

Preparando o ambiente

Para este curso, precisaremos instalar algumas ferramentas - caso você ainda não tenha feito isso anteriormente:

MySQL

É o banco SQL que utilizaremos para persistir os dados da nossa API. [Este tutorial](https://www.alura.com.br/artigos/mysql-do-download-e-instalacao-ate-sua-primeira-tabela) (<https://www.alura.com.br/artigos/mysql-do-download-e-instalacao-ate-sua-primeira-tabela>) contem o passo-a-passo para instalação no Windows e alguns comandos iniciais. Para instalação no MacOS [baixe o arquivo DMG](https://downloads.mysql.com/archives/community/) (<https://downloads.mysql.com/archives/community/>) e em seguida [siga as instruções de instalação](https://dev.mysql.com/doc/mysql-osx-excerpt/8.0/en/osx-installation-pkg.html) (<https://dev.mysql.com/doc/mysql-osx-excerpt/8.0/en/osx-installation-pkg.html>); se estiver usando Linux (Debian/Ubuntu), você pode [baixar direto via terminal](https://ubuntu.com/server/docs/databases-mysql) (<https://ubuntu.com/server/docs/databases-mysql>) com `sudo apt install mysql-server` .

Caso você já tenha o PostgreSQL instalado e prefira utilizá-lo no lugar do MySQL, pode pular esta etapa; porém será necessário fazer pequenos ajustes no script (ver próximas seções) e instalar a dependência do Postgres no projeto com `npm install pg` .

Cliente para gerenciamento de databases

Durante o curso faremos consultas ao banco direto pelo terminal, porém se você quiser, pode fazer isso usando um cliente como o [MySQL Workbench](#)

[\(https://dev.mysql.com/downloads/workbench/\)](https://dev.mysql.com/downloads/workbench/) ou o [dBeaver](https://dbeaver.io/)

NodeJS

Se você já fez outros cursos dessa formação, o NodeJS já deve estar instalado. Você pode conferir com o comando `node --version` ou `node -v` no terminal - caso retorne um número de versão, como por exemplo, `v14.10.1`, o NodeJS já está instalado. Caso precise instalar, os links estão na [página oficial](https://nodejs.org/en/download/) (<https://nodejs.org/en/download/>) do NodeJS. Este curso foi desenvolvido usando a versão 14.10.1 e o NPM versão 6.14.8.

Instalação do projeto

Baixe o repositório inicial (<https://github.com/alura-cursos/2036-graphql-prisma2> (<https://github.com/alura-cursos/2036-graphql-prisma2>)), navegue até a pasta pelo terminal e instale as dependências iniciais com `npm install`.

Por enquanto, utilizamos as seguintes dependências: `apollo-server`, `graphql`, `mysql2` e `nodemon`. Você pode ver como elas estão sendo utilizadas no arquivo `src/index.js`.

Já é possível subir o servidor GraphQL com o comando `npm start` e conferir no Playground em <http://localhost:4000> (<http://localhost:4000>). Caso você já tenha outro projeto em GraphQL rodando nesta porta (4000 é a porta padrão do GraphQL), é só modificar o número da porta no arquivo `src/index.js`:

```
server.listen({ port: 4001 }, () => console.log(`Servidor pronto
```

[COPIAR CÓDIGO](#)

Vamos fazer o restante das instalações e a criação do banco durante o desenvolvimento do projeto.

Além do `src/index.js`, deixamos alguns arquivos prontos: o `.gitignore` já com a pasta `node_modules` indicada e o arquivo `scriptSQL.sql` que vamos usar em seguida para criar e popular o banco que usaremos no projeto.

Agora podemos começar!