

Faça como eu fiz

Durante o vídeo criamos um novo método para restaurar registros que foram apagados da tabela Pessoas usando `_soft delete_`. Uma vez que adicionamos a opção `{ paranoid: true }` em todos os modelos, vamos incluir também este método nos outros controladores: Níveis, Turmas e também as matrículas que estão no mesmo controlador de Pessoas.

api/models/PessoasController.js

```
static async restauraMatricula(req, res) {
  const { estudanteId, matriculaId } = req.params
  try {
    await database.Matriculas.restore({
      where: {
        id: Number(matriculaId),
        estudante_id: Number(estudanteId)
      }
    })
    return res.status(200).json({ mensagem: `id ${id} restaurado` })
  } catch (error) {
    return res.status(500).json(error.message)
  }
}
```

api/models/NivelController.js

```
static async restauraNivel(req, res) {
  const { id } = req.params
  try {
    await database.Niveis.restore( {where: { id: Number(id) } } )
    return res.status(200).json({ mensagem: `id ${id} restaurado` })
  } catch (error) {
    return res.status(500).json(error.message)
  }
}
```

api/models/TurmasController.js

```
static async restauraTurma(req, res) {
  const { id } = req.params
  try {
    await database.Turmas.restore( {where: { id: Number(id) } } )
    return res.status(200).json({ mensagem: `id ${id} restaurado` })
  } catch (error) {
    return res.status(500).json(error.message)
  }
}
```

E não podemos esquecer de atualizar os arquivos de rotas para que a API tenha uma rota de restauração de registro para cada uma das tabelas:

peessoasRoute.js

```
.post('/peessoas/:id/restaura', PessoaController.restauraPessoa)
```

```
.post('/peessoas/:estudanteId/matricula/:matriculaId/restaura', PessoaController.restauraMatricula)
```



niveisRoute.js

```
.post('/niveis/:id/restaura', NivelController.restauraNivel)
```

turmasRoute.js

```
.post('/turmas/:id/restaura', TurmaController.restauraTurma)
```

Agora é possível restaurar todos os registros de todas as tabelas.