

Para saber mais: Transações com Sequelize

O Sequelize não implementa transações nas queries por padrão; mas é muito aconselhável que você as utilize, especialmente em produção.

Existem duas formas de fazer isso utilizando os métodos do Sequelize: a primeira é utilizando transações não gerenciadas (*unmanaged transactions*), onde quem está desenvolvendo é responsável por chamar os métodos apropriados de *rollback* e *commit*:

```
const transacao = await sequelize.transaction();

try {
  const personagem = await Personagem.create({
    nome: 'Bart',
    sobrenome: 'Simpson'
  }, { transaction: transacao });
  await personagem.addParente({
    nome: 'Lisa',
    sobrenome: 'Simpson'
  }, { transaction: transacao });
  await transacao.commit();
} catch (error) {
  await transacao.rollback();
}
```

No exemplo acima (da própria documentação do Sequelize) os métodos `t.commit()` e `t.rollback()` foram adicionados manualmente.

A outra forma, como fizemos no vídeo, foi utilizando transações gerenciadas (*managed transactions*) onde toda a operação a ser feita no banco é passada como *callback* do método `sequelize.transaction()`. Nesse caso, e como foi feito no código do nosso projeto, não há a necessidade de adicionar manualmente os métodos `t.commit()` e `t.rollback()`.

Segue o link para a [documentação \(https://sequelize.org/master/manual/transactions.html\)](https://sequelize.org/master/manual/transactions.html) do Sequelize sobre transações.