



**PROJETO DE DISCIPLINA:
PROJETO DE BANCO DE DADOS**

LEONARDO VERAS MASCENA OLIVEIRA LOPES

**SERRA TALHADA, PE
07/2021**

LEONARDO VERAS MASCENA OLIVEIRA LOPES

Projeto desenvolvido para efetivação da 2ª etapa
apresentado para avaliação na Disciplina Projeto de
Banco de Dados Relacionais ministrada pelo Prof.
Hidelberg Oliveira, período letivo 2020.1.

PROJETO DE BANCO DE DADOS

01. DEFINIÇÃO DA PROBLEMÁTICA (DOMÍNIO DA APLICAÇÃO)

“O Tratamento Fora de Domicílio – TFD, instituído pela Portaria nº 55 da Secretaria de Assistência à Saúde (Ministério da Saúde), é um instrumento legal que visa garantir, através do SUS, tratamento médico a pacientes portadores de doenças não tratáveis no município de origem por falta de condições técnicas.

Assim, o TFD consiste em uma ajuda de custo ao paciente, e em alguns casos, também ao acompanhante, encaminhados por ordem médica à unidades de saúde de outro município ou Estado da Federação, quando esgotados todos os meios de tratamento na localidade de residência do mesmo, desde que haja possibilidade de cura total ou parcial, limitado no período estritamente necessário a este tratamento e aos recursos orçamentários existentes.”

Com isso, o Health Transportation, é um sistema que será utilizado para organização, gerenciamento das viagens e seus respectivos processos, referentes a este programa.

02. DEFINIÇÃO DE REQUISITOS

2.1. Requisitos Funcionais:

Identificação:	[RF01] Manter HealthCenter (Centro/Secretaria de Saúde)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta dos centros de saúde (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF02] Manter Prefecture (Prefeitura)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta das prefeituras (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF03] Manter User (Usuário)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta dos usuários (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF04] Manter Passenger (Passageiro)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta dos passageiros (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF05] Manter Travel (Viagem)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta das viagens (CRUD).</i>

Prioridade:	(X) Essencial () Importante () Desejável
--------------------	--

Identificação:	[RF06] Manter Driver (Motorista)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta dos motoristas (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF07] Manter Vehicle (Veículo)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta dos veículos (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF08] Manter Location (Locação)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta das locações (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF09] Manter OrderResetPassword (Pedi. de Redefinição de Senha)
Descrição:	<i>O sistema deverá permitir a criação, atualização, desativação/remoção e consulta dos pedidos de redefinição de senha (CRUD).</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF10] Gerar Log
Descrição:	<i>O sistema deverá, para cada ação do sistema (Criar, atualizar, recuperar e deletar), gerar um log da modificação.</i>
Prioridade:	() Essencial () Importante (X) Desejável

Identificação:	[RF11] Gerar Relatório
Descrição:	<i>O sistema deverá gerar um relatório para todas as viagens que estiverem com status de “Em andamento” ou “Concluída”. Este relatório, deverá conter as informações da viagem e a lista de passageiros cadastrados registrados na mesma.</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RF12] Dashboard
Descrição:	<i>O sistema gerará gráficos estatísticos contendo:</i> <ul style="list-style-type: none"> - Quantidade de passageiros por bairro no ano corrente; - Quantidade de passageiros por meses no ano corrente;
Prioridade:	() Essencial () Importante (X) Desejável

2.2. Requisitos Não-Funcionais:

Identificação:	[RNF01] Frameworks utilizados
Tipo:	Implementação
RF Relacionado:	Nenhum
Descrição:	O Sistema deverá utilizar os frameworks Spring (Boot, JPA e MVC) e Thymeleaf (Template Engine).
Prioridade:	() Essencial (X) Importante () Desejável

Identificação:	[RNF02] SGBD utilizado
Tipo:	Implementação
Descrição:	O Sistema, deverá utilizar o sistema de gerenciamento de banco de dados (SGBD) PostgreSQL.
Prioridade:	() Essencial (X) Importante () Desejável

Identificação:	[RNF03] Linguagem de programação utilizada
Tipo:	Implementação
Descrição:	O Sistema, deverá utilizar a linguagem de programação Java.
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RNF04] Backup
Tipo:	Segurança
RF Relacionado:	Nenhum
Descrição:	O sistema deve ser configurado para fazer o backup do banco de dados uma vez por dia e em determinada hora especificada.
Prioridade:	() Essencial (X) Importante () Desejável

Identificação:	[RNF05] Padrão de Projeto
Tipo:	Implementação
RF Relacionado:	Nenhum
Descrição:	O sistema deve usar os padrões de projeto MVC e DAO.
Prioridade:	() Essencial (X) Importante () Desejável

Identificação:	[RNF06] Acesso em nível multi-usuário
Tipo:	Segurança
RF Relacionado:	[RF03] Manter User
Descrição:	O acesso ao sistema será feito em nível multiusuário, subdivididos em três grupos: - Admin, acesso total ao sistema;

	<ul style="list-style-type: none"> - Gestor, acesso parcial as funções do sistema; - Operador, acesso restrito às funções do sistema.
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RNF07] Senha
Tipo:	Segurança
RF Relacionado:	[RF03] Manter User
Descrição:	<i>O campo senha dos usuários deverá aceitar caracteres alfanuméricos, com tamanho entre 6 e 11, distinguindo caracteres maiúsculos de minúsculos. Além disso, a senha deve ser salva no banco de dados de forma criptografada.</i>
Prioridade:	() Essencial (X) Importante () Desejável

Identificação:	[RNF08] Resetar Senha
Tipo:	Segurança
RF Relacionado:	[RF03] Manter User
Descrição:	<i>Caso um usuário tenha esquecido sua senha de acesso ao sistema, um outro usuário com nível de acesso superior poderá resetar a senha esquecida para uma padrão. Está senha, será enviada para o email do usuário que fez solicitação.</i>
Prioridade:	() Essencial (X) Importante () Desejável

Identificação:	[RNF09] Acesso do Operador (User)
Tipo:	Segurança
RF Relacionado:	[RF03] Manter User
Descrição:	<i>Ao cadastrar um usuário com o papel de operador, o seu acesso ao sistema será restrito às permissões concedidas. São elas: manter travel; manter locação, manter passageiro, gerar relatórios.</i>
Prioridade:	(X) Essencial () Importante () Desejável

Identificação:	[RNF10] Acesso do Gestor (User)
Tipo:	Segurança
RF Relacionado:	[RF03] Manter User
Descrição:	<i>Ao cadastrar um usuário com o papel de gestor, o seu acesso ao sistema será restrito às permissões concedidas. São elas: manter User (operador) e todas as permissões relacionados ao papel de operador; manter motorista, manter veículo e acessar o dashboard.</i>
Prioridade:	(X) Essencial () Importante () Desejável

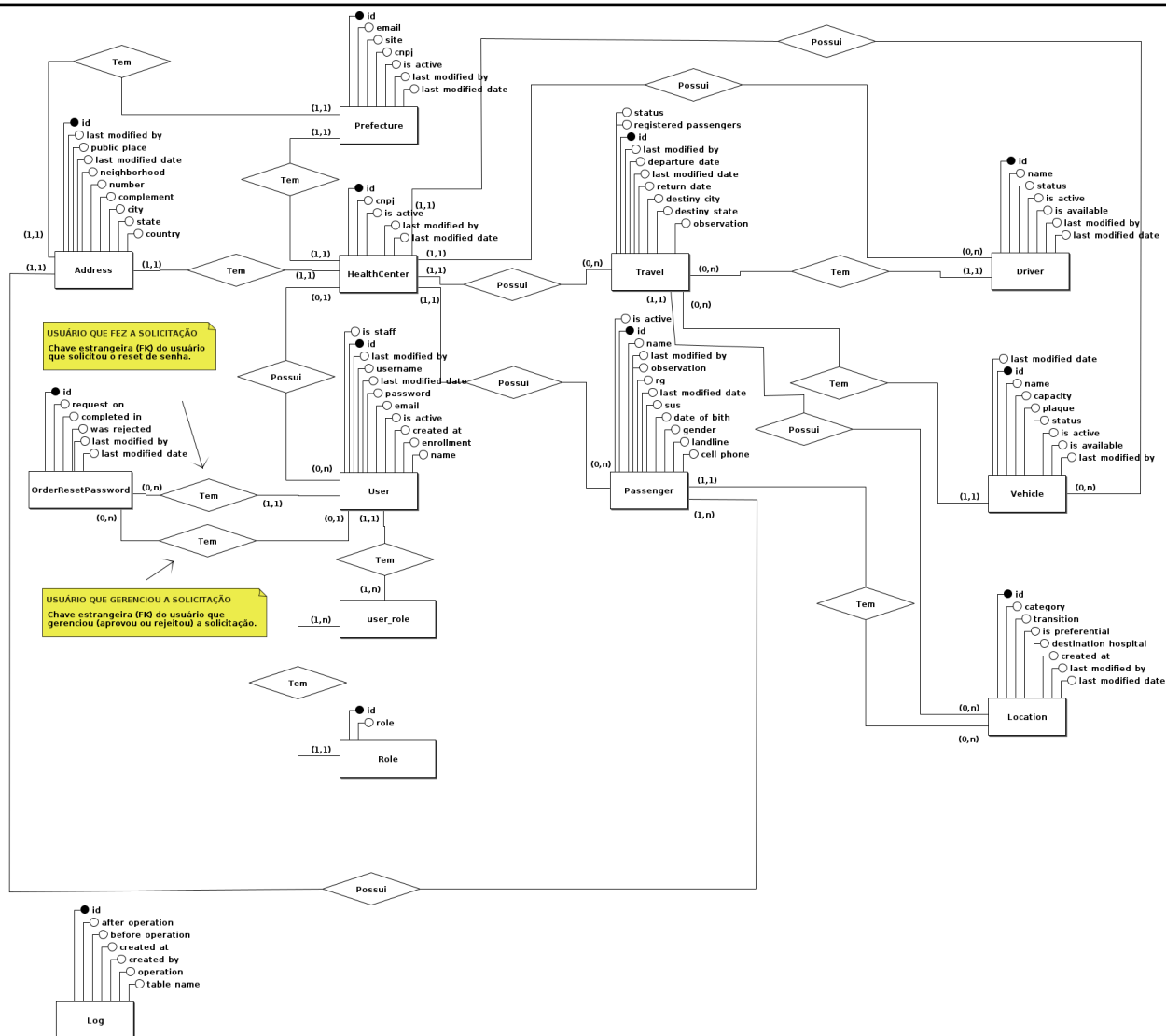
Identificação:	[RNF11] Acesso do Admin (User)
-----------------------	--------------------------------

Tipo:	Segurança
RF Relacionado:	[RF03] Manter User
Descrição:	Ao cadastrar um usuário com o papel de admin, o seu acesso ao sistema não terá restrição alguma (acesso total ao sistema). Elas incluem as permissões do operador e gestor, como também, manter o HealthCenter, Prefecture, Roles e o Users(gestor e operador).
Prioridade:	(X) Essencial () Importante () Desejável

2.3. Requisitos de Domínio (se existir):

Identificação:	[RD01] Regra para reset de senha
Descrição:	Ao aprovar uma solicitação de reset de senha, o sistema irá fornecer uma senha de forma aleatória contendo os requisitos necessários.
Prioridade:	() Essencial (X) Importante () Desejável

03. DIAGRAMA ENTIDADE-RELAÇONAMENTO



04. MAPEAMENTO DO ESQUEMA CONCEITUAL PARA O ESQUEMA LÓGICO-RELACIONAL

Address (id, public_place, neighborhood, number, complement, city, state, country, last_modified_by, last_modified_date)

Driver (id, name, id_health_center, status, is_active, is_available, last_modified_by, last_modified_date)

HealthCenter (id, id_address, id_prefecture, cnpj, last_modified_by, last_modified_date)

Location (id, id_travel, id_passenger, category, transition, is_preferential, destination_hospital, created_at, last_modified_by, last_modified_date)

Log (id, created_by, created_at, operation, before_operation, after_operation, table_name)

OrderResetPassword (id, id_user_who_requested, id_user_who_solved_order, request_on, completed_id, was_rejected, last_modified_by, last_modified_date)

Passenger (id, name, rg, sus, date_of_birth, gender, id_address, landline, cell_phone, id_health_center, is_active, observation, last_modified_by, last_modified_date)

Prefecture (id, id_address, email, site, cnpj, is_active, last_modified_by, last_modified_date)

Travel (id, departure_date, return_date, destiny_city, destiny_state, id_driver, id_vehicle, id_health_center, status, observation, registered_passengers, last_modified_by, last_modified_date)

User (id, username, password, email, is_staff, is_active, created_at, enrollment, name, id_health_center, last_modified_by, last_modified_date)

user_role (user_id, role_id)

Vehicle (id, name, capacity, id_health_center, plaque, is_active, is_available, status, last_modified_by, last_modified_date)

Role (id, role)

05. DICIONÁRIO DE DADOS

Address (Endereço)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
public_place	Rua / Avenida / etc ... referente ao local / residência.	VARCHAR	NOT NULL
neighborhood	Bairro referente ao local/residência.	VARCHAR	NOT NULL
number	Número referente ao local/residência.	INTEGER	
complement	Casa, Apartamento, Prédio, etc ...	VARCHAR	
city	Cidade referente ao local/residência.	VARCHAR	NOT NULL
state	Estado referente ao local/residência.	VARCHAR	NOT NULL
country	País referente ao local/residência.	VARCHAR	NOT NULL

last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Driver (Motorista)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
name	Nome do motorista.	VARCHAR	NOT NULL
id_health_center	Chave estrangeira referente ao centro de saúde onde o motorista trabalha.	BIGINT	NOT NULL
status	Status do motorista (Ocupado ou Livre).	VARCHAR	NOT NULL
is_active	Atributo referente ao status de ativo ou inativo no sistema.	BOOLEAN	
is_available	Atributo referente ao status de disponível ou não para viagem.	BOOLEAN	
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

HealthCenter (Centro de Saúde)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
id_address	Chave estrangeira referente ao endereço do centro de saúde.	BIGINT	NOT NULL, UNIQUE
id_prefecture	Chave estrangeira referente a prefeitura/convênio daquele centro de saúde.	BIGINT	NOT NULL, UNIQUE
cnpj	Número de identificação de uma pessoa jurídica.	VARCHAR	NOT NULL, UNIQUE
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL

last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL
--------------------	--	---------	----------

Location (Locação - É uma vaga/assento do veículo)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
id_travel	Chave estrangeira referente a viagem que esta locação pertence.	BIGINT	NOT NULL
id_passenger	Chave estrangeira referente ao passageiro que irá ocupar esta locação.	BIGINT	NOT NULL
category	Categoria do passageiro (Paciente ou Acompanhante)	VARCHAR	NOT NULL
transition	Transição do passageiro (Somente Ida, Somente Volta, Ida e Volta).	VARCHAR	NOT NULL
is_preferential	Se o passageiro é preferencial.	BOOLEAN	
destination_hospital	Hospital de destino daquele passageiro.	VARCHAR	NOT NULL
created_at	Data referente ao dia, mês e ano em que a locação foi criada.	DATE	NOT NULL
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Log (Registro de atividades no sistema)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
created_by	Nome do usuário que realizou a operação.	VARCHAR	
create_at	Data em que a opestação foi realizada.	DATE	NOT NULL
operation	Procedimento realizado (INSERT, UPDATE, DELETE)	VARCHAR	NOT NULL

before_operation	Estado do objeto antes da operação.	VARCHAR	
after_operation	Estado do objeto depois da operação.	VARCHAR	NOT NULL
table_name	Nome da tabela que foi realizada.	VARCHAR	NOT NULL

OrderResetPassword (Pedido de redefinição da Senha)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
id_user_who_requested	Chave estrangeira referente ao usuário que fez o pedido de redefinição da senha.	BIGINT	NOT NULL
id_user_who_solved_order	Chave estrangeira referente ao usuário que aprovou a redefinição da senha.	BIGINT	
request_on	Data referente ao dia, mês e ano em que o usuário solicitou o pedido de redefinição de senha.	DATE	NOT NULL
completed_in	Data referente ao dia, mês e ano em que o pedido de redefinição de senha foi aprovado.	DATE	
was_rejected	Atributo referente ao status de rejeitado ou não do pedido de redefinição de senha.	BOOLEAN	
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Passenger (Passageiro)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
name	Nome referente ao funcionário / passageiro.	VARCHAR	NOT NULL
rg	Carteira de identidade referente ao funcionário / passageiro.	VARCHAR	NOT NULL, UNIQUE

sus	Dado referente ao número do cartão do Sistema Único de Saúde.	VARCHAR	NOT NULL, UNIQUE
date_of_birth	Data de nascimento do funcionário / passageiro.	DATE	NOT NULL
gender	Gênero (biológico) do funcionário / passageiro.	VARCHAR	NOT NULL
id_address	Chave estrangeira referente ao endereço do funcionário / passageiro.	BIGINT	NOT NULL
landline	Telefone-fixo (para contato) referente ao funcionário / passageiro.	VARCHAR	
cell_phone	Telefone celular (para contato) referente ao funcionário / passageiro.	VARCHAR	
id_health_center	Chave estrangeira referente ao centro de saúde que funcionário / passageiro está ligado / contratado.	BIGINT	NOT NULL
is_active	Atributo referente ao status de ativo ou inativo no sistema.	BOOLEAN	
observation	Alguma informação adicional referente ao passageiros.	TEXT	
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Prefecture (Prefeitura)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
id_address	Chave estrangeira referente ao endereço da prefeitura.	BIGINT	NOT NULL, UNIQUE
email	Email de contato da prefeitura.	VARCHAR	NOT NULL, UNIQUE
site	Site da prefeitura.	VARCHAR	NOT NULL, UNIQUE
cnpj	Número de identificação de uma pessoa jurídica.	VARCHAR	NOT NULL, UNIQUE

is_active	Atributo referente ao status de ativo ou inativo no sistema.	BOOLEAN	
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Travel (Viagem)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador	BIGINT	NOT NULL, UNIQUE
departure_date	Data de partida da viagem.	DATE	NOT NULL
return_date	Data de retorno da viagem.	DATE	
destiny_city	Cidade de destino daquela viagem.	VARCHAR	NOT NULL
destiny_state	Estado de destino daquela viagem.	VARCHAR	NOT NULL
id_driver	Chave estrangeira referente ao motorista que irá dirigir o veículo dessa viagem.	BIGINT	NOT NULL
id_vehicle	Chave estrangeira referente ao veículo utilizado nessa viagem.	BIGINT	NOT NULL
id_health_center	Chave estrangeira referente ao centro de saúde a que esta viagem está ligada/pertence.	BIGINT	NOT NULL
status	Status da viagem (Em trânsito, aguardando ou concluída).	VARCHAR	NOT NULL
observation	Alguma informação adicional referente à viagem ou passageiros.	TEXT	
registered_passengers	Quantidade de passageiros registrados na viagem.	INTEGER	NOT NULL
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

User - (Usuário do sistema)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
username	Nome do usuário para acessar o sistema.	VARCHAR	NOT NULL, UNIQUE
password	Senha do usuário para acessar o sistema.	VARCHAR	NOT NULL
email	Email de contato do usuário.	VARCHAR	NOT NULL, UNIQUE
is_staff	Atributo referente ao grau de permissão do usuário no sistema.	BOOLEAN	
is_active	Atributo referente ao status de ativo ou inativo no sistema.	BOOLEAN	
created_at	Data referente ao dia, mês e ano que o usuário foi criado.	DATE	NOT NULL
enrollment	Atributo referente a matrícula do usuário de acordo com o contrato do mesmo com a secretaria de saúde.	VARCHAR	
name	Nome do usuário.	VARCHAR	NOT NULL
id_health_center	Chave estrangeira referente ao centro de saúde que o usuário pertence.	BIGINT	NOT NULL
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Vehicle - (Veículo)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
name	Nome do veículo.	VARCHAR	NOT NULL
capacity	Capacidade do veículo.	INTEGER	NOT NULL

id_health_center	Chave estrangeira referente ao centro de saúde que este veículo pertence.	BIGINT	NOT NULL
plaque	Placa de identificação do veículo.	VARCHAR	NOT NULL, UNIQUE
is_active	Atributo referente ao status de ativo ou inativo no sistema.	BOOLEAN	
is_available	Atributo referente ao status de disponível ou indisponível para viagem.	BOOLEAN	
status	Status do veículo (Em uso ou liberado).	VARCHAR	NOT NULL
last_modified_by	Atributo referente ao último usuário que modificou os dados.	VARCHAR	NOT NULL
last_modified_date	Atributo referente a data da última modificação na tabela.	VARCHAR	NOT NULL

Role - (Papel/Função)			
Atributo	Descrição	Tipo de Dado	Restrições
id	Identificador.	BIGINT	NOT NULL, UNIQUE
role	Nome da função/papel.	VARCHAR	NOT NULL, UNIQUE

user_role - (Usuário_Função) Tabela intermediária entre usuário e papel (relação n,n)			
Atributo	Descrição	Tipo de Dado	Restrições
user_id	Chave estrangeira referente ao usuário.	BIGINT	NOT NULL
role_id	chave estrangeira referente a permissão.	BIGINT	NOT NULL

05. SCRIPTS SQL

5.1 DDL

Address
<pre>CREATE TABLE addresses (id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY, last_modified_by varchar(255) NULL, last_modified_date timestamp NULL,</pre>

```

city varchar(255) NULL,
complement varchar(255) NULL,
country varchar(255) NULL,
neighborhood varchar(255) NULL,
"number" int4 NOT NULL,
public_place varchar(255) NULL,
state varchar(2) NOT NULL,
CONSTRAINT addresses_pkey PRIMARY KEY (id)

```

```
);
```

Driver

```

CREATE TABLE drivers (
  id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  last_modified_by varchar(255) NULL,
  last_modified_date timestamp NULL,
  is_active bool NULL,
  is_available bool NULL,
  "name" varchar(255) NULL,
  health_center_id int8 NOT NULL,
  CONSTRAINT drivers_pkey PRIMARY KEY (id)

```

```
);
```

```

ALTER TABLE drivers ADD CONSTRAINT fk96tt889yoovrmgm1shjxdgwww FOREIGN KEY
(health_center_id) REFERENCES health_centers(id);

```

HealthCenter

```

CREATE TABLE health_centers (
  id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  last_modified_by varchar(255) NULL,
  last_modified_date timestamp NULL,
  is_active bool NULL,
  cnpj varchar(18) NOT NULL,
  address_id int8 NULL,
  prefecture_id int8 NULL,
  CONSTRAINT health_centers_pkey PRIMARY KEY (id),
  CONSTRAINT uk_a2wio9at225bgog42g0sy9nse UNIQUE (cnpj)

```

```
);
```

```

ALTER TABLE health_centers ADD CONSTRAINT fkp40p1gpu1jmt904c4fsrpentp FOREIGN
KEY (address_id) REFERENCES addresses(id);

```

```

ALTER TABLE public.health_centers ADD CONSTRAINT fkt33bbvd3rmvikshdggvanvw19
FOREIGN KEY (prefecture_id) REFERENCES public.prefectures(id);

```

Location

```

CREATE TABLE locations (
  id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  last_modified_by varchar(255) NULL,
  last_modified_date timestamp NULL,
  category varchar(13) NOT NULL,
  created_at date NOT NULL,

```



```

destination_hospital varchar(255) NOT NULL,
is_preferential bool NULL,
transition varchar(2) NOT NULL,
passenger_id int8 NOT NULL,
travel_id int8 NOT NULL,
CONSTRAINT locations_pkey PRIMARY KEY (id)
);

ALTER TABLE locations ADD CONSTRAINT fk7dp4k51hx6b08cfk4fm268sex FOREIGN KEY
(travel_id) REFERENCES travels(id);
ALTER TABLE locations ADD CONSTRAINT fkt9mjk3dx6jgoxxdvqwxymhs1 FOREIGN
KEY (passenger_id) REFERENCES passengers(id);

```

Log

```

CREATE TABLE log (
  id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  after_operation text NULL,
  before_operation text NULL,
  created_at date NULL,
  created_by varchar(255) NULL,
  operation text NULL,
  table_name varchar(255) NULL,
  CONSTRAINT log_pkey PRIMARY KEY (id)
);

```

OrderResetPassword

```

CREATE TABLE order_reset_passwords (
  id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  last_modified_by varchar(255) NULL,
  last_modified_date timestamp NULL,
  completed_in date NULL,
  request_on date NULL,
  was_rejected bool NULL,
  id_user_who_requested int8 NULL,
  id_user_who_solved_order int8 NULL,
  CONSTRAINT order_reset_passwords_pkey PRIMARY KEY (id)
);

ALTER TABLE order_reset_passwords ADD CONSTRAINT fkbjhm8xucro55oaf04v211186w
FOREIGN KEY (id_user_who_requested) REFERENCES users(id);
ALTER TABLE order_reset_passwords ADD CONSTRAINT fksfooku7xwkhfdx2sijwx4wklv
FOREIGN KEY (id_user_who_solved_order) REFERENCES users(id);

```

Passenger

```

CREATE TABLE passengers (
  id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  last_modified_by varchar(255) NULL,
  last_modified_date timestamp NULL,
  is_active bool NULL,

```

```

is_blocked bool NULL,
cell_phone varchar(15) NULL,
date_of_birth date NOT NULL,
gender varchar(1) NOT NULL,
in_travel bool NULL,
landline varchar(14) NULL,
"name" varchar(255) NOT NULL,
observation text NULL,
rg varchar(9) NOT NULL,
sus varchar(18) NOT NULL,
warning_count int4 NULL,
address_id int8 NOT NULL,
health_center_id int8 NOT NULL,
CONSTRAINT passengers_pkey PRIMARY KEY (id),
CONSTRAINT uk_rmvc9ssd5o956t7vgkltkjs9 UNIQUE (sus),
CONSTRAINT uk_tlp5vpggq3y02cm022ddc18a2 UNIQUE (rg)
);

ALTER TABLE passengers ADD CONSTRAINT fknherxk6jewocu24bbhsrcln1 FOREIGN
KEY (health_center_id) REFERENCES health_centers(id);
ALTER TABLE passengers ADD CONSTRAINT fks4tsmh5brqxvotas77e1vyou FOREIGN KEY
(address_id) REFERENCES addresses(id);

```

Prefecture

```

CREATE TABLE prefectures (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    last_modified_by varchar(255) NULL,
    last_modified_date timestamp NULL,
    active bool NOT NULL,
    cnpj varchar(18) NOT NULL,
    email varchar(255) NOT NULL,
    site varchar(255) NOT NULL,
    address_id int8 NULL,
    CONSTRAINT prefectures_pkey PRIMARY KEY (id),
    CONSTRAINT uk_a8wosnmwuumeynbvlbstbn84a UNIQUE (email),
    CONSTRAINT uk_l0s8kqq6968s24xwi5or6o1ye UNIQUE (site),
    CONSTRAINT uk_q42en1a3nkk4mp58djuvmfyn8 UNIQUE (cnpj)
);
ALTER TABLE prefectures ADD CONSTRAINT fkmvcjvol68elcqf5lgfh5kuhf9 FOREIGN KEY
(address_id) REFERENCES addresses(id);

```

Role

```

CREATE TABLE roles (
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    "role" varchar(255) NULL,
    CONSTRAINT roles_pkey PRIMARY KEY (id)
);

```

Travel

```
CREATE TABLE travels (  
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,  
    last_modified_by varchar(255) NULL,  
    last_modified_date timestamp NULL,  
    is_active bool NULL,  
    departure_date date NOT NULL,  
    destiny_city varchar(255) NULL,  
    destiny_state varchar(2) NOT NULL,  
    observation text NULL,  
    registered_passengers int4 NOT NULL,  
    return_date date NULL,  
    status varchar(15) NOT NULL,  
    driver_id int8 NOT NULL,  
    health_center_id int8 NOT NULL,  
    vehicle_id int8 NOT NULL,  
    CONSTRAINT travels_pkey PRIMARY KEY (id)  
);  
  
ALTER TABLE travels ADD CONSTRAINT fke6pm03eicduvcxlcx9e5rfe8y FOREIGN KEY  
(driver_id) REFERENCES drivers(id);  
ALTER TABLE travels ADD CONSTRAINT fklspsvxlrfk1u8mufpnhr4x3y FOREIGN KEY  
(vehicle_id) REFERENCES vehicles(id);  
ALTER TABLE travels ADD CONSTRAINT fkqxkktvt7d8q6ac53ku5pt6prk FOREIGN KEY  
(health_center_id) REFERENCES health_centers(id);
```

UserRole

```
CREATE TABLE user_role (  
    user_id int8 NOT NULL,  
    role_id int8 NOT NULL,  
    CONSTRAINT user_role_pkey PRIMARY KEY (user_id, role_id)  
);  
  
ALTER TABLE user_role ADD CONSTRAINT fkj345gk1bovqvfname88rcx7yyx FOREIGN KEY  
(user_id) REFERENCES users(id);  
ALTER TABLE user_role ADD CONSTRAINT fkt7e7djp752sqn6w22i6ocqy6q FOREIGN KEY  
(role_id) REFERENCES roles(id);
```

User

```
CREATE TABLE users (  
    id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY,  
    last_modified_by varchar(255) NULL,  
    last_modified_date timestamp NULL,  
    is_active bool NULL,  
    created_at date NULL,  
    email varchar(255) NOT NULL,  
    enrollment varchar(7) NULL,  
    "name" varchar(255) NOT NULL,  
    "password" varchar(255) NOT NULL,
```

```

is_staff bool NULL,
username varchar(255) NOT NULL,
health_center_id int8 NULL,
CONSTRAINT uk_6dotkott2kjsp8vw4d0m25fb7 UNIQUE (email),
CONSTRAINT uk_r43af9ap4edm43mmtq01oddj6 UNIQUE (username),
CONSTRAINT users_pkey PRIMARY KEY (id)
);

ALTER TABLE users ADD CONSTRAINT fk86p8hntpqj2qjj3i074ck325i FOREIGN KEY
(health_center_id) REFERENCES health_centers(id);

```

Vehicle
<pre> CREATE TABLE vehicles (id int8 NOT NULL GENERATED BY DEFAULT AS IDENTITY, last_modified_by varchar(255) NULL, last_modified_date timestamp NULL, is_active bool NULL, is_available bool NULL, capacity int4 NOT NULL, "name" varchar(255) NULL, plaque varchar(255) NULL, health_center_id int8 NOT NULL, CONSTRAINT vehicles_pkey PRIMARY KEY (id)); ALTER TABLE vehicles ADD CONSTRAINT fk6l20i2d7by0r5e7u741svdnjw FOREIGN KEY (health_center_id) REFERENCES health_centers(id); </pre>

5.2. DML

findByHealthCenters
(Buscando todos os OrderResetPassword de acordo com o centro de saúde e usuário que fez a solicitação)
<pre> select orp.* from order_reset_passwords orp, users u where id_user_who_requested != ?2 and u.health_center_id = ?1 and u.id = id_user_who_requested " + order by orp.request_on desc; </pre>

findHealthCentersByCity
(Trás todos os centros de saúde de acordo com a cidade informada)
<pre> select hc.* from health_centers hc, prefectures p, addresses a WHERE UPPER(a.city) LIKE UPPER(CONCAT('%',?1,'%')) and hc.prefecture_id = p.id and p.address_id = a.id; </pre>

findLocationByPassengerAndTravelStatus
(Trás todas as locações de acordo com o status da viagem e pelo id do passageiro)
<pre> select * from locations l, travels t where l.passenger_id = ?1 and t.status = ?2 and t.id = l.travel_id; </pre>

findByHealthCenters

(Buscando todos os OrderResetPassword de acordo com o centro de saúde e usuário que fez a solicitação)

```
select orp.* from order_reset_passwords orp, users u where id_user_who_requested != ?2 and u.health_center_id = ?1 and u.id = id_user_who_requested " + order by orp.request_on desc
```

findUsersByHealthCenter

(Buscando todos os usuários de acordo com o centro de saúde e exceto o autenticado)

```
SELECT * FROM USERS WHERE health_center_id = ?1 EXCEPT SELECT * FROM USERS WHERE id = ?2;
```

findDriversByHealthCenterAndAvailableAndActive

(Trás todos os motoristas de acordo com o centro de saúde, status disponível e se está ativo.)

```
SELECT * FROM DRIVERS WHERE health_center_id = ?1 and is_available = ?2 and is_active = ?3;
```

findOrderResetPasswordByActive

(Trás todos OrderResetPassword do usuário informado e que o completed_in for null)

```
select * from order_reset_passwords where id_user_who_requested = ?1 and completed_in is null;
```

findAllOrderResetpasswordByOrderDesc

(Trás todos os OrderResetPassword em ordem “desc”)

```
select * from order_reset_passwords order by request_on desc;
```

findRolesExceptAdmin

(Trás todas as “ROLES” exceto o admin)

```
"SELECT * from roles where role != 'ADMIN'"
```

findUserExceptAuthenticated

(Trás todos os usuários menos o que está logado)

```
SELECT * FROM USERS EXCEPT SELECT * FROM USERS WHERE id = ?1
```