


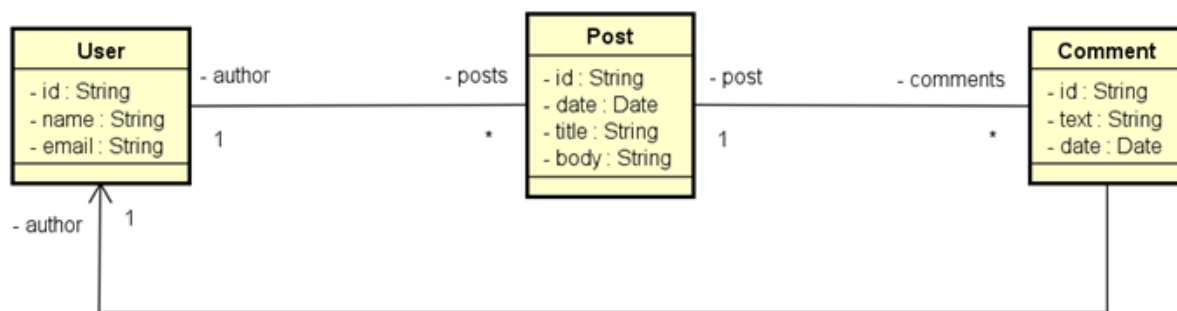
Instalando o servidor do MongoDB via Docker

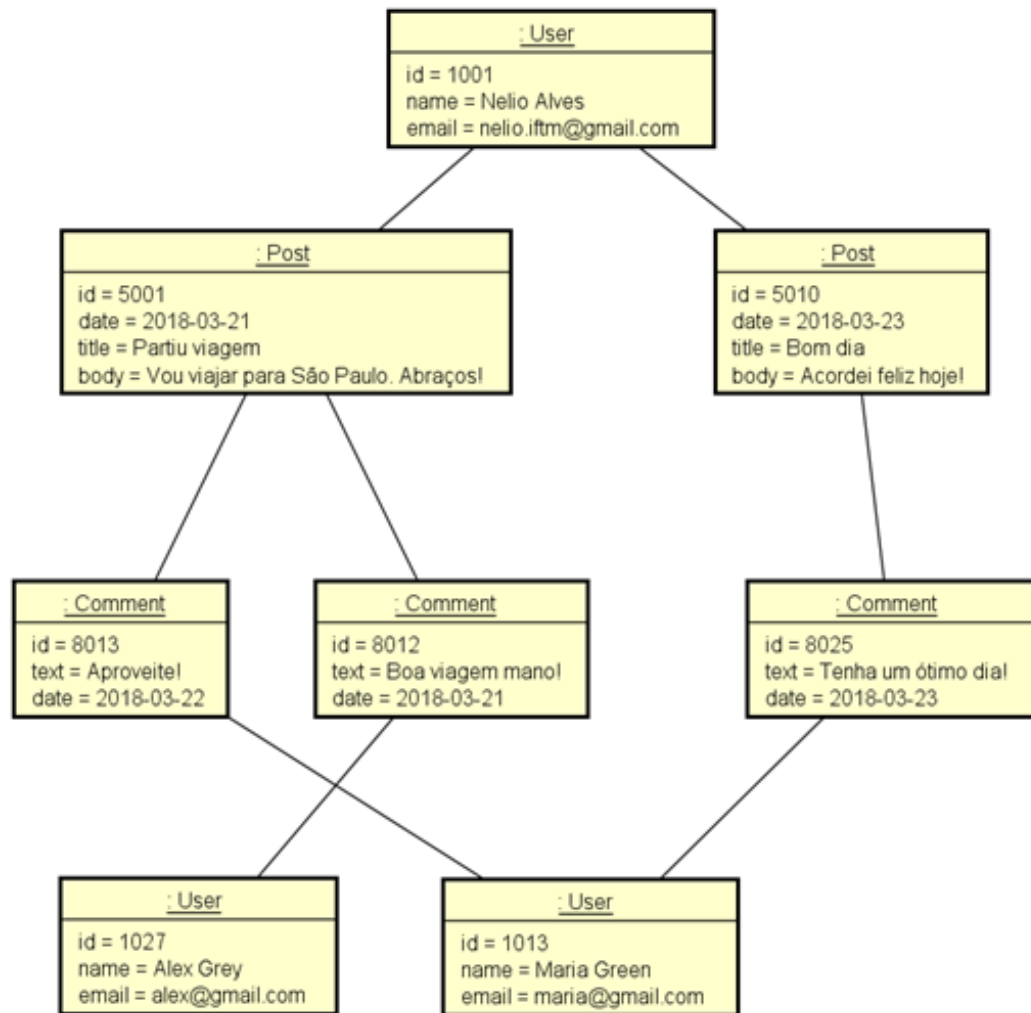
👤 Criado por	 Cleber Alves Feitosa
🕒 Hora da criação	@14 de novembro de 2023 14:15
🏷 Tags	

Projeto Posts (MongoDB)

Material de apoio para alunos

Modelo Conceitual





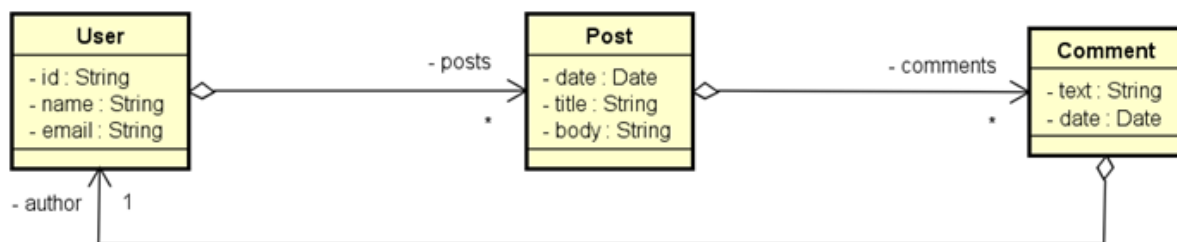
Como seria em um banco de dados relacional

user		
id	name	email
x-o-x	x-o-x	x-o-x
1001	Maria Brown	maria@gmail.com
x-o-x	x-o-x	x-o-x
x-o-x	x-o-x	x-o-x
1013	Alex Green	alex@gmail.com
x-o-x	x-o-x	x-o-x
1027	Bob Grey	bob]@gmail.com

post				
id	date	title	body	author_id
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
5001	2018-03-21	Partiu viagem	Vou viajar para São Paulo. Abraços!	1001
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
5010	2018-03-23	Bom dia	Acordei feliz hoje!	1001
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X

comment				
id	text	date	post_id	author_id
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
8012	Boa viagem mano!	2018-03-21	5001	1013
8013	Aproveite!	2018-03-22	5001	1027
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X
8025	Tenha um ótimo dia!	2018-03-23	5010	1013
X-O-X	X-O-X	X-O-X	X-O-X	X-O-X

Opção 1 de agregação



```

{
  "id": "1001",
  "name": "Maria Brown",
  "email": "maria@gmail.com",
  "posts": [
    {
      "date": "2018-03-21",
      "title": "Partiu viagem",
      "body": "Vou viajar para São Paulo. Abraços!",
      "comments": [
        {
          "text": "Boa viagem mano!",

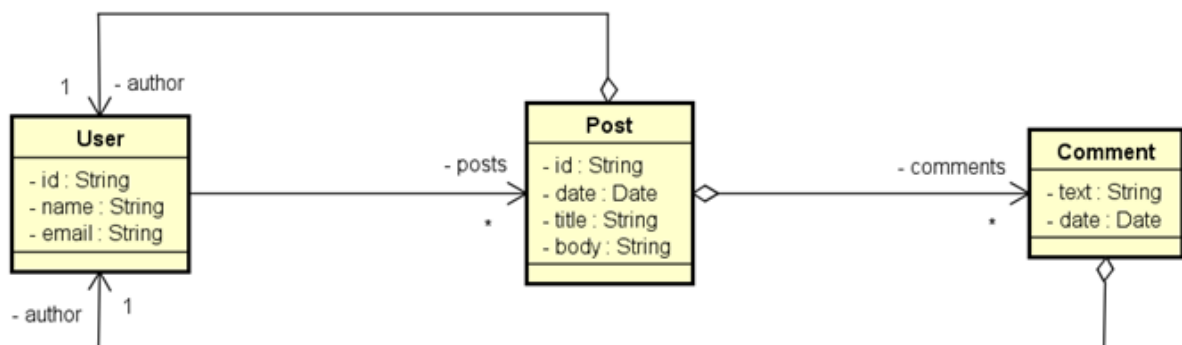
```

```

        "date": "2018-03-21",
        "author": {
          "id": "1013",
          "name": "Alex Green"
        }
      },
      {
        "text": "Aproveite!",
        "date": "2018-03-22",
        "author": {
          "id": "1027",
          "name": "Bob Grey"
        }
      }
    ]
  },
  {
    "date": "2018-03-23",
    "title": "Bom dia",
    "body": "Acordei feliz hoje!",
    "comments": [
      {
        "text": "Tenha um ótimo dia!",
        "date": "2018-03-23",
        "author": {
          "id": "1013",
          "name": "Alex Green"
        }
      }
    ]
  }
]
}

```

Opção 2 de agregação



```

{
  "id": "1001",

```

```

    "name": "Maria Brown",
    "email": "maria@gmail.com",
    "posts": ["5001", "5010"]
  }
  {
    "id": "5001",
    "date": "2018-03-21",
    "title": "Partiu viagem",
    "body": "Vou viajar para São Paulo. Abraços!",
    "author": {
      "id": "1001",
      "name": "Maria Brown"
    },
    "comments": [
      {
        "text": "Boa viagem mano!",
        "date": "2018-03-21",
        "author": {
          "id": "1013",
          "name": "Alex Green"
        }
      },
      {
        "text": "Aproveite!",
        "date": "2018-03-22",
        "author": {
          "id": "1027",
          "name": "Bob Grey"
        }
      }
    ]
  }
}
{
  "id": "5010",
  "date": "2018-03-23",
  "title": "Bom dia",
  "body": "Acordei feliz hoje!",
  "author": {
    "id": "1001",
    "name": "Maria Brown"
  },
  "comments": [
    {
      "text": "Tenha um ótimo dia!",
      "date": "2018-03-23",
      "author": {
        "id": "1013",
        "name": "Alex Green"
      }
    }
  ]
}
}

```

Container Docker do MongoDB para desenvolvimento

```
docker run -d -p 27017:27017 -v /data/db --name mongo1 mongo:latest -bionic
```

Vamos verificar se está rodando docker ps

testando

docker exec -it mongo1 bash

Pronto feito isso, vamos entra no nosso container, basta digitar **mongo**

Se você digitar help, ele mostra todos os comando de ajuda do Mongo.

Se digitarmos show dbs, veremos os bancos de dados exemplos que já vem no Mongo.

Beleza...

Para sair do teminal do Mongo digitamos, exit, e voltamos para o teminal linux.

Exit mais uma vez saímos do terminal do Linux, e o servidor está funcionando na máquina local.

Para inspecionarmos nosso conteinner, digitamos **docker inspect** e informamos o id do nosso conteinner, podemos informar apenas os três primeiros digitos.

```

C:\WINDOWS\system32> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
75d3b8959c74   mongo:4.4.3-bionic   "docker-entrypoint.s..."   27 minutes ago   Up 27 minutes   0.0.0.0:27017->27017/tcp   mongo1
PS C:\WINDOWS\system32> docker inspect 75d
docker: 'inspect' is not a docker command.
See 'docker --help'
PS C:\WINDOWS\system32> docker inspect 75d
[
  {
    "Id": "75d3b8959c7453153efae479251160ec0afadd51df0d4c186404d7bfb5aa282",
    "Created": "2023-11-14T19:08:58.750231406Z",
    "Path": "docker-entrypoint.sh",
    "Args": [
      "mongod"
    ],
    "State": {
      "Status": "running",
      "Running": true,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 42239,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2023-11-14T19:09:01.582432614Z",
      "FinishedAt": "0001-01-01T00:00:00Z"
    },
    "Image": "sha256:c8a8e14b1fd68aedb435fec2a6eaa326cf5633fc57b7e28b5cc37d938ead9edd",
    "ResolvConfPath": "/var/lib/docker/containers/75d3b8959c7453153efae479251160ec0afadd51df0d4c186404d7bfb5aa282/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/75d3b8959c7453153efae479251160ec0afadd51df0d4c186404d7bfb5aa282/hostname",
    "HostsPath": "/var/lib/docker/containers/75d3b8959c7453153efae479251160ec0afadd51df0d4c186404d7bfb5aa282/hosts",
    "LogPath": "/var/lib/docker/containers/75d3b8959c7453153efae479251160ec0afadd51df0d4c186404d7bfb5aa282/json.log",
    "Name": "/mongo1",
    "RestartCount": 0,
    "Driver": "overlay2",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": null,
    "HostConfig": {
      "Binds": null,
      "ContainerIDFile": "",
      "LogConfig": {
        "Type": "json-file",
        "Config": {}
      },
      "NetworkMode": "default",
      "PortBindings": {
        "27017/tcp": [
          {
            "HostIp": "",
            "HostPort": "27017"
          }
        ]
      },
      "RestartPolicy": {
        "Name": "no",
        "MaximumRetryCount": 0
      }
    }
  }
]

```

Beleza, agora vamos procurar uma informação chamada “Mounts”:
 essa informação vai nos dizer que nossa unidade foi mapeada para a máquina hospedeira

```

"Mounts": [
  {
    "Type": "volume",
    "Name": "1427alfca1d8cb2f1495e1f6688cb170bce9c8ff972e3d478e8d5db292f8755b",
    "Source": "/var/lib/docker/volumes/1427alfca1d8cb2f1495e1f6688cb170bce9c8ff972e3d478e8d5db292f8755b/_data",
    "Destination": "/data/db",
    "Driver": "local",
    "Mode": "",
    "RW": true,
    "Propagation": ""
  }
],

```

Na pasta source teremos acesso aos dados gerados e manipulados pelo nosso MongoDB, porém, isso só funciona para o Linux e o Mac, para o Windows precisamos fazer um macetizinho aqui oh:

Localização dos volumes Docker no Windows

<https://stackoverflow.com/questions/43181654/locating-data-volumes-in-docker-desktop-windows>

\\wsl\$\docker-desktop-data\data\docker\volumes

Postman collection

<https://www.getpostman.com/collections/2335420a92e700cc24fa>