**Centro Universitário UNISATC**

Engenharia de Software 3a fase – Banco de Dados II – Prof.Jorge Luiz da Silva

**TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM**

**Projeto de banco de dados para um sistema de Gerenciamento de Estacionamento**

Realdo Justino Junior - https://github.com/Realdo-Justino

Paulo Ronchi Francisconi- https://github.com/pauloFrancisconi

Rhyan Willemann Orben - https://github.com/rhyanorben

Igor Rocha Speck - https://github.com/igrocha

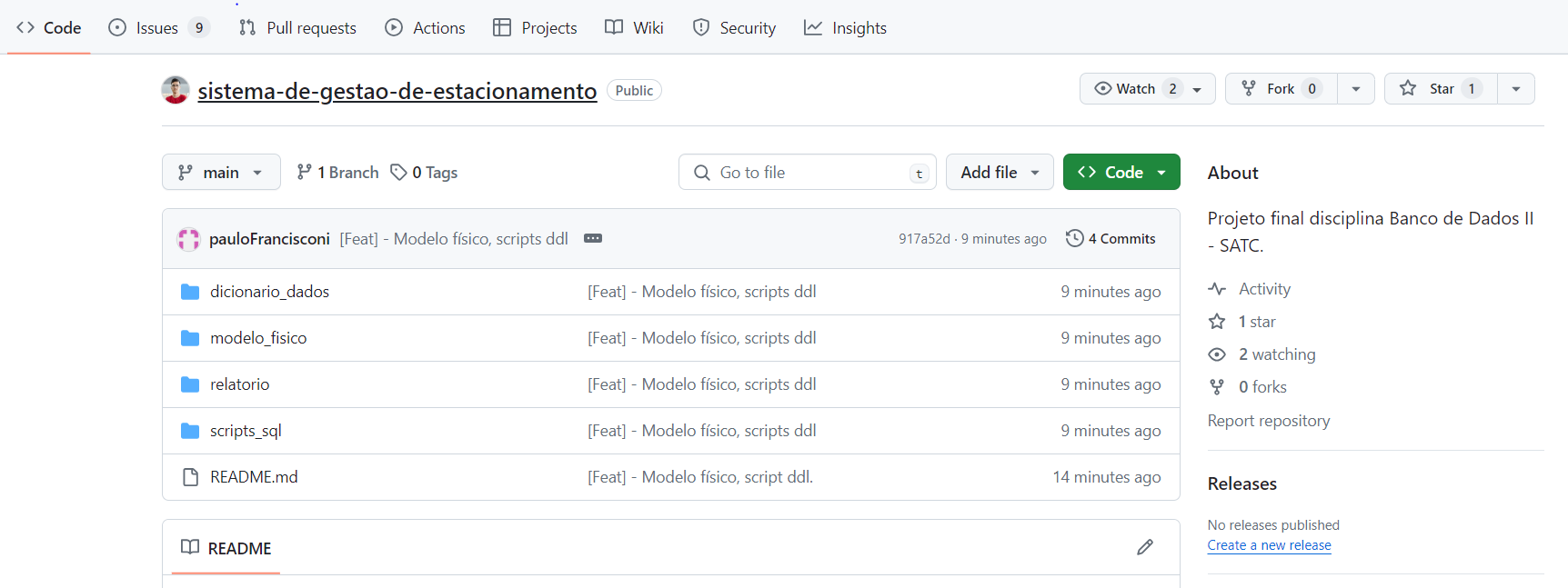
João Augusto - https://github.com/Joao-AugustoPF

Wesley Schardosim Machado - https://github.com/Wesley-SM

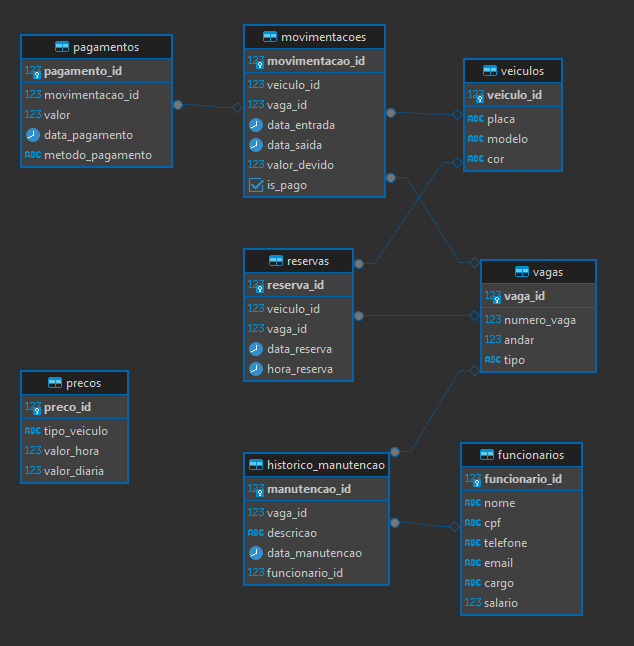
Criciúma, 09/06/2024

**URL do projeto no GitHub**

<https://github.com/Joao-AugustoPF/sistema-de-gestao-de-estacionamento>



**Modelo ER Físico**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tabela** | funcionarios | | | | | | |
| **Descrição** | Tabela responsável por armazenar os funcionários cadastrados | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| funcionario\_id | | int | 1 – sem limite | NOT NULL | X |  | Código identificador do funcionário |
| nome | | varchar(70) | 1 – 70 | NOT NULL |  | X | Nome do funcionário |
| cpf | | varchar(11) | 0 – 11 |  |  |  | CPF do funcionário |
| telefone | | varchar(11) | 0 – 11 |  |  |  | Telefone do funcionário |
| email | | varchar(40) | 0 – 40 |  |  |  | E-mail do funcionário |
| cargo | | varchar(30) | 0 – 30 |  |  |  | Cargo do funcionário |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |
|  | |  |  |  |  | | |

**Dicionário de Dados**

**Script dos comandos DDL para criação do Banco de dados**

-- Tabela De Permissoes

CREATE TABLE Permissoes (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

descricao VARCHAR(50) NOT NULL

);

-- Tabela de funcionarios

CREATE TABLE Funcionarios (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

nome VARCHAR(100) NOT NULL,

cpf VARCHAR(14) NOT NULL,

email VARCHAR(40) NOT NULL,

telefone VARCHAR(11) NOT NULL,

ativo BOOLEAN DEFAULT TRUE,

cargo VARCHAR(50),

permissao\_id INT UNSIGNED,

FOREIGN KEY (permissao\_id) REFERENCES Permissoes(id)

);

-- Tabela De Veiculos

CREATE TABLE Veiculos (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

placa VARCHAR(10) NOT NULL,

tipo\_veiculo ENUM('Moto', 'Caminhão', 'Carro') NOT NULL

);

--Tabela Tipo de Vagas

CREATE TABLE TipoVaga (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

descricao VARCHAR(50) NOT NULL

);

-- Tabela Vagas

CREATE TABLE Vagas (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

numero INT UNSIGNED NOT NULL,

tipo\_id INT UNSIGNED,

veiculo\_id INT UNSIGNED,

disponivel TINYINT(1) DEFAULT 1,

FOREIGN KEY (tipo\_id) REFERENCES TipoVaga(id),

FOREIGN KEY (veiculo\_id) REFERENCES Veiculos(id),

INDEX (tipo\_id),

INDEX (veiculo\_id)

);

-- Tabela Tarifa

CREATE TABLE Tarifas (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

tipo\_id INT UNSIGNED,

valor\_por\_hora DECIMAL(10, 2),

FOREIGN KEY (tipo\_id) REFERENCES TipoVaga(id),

INDEX (tipo\_id)

);

-- Tabela Pagamentos

CREATE TABLE Pagamentos (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

movimentacao\_id INT UNSIGNED,

metodo\_pagamento ENUM('Cartão', 'Dinheiro', 'Pix') NOT NULL,

valor\_pago DECIMAL(10, 2) NOT NULL,

data\_pagamento TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (movimentacao\_id) REFERENCES Movimentacoes(id),

INDEX (movimentacao\_id)

);

-- Tabela Movimentacoes

CREATE TABLE Movimentacoes (

id INT UNSIGNED PRIMARY KEY AUTO\_INCREMENT,

vaga\_id INT UNSIGNED,

veiculo\_id INT UNSIGNED,

entrada TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

saida TIMESTAMP NULL,

FOREIGN KEY (vaga\_id) REFERENCES Vagas(id),

FOREIGN KEY (veiculo\_id) REFERENCES Veiculos(id),

INDEX (vaga\_id),

INDEX (veiculo\_id)

);

**Script que popula as tabelas do Banco de dados**

/\* ADICIONA AS PERMISSÕES\*/

INSERT INTO Permissoes (descricao) VALUES

('Administrador'),

('Gerente'),

('Atendente'),

('Segurança'),

('Limpeza');

/\* \*/

INSERT INTO Funcionarios (nome, cpf, email, telefone, ativo, cargo, permissao\_id) VALUES

('João Silva', '123.456.789-00', 'joao.silva@example.com', '11999999999', TRUE, 'Gerente de Operações', 2),

('Maria Souza', '987.654.321-00', 'maria.souza@example.com', '11988888888', TRUE, 'Atendente', 3),

('Carlos Pereira', '111.222.333-44', 'carlos.pereira@example.com', '11977777777', TRUE, 'Supervisor', 1),

('Ana Costa', '222.333.444-55', 'ana.costa@example.com', '11966666666', TRUE, 'Segurança', 4),

('Pedro Lima', '333.444.555-66', 'pedro.lima@example.com', '11955555555', TRUE, 'Limpeza', 5),

('Laura Mendes', '444.555.666-77', 'laura.mendes@example.com', '11944444444', TRUE, 'Atendente', 3);

/\* \*/

INSERT INTO Veiculos (placa, tipo\_veiculo) VALUES

('ABC-1234', 'Carro'),

('XYZ-5678', 'Moto'),

('JKL-9101', 'Caminhão'),

('MNO-2345', 'Carro'),

('PQR-6789', 'Moto'),

('STU-3456', 'Caminhão'),

('VWX-7890', 'Carro');

/\* \*/

INSERT INTO TipoVaga (descricao) VALUES

('Carro'),

('Moto'),

('Caminhão'),

('Elétrico'),

('Especial');

/\* \*/

INSERT INTO Vagas (numero, tipo\_id, veiculo\_id, disponivel) VALUES

(1, 1, 1, 1),

(2, 2, 2, 1),

(3, 3, 3, 1),

(4, 1, 4, 1),

(5, 2, 5, 1),

(6, 3, 6, 1),

(7, 1, 7, 1),

(8, 4, NULL, 1),

(9, 5, NULL, 1);

/\* \*/

INSERT INTO Tarifas (tipo\_id, valor\_por\_hora) VALUES

(1, 5.00),

(2, 3.00),

(3, 7.00),

(4, 6.50),

(5, 10.00);

/\* \*/

INSERT INTO Movimentacoes (vaga\_id, veiculo\_id, entrada) VALUES

(1, 1, '2024-06-11 08:00:00'),

(2, 2, '2024-06-11 09:00:00'),

(3, 3, '2024-06-11 10:00:00'),

(4, 4, '2024-06-11 11:00:00'),

(5, 5, '2024-06-11 12:00:00'),

(6, 6, '2024-06-11 13:00:00'),

(7, 7, '2024-06-11 14:00:00');

/\* \*/

INSERT INTO Pagamentos (movimentacao\_id, metodo\_pagamento, valor\_pago) VALUES

(1, 'Cartão', 15.00),

(2, 'Dinheiro', 9.00),

(3, 'Pix', 21.00),

(4, 'Cartão', 30.00),

(5, 'Dinheiro', 18.00),

(6, 'Pix', 42.00),

(7, 'Cartão', 25.00);GO

**Principais consultas mapeadas baseadas em regras de negócio (mínimo 4)**

--13. (Aberta) Utilizando a linguagem SQL, execute consultas que possam responder a seguinte questão: Qual(is) Carro(s) possui(em) mais sinistros cadastrados?

select s.placa, COUNT(\*)

from sinistro s inner join carro c

on s.placa = c.placa

group by s.placa

order by COUNT(\*) desc

--14. (Aberta) Quantos clientes cadastrados possuem letra “a” no meio e não no final?

select \* --count(\*)

from cliente

where nome like '%a%'

and nome not like '%a'

--15. (Aberta) Quantos clientes não possuem apólice?

select nome, cod\_apolice --\* --count(\*)

from cliente c left join apolice a

on c.cod\_cliente = a.cod\_cliente

where cod\_apolice is null

select nome, cod\_apolice

from cliente c inner join apolice a on c.cod\_cliente = a.cod\_cliente

--16. (Aberta) Quais carros não possuem apólice?

select \* --count(\*)

from carro c left join apolice a on c.placa = a.placa

where cod\_apolice is null

select \*

from apolice

where placa not in (select placa from carro)