**Implementando o backend da aplicação**

Implemente uma aplicação em NodeJS com rotas, models e controllers, conforme Abaixo:

Crie um banco de dados com o mongodb chamado bdsi4oano

**Cadastro de Usuários**

**Model**

Conterá o campo fotoUsuario, nome, email, senha e cpf como string, grave os campos createdAt e updatedAt

**Rotas**

Crie as rotas com os respectivos métodos:

|  |  |  |
| --- | --- | --- |
| **Método Http** | **Rota** | **Método do controller** |
| get | /usuarios | index – lista todos os usuários |
| get | /usuarios/:id | show – lista somente um usuário |
| post | /usuarios | store – grava o usuário |
| put | /usuarios/:id | update – altera o usuário |
| delete | /usuarios/:id | delete – apaga o usuário |

**Controller**

**Métodos store e update**

Nestes métodos deverá ser feita as seguintes validações:

Se existir o cpf, retornar o code status 401 e um json com a mensagem: CPF existente. Exemplo:

return status(401).json({ error: ‘CPF existente‘})

Se existir o email, retornar o code status 401 e um json com a mensagem: Email existente. Exemplo:

return status(401).json({ error: ‘Email existente‘})

Retornar para o frontend somente os campos fotoUsuario, nome, email, senha e cpf.

**Método index**

Deverá retornar para o frontend todos os usuários, mas somente os campos fotoUsuario, nome, email e senha.

**Método show**

Deverá receber o id do usuário e retornar para o frontend o usuário com o respectivo id e somente os campos fotoUsuario, nome, email e senha.

**Método delete**

Deverá receber o id do usuário e excluí-lo do banco de dados.