**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**
**INSTITUTO METRÓPOLE DIGITAL - IMD**
**BANCO DE DADOS**

# MODELO DE BANCO DE DADOS PARA APLICATIVO DE AGENDAMENTO DE SERVIÇOS AUTOMOTIVOS

PROFESSOR: RODRIGO TEIXEIRA RAMOS

ALUNOS: JOÃO VICTOR MARQUES DE OLIVEIRA
MATHEUS FELLIPE DA COSTA ANDRADE

NATAL, NOVEMBRO DE 2017

## INTRODUÇÃO

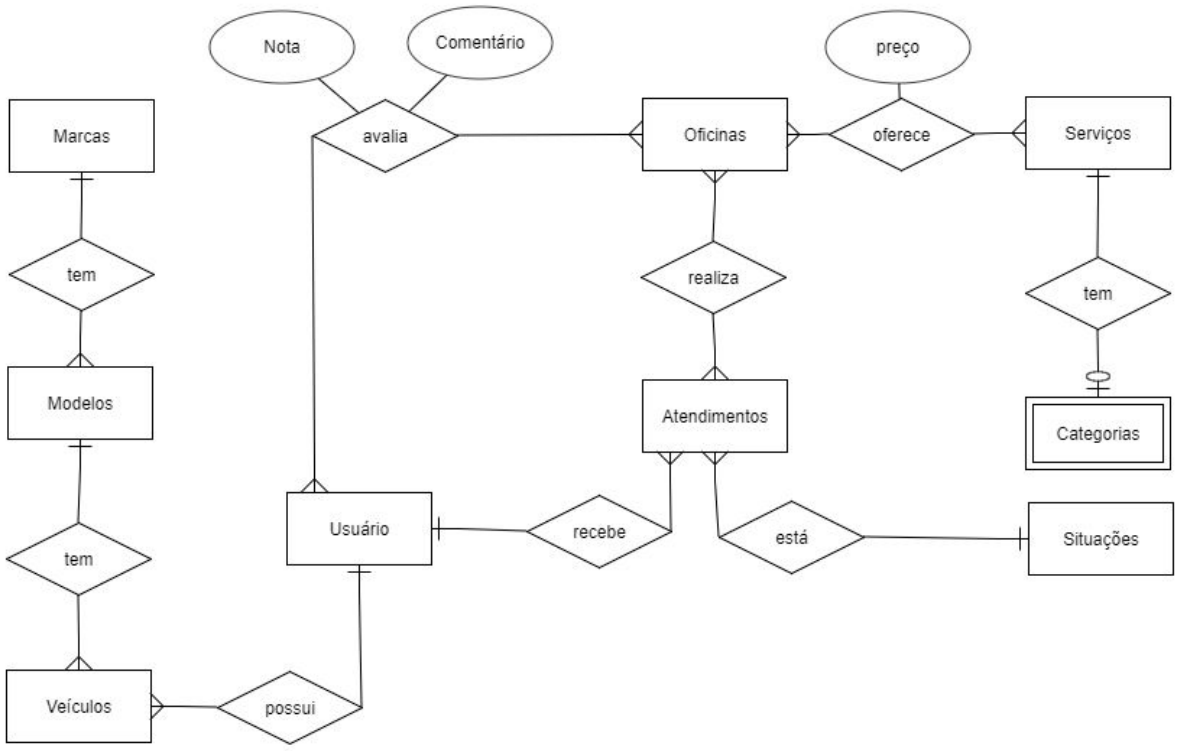
A inserção dos modos de gerenciamento de negócios no mundo da tecnologia é de grande importância para o crescimento e consolidação de um projeto em seu mercado. É nessa perspectiva que o presente trabalho se propõe a apresentar um projeto de banco de dados relacional no SGBD PostgreSQL e com suas regras de negócio geridas internamente por constraints, triggers e procedures.

## 1. OBJETIVOS

O banco de dados a ser descrito apresenta uma solução para o gerenciamento de um modelo de negócio com o intuito de disponibilizar um serviço de atendimento ágil, prático e seguro para automóveis em oficinas mecânicas. Esse serviço fornece ao dono de um veículo uma interface de agendamento com uma oficina, onde é possível realizar o cadastramento de veículos, solicitação de serviços, acompanhamento de agendamentos e avaliação de atendimentos já realizados.

As regras de negócio da aplicação são todas gerenciada através de técnicas de controle do banco de dados. A trigger ***verifica_atendimento***, por exemplo, é extremamente importante para as etapa de realização de atendimentos. Com seu uso, é possível restringir a realização de atendimentos por um usuário caso ele ainda não tenha realizado a avaliação de um atendimento já finalizado. Todo o controle de cadastro de atendimentos e de seus respectivos serviços é tratado por meio do uso de um procedure denominado ***InsereAtendimento***. Essa função possui a responsabilidade de após a inserção de dados na tabela atendimentos, os dados dos serviços daquele determinado atendimento serem incluídos automaticamente. Além disso, o uso desses procedimentos nos permite garantir uma maior integridade dos dados do banco de dados.

## 2. MODELO CONCEITUAL



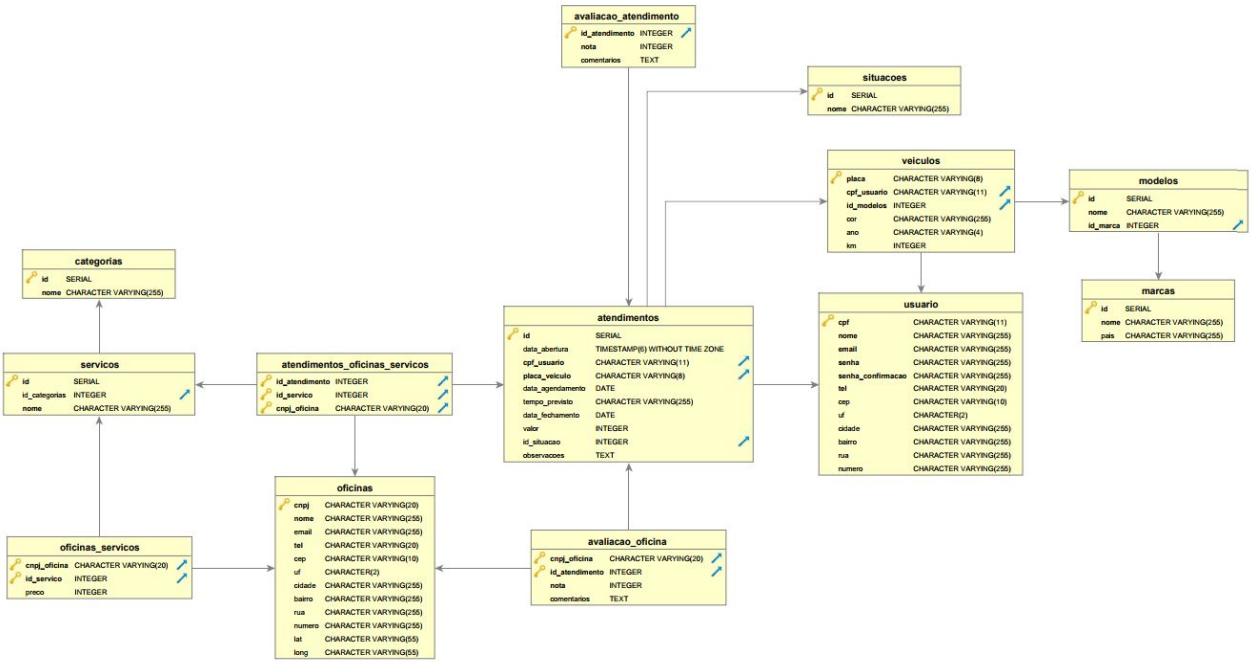

O modelo entidade-relacionamento define o funcionamento de cada entidade no projeto. Vale destacar, aqui, o objetivo de servir de intermediário entre oficinas e usuários, o que pode ser notado ao analisar o modelo. As entidades ***Usuário*** e ***Oficina*** são as principais e o motivo das demais existirem. ***Atendimento***, por exemplo, tem função de relacionar um usuário e várias oficinas.

Como requisitos dos modelo conceitual, podemos citar, entre outros:

a. Entidades Fortes: ***Usuário***, ***Oficina***, ***Modelos*** e ***Marcas***;
b. Entidade Fraca: ***Categorias***;
c. Relacionamento com atributos: ***avalia***, entre ***Usuário*** e ***Oficina***, e ***oferece***, entre ***Oficinas*** e ***Serviços***;
d. Relacionamento **n**x**m**: ***oferece***, entre ***Oficinas*** e ***Serviços***; Relacionamento **1**x**n**: ***tem***, entre ***Marcas*** e ***Modelos***.

Vale ressaltar que, para fins de legibilidade, os atributos de cada entidade serão mostrados apenas no próximo diagrama, que define o modelo lógico da aplicação.

## 3. MODELO LÓGICO




O modelo lógico está definido no diagrama acima, que pode ser melhor visualizado no final do relatório. Está devidamente normalizado na 3FN, especificando todas as tabelas e campos do sistema.

## 4. MODELO FÍSICO

Sobre os requisitos do modelo físico, é possível citar os seguintes exemplos:

(i) 10 tabelas: ***oficinas***, ***usuario***, ***marcas***, ***modelos***, ***servicos***, ***atendimentos***, ***avaliacao_oficina***, ***avaliacao_atendimento***, ***atendimentos_oficinas_servicos***, ***oficinas_servicos***.

(ii) Observando o modelo apresentado, pode-se notar as seguintes propriedades:

(a) Chave-primária simples: **cpf**, na tabela ***usuario***;

(b) Chave-primária composta: **cnpj_oficina** e **id_atendimento**, na tabela ***avaliacao_oficina***;

(c) Chaves naturais, quando possível: **cpf** (tabela ***usuário***), **cnpj** (tabela ***oficina***), **placa** (tabela ***veiculos***).

(iii) Chaves estrangeiras: observar modelo.

(iv) Contraints: 2 do tipo **CHECK** para restringir **nota**, na tabela ***avaliacao***, e **data_fechamento**, na tabela ***atendimento***.

Triggers: **verifica_atendimento** não permite criar novo atendimento sem avaliar anterior e **verifica_senha** valida a senha do usuário.

(v) Procedure: **InsereAtendimento** é responsável por inserir pelo menos um serviço na criação de cada atendimento, vinculando a inserção na tabela ***atendimento*** a ***atendimento_servico***.

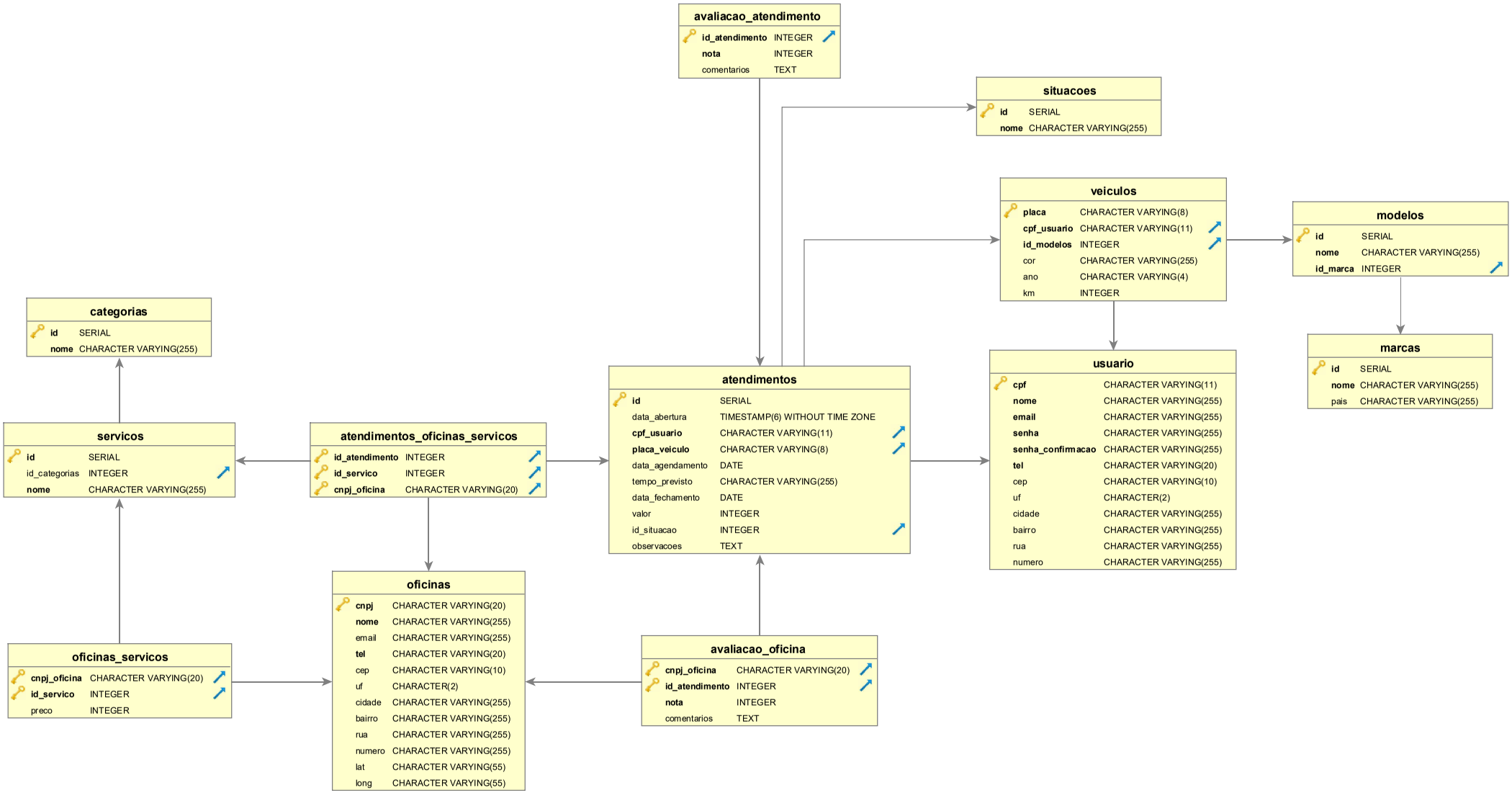
A instruções INSERT, UPDATE e DELETE podem ser observadas no arquivo “DML”, enviado via SIGAA.

(vi) As consultas podem ser observadas no arquivo “DQL” enviado via SIGAA.

## 5. CONCLUSÃO

O ensino tradicional pode adquirir um viés demasiadamente teórico. Desse modo, criar um banco de dados observando os critérios propostos e seguindo as normas possibilita consolidar o conhecimento desenvolvido ao longo do curso de Banco de Dados de maneira bastante plena.

O projeto conseguiu cumprir boa parte dos objetivos de maneira sólida e, muito provavelmente, serviria de base de dados para uma aplicação real. Além disso, possibilitou melhor conhecimento acerca das ferramentas de desenvolvimento com banco de dados e da criação de diagramas, que organizam e fundamentam qualquer projeto do tipo.


avaliacao_atendimento
id_atendimento INTEGER
nota INTEGER
comentarios TEXT
situacoes
id SERIAL
nome CHARACTER VARYING(255)
veiculos
placa CHARACTER VARYING(8)
cpf_usuario CHARACTER VARYING(11)
id_modelos INTEGER
cor CHARACTER VARYING(255)
ano CHARACTER VARYING(4)
km INTEGER
modelos
id SERIAL
nome CHARACTER VARYING(255)
id_marca INTEGER
categorias
id SERIAL
nome CHARACTER VARYING(255)
marcas
id SERIAL
nome CHARACTER VARYING(255)
pais CHARACTER VARYING(255)
atendimentos
id SERIAL
data_abertura TIMESTAMP(6) WITHOUT TIME ZONE
cpf_usuario CHARACTER VARYING(11)
placa_veiculo CHARACTER VARYING(8)
data_agendamento DATE
tempo_previsto CHARACTER VARYING(255)
data_fechamento DATE
valor INTEGER
id_situacao INTEGER
observacoes TEXT
usuario
cpf CHARACTER VARYING(11)
nome CHARACTER VARYING(255)
email CHARACTER VARYING(255)
senha CHARACTER VARYING(255)
senha_confirmacao CHARACTER VARYING(255)
tel CHARACTER VARYING(20)
cep CHARACTER VARYING(10)
uf CHARACTER(2)
cidade CHARACTER VARYING(255)
bairro CHARACTER VARYING(255)
rua CHARACTER VARYING(255)
numero CHARACTER VARYING(255)
servicos
id SERIAL
id_categorias INTEGER
nome CHARACTER VARYING(255)
atendimentos_oficinas_servicos
id_atendimento INTEGER
id_servico INTEGER
cnpj_oficina CHARACTER VARYING(20)
oficinas
cnpj CHARACTER VARYING(20)
nome CHARACTER VARYING(255)
email CHARACTER VARYING(255)
tel CHARACTER VARYING(20)
cep CHARACTER VARYING(10)
uf CHARACTER(2)
cidade CHARACTER VARYING(255)
bairro CHARACTER VARYING(255)
rua CHARACTER VARYING(255)
numero CHARACTER VARYING(255)
lat CHARACTER VARYING(55)
long CHARACTER VARYING(55)
oficinas_servicos
cnpj_oficina CHARACTER VARYING(20)
id_servico INTEGER
preco INTEGER
avaliacao_oficina
cnpj_oficina CHARACTER VARYING(20)
id_atendimento INTEGER
nota INTEGER
comentarios TEXT