



PROF^o JADER JACKSON

ESPECIALISTA DESENVOLVIMENTO WEB – UFC

MIGRATIONS



As migrações (migrations) no contexto do Node.js, especialmente quando se trata de bancos de dados, são um conceito importante para gerenciar as alterações na estrutura do banco de dados de uma forma controlada e automatizada. São frequentemente usadas em conjunto com frameworks e bibliotecas de bancos de dados, como o Sequelize para bancos de dados SQL ou o Mongoose para bancos de dados NoSQL.



PRINCIPAIS OBJETIVOS E BENEFÍCIOS DAS MIGRAÇÕES NODE.JS



1. Controle de versão do esquema do banco de dados:

À medida que o projeto evolui, pode ser necessário fazer alterações na estrutura do banco de dados, como adicionar, remover ou modificar tabelas, colunas, índices etc... As migrações permitem que você rastreie essas alterações e as aplique de forma ordenada e controlada em diferentes ambientes (desenvolvimento, teste, produção).



PRINCIPAIS OBJETIVOS E BENEFÍCIOS DA MIGRAÇÕES NODE.JS



2. Colaboração em equipe:

Quando várias pessoas estão trabalhando no mesmo projeto, é importante ter um método para coordenar e sincronizar as alterações no banco de dados. As migrações permitem que todos na equipe estejam cientes das mudanças e possam aplicá-las em seus ambientes locais de maneira consistente.



PRINCIPAIS OBJETIVOS E BENEFÍCIOS DA MIGRAÇÕES NODE.JS



3. Reprodutibilidade: As migrações garantem que as alterações no banco de dados possam ser reproduzidas em diferentes ambientes, evitando discrepâncias entre as bases de dados em desenvolvimento e produção, por exemplo.



PRINCIPAIS OBJETIVOS E BENEFÍCIOS DA MIGRAÇÕES NODE.JS



4. Rollback (reverter mudanças): Se uma migração causar problemas ou se você precisar voltar para uma versão anterior da estrutura do banco de dados, as migrações normalmente permitem reverter as alterações aplicando migrações anteriores.



PRINCIPAIS OBJETIVOS E BENEFÍCIOS DA MIGRAÇÕES NODE.JS



5. Facilitação da implantação: Ao automatizar o processo de aplicação de alterações no banco de dados, as migrações ajudam a simplificar a implantação de novas versões do aplicativo, garantindo que o esquema do banco de dados esteja sempre em sincronia com o código.



PRINCIPAIS OBJETIVOS E BENEFÍCIOS DA MIGRAÇÕES NODE.JS



6. Histórico e auditoria: As migrações mantêm um histórico das alterações feitas no esquema do banco de dados ao longo do tempo, o que pode ser útil para fins de auditoria e rastreamento.

Em resumo, as migrações no Node.js são uma prática recomendada para gerenciar as mudanças na estrutura do banco de dados, garantindo consistência, controle e facilidade de colaboração em projetos de desenvolvimento de software.



IMPLEMENTADO A MIGRATION



OBS. Certifique-se que você tem o diretório (pasta) migration.

`npm sequelize-cli init`



IMPLEMENTADO A MIGRATION



1. CRIE UM ARQUIVO na raiz do projeto COM SEGUINTE NOME:

`.sequelizerc`

2. DIGITE ESSA INSTRUÇÃO DENTRO DO ARQUIVO `.sequelizerc`

`//Caminho dos diretórios`

`const path = require('path')`

`//indica o caminho`

`module.exports = {`

`'config': path.resolve('db', 'config', 'database.js'),`

`'models-path': path.resolve('db', 'models'),`

`'seeders-path': path.resolve('db', 'seeders'),`

`'migrations-path': path.resolve('db', 'migrations')`



CRIANDO A MIGRATION



//CRIAR A MIGRATION DA PÁGINA HOME DO CONTEÚDO DO TOPO

`npx sequelize-cli migration:generate --name create-homes-tops`



EXECUTANDO A MIGRATION



```
npx sequelize-cli db:migrate
```



NODE.JS



Atividade: Implemente uma rota que recupere registro do banco de dados.

