

tryber / **sd-010-a-mysql-all-for-one** Publicgenerated from [betrybe/sd-0x-mysql-all-for-one](#)

★ 1 star 🍴 13 forks

★ Star

👁 Watch ▼

Code

Issues

Pull requests 128

Actions

Projects

Wiki

Security

Insights

🔗 master ▼

...



vinicius-vasconcelos ...

on 23 Jul

[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 All For One

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 desse 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](#)
 - [Instruções para restaurar o banco de dados Northwind](#)
 - [Instruções para testar suas queries](#)
- [Requisitos do projeto](#)
 - [Lista de requisitos](#)
 - [Desafios Iniciais](#)
 - [Desafios sobre filtragem de dados](#)
 - [Desafios de manipulação de tabelas](#)
- [Depois de terminar o desenvolvimento \(opcional\)](#)
- [Revisando um pull request](#)
- [Avisos Finais](#)

Habilidades

Nesse projeto, você será capaz de:

- Entender o que são bancos de dados
- Entender como a linguagem de consulta estruturada (SQL) é usada
- Compreender como as tabelas se encaixam no conceito de banco de dados
- Montar um ambiente de desenvolvimento local para praticar SQL
- Entender como usar o MySQL Workbench
- Compreender o que é uma query SQL e quais são seus tipos de comando
- Gerar valores com `SELECT`
- Selecionar colunas individualmente com `SELECT`
- Renomear e gerar colunas em uma consulta com `AS`
- Concatenar colunas e valores com `CONCAT`
- Remover dados duplicados em uma consulta com `DISTINCT`
- Contar a quantidade de resultados em uma consulta com `COUNT`
- Limitar a quantidade de resultados em uma consulta com `LIMIT`
- Pular resultados em uma consulta com `OFFSET`
- Ordenar os resultados de uma consulta com `ORDER BY`
- Filtrar resultados de consultas com o `WHERE`
- Utilizar operadores booleanos e relacionais em consultas

- Criar consultas mais dinâmicas e maleáveis com `LIKE`
- Fazer consultas que englobam uma faixa de resultados com `IN` e `BETWEEN`
- Encontrar e separar resultados que incluem datas.
- Inserir dados em tabelas com `INSERT`
- Atualizar dados em tabelas com `UPDATE`
- Apagar dados em tabelas com `DELETE`

Entregáveis

O que deverá ser desenvolvido

Hoje você fará um projeto com o codinome *All For One* em que praticará todos os conceitos de SQL já ensinados até aqui. Porém, você vai usar um banco de dados totalmente diferente, então dê tchau para o `sakila` e dê boas vindas ao `Northwind`, que será usado neste projeto. As instruções de como restaurar o banco podem ser lidas a seguir.

Desenvolvimento

Temos, nesse projeto, uma série de desafios com diferentes níveis de complexidade que devem ser resolvidos cada um em seu arquivo próprio.

1. Leia a pergunta e crie na raiz do projeto um arquivo chamado `desafioN.sql`, em que `N` é o número do desafio;
2. O arquivo deve conter apenas o código SQL do desafio resolvido. **Não se esqueça de incluir o ponto e vírgula (";")** no final de suas queries e também de colocar o nome do **banco_de_dados.tabela_por_completo**, como no exemplo a seguir:

```
SELECT * FROM northwind.orders;
```

3. Faça isso até finalizar todos os desafios.
4. Para entregar o seu projeto você deverá criar um *Pull Request* neste repositório. Este *Pull Request* deverá conter os arquivos `desafio1.sql`, `desafio2.sql` e assim por diante até o `desafio27.sql`, que conterão seu código SQL de cada desafio, respectivamente.

5. **Não é necessário colocar** `USE northwind;` OU `SET SQL_SAFE_UPDATES = 0;` no início dos seus arquivos.

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

Você pode adicionar outros arquivos se julgar necessário. 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 1 dia de projeto.
- Data de entrega para avaliação final do projeto: 29/07/2021 - 14:00h .

Instruções para entregar seu projeto

Antes de começar a desenvolver

1. Clone o repositório

- `git clone https://github.com/tryber/sd-010-a-mysql-all-for-one.git` .
- Entre na pasta do repositório que você acabou de clonar:
 - `cd sd-010-a-mysql-all-for-one`

2. Instale as dependências [**Caso existam**]

- `npm install` [**exemplo**]

3. 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 onde você vai guardar os `commits` do seu projeto
 - Você deve criar uma branch no seguinte formato: `nome-de-usuario-nome-do-projeto`

- Exemplo: `git checkout -b seunome-mysql-all-for-one`
4. Para cada exercício você deve criar um novo arquivo sql seguindo a seguinte estrutura:
- desafio1.sql, desafio2.sql...desafioN.sql
5. 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.sql)
 - Adicione o novo arquivo ao *stage* do Git
 - Exemplo:
 - `git add .` (adicionando solução para desafio 1)
 - `git status` (deve aparecer listado o arquivo *desafio1.sql* em verde)
 - Faça o `commit` inicial
 - Exemplo:
 - `git commit -m 'iniciando o projeto MySQL All For One'` (fazendo o primeiro commit)
 - `git status` (deve aparecer uma mensagem tipo *nothing to commit*)
6. Adicione a sua branch com o novo `commit` ao repositório remoto
- Usando o exemplo anterior: `git push -u origin seunome-mysql-all-for-one`
7. 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 que garantir que todas as *issues* comentadas pelo CodeClimate estão resolvidas! ⚠

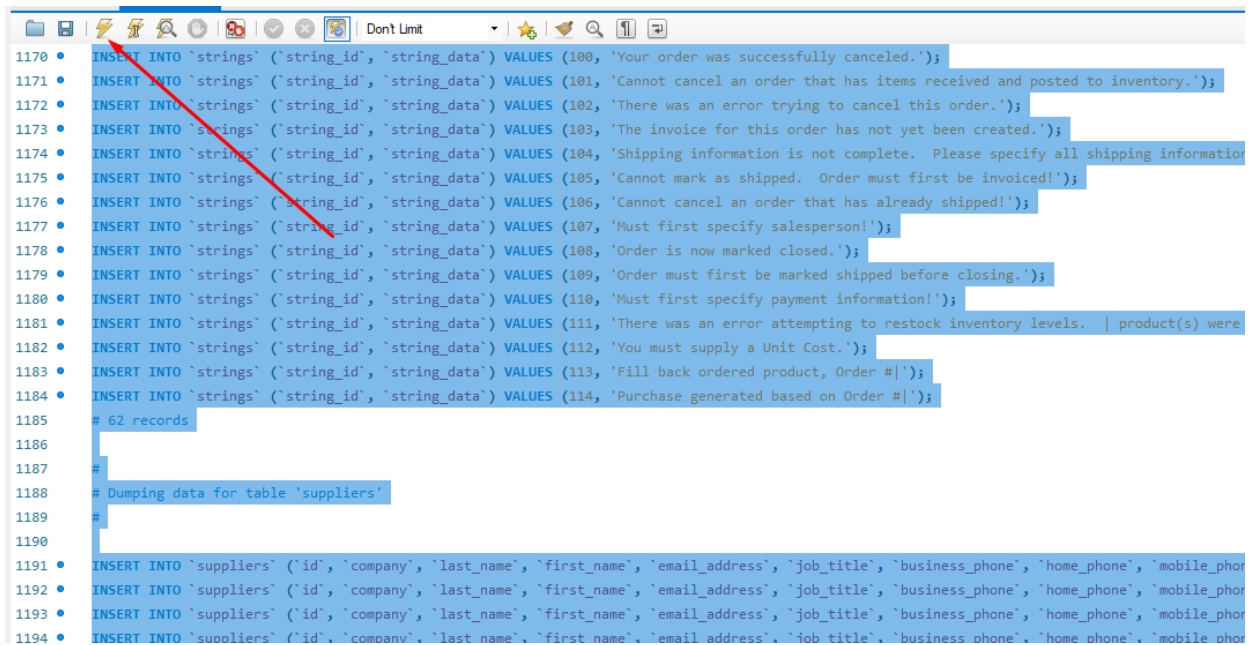
⚠ É importante que seus arquivos tenham exatamente estes nomes e esteja na raiz do projeto! ⚠

- 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

Instruções para restaurar o banco de dados Northwind

1. Faça o download do arquivo de backup [aqui](#) clicando em "Raw", depois clicando com botão direito e selecionando "Salvar como" para salvar o arquivo em seu computador.
2. Abra o arquivo com algum editor de texto, e selecione todo o conteúdo do arquivo usando `CTRL-A`.
3. Abra o MySQL Workbench.
4. Abra uma nova janela de query e cole dentro dela todo o conteúdo do arquivo `northwind.sql`.
5. Selecione todo o código com o atalho `CTRL-A` e depois clique no ícone de trovão para executar a query.



```

1170 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (100, 'Your order was successfully canceled.');
```

```

1171 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (101, 'Cannot cancel an order that has items received and posted to inventory.');
```

```

1172 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (102, 'There was an error trying to cancel this order.');
```

```

1173 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (103, 'The invoice for this order has not yet been created.');
```

```

1174 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (104, 'Shipping information is not complete. Please specify all shipping information.');
```

```

1175 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (105, 'Cannot mark as shipped. Order must first be invoiced!');
```

```

1176 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (106, 'Cannot cancel an order that has already shipped!');
```

```

1177 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (107, 'Must first specify salesperson!');
```

```

1178 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (108, 'Order is now marked closed.');
```

```

1179 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (109, 'Order must first be marked shipped before closing.');
```

```

1180 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (110, 'Must first specify payment information!');
```

```

1181 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (111, 'There was an error attempting to restock inventory levels. | product(s) were');
```

```

1182 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (112, 'You must supply a Unit Cost.');
```

```

1183 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (113, 'Fill back ordered product, Order #|');
```

```

1184 • INSERT INTO `strings` (`string_id`, `string_data`) VALUES (114, 'Purchase generated based on Order #|');
```

```

1185 # 62 records
```

```

1186
```

```

1187 #
```

```

1188 # Dumping data for table 'suppliers'
```

```

1189 #
```

```

1190
```

```

1191 • INSERT INTO `suppliers` (`id`, `company`, `last_name`, `first_name`, `email_address`, `job_title`, `business_phone`, `home_phone`, `mobile_phon
```

```

1192 • INSERT INTO `suppliers` (`id`, `company`, `last_name`, `first_name`, `email_address`, `job_title`, `business_phone`, `home_phone`, `mobile_phon
```

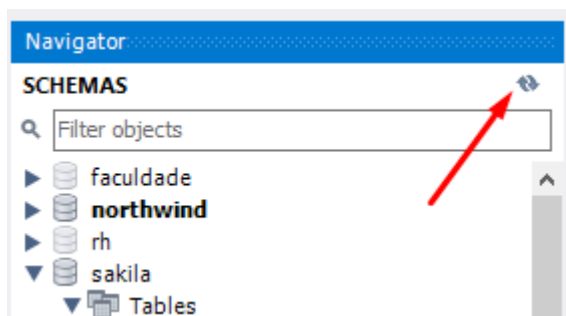
```

1193 • INSERT INTO `suppliers` (`id`, `company`, `last_name`, `first_name`, `email_address`, `job_title`, `business_phone`, `home_phone`, `mobile_phon
```

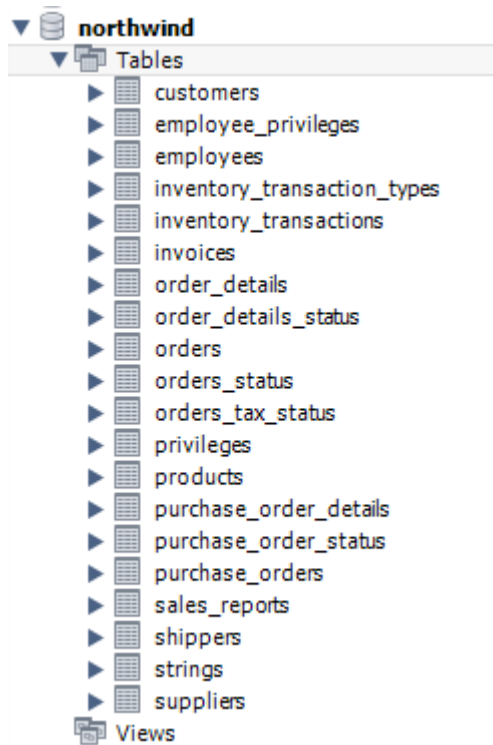
```

1194 • INSERT INTO `suppliers` (`id`, `company`, `last_name`, `first_name`, `email_address`, `job_title`, `business_phone`, `home_phone`, `mobile_phon
```

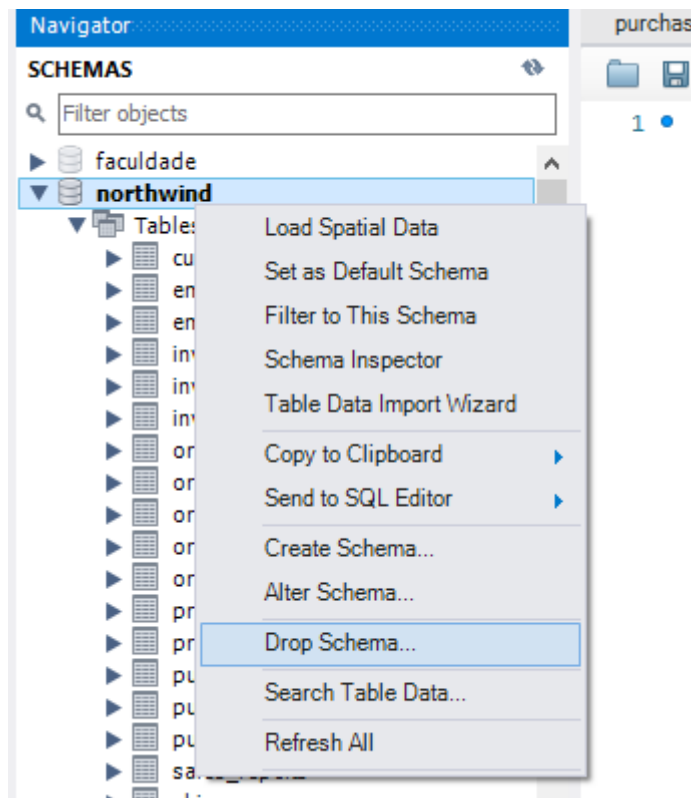
6. Aguarde alguns segundos (espere em torno de 30 segundos antes de tentar fazer algo).
7. Clique no botão apontado na imagem a seguir para atualizar a listagem de banco de dados.



8. Verifique se o banco restaurado possui todas as seguintes tabelas:



9. Clique com botão direito em cada tabela e selecione "Select Rows" e certifique-se que todas as tabelas possuem registros. Caso tenha alguma faltando, faça o passo a seguir. Caso contrário, pode ir para próxima seção.
10. Caso existam tabelas faltando, drope o banco de dados, clicando com o botão direito em cima do banco de dados northwind e selecionando "Drop Schema", e refaça os passos novamente, dessa vez aguardando um tempo maior quando executar o script de restauração.



Instruções para testar suas queries

Para executar localmente os testes, é preciso escrever o seguinte no seu terminal:



```
MYSQL_USER=<SEU_NOME_DE_PESSOA_USUARIA> MYSQL_PASSWORD=<SUA_SENHA> HOSTNAME=<N
```

Ou seja, suponha que para poder acessar a base de dados feita neste projeto você tenha `root` como seu nome de pessoa usuária, `password` como senha e `localhost` como host. Logo, você executaria:

```
MYSQL_USER=root MYSQL_PASSWORD=password HOSTNAME=localhost npm test
```

Usando o exemplo anterior de base, suponha que você não tenha setado uma senha para `root`. Neste caso, você executaria:

```
MYSQL_USER=root MYSQL_PASSWORD= HOSTNAME=localhost npm test
```

 **Não é necessário colocar** `USE northwind; OU SET SQL_SAFE_UPDATES = 0;` no início dos seus arquivos 

Requisitos do projeto

Monte queries para encontrar as informações esperadas pelos desafios:

Lista de Requisitos

Desafios Iniciais

- 1 - Exiba apenas os nomes dos produtos na tabela `products`.
- 2 - Exiba os dados de todas as colunas da tabela `products`.
- 3 - Escreva uma query que exiba os valores da coluna que representa a primary key da tabela `products`.
- 4 - Conte quantos registros existem em `product_name` de `products`.

5 - Monte uma query que exiba os dados da tabela `products` a partir do quarto registro até o décimo terceiro, incluindo tanto um quanto o outro. Obs.: não use `where` ou `order by`.

6 - Exiba os dados das colunas `product_name` e `id` da tabela `products` de maneira que os resultados estejam em ordem alfabética dos nomes.

7 - Mostre apenas os ids dos 5 últimos registros da tabela `products` (a ordenação deve ser baseada na coluna `id`).

8 - Faça uma consulta que retorne três colunas. Na primeira coluna, exiba a soma de `5 + 6` (essa soma deve ser realizada pelo SQL). Na segunda coluna deve haver a palavra "de". E por fim, na terceira coluna, exiba a soma de `2 + 8` (essa soma deve ser realizada pelo SQL). A primeira coluna deve se chamar "A", a segunda coluna deve se chamar "Trybe" e a terceira coluna deve se chamar "eh". Não use colunas pre-existentes, apenas o que for criado na hora.

Desafios sobre filtragem de dados

9 - Mostre todos os valores de `notes` da tabela `purchase_orders` que não são nulos.

10 - Mostre todos os dados da tabela `purchase_orders` em ordem decrescente ordenados por `created_by` em que o `created_by` é maior ou igual a 3. E como critério de desempate para a ordenação, ordene também os resultados pelo `id` de forma crescente.

11 - Exiba os dados de `notes` da tabela `purchase_orders` em que seu valor de "Purchase generated based on Order" está entre 30 e 39, incluindo tanto o valor de 30 quanto de 39.

12 - Mostre as `submitted_date` de `purchase_orders` em que a `submitted_date` é do dia 26 de abril de 2006.

13 - Mostre o `supplier_id` das `purchase_orders` em que o `supplier_id` seja 1 ou 3.

14 - Mostre os `supplier_id` da `purchase_orders` em que o `supplier_id` seja de 1 a 3, incluindo tanto o 1 quanto o 3.

15 - Mostre somente as horas (sem os minutos e os segundos) da `submitted_date` de todos registros de `purchase_orders`. Chame essa coluna de `submitted_hour`.

16 - Exiba a `submitted_date` das `purchase_orders` que estão entre 2006-01-26 00:00:00 e 2006-03-31 23:59:59.

17 - Mostre os registros das colunas `id` e `supplier_id` das `purchase_orders` em que os `supplier_id` sejam tanto 1, ou 3, ou 5, ou 7.

18 - Mostre todos os registros de `purchase_orders` que tem o `supplier_id` igual a 3 e `status_id` igual a 2.

19 - Mostre a quantidade de pedidos que foram feitos na tabela `orders` pelo `employee_id` igual a 5 ou 6, e que foram enviados através do método `shipper_id` igual a 2. Chame a coluna de `orders_count`.

Desafios de manipulação de tabelas

20 - Adicione ao `order_details` uma linha com os seguintes dados: `order_id`: 69, `product_id`: 80, `quantity`: 15.0000, `unit_price`: 15.0000, `discount`: 0, `status_id`: 2, `date_allocated`: NULL, `purchase_order_id`: NULL e `inventory_id`: 129. Obs.: o `id` deve ser incrementado automaticamente.

21 - Adicione, com um único `INSERT`, duas linhas ao `order_details` com os mesmos dados. Esses dados são novamente `order_id`: 69, `product_id`: 80, `quantity`: 15.0000, `unit_price`: 15.0000, `discount`: 0, `status_id`: 2, `date_allocated`: NULL, `purchase_order_id`: NULL e `inventory_id`: 129 (o `id` deve ser incrementado automaticamente).

22 - Atualize os dados de `discount` do `order_details` para 15. (Não é necessário utilizar o `SAFE UPDATE` em sua query).

23 - Atualize os dados de `discount` da tabela `order_details` para 30 cuja `unit_price` seja menor que 10.0000. (Não é necessário utilizar o `SAFE UPDATE` em sua query).

24 - Atualize os dados de `discount` da tabela `order_details` para 45 cuja `unit_price` seja maior que 10.0000 e o `id` seja um número entre 30 a 40. (Não é necessário utilizar o `SAFE UPDATE` em sua query).

25 - Delete todos os dados em que a `unit_price` da tabela `order_details` seja menor que 10.0000.

26 - Delete todos os dados em que a `unit_price` da tabela `order_details` seja maior que 10.0000.

27 - Delete todos os dados da tabela `order_details`.

Depois de terminar o desenvolvimento

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-010-a` .

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



GabrielCoruja Gabriel Dalseco



vinicius-vasconcelos Vinicius-Vasconcelos

Languages

● **JavaScript** 100.0%