08

Para saber mais: Outras ferramentas

Para este projeto, utilizamos principalmente as ferramentas do ApolloServer para construir uma API GraphQL, com as libs `apollo-server` para subir o servidor e as `apollo-datasource` para conectar com as bases de dados.

Apesar de ser uma ferramenta bem completa, os serviços do Apollo não são os únicos disponíveis para trabalharmos com GraphQL. Algumas outras opções são:

Faunadb (https://fauna.com/)

Plataforma de database que oferece soluções para aplicações *serverless*, utilizando o GraphQL como linguagem de query.

Express (https://expressjs.com/)

Bastante utilizado em APIs REST com NodeJS, o Express também pode ser utilizado com GraphQL. A página inicial do GraphQL tem este tutorial (https://graphql.org/graphql-js/running-an-express-graphql-server/) (em inglês) que utiliza as libs `express` e `express-graphql` para subir um servidor.

graphql-yoga (https://github.com/prisma-labs/graphql-yoga)

Servidor GraphQL focado em performance e com um setup parecido com o do Apollo. É baseado em parte nas libs do Apollo, Express e GraphQL Tools.

Prisma (https://www.prisma.io/)

Ferramenta para interface com bases de dados (MySQL, SQLite ou Postgres) e que pode ser utilizada tanto para construir APIs em GraphQL quanto REST. Tem seu próprio sistema para trabalhar com query building e escrever schemas GraphQL, além de se propor a substituir os ORMs tradicionais, com gerenciamento de migrações e outras ferramentas como paginação, filtros e transações. Para saber mais sobre o que são ORMs, confira este Alura+ (https://youtu.be/x39vqeBTUmE) no canal da Alura no YouTube.

Independentemente de qual ferramenta escolher, os conceitos-chave do GraphQL — definição de schema, resolução de campos, tipos scalar, objeto e raiz, funcionamento do servidor GraphQL e montagem de queries — permanecem.