

## Desenvolvendo uma API em Node.js consumindo MySQL como Banco de Dados

### MySQL:

- Instalar o MySQL em: <https://dev.mysql.com/downloads/>
- Usando o MySQL Workbench, desenvolva e execute o script Contato.sql que está no Git na pasta Aulas/Programação Orientada à Objetos/Aula 03
- Certifique-se que o banco e a tabela foram criados e insira ao menos dois registros estão na tabela
- Se a sua versão de MySQL é igual ou superior a 8.0, executar o seguinte comando: `ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'Sua senha aqui!'`

### VS Code:

- Acesse o terminal (cmd) do windows e vá até a pasta C:\Projetos\AgendaContatos\backend
- Para abrir o VS Code com esse projeto digite o comando: code .
- No terminal execute o comando: yarn add mysql  
(Esse comando irá instalar o driver para comunicação entre Node.js e MySQL)
- No terminal execute o comando: yarn add nodemon -D  
(O Nodemon é um pacote utilizado para sempre “restartar” o servidor Node.js quando algum arquivo for alterado. E o parâmetro -D significa que o nodemon será utilizado apenas em ambiente de Desenv)
- Alterar o arquivo package.json, incluindo o seguinte código abaixo de "license":

```
"scripts": {  
  "dev": "nodemon src/server.js"  
}
```
- No terminal executar o comando: yarn dev  
(O comando “yarn dev” substitui o comando “node server.js”, ou seja, o “yarn dev” está executando o nodemon e o nodemon executa o nosso servidor Node server.js e qualquer alteração que houver em algum arquivo do servidor o nodemon automaticamente reinicia o servidor. Faça um teste altere alguma linha no arquivo routes.js ou server.js e acompanhe o que acontece no terminal)
- Alterar o conteúdo dos arquivos server.js e routes.js, conforme o conteúdo que está na pasta Aula 03 no Git
- Apague o arquivo bancoDadosContato.json e retire as suas referências no routes.js

### Postman:

- Crie uma pasta no Postman e chame-a de AgendaContatos

- Dentro dessa pasta crie 5 requests para as seguintes APIs que desenvolvemos em Node.js com o VS Code
- As requests devem ser essas:
  - GET: <http://localhost:3000/contatos>
  - GET: <http://localhost:3000/contatos/1>
  - POST: <http://localhost:3000/contatos>
  - DEL: <http://localhost:3000/contatos>
  - PUT: <http://localhost:3000/contatos/2>
- Vamos testar cada API e vamos dar SELECT nas tabelas do Banco de Dados e vamos acompanhar se tudo está funcionando!