

CENTRO UNIVERSITÁRIO CESMAC  
SISTEMA DE INFORMAÇÃO  
PROGRAMAÇÃO PARA WEB

## **ENTREGA DO SISTEMA DE BOLÃO**

Italo Wanderson Costa Oliveira

Maceió, Novembro de 2019

## **RESUMO**

Neste documento se encontra a entrega do trabalho final da disciplina de Programação para Web, onde irá constar os passos para configuração e execução do sistema para validar entrega do trabalho.

**Aluno:**

Luiz Paulo de Melo Barroca

## INTRODUÇÃO

O projeto é um sistema para gerenciamento de uma clínica veterinária, onde são cadastrados os clientes e os pets que serão atendidos na clínica.

Para o desenvolvimento do backend foi utilizada a linguagem de programação Java 12 com o Framework Spring Boot que usa o modelo de ORM que faz parte do requisito do projeto, utilizando o banco de dados PostgreSQL.

No front foi utilizado Javascript com o Framework Vue.JS para consumir os dados da aplicação Backend.

O projeto está em repositório aberto no seguinte link:

<https://github.com/luizpaulobarroca/prog-web>

Primeiramente para executar o projeto é necessário instalar o Java 12 para o backend e o Node.js para o frontend. Instale o rails usando o Java 12 com o arquivo baixado no site da oracle, o Node em (<https://nodejs.org/en/>) caso esteja usando Windows (como este que vos escreve) e o PostgreSQL.

Configurando o banco de dados e crie um banco de dados chamado **progweb**, acesse o arquivo **application.properties** em /api/src/main/resources e edite o username e o password para o usuário do seu banco de dados postgresQL

Após instalar as dependências do projeto, baixe o projeto do repositório execute os seguintes comandos.

Acesse o banco de dados criado para a aplicação e execute os seguintes comandos SQL:

```
create table addresses (id bigserial not null, cep varchar(255) not null, city varchar(255) not null, complement varchar(255), number varchar(255) not null, street varchar(255) not null, uf varchar(2) not null, primary key (id));
```

```
create table customers (id bigserial not null, cpf varchar(255) not null, email
varchar(255) not null, name varchar(255) not null, phone varchar(255) not null, type
varchar(255) not null, address_id int8 not null, primary key (id));
```

```
create table pets (id bigserial not null, birth_date date not null, name varchar(255)
not null, specie varchar(255) not null, weight float8 not null, customer_id int8, primary
key (id));
```

```
alter table customers add constraint FKm07rwxbyxjv8r4fcye1ff6rmq foreign key
(address_id) references addresses;
```

```
alter table pets add constraint FKhili9xtogoo1wl2rx11t95u2e foreign key
(customer_id) references customers;
```

Acesse a pasta **api** pelo terminal e execute o seguinte comando:

```
./mvnw spring-boot:run (Unix)
```

```
./mvnw.cmd spring-boot:run (windows)
```

E estas mensagens devem aparecer no terminal:

```
2019-12-06 16:52:27.604 INFO 19905 --- [ restartedMain]
o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080
(http) with context path '/api/v1'
```

```
2019-12-06 16:52:27.607 INFO 19905 --- [ restartedMain]
com.cesmac.api.ApiApplication : Started ApiApplication in 4.607 seconds
(JVM running for 5.216)
```

Provavelmente o servidor estará funcionando neste momento acesse <http://localhost:8080/api/v1/> se a imagem abaixo aparecer é porque o servidor está funcionando e agora iremos partir para o frontend.

---

# Whitelabel Error Page

This application has no explicit mapping for /error, so you are seeing this as a fallback.

Fri Dec 06 16:52:29 BRT 2019

There was an unexpected error (type=Not Found, status=404).

No message available

Para executar o frontend acesse a pasta **front** e execute os seguintes comandos:

npm install

npm serve

ou

yarn install

yarn serve

Se tudo ocorrer bem acesse o <http://localhost:8081/> e no canto superior esquerdo em clientes. Pronto, você estará no módulo implementado.

Nesta tela haverá a listagem dos clientes onde cada cliente poderá ser editado ou excluído e também o botão de **Novo** para criar um novo cliente.