**Back End**

1. **O que é Back-End**
2. **JavaScript**
3. **Node.js**
4. **Sequelize**
5. **Cors**
6. **Express**
7. **Postman**
8. **Mysql Workbench**
9. **Xampp**

**1. O que é Back-End**

A tradução de Back-End é denominada 'Processamento Interno', como o nome já diz, é a parte interna de uma aplicação. Pode ficar meio abstrato num primeiro momento, mas pense que para conseguir usar o Facebook no dia a dia, os dados (as informações) do seu perfil, amigos e publicações precisam estar salvos em algum lugar e serem processados a partir dele, sendo este lugar um banco de dados.

O Back-End trabalha em boa parte dos casos fazendo a ponte entre os dados que vem do navegador rumo ao banco de dados e vice-versa, sempre aplicando as devidas regras de negócio, validações e garantias num ambiente restrito ao usuário final (ou seja, onde ele não consegue acessar ou manipular algo).

Algumas linguagens de Back-End:

* PHP
* Ruby
* Python
* JavaScript
* Java
* C#
* C++
* C

**2. JavaScript**

O JavaScript é uma linguagem de script que trazendo um site dinâmico ( junto ao HTML e ao CSS ) ,e muito útil em aplicativos do lado do servidor, tais como a criação de servidores web.



O Jogo dos Dinossauros, um exemplo de um jogo de navegador web integrado criado usando JavaScript

**3. Node.js**

O node.js é um ambiente de execução do JavaScript. Com ele é possível criar aplicações em JavaScript sem a necessidade de um browser, criando a opção de utilizar a linguagem no Back-End.

**4. Sequelize**

É um framework do JavaScript que permite relacionar o banco com o Back-End. Ele faz a abstração do Banco de Dados, podendo manipular os Dados no Back-End sem precisar entrar no Banco.

Suporta diversos Bancos de Dados como PostgreSQL, MySQL, SQLite e MSSQL.

**5. Cors**

CORS é o acrônimo de Cross-origin Resource Sharing (Compartilhamento de Recursos de Origem Cruzada). Ele informa ao navegador se um recurso pode ou não ser acessado. Todo navegador possui um mecanismo de segurança chamado de same-origin policy. Portanto, ele é usado para garantir mais segurança e limitar como um documento ou script de uma determinada origem pode interagir com recursos de outra origem diferente.

Utilizamos ele no Back-End para enviarmos dados para nosso cliente.

**6. Express**

+

|  |  |  |
| --- | --- | --- |
| **GESTORES** | | |
| **Campo** | **Descrição** | **Formato** |
| IDGESTOR | Cada gestor terá o seu número de identificação. | INTEGER |
| Nome | Nome do gestor. | VARCHAR |
| CPF | CPF do gestor. | CHAR |
| E-mail | E-mail para ingresso na plataforma. | VARCHAR |
| Senha | Senha pessoal de ingresso na plataforma. | VARCHAR |
| Sexo | Gênero do gestor. | ENUM |
| Cargo | Ocupação do usuário na plataforma. | VARCHAR |
| ID\_ADMIN | Identificação do gestor administrador responsável pelo cadastro. | INTEGER |

|  |  |  |
| --- | --- | --- |
| **MEDICOS** | | |
| **Campo** | **Descrição** | **Formato** |
| IDMEDICO | Cada médico terá o seu  número e identificação. | INTEGER |
| Nome | Nome do médico. | VARCHAR |
| CPF | CPF do médico. | CHAR |
| CRM | O registro do profissional no Conselho Regional de Medicina (CRM). | CHAR |
| Sexo | Gênero do médico. | ENUM |
| E-mail | E-mail para ingresso na plataforma. | VARCHAR |
| Senha | Senha pessoal de ingresso na plataforma. | ENUM |
| Especialidade | Campo de conhecimento do médico. | VARCHAR |
| Cargo | Ocupação do usuário na plataforma. | VARCHAR |
| ID\_GESTOR\_MED | Identificação do gestor responsável pelo cadastro. | INTEGER |

|  |  |  |
| --- | --- | --- |
| **ENDERECO\_MEDICOS** | | |
| **Campo** | **Descrição** | **Formato** |
| IDEND\_MED | Todo endereço cadastrado terá seu número de identificação. | INTEGER |
| CEP | (Código de Endereçamento Postal), para identificar melhor o endereço do médico. | CHAR |
| Logradouro | Campo que contém o nome da rua da qual o médico mora. | VARCHAR |
| NUM\_END\_MED | Identificação de moradia do médico. | VARCHAR |
| Bairro | Comunidade ou região dentro da cidade onde o médico mora. | VARCHAR |
| Complemento | Informação que indica com precisão a localização do endereço. | VARCHAR |
| PT\_REF | Local que serve como referência indicando  orientação. | VARCHAR |
| Cidade | Nome da cidade em que o médico habita. | VARCHAR |
| UF | (Unidade federativa). Refere-se ao estado em que o médico mora. | CHAR |
| País | Nome do país em que o médico habita. | VARCHAR |
| TIPO\_END\_MED | Classificação do lugar onde o médico mora. | ENUM |
| ID\_END\_MED | Identificação do médico que cadastrou o endereço. | INTEGER |

|  |  |  |
| --- | --- | --- |
| **TEL\_MEDICOS** | | |
| **Campo** | **Descrição** | **Formato** |
| IDTEL\_MED | Todo telefone cadastrado terá seu número de identificação. | INTEGER |
| DDD | Dígitos que identificam a cidade do país da qual o médico mora. | CHAR |
| NUM\_TEL\_MED | Número de telefone do médico. | VARCHAR |
| TIPO\_TEL\_MED | Classificação do telefone cadastrado. | ENUM |
| ID\_TEL\_MED | Identificação do médico que cadastrou o telefone. | INTEGER |

|  |  |  |
| --- | --- | --- |
|  | **MED\_CONVS** |  |
| **Campo** | **Descrição** | **Formato** |
| ID\_MED\_CONV | Cada médico convidado terá o seu número e identificação. | INTEGER |
| Nome | Nome do médico. | VARCHAR |
| CPF | CPF (Cadastro de Pessoa Física) do médico. | CHAR |
| Especialidade | Campo de conhecimento médico. | VARCHAR |
| E-mail | E-mail para entrar na plataforma. | VARCHAR |
| Senha | Senha para entrar na plataforma. | VARCHAR |
| CRM | O registro do profissional no Conselho Regional de Medicina (CRM). | CHAR |
| Sexo | Gênero do(a) médico(a) convidado(a). | ENUM |
| Cargo | Ocupação do usuário na plataforma. | VARCHAR |
| ID\_GEST\_CONV | Identificação do gestor responsável pelo cadastro. | INTEGER |
| ID\_EXAME\_CONV | Todos os exames que necessitem de um especialista terão um número de identificação. | VARCHAR |

|  |  |  |
| --- | --- | --- |
|  | **ENDERECO\_****MEDICO\_CONVS** |  |
| **Campo** | **Descrição** | **Formato** |
| ID\_END\_MED\_CONV | Todo endereço cadastrado terá seu número de identificação. | INTEGER |
| CEP | (Código de endereçamento postal), para identificar o endereço do médico. | CHAR |
| Logradouro | Nome da rua do médico convidado. | VARCHAR |
| NUMERO\_END\_MED\_CON | Identificação da moradia do médico convidado. | VARCHAR |
| Bairro | Comunidade ou região dentro da cidade onde o médico mora. | VARCHAR |
| Complemento | Informação que indica com precisão a localização do endereço. | VARCHAR |
| PT\_REF | Local que serve como referência indicando  orientação. | VARCHAR |
| Cidade | Nome da cidade que o médico convidado mora. | VARCHAR |
| UF | (Unidade Federativa), refere-se ao estado em que o médico mora. | CHAR |
| País | Pais em que o médico convidado mora. | VARCHAR |
| TIPO\_END\_MED\_CONV | Classificação do lugar onde o médico mora. | ENUM |
| ID\_END\_CONV | Identificação do médico que cadastrou o endereço. | INTEGER |

|  |  |  |
| --- | --- | --- |
|  | **TEL\_****MEDICO\_CONVS** |  |
| **Campo** | **Descrição** | **Formato** |
| IDTEL\_MED\_CONVS | Todo telefone cadastrado terá seu número de identificação. | INTEGER |
| DDD | Dígitos que identificam a cidade do país/estado no qual o médico mora. | CHAR |
| NUM\_TEL\_CONVS | Número de telefone do médico. | VARCHAR |
| TIPO\_TEL\_CONVS | Classificação do telefone cadastrado. | ENUM |
| ID\_TEL\_CONVS | Identificação do médico que cadastrou o telefone. | INTEGER |

|  |  |  |
| --- | --- | --- |
|  | **ATLETAS** |  |
| **Campo** | **Descrição** | **Formato** |
| IDATLETA | Todo atleta cadastrado terá seu número de identificação. | INTEGER |
| Nome | Nome do atleta cadastrado no sistema | VARCHAR |
| CPF | Cadastro de Pessoa Física do atleta. Necessário para ingresso na plataforma. | CHAR |
| E-mail | E-mail para identificação na plataforma. | VARCHAR |
| DT\_NASC | Data de nascimento do atleta na plataforma. | DATE |
| Senha | Senha utilizada para entrar na plataforma. | VARCHAR |
| Sexo | Sexo do atleta. | ENUM |
| Categoria | Categoria na qual o atleta compete. | VARCHAR |
| Posição | Posição na qual o atleta faz parte em seu esporte. | VARCHAR |
| Cargo | Ocupação do usuário na plataforma. | VARCHAR |
| Modalidade | Modalidade específica a qual o atleta compete em seu esporte. | VARCHAR |
| Solicitação | Status no qual o exame se encontra, se foi solicitado ou não. | ENUM |
| Situação | Situação do atleta, se foi aprovado, está em análise ou se foi reprovado. | ENUM |
| ID\_GESTOR\_ATL | Todo atleta cadastrado na plataforma possui um gestor. | INTEGER |

|  |  |  |
| --- | --- | --- |
|  | **ENDERECO\_ATLETAS** |  |
| **Campo** | **Descrição** | **Formato** |
| IDEND\_ATL | Todo endereço cadastrado terá seu número de identificação. | INTEGER |
| CEP | (Código de Endereçamento Postal), para identificar melhor o endereço do atleta. | CHAR |
| Logradouro | Nome da rua do atleta. | VARCHAR |
| NUM\_END\_ATL | Identificação da moradia do atleta. | VARCHAR |
| Bairro | Comunidade ou região dentro da cidade onde o atleta mora. | VARCHAR |
| Complemento | Informação que indica com precisão a localização do endereço. | VARCHAR |
| PT\_REF | Local que serve como referência indicando orientação. | VARCHAR |
| Cidade | Nome da cidade em que o atleta mora. | VARCHAR |
| UF | (Unidade Federativa), refere-se ao estado que o atleta mora. | CHAR |
| País | País que o atleta convidado mora, | VARCHAR |
| TIPO\_END\_ATL | Classificação do lugar onde o atleta mora. | ENUM |
| ID\_END\_ATL | Identificação do atleta em que cadastrou o endereço. | INTEGER |

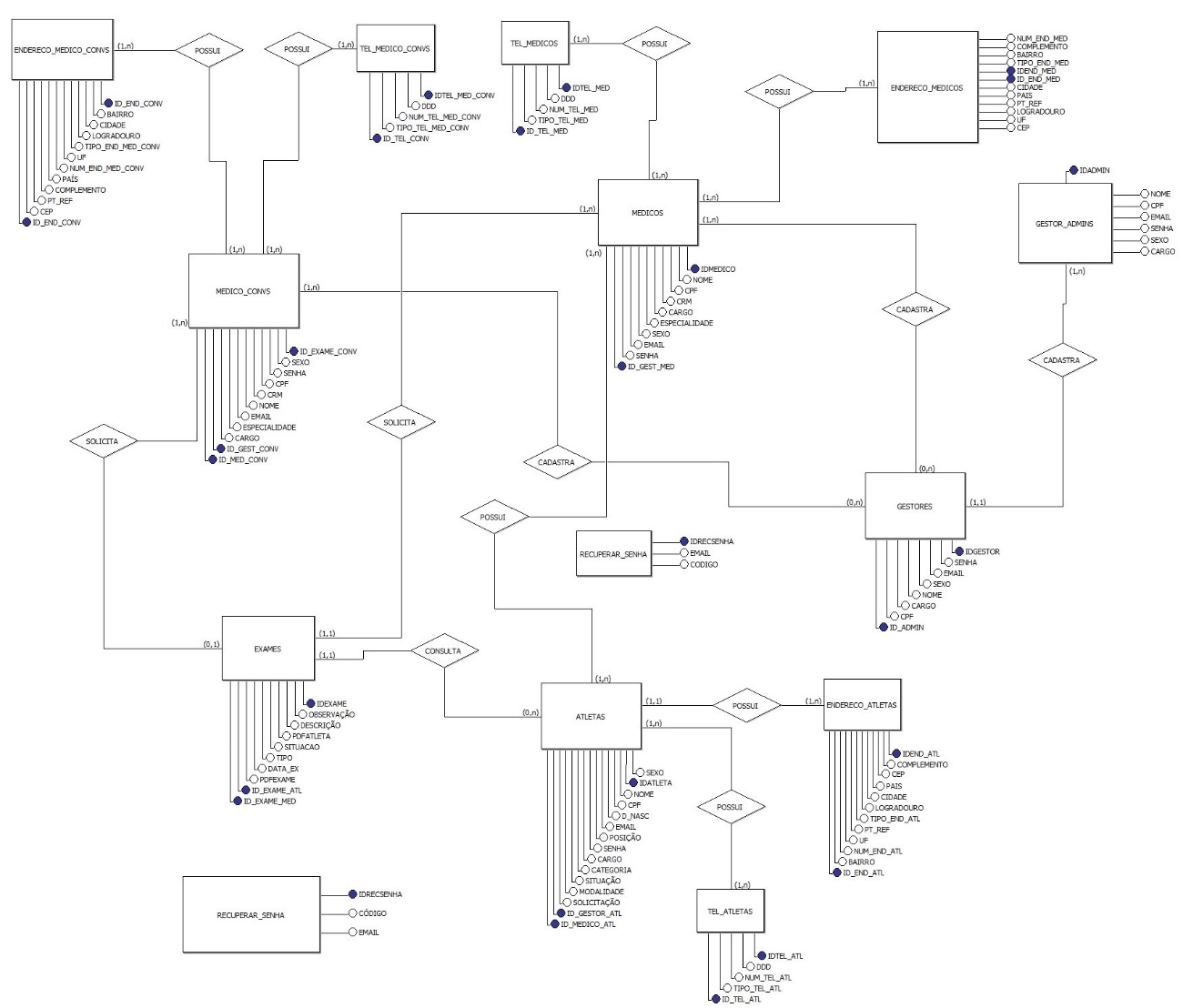
|  |  |  |
| --- | --- | --- |
|  | **TEL\_ATLETAS** |  |
| **Campo** | **Descrição** | **Formato** |
| IDTEL\_ATL | Todo telefone cadastrado terá seu número de identificação. | INTEGER |
| DDD | Dígitos que identificam a cidade do país/estado da qual o atleta mora. | CHAR |
| NUM\_TEL\_ATL | Número de telefone do atleta. | VARCHAR |
| TIPO\_TEL\_ATL | Classificação do telefone cadastrado. | ENUM |
| ID\_TEL\_ATL | Identificação do atleta em que cadastrou o telefone. | INTEGER |

|  |  |  |
| --- | --- | --- |
|  | **EXAMES** |  |
| **Campo** | **Descrição** | **Formato** |
| IDEXAME | Todo exame cadastrado terá seu número de identificação | INTEGER |
| PDFEXAME | PDF gerado com informações e descrição do exame enviados pelo médico. | LONGBLOB |
| PDFATLETA | PDF gerado com informações e descrição do exame para o atleta. | LONGBLOB |
| Tipo | Tipo do exame. | VARCHAR |
| Descrição | Descrição do exame. | TEXT |
| DATA\_EX | Data do exame. | DATETIME |
| Situação | Situação do exame. | ENUM |
| Observação | Informações adicionais no exame pelo médico. | TEXT |

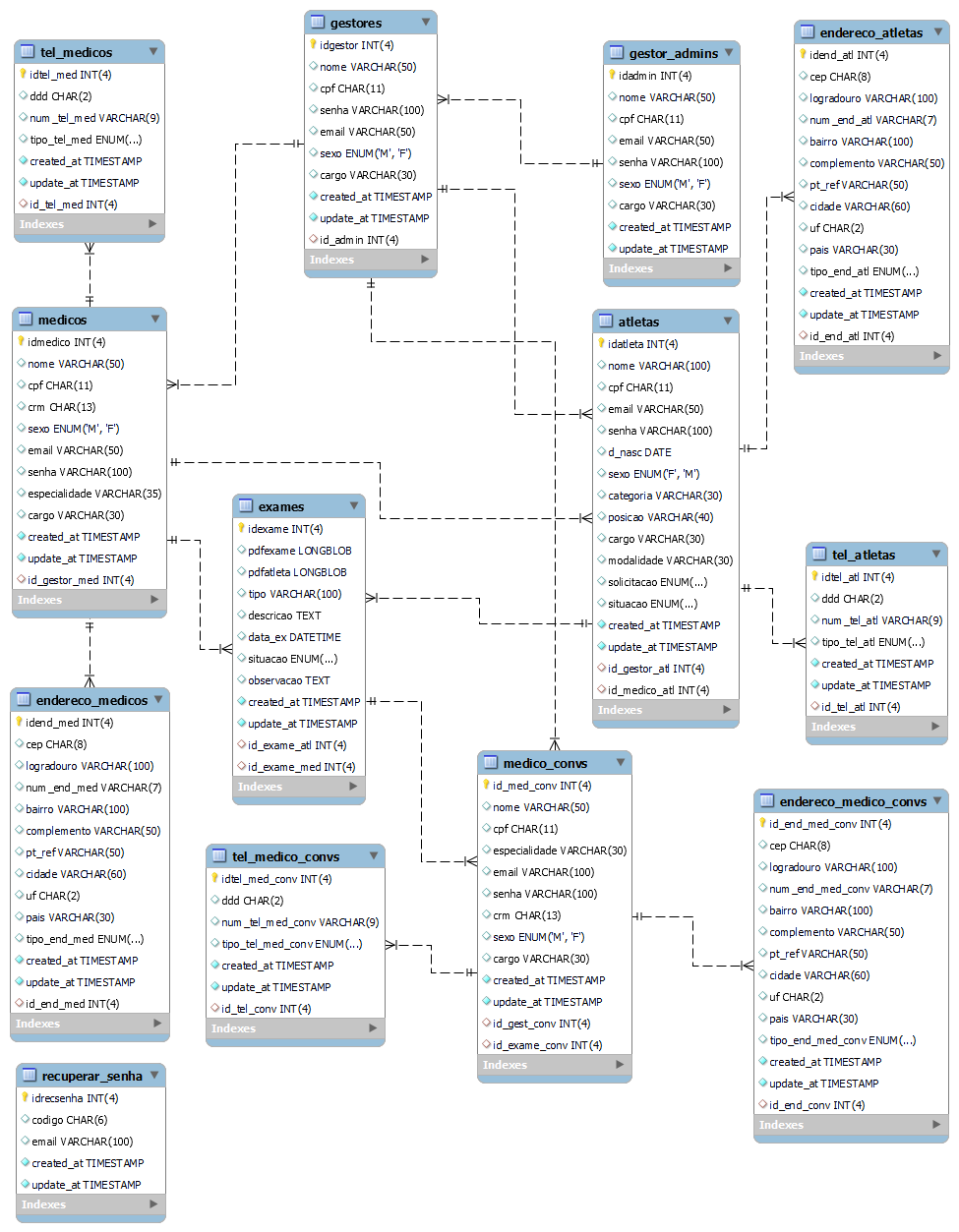
|  |  |  |
| --- | --- | --- |
|  | **RECUPERAR\_SENHA** |  |
| **Campo** | **Descrição** | **Formato** |
| IDRECSENHA | Todo código de recuperação terá seu número de identificação. | INTEGER |
| Código | Código que será enviado para o usuário para autenticação. | CHAR |
| Email | E-mail para recebimento da do código. | VARCHAR |

**7. Modelagens**

**7.1 Modelo Conceitual**



**7.2 Modelagem Lógica**



**7.3 Modelagem Física**

|  |  |
| --- | --- |
|  | |
|  | | CREATE DATABASE SESI; | |
|  | | use SESI; | |
|  | | /\*------------------------------------ TABELAS ------------------------------------\*/ | |
|  | |  | |
|  | | /\* -- TABELAS GESTOR -- \*/ | |
|  | |  | |
|  | | CREATE TABLE gestor\_admins ( | |
|  | | idadmin INT(4) PRIMARY KEY auto\_increment, | |
|  | | nome VARCHAR(50), | |
|  | | cpf CHAR(11), | |
|  | | email VARCHAR(50), | |
|  | | senha VARCHAR(100), | |
|  | | sexo ENUM("M","F"), | |
|  | | cargo VARCHAR(30), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp() | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE gestores ( | |
|  | | idgestor INT (4) PRIMARY KEY auto\_increment, | |
|  | | nome VARCHAR(50), | |
|  | | cpf CHAR(11), | |
|  | | senha VARCHAR(100), | |
|  | | email VARCHAR(50), | |
|  | | sexo ENUM("M","F"), | |
|  | | cargo VARCHAR(30), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_admin INT(4), | |
|  | | FOREIGN KEY (id\_admin) | |
|  | | references gestor\_admins (idadmin) | |
|  | | ); | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | | /\* -- TABELAS MEDICO -- \*/ | |
|  | | CREATE TABLE medicos ( | |
|  | | idmedico INT(4) PRIMARY KEY auto\_increment, | |
|  | | nome VARCHAR(50), | |
|  | | cpf CHAR(11), | |
|  | | crm CHAR(13), | |
|  | | sexo ENUM("M","F"), | |
|  | | email VARCHAR(50), | |
|  | | senha VARCHAR(100), | |
|  | | especialidade VARCHAR(35), | |
|  | | cargo VARCHAR (30), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_gestor\_med INT(4), | |
|  | | FOREIGN KEY (id\_gestor\_med) | |
|  | | references gestores (idgestor) | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE endereco\_medicos ( | |
|  | | idend\_med INT(4) PRIMARY KEY auto\_increment, | |
|  | | cep CHAR(8), | |
|  | | logradouro VARCHAR(100), | |
|  | | num\_end\_med varchar(7), | |
|  | | bairro VARCHAR(100), | |
|  | | complemento VARCHAR(50), | |
|  | | pt\_ref VARCHAR(50), | |
|  | | cidade VARCHAR(60), | |
|  | | uf CHAR(2), | |
|  | | pais VARCHAR(30), | |
|  | | tipo\_end\_med ENUM ('RESIDENCIAL', 'COMERCIAL'), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_end\_med INT(4), | |
|  | | FOREIGN KEY (id\_end\_med) | |
|  | | references medicos (idmedico) | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE tel\_medicos ( | |
|  | | idtel\_med INT(4) PRIMARY KEY auto\_increment, | |
|  | | ddd CHAR(2), | |
|  | | num\_tel\_med VARCHAR(9), | |
|  | | tipo\_tel\_med ENUM('RESIDENCIAL', 'COMERCIAL','CELULAR'), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_tel\_med INT(4), | |
|  | | FOREIGN KEY (id\_tel\_med) | |
|  | | references medicos (idmedico) | |
|  | | ); | |
|  | | /\* -- TABELAS MEDICO -- \*/ | |
|  | |  | |
|  | |  | |
|  | | /\* -- TABELAS ATLETA -- \*/ | |
|  | | CREATE TABLE atletas ( | |
|  | | idatleta INT(4) PRIMARY KEY auto\_increment, | |
|  | | nome VARCHAR(100), | |
|  | | cpf CHAR(11), | |
|  | | email varchar(50), | |
|  | | senha VARCHAR(100), | |
|  | | d\_nasc DATE, | |
|  | | sexo ENUM('F','M'), | |
|  | | categoria VARCHAR(30), | |
|  | | posicao VARCHAR(40), | |
|  | | cargo VARCHAR(30), | |
|  | | modalidade VARCHAR(30), | |
|  | | solicitacao ENUM("SOLICITADO","NÃO SOLICITADO"), | |
|  | | situacao enum("APROVADO","EM ANÁLISE","PENDENTE","REPROVADO"), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA 1,N \*/ | |
|  | | id\_gestor\_atl INT(4), | |
|  | | FOREIGN KEY (id\_gestor\_atl) | |
|  | | references gestores (idgestor), | |
|  | |  | |
|  | | id\_medico\_atl INT(4), | |
|  | | FOREIGN KEY (id\_medico\_atl) | |
|  | | references medicos (idmedico) | |
|  | |  | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE endereco\_atletas ( | |
|  | | idend\_atl INT(4) PRIMARY KEY auto\_increment, | |
|  | | cep CHAR(8), | |
|  | | logradouro VARCHAR(100), | |
|  | | num\_end\_atl varchar(7), | |
|  | | bairro VARCHAR(100), | |
|  | | complemento VARCHAR(50), | |
|  | | pt\_ref VARCHAR(50), | |
|  | | cidade VARCHAR(60), | |
|  | | uf CHAR(2), | |
|  | | pais VARCHAR(30), | |
|  | | tipo\_end\_atl ENUM ('RESIDENCIAL', 'COMERCIAL'), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | id\_end\_atl INT(4), | |
|  | | FOREIGN KEY (id\_end\_atl) | |
|  | | references atletas (idatleta) | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE tel\_atletas ( | |
|  | | idtel\_atl INT(4) PRIMARY KEY auto\_increment, | |
|  | | ddd CHAR(2), | |
|  | | num\_tel\_atl VARCHAR(9), | |
|  | | tipo\_tel\_atl ENUM('RESIDENCIAL', 'COMERCIAL','CELULAR'), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | id\_tel\_atl INT(4), | |
|  | | FOREIGN KEY (id\_tel\_atl) | |
|  | | references atletas (idatleta) | |
|  | | ); | |
|  | | /\* -- TABELAS ATLETA -- \*/ | |
|  | |  | |
|  | |  | |
|  | | /\* -- TABELAS EXAME -- \*/ | |
|  | | CREATE TABLE exames( | |
|  | | idexame INT(4) PRIMARY KEY auto\_increment, | |
|  | | pdfexame longblob, | |
|  | | pdfatleta longblob, | |
|  | | tipo VARCHAR(100), | |
|  | | descricao TEXT, | |
|  | | data\_ex DATETIME, | |
|  | | situacao ENUM("CONCLUIDO","EM ANALISE","PENDENTE"), | |
|  | | observacao TEXT, | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_exame\_atl INT(4), | |
|  | | FOREIGN KEY (id\_exame\_atl) | |
|  | | references atletas (idatleta), | |
|  | |  | |
|  | | id\_exame\_med INT(4), | |
|  | | FOREIGN KEY (id\_exame\_med) | |
|  | | references medicos (idmedico) | |
|  | | ); | |
|  | | /\* -- TABELAS EXAME -- \*/ | |
|  | |  | |
|  | |  | |
|  | | /\* TABELAS MEDICO CONVIDADO\*/ | |
|  | | CREATE TABLE medico\_convs ( | |
|  | | id\_med\_conv INT(4) PRIMARY KEY auto\_increment, | |
|  | | nome VARCHAR(50), | |
|  | | cpf CHAR(11), | |
|  | | especialidade VARCHAR(30), | |
|  | | email VARCHAR(100), | |
|  | | senha VARCHAR(100), | |
|  | | crm CHAR(13), | |
|  | | sexo ENUM("M","F"), | |
|  | | cargo VARCHAR(30), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_gest\_conv INT(4), | |
|  | | FOREIGN KEY (id\_gest\_conv) | |
|  | | references gestores (idgestor), | |
|  | |  | |
|  | | id\_exame\_conv INT(4), | |
|  | | FOREIGN KEY (id\_exame\_conv) | |
|  | | references exames (idexame) | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE endereco\_medico\_convs ( | |
|  | | id\_end\_med\_conv INT(4) PRIMARY KEY auto\_increment, | |
|  | | cep CHAR(8), | |
|  | | logradouro VARCHAR(100), | |
|  | | num\_end\_med\_conv varchar(7), | |
|  | | bairro VARCHAR(100), | |
|  | | complemento VARCHAR(50), | |
|  | | pt\_ref VARCHAR(50), | |
|  | | cidade VARCHAR(60), | |
|  | | uf CHAR(2), | |
|  | | pais VARCHAR(30), | |
|  | | tipo\_end\_med\_conv ENUM ('RESIDENCIAL', 'COMERCIAL'), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_end\_conv INT(4), | |
|  | | FOREIGN KEY (id\_end\_conv) | |
|  | | references medico\_convs (id\_med\_conv) | |
|  | | ); | |
|  | |  | |
|  | | CREATE TABLE tel\_medico\_convs ( | |
|  | | idtel\_med\_conv INT(4) PRIMARY KEY auto\_increment, | |
|  | | ddd CHAR(2), | |
|  | | num\_tel\_med\_conv VARCHAR(9), | |
|  | | tipo\_tel\_med\_conv ENUM('RESIDENCIAL', 'COMERCIAL','CELULAR'), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp(), | |
|  | |  | |
|  | | /\* CHAVE ESTRANGEIRA\*/ | |
|  | | id\_tel\_conv INT(4), | |
|  | | FOREIGN KEY (id\_tel\_conv) | |
|  | | references medico\_convs (id\_med\_conv) | |
|  | | ); | |
|  | |  | |
|  | | /\* TABELAS RECUPERAR A SENHA \*/ | |
|  | |  | |
|  | | CREATE TABLE recuperar\_senha ( | |
|  | | idrecsenha INT(4) PRIMARY KEY auto\_increment, | |
|  | | codigo CHAR(6), | |
|  | | email VARCHAR(100), | |
|  | | created\_at timestamp default current\_timestamp() ON UPDATE current\_timestamp(), | |
|  | | update\_at timestamp default current\_timestamp() | |
|  | | ); | |
|  | |  | |
|  | |  | |
|  | | /\*------------------------------------ TABELAS ------------------------------------\*/ | |
|  | |  | |
|  | | /\*------------------------------------ INSERTS ------------------------------------\*/ | |
|  | | SET foreign\_key\_checks = 0; | |
|  | |  | |
|  | | /\*------------------------- INSERT GESTOR ADM -------------------------\*/ | |
|  | |  | |
|  | | insert into gestor\_admins values (default,"rayane","54639920881","ray@gmail.com","ray1234",'F',"gestor admin", NOW(), NOW()); | |
|  | |  | |
|  | | /\*------------------------- INSERT GESTOR -------------------------\*/ | |
|  | |  | |
|  | | insert into gestores values (default,"Vinicius","53899079817","zile1234","vinicius@gmail.com",'M',"gestor", NOW(), NOW(),"1"); | |
|  | | insert into gestores values (default,"Wesley","25694874123","ws1234","ws@gmail.com",'M',"gestor",NOW(), NOW(),"1"); | |
|  | | insert into gestores values (default,"Stefanny","25694874123","stefanny1234","stefanny@gmail.com",'F',"gestor",NOW(), NOW(),"1"); | |
|  | | insert into gestores values (default,"Paula","45899256374","paula1234","paula@gmail.com",'F',"gestor",NOW(), NOW(),"1"); | |
|  | |  | |
|  | | /\*------------------------- INSERT ATLETA -------------------------\*/ | |
|  | |  | |
|  | | INSERT INTO ATLETAS VALUES (DEFAULT, 'ISABEL FERREIRA', '25698745632',"ISA@GMAIL.COM", 'ISABEL123', '2000/05/20', 'F','VOLÊI','LÍBERO','ATLETA', 'VOLÊI PRAIA',"SOLICITADO",'PENDENTE', NOW(), NOW(), '1', '1'); | |
|  | | INSERT INTO ENDERECO\_ATLETAS VALUES (DEFAULT, '06814090', 'SÃO BERNARDO', 336, 'SÃO MARCOS', 'SERRALHERIA MM', 'EMBU DAS ARTES','São Paulo', 'SP', 'BRASIL','RESIDENCIAL', NOW(), NOW(), '1'); | |
|  | | INSERT INTO TEL\_ATLETAS VALUES (DEFAULT, '11', 934664070, 'CELULAR',NOW(), NOW(), '1'); | |
|  | |  | |
|  | | INSERT INTO ATLETAS VALUES (DEFAULT, 'PEDRO', '45687954622',"PEDRO@GMAIL.COM", 'PEDRO123', '2000/04/7', 'M','FUTEBOL','LATERAL','ATLETA','FUTEBOL DE CAMPO',"SOLICITADO",'APROVADO',NOW(), NOW(),'2', '2'); | |
|  | | INSERT INTO ENDERECO\_ATLETAS VALUES (DEFAULT, '04467598', 'RUA 25 DE JANEIRO', 11, 'PARQUE PRIMAVERA', 'EM FRENTE A ARVORE', 'COLEGIO ISABEL VIEIRA FERREIRA', 'SAO PAULO', 'SP', 'BRASIL', 'COMERCIAL',NOW(), NOW(), '2'); | |
|  | | INSERT INTO TEL\_ATLETAS VALUES (DEFAULT, '11', 978456679, 'CELULAR',NOW(), NOW(), '2'); | |
|  | |  | |
|  | | /\*------------------------- INSERT MEDICO -------------------------\*/ | |
|  | | INSERT INTO MEDICOS VALUES (DEFAULT, 'FABIO', '78955623145',"123456987",'M','FABIO@GMAIL.COM','FABIO123','CARDIOLOGISTA','Médico',NOW(), NOW(),1); | |
|  | | INSERT INTO ENDERECO\_MEDICOS VALUES (DEFAULT, '04467565', 'RUA 7 DE ABRIL', 30, 'JARDIM APURA', 'COLEGIO THIAGO ABERIONE','avenida carminha', 'SAO PAULO', 'SP', 'BRASIL','RESIDENCIAL' , NOW(), NOW(),'1'); | |
|  | | INSERT INTO TEL\_MEDICOS VALUES (DEFAULT, '11', 951234788, 'CELULAR', NOW(), NOW(),'1'); | |
|  | |  | |
|  | | INSERT INTO MEDICOS VALUES (DEFAULT, 'REGINA', '45615987654',"987654321",'F','REGINA@GMAIL.COM','REGINA123','ORTOPEDISTA','Médico', NOW(), NOW(),2); | |
|  | | INSERT INTO ENDERECO\_MEDICOS VALUES (DEFAULT, '23569874', 'PORIPORÁ', 305, 'SÃO FRANCISCO', 'avenida zile', 'PAPELARIA ENALDO', 'SAO PAULO', 'SP', 'BRASIL','COMERCIAL',NOW(), NOW(), '2'); | |
|  | | INSERT INTO TEL\_MEDICOS VALUES (DEFAULT, '11', 41492311, 'RESIDENCIAL',NOW(), NOW(), '2'); | |
|  | |  | |
|  | | /\*------------------------- INSERT EXAME -------------------------\*/ | |
|  | | insert into exames values (default,'','','Exame de sangue','lorem', '2022/09/16', 'Pendente', 'lorem', NOW(), NOW(),'1', '1'); | |
|  | | insert into exames values (default,'','','Exame cardiologico','lorem', '2022/05/13', 'Concluido','lorem', NOW(), NOW(),'2', '2'); | |
|  | |  | |
|  | | /\*------------------------- INSERT MEDICO CONVIDADO -------------------------\*/ | |
|  | |  | |
|  | | insert into medico\_convs value(default,'Wesley','10283268584','Cardiologia','wesley12@Hotmail.com','ws123456',423843258,'m','Médico convidado', NOW(), NOW(),'1','1'); | |
|  | | insert into endereco\_medico\_convs value(default,'04470389','Travessa São Sebastião',1005,'Parque Primavera','Casa A','lado do Banco','Sâo Paulo','SP','Basil','Residencial', NOW(), NOW(),'1'); | |
|  | | insert into TEL\_MEDICO\_CONVS value(default,11,956923807,'Celular', NOW(), NOW(), '1'); | |
|  | |  | |
|  | | insert into medico\_convs value(default,'Gabriel',81382590563,'Traumatologia','Gabriel1@gmail.com','gaab1234',455326132,'m','Médico convidado',NOW(), NOW(),'2','2'); | |
|  | | insert into endereco\_medico\_convs value(default,'04467270','Rua Adauto Lemos',36,'Vila missionaria','Casa B','lado do MC','Rio de Janeiro','Rj','Basil','Comercial', NOW(), NOW(),'2'); | |
|  | | insert into TEL\_MEDICO\_CONVS value(default,11,956389091,'Comercial',NOW(), NOW(), '2'); | |
|  | |  | |
|  | |  | |
|  | | /\*------------------------------------ INSERTS ------------------------------------\*/ | |

**Infraestrutura**

1. **Definição**

A Squad DevOps tem como função arquitetar um ambiente para cada funcionalidade de uma página (Front-End, Back-End e Banco de Dados), sua finalidade é realizar a virtualização e agrupamento de um software. O trio responsável por esta fase foi composto de três alunos: Luccas Thiago, Paula Santana Barboza e Pedro Henrique Lira da Silva Bispo.

1. **Recursos**
2. Ubuntu Server 16.04: sistema operacional destinado para servidores.
3. VirtualBox: programa de virtualização, que permite executar diferentes sistemas operacionais em um computador.
4. Docker: agrupa recursos no sistema hospedeiro com as configurações individuais de cada componente de um software.
5. SQL Server: sistema gerenciador de Banco de Dados, que armazena, atualiza e recupera dados computacionais.
6. NodeJS: permite que aplicações Javascript rodem sem a dependência de um navegador.
7. NPM: gerenciador de pacotes do Node, que auxilia na instalação e remoção de pacotes.
8. Nginx: servidor web, que possui a capacidade de receber uma ampla quantidade de clientes com menos recursos.
9. **Ambiente Operacional**

O sistema operacional utilizado foi o Ubuntu 16.04 LTS (Xenial Xerus), uma distribuição Linux Open Source. Suas especificações são:

1. Arquitetura de processador suportadas:
2. amd64 (Intel/AMD 64-bit);
3. arm64 (64-bit ARM);
4. ppc64el (POWER8 and POWER9);
5. s398 (IBM Z and LinuxONE);
6. **Processo de Desenvolvimento Técnico**

Nesta área, os alunos definiram o manual de uso do ambiente para auxiliar a instalação, o passo a passo da utilização de arquivos e programas.

**4.1 Docker:**

* + O Docker realiza a virtualização e faz o agrupamento do sistema, com economia de recursos, como energia, memória e processamento.
  + Docker se assemelha a uma máquina virtual, com a possibilidade de ser executado em qualquer sistema operacional (Windows e Linux).

**4.1.1 Dockerfile:**

* + Dockerfile é um arquivo que permite criar a Imagem Docker.

**4.1.2 Imagens Docker:**

* As imagens Docker são as principais responsáveis por permitir que determinada funcionalidade da página (Back-End, Front-End e Banco de Dados), sejam executadas de forma correta.

**Imagem Front-End:**

FROM node:16-alpine as build //importação do node pelo docker hub

WORKDIR /app //criação da pasta

COPY . /app //informação será encaminhada para pasta criada

ENV PATH /app/node\_modules/.bin:$PATH //encaminha valores no arquivo docker-compose

RUN npm install //instala as dependências

RUN npm run build //cria uma instância

FROM nginx:1.21.6-alpine //importação do nginx pelo docker hub

COPY --from=build /app/build /usr/share/nginx/html

RUN rm /etc/nginx/conf.d/default.conf

COPY nginx/nginx.conf /etc/nginx/conf.d

EXPOSE 8080 //porta em que o código irá rodar

CMD ["nginx", "-g", "daemon off;"] //comando do prompt de comando

#https://medium.com/swlh/dockerizing-a-react-application-with-docker-and-nginx-19e88ef8e99a

**Imagem Back-End:**

FROM node:16 //importação do node pelo docker hub

WORKDIR /usr/src/sesi\_be //criação das pastas

COPY package\*.json ./ //as informações serão encaminhadas ao arquivo package\*.json

RUN npm install //comando para executar a instalação dos comandos acima

COPY . .

**Imagem Banco de Dados:**

FROM mysql //importação do mysql via Docker Hub

ENV MYSQL\_ROOT\_PASSWORD passwd //senha definida para realizar a conexão com o banco

ENV MYSQL\_USER api //usuário

ENV MYSQL\_PASSWORD passwd

COPY ./BANCO\_DE\_DADOS.sql /docker-entrypoint-initdb.d/ //copia o script do banco e o encaminha para a ->

// -> pasta, onde irá executar o mesmo no mysql Server

WORKDIR /var/lib/mysqld //criação das pastas

#https://stackoverflow.com/questions/62999267/how-to-run-mysql-scripts-in-docker-image#62999651

**4.1.3 Docker Hub:**

* Docker Hub é um serviço para hospedar e instalar diversas imagens Docker.
* Este serviço disponibiliza, por meio do Prompt de comando (ou terminal Linux), a instalação da imagem, conforme exemplo abaixo:

C:\Users\I>docker pull node

* Node é uma das diversas tecnologias disponíveis, como imagem no Docker Hub.

**4.1.4 Docker-Compose:**

* Docker Compose é uma ferramenta que agiliza o processo do deploy de um software. Agrupa todos os containers da aplicação e os executa no mesmo instante.

**Docker-Compose Banco de Dados:**

version: "3.7"

services:

# Banco de dados (database)

database:

build:

context: ./SESI\_Database

dockerfile: db.dockerfile

image: sesi\_db:1.0.0

container\_name: sesi\_db

command: --default-authentication-plugin=mysql\_native\_password

restart: always

volumes:

- ./SESI\_Database:/var/lib/mysqld/

ports:

- "30306:3306"

expose:

- "30306"

networks:

- db\_be

**Docker-Compose Back-End:**

backend:

build:

context: ./SESI\_Backend

dockerfile: be.dockerfile

image: sesi\_be:1.0.0

container\_name: sesi\_be

restart: always

command: node app.js

volumes:

- ./SESI\_Backend:/usr/sesi\_be

ports:

- "24:3000"

expose:

- "24"

depends\_on:

- database

networks:

- db\_be

**Docker-Compose Front-End:**

frontend:

build:

context: ./SESI\_Frontend

dockerfile: fe.dockerfile

image: sesi\_fe:1.0.0

container\_name: sesi\_fe

restart: always

volumes:

- ./SESI\_Frontend:/var/www/html/

ports:

- "8080:8080"

environment:

- NODE\_ENV=development

depends\_on:

- database

- backend

networks:

db\_be:

driver: bridge

**5. Comando do software**

* O comando, ilustrado abaixo, tem como objetivo construir imagens, iniciar os containers, deixando-os em segundo plano e executando-os.

docker-compose up --build

**Referências Bibliográficas**

DATE, C. J. **Projeto de Banco de Dados e Teoria.** Novatec, 1° ED, 2015, São Paulo.