



CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE
E DESENVOLVIMENTO DE SISTEMAS -
Enterprise Architectures Development

Challenge Plusoft - Sprint 3

SigmaBin

Integrantes Grupo Sigma

2TDSG - Diogo Amaral RM: 85708

2TDSG - Gustavo Pereira dos Santos RM: 85937

2TDSR - Laura Luz Cabral RM: 85385

2TDSR - Lucas Castro RM: 85164

2TDSG - Paulo Fernando Moncaio Avelar Muniz RM: 84372

2TDSG - Víctor Madrid Davanço RM: 85824

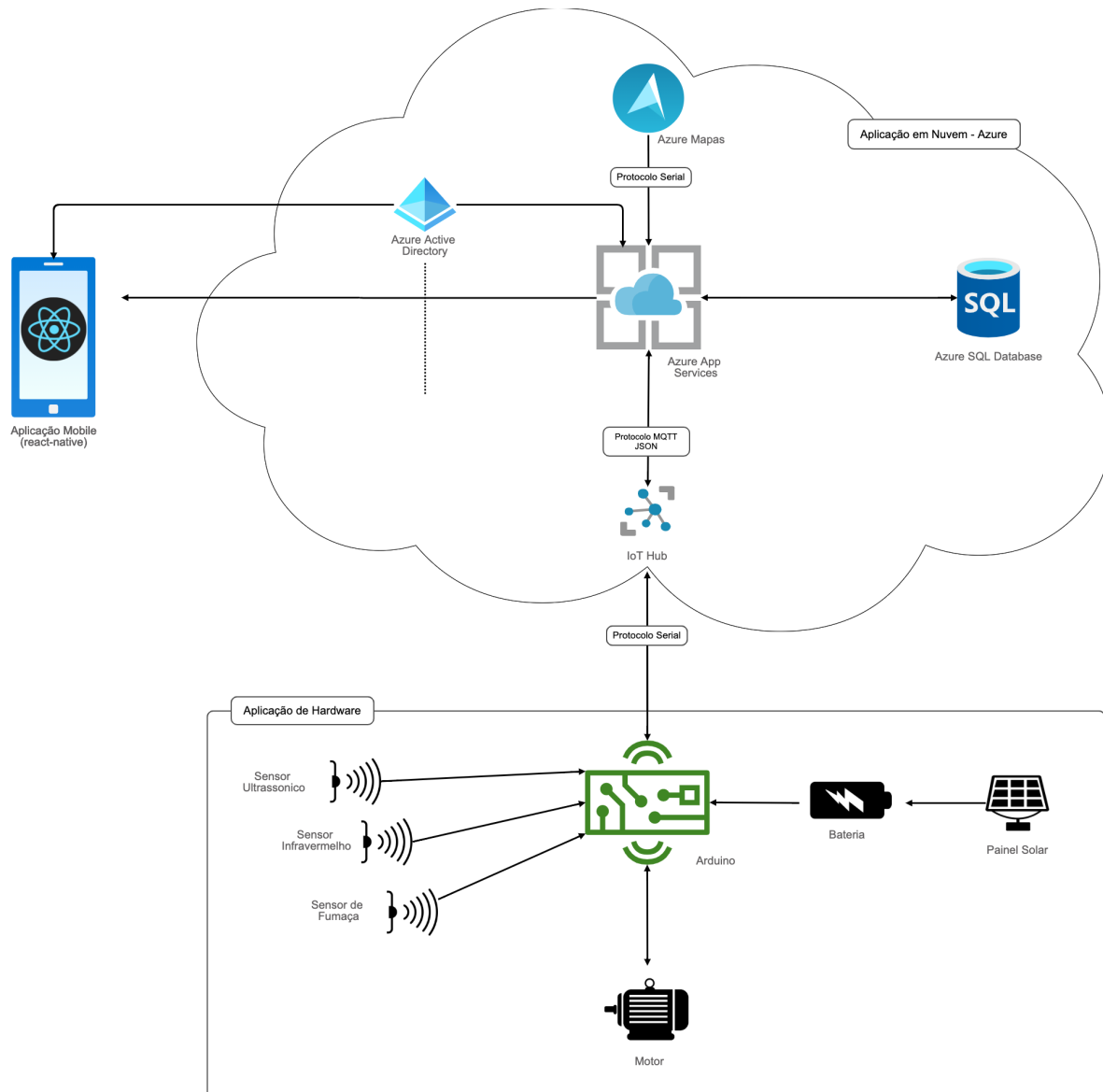
São Paulo 2021

1. OBJETIVO E ESCOPO DO PROJETO

O aumento da concentração da população urbana enfrenta um grande dilema quando se trata de geração de resíduos e seu descarte. Quando não há mecanismo eficiente de gestão urbana, contribuímos para um desenfreio de descarte impróprio que conseqüentemente acarreta em poluição ao meio ambiente. Para evitar esse tipo de cenário e em congruência com os conceitos de Smart Cities, o grupo Sigma visa propor uma solução que integre a gestão de descarte de resíduos a tecnologias. Por meio de IoT, Sensores, Dispositivos, Cloud Storage e Aplicativo buscaremos desenvolver o SigmaBin, uma lixeira inteligente que será capaz de mensurar o volume de resíduo contido, compacta-lo quando necessário, e enviar status ao servidor para que remotamente uma equipe responsável consiga monitorar e melhor decidir em qual momento recolher o resíduo ou realizar manutenções das lixeiras de descarte desses resíduos. Como medidas de segurança, adicionaremos um sensor de fumaça em caso da lixeira pegar fogo por algum descarte de bitucas, e um sensor de presença que impede o processo de compactação caso acionado.

A solução proposta resultará em uma aplicação mobile que se comunicará com um servidor e um banco de dados hospedados na nuvem. Como servidor propomos a utilização do Azure da Microsoft e como banco de dados o SQL da Azure. Para a integração da API entre o servidor e a plataforma de prototipação utilizaremos o IoT-hub (também da Azure). Sendo que a comunicação entre o Node-red e o servidor ocorrerá por intermédio do Protocolo MQTT JSON e a comunicação com o arduino (placa de prototipação) se valerá do protocolo serial. Já o arduino será o responsável por receber os inputs gerados pelos sensores e comunicá-los às demais instâncias da aplicação. Finalmente, como sensores, utilizaremos: sensores infravermelho, sensores ultrassônicos, sensores de fumaça, braço compactador de lixo, e baterias de armazenamento de energia.

2. ARQUITETURA ATUALIZADA



3. TABELA DOS ENDPOINTS

```
--drop table tb_usuario;
--drop table tb_admin;
--drop table tb_lixeira;
--drop table tb_complexo;
--drop table tb_endereco;

CREATE TABLE tb_endereco (
  id_endereco bigint PRIMARY KEY ,
  rua VARCHAR(50) NOT NULL,
  numero VARCHAR(50) NOT NULL,
  cep VARCHAR(50) NOT NULL,
  cidade VARCHAR(50) NOT NULL,
  estado VARCHAR(50) NOT NULL,
  pais VARCHAR(50) NOT NULL);
CREATE TABLE tb_complexo (
  id_complexo bigint PRIMARY KEY ,
  id_endereco bigint NOT NULL
REFERENCES tb_endereco,
  nome VARCHAR(50) NOT NULL);
CREATE TABLE tb_usuario (
  id_usuario bigint PRIMARY KEY ,
  id_complexo bigint NOT NULL
REFERENCES tb_complexo,
  nome VARCHAR(50) NOT NULL,
  email VARCHAR(50) NOT NULL,
  telefone VARCHAR(20) NOT NULL,
  senha VARCHAR(30)
);
CREATE TABLE tb_admin (
  id_admin bigint PRIMARY KEY ,
  id_complexo bigint NOT NULL
REFERENCES tb_complexo,
  nome VARCHAR(50) NOT NULL,
  email VARCHAR(50) NOT NULL,
  telefone bigint(11) NOT NULL,
  senha VARCHAR(100)
);
CREATE TABLE tb_lixeira (
  id_lixeira bigint PRIMARY KEY ,
  id_endereco bigint NOT NULL
REFERENCES tb_endereco,
  status VARCHAR(50)
CHECK( status IN ('cheio','vazio','meio cheio', 'manutenção') ),
  id_complexo bigint NOT NULL
REFERENCES tb_complexo,
  capacidade VARCHAR(50) NOT NULL,
  nivel_residuo bigint NOT NULL,
  tipoResiduo VARCHAR(50)
CHECK( tipoResiduo IN ('orgânico', 'reciclável'))
);
```

```

-- INSERTS
-- tb_endereco
INSERT INTO tb_endereco (id_endereco, rua, numero, cep, cidade, estado, pais) VALUES (
1,
'Rua das alpineias',
'21',
'75-887-091',
'Goiania',
'GO',
'Brasil');
INSERT INTO tb_endereco (id_endereco, rua, numero, cep, cidade, estado, pais) VALUES (
2,
'Rua das rosas',
'11',
'75-887-000',
'Goiania',
'GO',
'Brasil');
INSERT INTO tb_endereco (id_endereco, rua, numero, cep, cidade, estado, pais) VALUES (
3,
'Rua dos pinheiros',
'13',
'05-887-000',
'São Paulo',
'SP',
'Brasil');

```

```

-- INSERTS
-- tb_complexo
INSERT INTO tb_complexo (id_complexo, id_endereco, nome) VALUES (
1,
1,
'Parque da Lagoa');
INSERT INTO tb_complexo (id_complexo, id_endereco, nome) VALUES (
2,
2,
'Parque das Rosas');
INSERT INTO tb_complexo (id_complexo, id_endereco, nome) VALUES (
3,
3,
'Parque Ibirapuera');

```

```

-- INSERTS
-- tb_usuario
INSERT INTO tb_usuario (id_usuario, id_complexo, nome, email, telefone, senha) VALUES (
1,
1,
'Laura Luz',
'lauraluza@gmail.com',
'62999996767',
'senha1');
INSERT INTO tb_usuario (id_usuario, id_complexo, nome, email, telefone, senha) VALUES (
2,
1,
'Lucas Castro',
'lucass@gmail.com',
'62999992222',
'senha2');
INSERT INTO tb_usuario (id_usuario, id_complexo, nome, email, telefone, senha) VALUES (
3,
2,
'Diogo',
'diogo@gmail.com',
'62999994444',
'senha3');

```

4. DESCRIÇÃO FUNCIONALIDADE

| Nome | Path | Verbo | Ação |
|---------|----------------|--------|-------------------------------------|
| Index | /lixreira | GET | Listar todas as lixeira cadastradas |
| Create | /lixreira | POST | Cria um novo cadastro de lixeira |
| Show | /lixreira/{id} | GET | Mostra dados de uma lixeira |
| Update | /lixreira/{id} | PUT | Atualiza os dados de uma lixeira |
| Destroy | /lixreira/{id} | DELETE | Apaga uma lixeira |

A aplicação consiste em uma API para manutenção e tratamento de resíduos produzidos em complexos/ parques públicos e privados. Para tal, foram implementados os métodos relativos ao CRUD para acessos de informações sobre:

- Endereços, Complexos, Usuários, Adminidtradores e Lixeiras

Já para o desenvolvimento da aplicação foram utilizados:

- Spring Boot
- JPA
- Maven
- Hibernate