

UNIVERSIDADE DE BRASÍLIA

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Disciplina: Banco de Dados

**Projeto de Banco de Dados**

**Discentes:** Luís Felipe de Assis Medeiros - 170195503

Valentin Ferreira Paes - 180037901

Repositório do Projeto: [luismed05/ProjetoFinalDB (github.com)](https://github.com/luismed05/ProjetoFinalDB)

**Introdução**

O projeto batizado como “CoronaHelp” é um sistema de atendimento emergencial para pacientes com sintomas de COVID-19. Por meio de uma interface, o usuário pode informar sua localização, seu plano de saúde e o nível de sua urgência. O sistema escolherá o melhor hospital de acordo com o plano de saúde do usuário, sua localização, e principalmente o nível de urgência. Quando chegar a um hospital, o paciente será alocado a um dos leitos disponíveis.

A estrutura CRUD escolhida para o projeto foi uma arquitetura MVC com a Model e Controller sendo feita na Framework NodeJS e o View com a Framework ReactJS, onde o NodeJS funciona como uma API acessando o Banco de Dados e fornecendo informações a View. E o SGBD usado foi o MYSQL.

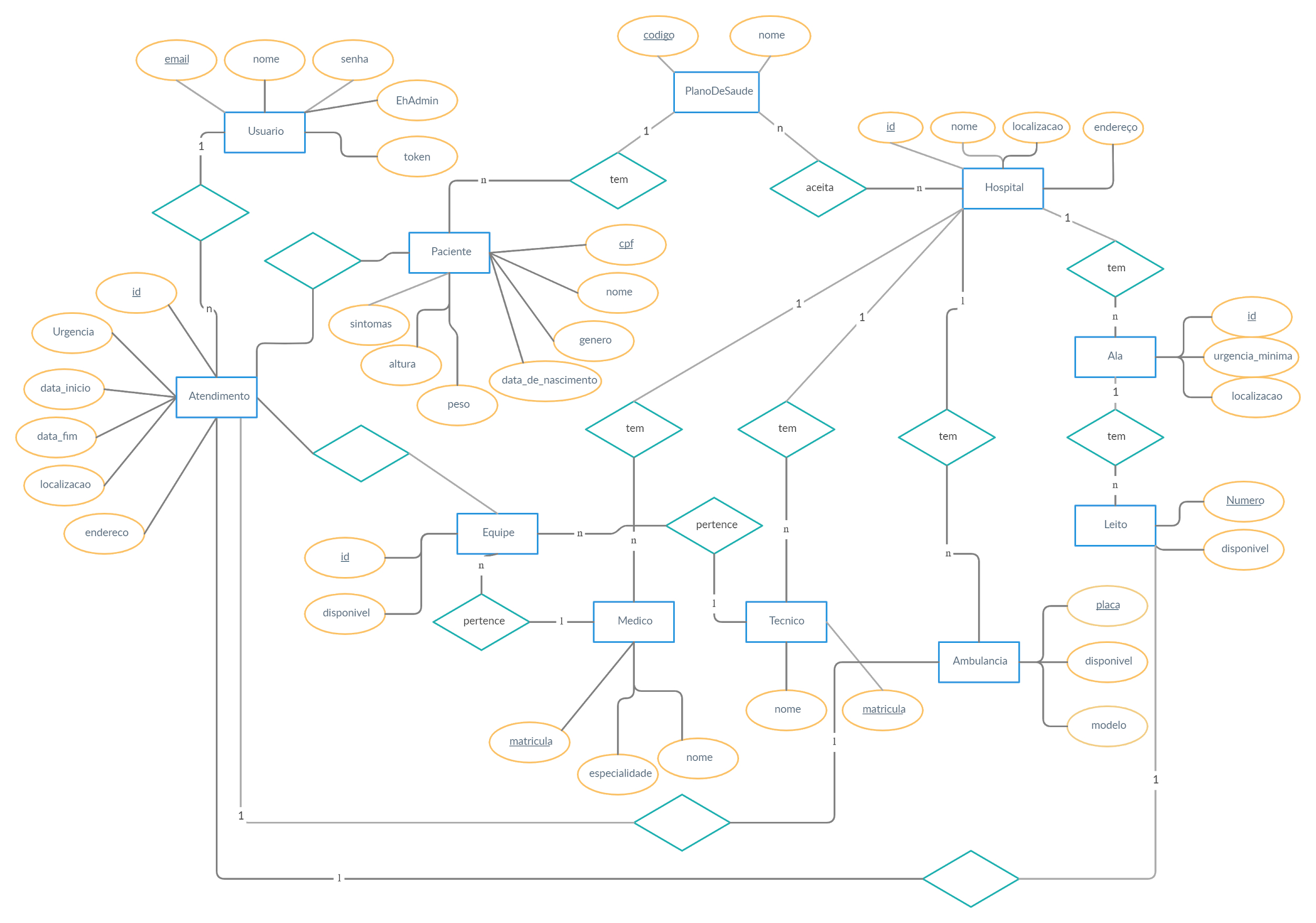
Model & Controller:

|  |
| --- |
| const Usuario = function(usuario){  this.email = usuario.email,  this.senha = usuario.senha,  this.nome = usuario.nome,  this.token = usuario.tokenAcesso,  this.admin = usuario.Ehadmin };  //Metodo da Controller para trazer todos os usuários cadastrados Usuario.index = async (req, result) => {  await sql.query("SELECT \* FROM Usuario", (err,res) => {  if(err){  if(debug == true) console.log(err)  console.log('erro ao realizar consulta');  result.status(500).send({message: "erro ao realizar consulta no banco"});  return false;  }  console.log("consulta realizada com sucesso");  result.status(200).send(res)  }) } |

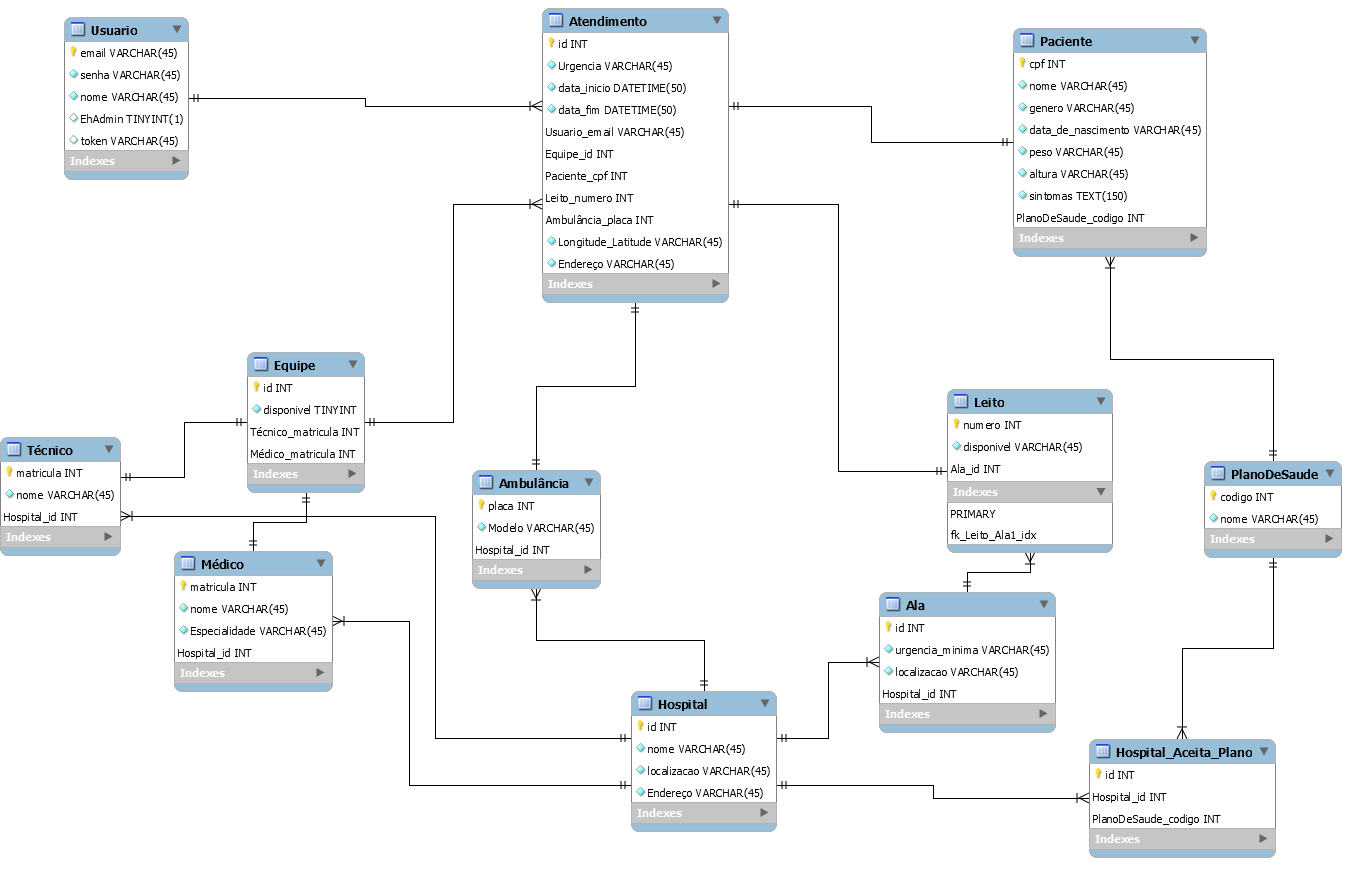
Exemplo da model de usuário e controller para trazer todos os usuários, sql.query é comando usado para rodar comandos DML dentro do Banco.

Logo após o sql.query é usado o Comando DML SELECT para buscar todas as informações de usuários.

Modelo de Entidade Relacionamento

****

Modelo Relacional

****

**Consultas em Álgebra relacional**

Pesquisa para trazer o nome dos hospital, a localização da ala e o número do leito:

Algebra Relacional:

hospital.nome, Ala.localizacao, leito.numero(hospital.id = Ala.hospital\_id (hospital x Ala) ^

Ala.id = leito.Ala\_id (leito x Ala) ^ disponivel = 1 (leito))(hospital,Ala,leito)

Codigo DML:

|  |
| --- |
| SELECT nome as 'Nome do Hospital',   a.localizacao as 'Setor de localização',   l.Numero as 'codigo do leito' FROM Hospital AS h JOIN Ala AS a ON h.id = a.hospital\_id JOIN Leito AS l ON a.id = l.Ala\_id WHERE (l.disponivel = "sim");  Retorno: |

Pesquisa para trazer o número do leito disponível:

Algebra Relacional:

Numero (Ala.id = leito.Ala\_id (leito x Ala) ^

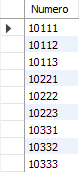
hospital.id = Ala.hospital\_id (hospital x Ala) ^

disponivel = 1 (leito))(leito)

Codigo DML:

|  |
| --- |
| SELECT l.Numero  FROM Leito AS l  JOIN Ala AS a ON a.id = l.Ala\_id  WHERE (l.disponivel = "sim" AND Hospital\_id = a.hospital\_id) LIMIT 1; |

Retorno:



Pesquisa para consultar codigo da equipe e nome do médico responsável:

Algebra Relacional:

Equipe.id, medico.nome (hopital.id = medico.hospital\_id (hospital x medico) ^

Equipe.medico\_matricula = medico.matricula (medico x Equipe) ^

disponivel = 1 (equipe))(Equipe,medico)

Codigo DML:

|  |
| --- |
| SELECT e.id as 'Codigo da Equipe',   m.nome as 'Nome do medico Responsavel'  FROM Hospital AS h INNER JOIN Medico AS m ON h.id = m.hospital\_id INNER JOIN Equipe AS e ON e.medico\_matricula = m.matricula WHERE (e.disponivel = 1);  Retorno: |

