



Faça como eu fiz: Refatorando o DataLoader

Durante o vídeo, implementamos o DataLoader de forma mais “curta” para focarmos na funcionalidade, mas agora podemos refatorar o código para uma forma mais usual.

Em `./api/matricula/datasource/matricula.js`, vamos modificar o método `getMatriculasPorEstudante` para que seja a nova instância de `DataLoader`, passando a query para o banco e a array de retorno diretamente nos parâmetros.

```
getMatriculasPorEstudante = new DataLo
const matriculas = await this.db
  .select('*')
  .from('matriculas')
  .whereIn('estudante_id', idsEstuda

return idsEstudantes.map(id =>
```

```
matriculas.filter(matricula =>  
  matricula.estudante_id === id)  
})
```

[COPIAR CÓDIGO](#)

Não podemos Sem esquecer de atualizar o
resolver em

`./api/user/resolvers/userResolvers.js` ; agora, o
método `.load()` passa a ser um método de
`getMatriculasPorEstudante` :

```
User: {  
  matriculas: (parent, _, { dataSource  
    dataSources.matriculasAPI.getMatri  
      .load(parent.id)  
  })  
}
```

[COPIAR CÓDIGO](#)

Feito!

