes dos sanduíches Big Mac e Quarteirão com Queijo.

Para isso, escreva no arquivo `desafio6.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a inclusão de `bacon` no final da lista de `ingredientes` dos sanduíches `Big Mac` e `Quarteirão com Queijo`.
2. Crie uma query que retorne o `nome` e `ingredientes` de todos os documentos.

7 - Remova o item `cebola` de todos os sanduíches.

Para isso, escreva no arquivo `desafio7.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a remoção do item `cebola` em todos os sanduíches.
2. Crie uma query que retorne o `nome` e `ingredientes` de todos os documentos.

8 - Remova o primeiro ingrediente do sanduíche Quarteirão com Queijo.

Para isso, escreva no arquivo `desafio8.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a remoção do **primeiro** ingrediente no sanduíche Quarteirão com Queijo.
2. Crie uma query que retorne o `nome` e `ingredientes` de todos os documentos.

9 - Remova o último ingrediente do sanduíche Cheddar McMelt.

Para isso, escreva no arquivo `desafio9.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a remoção do **último** ingrediente no sanduíche Cheddar McMelt.
2. Crie uma query que retorne o `nome` e `ingredientes` de todos os documentos.

10 - Adicione a quantidade de vendas dos sanduíches por dia da semana.

Para isso, escreva no arquivo `desafio10.js` quatro queries, **nesta ordem**:

1. Crie uma query que inclua um *array* com sete posições em todos os sanduíches. Cada uma delas representará um dia da semana, e cada posição iniciará em `0`. O *array* deve se parecer como abaixo:

```
"vendasPorDia": [0, 0, 0, 0, 0, 0, 0]
```

- O primeiro item desse *array* representa as vendas no **domingo**, o segundo item representa as vendas na **segunda-feira**, e assim até chegar ao último item, que representa as vendas no **sábado**.
2. Crie uma query que incremente as vendas de `Big Mac` às **quartas-feiras** em `60`.
 3. Crie uma query que incremente as vendas de todos os sanduíches de carne do tipo `bovino` e `pão` aos **sábados** em `120`.
 4. Crie uma query que retorne o `nome` e `vendasPorDia` de todos os documentos.

11 - Insira os elementos `combo` e `tasty` no `array tags` de todos os sanduíches e aproveite para deixar os elementos em ordem alfabética ascendente.

Para isso, escreva no arquivo `desafio11.js` duas queries, **nesta ordem**:

1. Crie uma query que faça tanto a inserção dos elementos `combo` e `tasty` no `array tags` de todos os sanduíches quanto a ordenação dos elementos de `tags` em ordem alfabética ascendente.
2. Crie uma query que retorne o `nome` e `tags` de todos os documentos.

12 - Ordene em todos os documentos os elementos do `array valoresNutricionais` pelo campo `percentual` de forma descendente.

Para isso, escreva no arquivo `desafio12.js` duas queries, **nesta ordem**:

1. Crie uma query que faça em todos os documentos a ordenação dos elementos do `array valoresNutricionais` pelo campo `percentual` de forma descendente. Dica: mesmo sem adicionar nenhum novo elemento, para essa operação é necessário utilizar também o modificador `$each`.
2. Crie uma query que retorne o `nome` e `valoresNutricionais` de todos os documentos.

13 - Adicione o elemento `muito sódio` ao final do `array tags` nos produtos em que o `percentual` de `sódio` seja maior ou igual a `40`.

Para isso, escreva no arquivo `desafio13.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a adição do elemento `muito sódio` ao final do `array tags` nos produtos em que o `percentual` de `sódio` seja maior ou igual a `40`.
2. Crie uma query que retorne o `nome` e `tags` de todos os documentos.

14 - Adicione o elemento `contém sódio` ao final do `array tags` nos produtos em que o `percentual` de `sódio` seja maior do que `20` e menor do que `40`.

Para isso, escreva no arquivo `desafio14.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a adição do elemento `contém sódio` ao final do `array tags` nos produtos em que o `percentual` de `sódio` seja maior do que `20` e menor do que `40`.

2. Crie uma query que retorne o `nome` e `tags` de todos os documentos.

15 - Conte quantos produtos contêm Mc no nome, sem considerar letras maiúsculas ou minúsculas.

16 - Conte quantos produtos têm 4 ingredientes.

17 - Conte quantos documentos contêm as palavras `frango` ou `hamburger` utilizando o operador `$text`.

Para isso, escreva no arquivo `desafio17.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a criação de um índice do tipo `text` no campo `descricao` com o idioma padrão `portuguese`.
2. Crie uma query que retorne a quantidade de documentos que contêm as palavras `frango` ou `hamburger` utilizando o operador `$text`.

18 - Conte quantos documentos contêm a expressão `feito com` utilizando o operador `$text`.

Para isso, escreva no arquivo `desafio18.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a criação de um índice do tipo `text` no campo `descricao` com o idioma padrão `portuguese`.
2. Crie uma query que retorne a quantidade de documentos que contêm a expressão `feito com` utilizando o operador `$text`.

19 - Renomeie o campo `descricao` para `descricaoSite` em todos os documentos.

Para isso, escreva no arquivo `desafio19.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a renomeação do campo `descricao` para `descricaoSite` em todos os documentos.
2. Crie uma query que retorne o `nome`, `descricao` e `descricaoSite` de todos os documentos.

20 - Remova o campo `curtidas` do item `Big Mac`.

Para isso, escreva no arquivo `desafio20.js` duas queries, **nesta ordem**:

1. Crie uma query que faça a remoção do campo `curtidas` do item `Big Mac`.

2. Crie uma query que retorne o `nome` e `curtidas` de todos os documentos.

21 - Retorne o nome dos sanduíches em que o número de curtidas é maior que o número de sanduíches vendidos .

22 - Retorne o nome e a quantidade de vendas (vendidos) dos sanduíches em que o número de vendas é múltiplo de 5 .

Depois de terminar o desenvolvimento (OPCIONAL)

Para sinalizar que o seu projeto está pronto para o *"Code Review"* dos seus colegas, faça o seguinte:

- Vá até a página **DO SEU Pull Request**, adicione a label de *"code-review"* e marque seus colegas:
 - No menu à direita, clique no link **"Labels"** e escolha a label **code-review**;
 - No menu à direita, clique no link **"Assignees"** e escolha **o seu usuário**;
 - No menu à direita, clique no link **"Reviewers"** e digite `students` , selecione o time `tryber/students-sd-00` .

Caso tenha alguma dúvida, [aqui tem um video explicativo](#).

Revisando um pull request

Use o conteúdo sobre [Code Review](#) para te ajudar a revisar os *Pull Requests*.

#VQV 🚀

Avisos Finais

Ao finalizar e submeter o projeto, não se esqueça de avaliar sua experiência preenchendo o formulário. Leva menos de 3 minutos!

Link: [FORMULÁRIO DE AVALIAÇÃO DE PROJETO](#)

O avaliador automático não necessariamente avalia seu projeto na ordem em que os requisitos aparecem no readme. Isso acontece para deixar o processo de avaliação mais rápido. Então, não se assuste se isso acontecer, ok?

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

Contributors 3



jeanpsv Jean Paulo Silva Vasconcelos



phramos07 Pedro Ramos



vinicius-vasconcelos Vinicius-Vasconcelos

Languages

● **Shell** 100.0%