**UFF - Universidade Federal Fluminense**

**Relatório do Trabalho Prático – Disciplina “Internet das Coisas”**

**Projeto “Qualidade da Água”**

**Alunos: Myrela França de Siqueira e Adriano Busson**

# **Objetivo**

O objetivo do trabalho é construir uma solução IoT para aplicar os conhecimentos adquiridos nas aulas da disciplina ‘Internet das Coisas” utilizando os principais componentes habilitadores da plataforma FIWARE.

Os requisitos mínimos solicitados para a implementação do trabalho são:

1. Utilizar um gerenciador de contexto (Orion ou FogFlow)
2. Utilizar um agente IoT com aquisição de dados (podendo ser dados simulados),
3. Possuir um processamento e uma saída podendo ser alarme, cálculos, atuação etc.

# **Visão do Projeto**

O acesso à água potável é essencial para a saúde, um direito humano básico e um componente de uma política eficaz de proteção da saúde e desenvolvimento a nível nacional, regional e local.

O Brasil ainda apresenta números preocupantes relacionados ao saneamento básico. Milhões de pessoas vivem sem água tratada e sem acesso à coleta de esgoto resultando em doenças que poderiam ser evitadas e que podem levar à morte por contaminação.

Em algumas regiões, foi demonstrado que os investimentos no abastecimento de água e no saneamento podem produzir um benefício econômico líquido, uma vez que as reduções nos efeitos adversos para a saúde e nos custos dos cuidados de saúde superam os custos da realização das intervenções.

A proposta do nosso projeto é desenvolver um sistema IoT para auxiliar no monitoramento da qualidade da água dentro de uma Estação de Tratamento de Água (ETA), ou em locais onde a qualidade das águas é fundamental para um determinado uso (especialmente para consumo humano) ou em locais críticos associados ao uso da água.

Os locais de monitoramento serão equipados de sensores para envio dos dados às equipes de operação facilitando, assim a análise dos dados. Através da coleta de amostras da água, o controle e monitoramento seguirá o método de vigilância permitindo identificar alterações específicas, exigindo ou não a tomada de providências.

O sistema, através de sensores realizará a medição de diversos parâmetros e classificará em tempo real a qualidade da água como potável (própria para o consumo) ou não potável (imprópria para o consumo).

Parâmetros de medição:

1. Valor de pH:

O pH é um parâmetro importante na avaliação do equilíbrio ácido-base da água. É também o indicador da condição ácida ou alcalina do estado da água. A OMS recomendou o limite máximo permitido de pH de 6,5 a 8,5.

1. Dureza:

A dureza é causada principalmente por sais de cálcio e magnésio. Esses sais são dissolvidos em depósitos geológicos através dos quais a água viaja. O período de tempo que a água fica em contato com o material produtor de dureza ajuda a determinar quanta dureza existe na água bruta. A dureza foi originalmente definida como a capacidade da água de precipitar sabão causada pelo Cálcio e pelo Magnésio.

1. Sólidos (Total de Sólidos Dissolvidos - TDS):

A água tem a capacidade de dissolver uma ampla gama de minerais ou sais inorgânicos e alguns orgânicos, como potássio, cálcio, sódio, bicarbonatos, cloretos, magnésio, sulfatos, etc. Esses minerais produziam sabor indesejado e cor diluída na aparência da água. Este é o parâmetro importante para o uso da água. A água com alto valor de TDS indica que a água é altamente mineralizada. O limite desejável para TDS é 500 mg/le o limite máximo é 1000 mg/l, prescrito para beber.

1. Cloraminas:

O cloro e a cloramina são os principais desinfetantes utilizados nos sistemas públicos de água. As cloraminas são mais comumente formadas quando a amônia é adicionada ao cloro para tratar a água potável. Níveis de cloro de até 4 miligramas por litro (mg/L ou 4 partes por milhão (ppm)) são considerados seguros na água potável.

1. Sulfato:

Os sulfatos são substâncias naturais encontradas em minerais, solo e rochas. Eles estão presentes no ar ambiente, nas águas subterrâneas, nas plantas e nos alimentos. O principal uso comercial do sulfato é na indústria química. A concentração de sulfato na água do mar é de cerca de 2.700 miligramas por litro (mg/L). Varia de 3 a 30 mg/L na maioria das fontes de água doce, embora concentrações muito mais altas (1000 mg/L) sejam encontradas em algumas localizações geográficas.

1. Condutividade:

A água pura não é um bom condutor de corrente elétrica, mas sim um bom isolante. O aumento na concentração de íons aumenta a condutividade elétrica da água. Geralmente, a quantidade de sólidos dissolvidos na água determina a condutividade elétrica. A condutividade elétrica (CE), na verdade, mede o processo iônico de uma solução que lhe permite transmitir corrente. De acordo com os padrões da OMS, o valor CE não deve exceder 400 μS/cm.

1. Carbono\_orgânico:

O carbono orgânico total (COT) nas águas de origem provém da matéria orgânica natural em decomposição (MON), bem como de fontes sintéticas. TOC é uma medida da quantidade total de carbono em compostos orgânicos em água pura. De acordo com a EPA dos EUA, < 2 mg/L como TOC em água tratada/potável e < 4 mg/Lit em água de origem usada para tratamento.

1. Trihalometanos:

THMs são produtos químicos que podem ser encontrados na água tratada com cloro. A concentração de THMs na água potável varia de acordo com o nível de matéria orgânica na água, a quantidade de cloro necessária para tratar a água e a temperatura da água que está sendo tratada. Níveis de THM de até 80 ppm são considerados seguros na água potável.

1. Turbidez:

A turbidez da água depende da quantidade de matéria sólida presente no estado suspenso. É uma medida das propriedades de emissão de luz da água e o teste é usado para indicar a qualidade da descarga de resíduos em relação à matéria coloidal. O valor médio de turbidez obtido para Wondo Genet Campus (0,98 NTU) é inferior ao valor recomendado pela OMS de 5,00 NTU

Para elaboração do trabalho usaremos as seguintes faixas de valores de referência para cada um dos parâmetros.

Os valores compreendidos entre cada faixa de referência contribuem para que a classificação da água seja ‘Potável”, ou seja aceitável para o consumo humano.

Se pelo menos um dos parâmetros apresentar valores fora da faixa de referência, a classificação da água passa a ser “não potável” (imprópria para o consumo humano).

|  |  |
| --- | --- |
| ***Parâmetros*** | ***Valores de referência (aceitáveis)*** |
| *Ph* | *de 6 até 8,5* |
| *Dureza* | *ppm de 47 até 324* |
| *Sólidos* | *de 500 até 1000 mg/l* |
| *Cloraminas* | *< 4 mg/l* |
| *Sulfato* | *de 3 até 30 mg/l* |
| *Condutividade* | *< 400 μS/cm* |
| *Carbono* | *< 2 mg/L* |
| *Trihalometanos* | *< 80 ppm* |
| *Turbidez* | *< 5,00 NTU* |

*Obs: Valores retirados do dataset “Water Quality -*  *Drinking water potability” do Kaggle, na qual o projeto de Machine Learning foi baseado. Estes valores não representam valores científicos aceitáveis, são apenas para efeito didático.*

[*https://www.kaggle.com/datasets/adityakadiwal/water-potability*](https://www.kaggle.com/datasets/adityakadiwal/water-potability)

# **Plataformas, tecnologias e ferramentas utilizadas no desenvolvimento**

1. **FIWARE**

FIWARE é uma plataforma de código aberto criada para facilitar o desenvolvimento de soluções inteligentes especialmente na área de IoT. A plataforma fornece vários componentes e ferramentas que permitem a criação de soluções que podem coletar, processar e gerenciar dados em tempo real de dispositivos diversos.

Para elaboração desse trabalho utilizamos os seguintes componentes habilitadores do Fiware e ferramentas de terceiros:

* **Orion Context Broker** – componente habilitador para gerenciamento de contexto que implementa um servidor NGSIv2. Segue o padrão publish/subscriber e gerencia o ciclo de vida das informações de contexto contemplando registros, atualizações, consultas e assinaturas.
  + **MongoDB** – banco de dados de código aberto utilizado pelo Orion Context Broker.
* **IoT Agent for Json** – componente que atua como um gateway de comunicação dos dispositivos que utilizam protocolo json com o Orion Context Broker (protocolo NGSI).
* **Cygnus** – conector que recebe os dados de contexto registrados no Orion Context Broker e persiste em um banco de dados de terceiros (third-party data base) com uma visão de histórico dos dados.
  + MySQL – banco de dados relacional de terceiros que usaremos para persistir os dados de contexto gerenciados pelo Cygnus.
* **Grafana** – é um software Open Source de terceiros (third-party) para visualização que auxilia a análise e monitoramento de dados por meio de gráficos. Tem suporte para vários tipos de banco de dados. Usaremos o Grafana para exibição dos dados históricos de contexto registrados no MySQL.

O Grafana pode ser acessado utilizando qualquer browser instalado no computador local.

* **Dummy-device** – programa em Python para simular a geração de dados do dispositivo e simulação de comandos.

1. **Docker**

Para desenvolvimento do trabalho, seguimos um dos modelos de montagem de ambiente sugerido nos tutoriais do Fiware: utilizamos a plataforma Docker rodando em cima do WSL 2.0 (Windows Subsystem for Linux 2) com a distribuição Ubuntu do Linux.

Docker é uma plataforma para desenvolvimento de soluções baseada em containers. Containers são unidades de software que encapsulam o código e suas dependências.

Dessa forma, organizamos cada componente da nossa solução, mencionados no item *“a. FIWARE”* anterior, em containers isolados se comunicando através de portas definidas em uma rede interna criada dentro do Docker.

As imagens dos componentes de cada container foram carregadas a partir do Docker Hub, que é um serviço disponibilizado pela plataforma Docker para acessar um repositório de imagens para os containers.

1. **Postman**

Utilizamos a ferramenta Postman como ferramenta auxiliar para execução das APIs de checagem dos componentes, criação de entidades, grupos de serviços, dispositivos e assinaturas.

O Postman é uma ferramenta utilizada por desenvolvedores para testar, documentar e colaborar APIs. A ferramenta permite enviar solicitações HTTP para APIs e receber as respostas correspondentes.

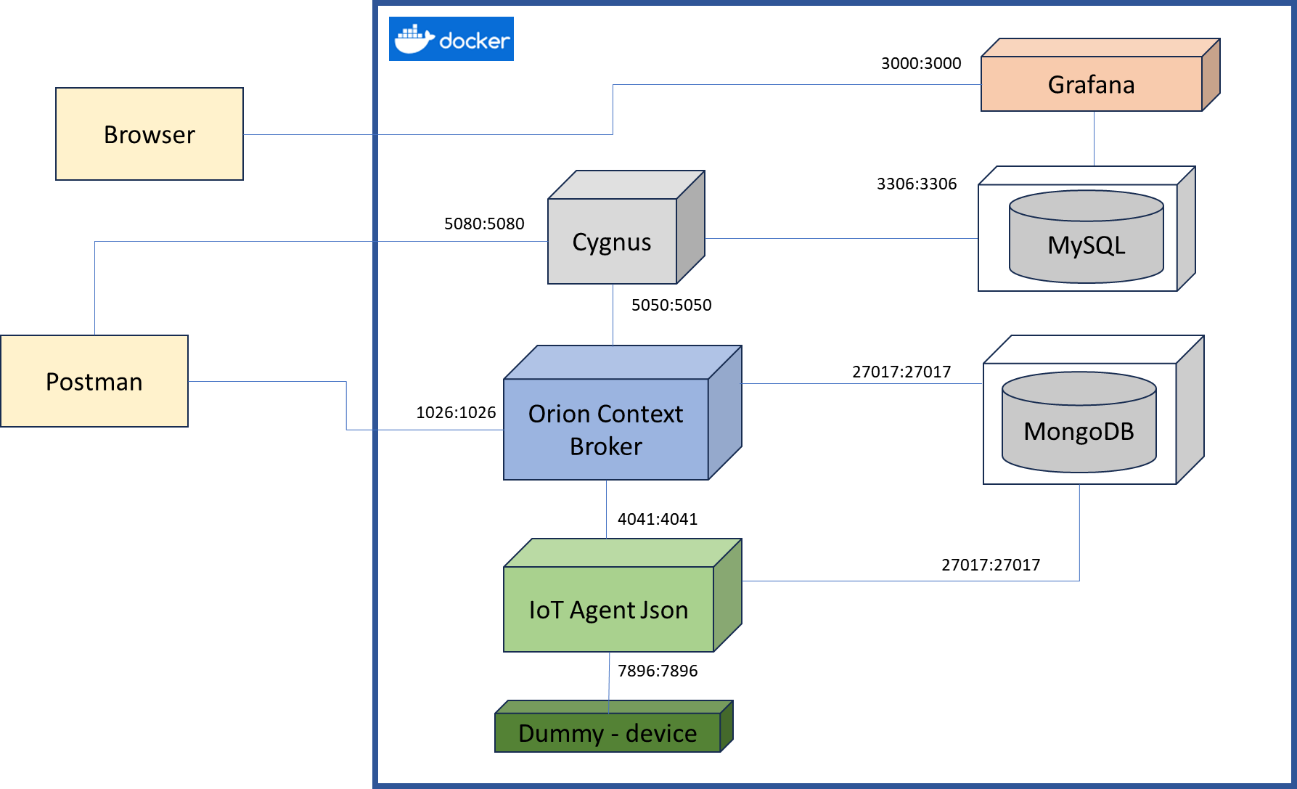
1. **Blokchain**

Utilizamos a ferramenta Hardhat para emular um nó da rede Ethereum, e utilizamos a ferramenta CARTESI para rodar um algoritmo de Machine Learning em Pytho como layer 2.

Os dados são enviados para o Hardhat através de um programa em python que lê os últimos dados gravados no MySQL.

1. **Visão da arquitetura dos componentes em containers**

Segue abaixo figura com a visão de como foram organizados os componentes e ferramentas mencionados nos itens anteriores (*“a. FIWARE”, “b. Docker” e “c. Postman”).*



1. **Visão da arquitetura dos componentes com interligação ao blockchain**

Segue abaixo figura com a visão de como interagem os componentes e ferramentas com o blockchain layer 1 e layer 2.

Diagrama

Descrição gerada automaticamente

# **Implementação**

Seguimos as seguintes etapas para implementar a solução seguindo a arquitetura mostrada acima:

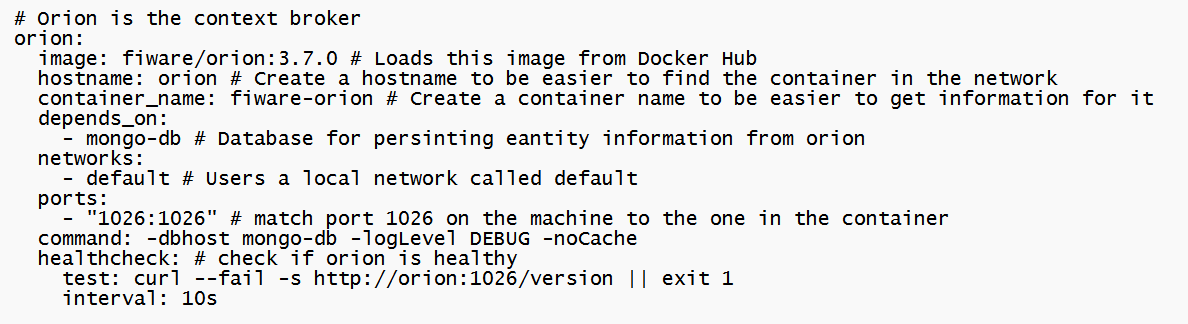
1. **Criação dos containers dentro do Docker - Docker Compose**

O [Docker Compose](https://docs.docker.com/compose/) é uma ferramenta usada para definir e executar aplicativos de vários containers do Docker a partir de um arquivo YAML usado para configurar os serviços executando um só comando. Além disso, o Docker Compose contém comandos para iniciar, baixar e visualizar o status dos serviços.

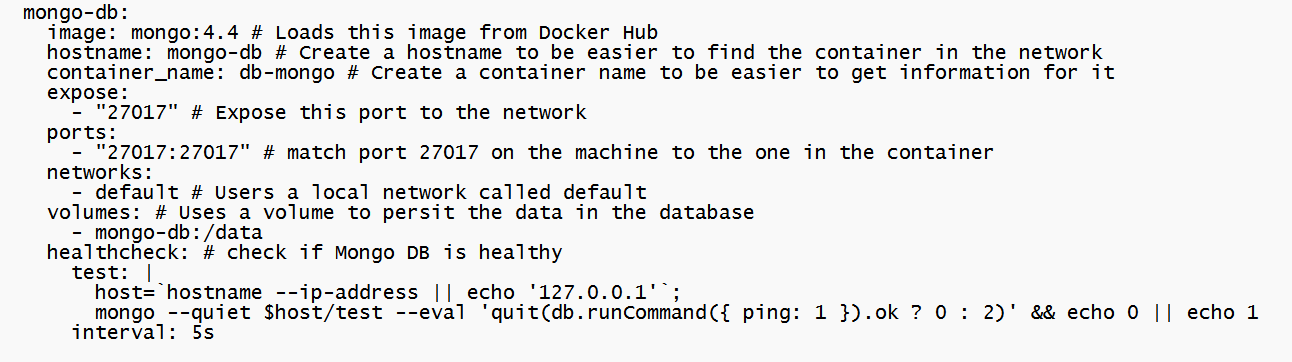
Utilizamos o Docker Compose para criar a estrutura dos containers, redes e volumes das bases de dados dentro do Docker.

Abaixo seguem os blocos com as definições de cada um dos containers configurados dentro do arquivo docker-compose.yml utilizado no projeto.

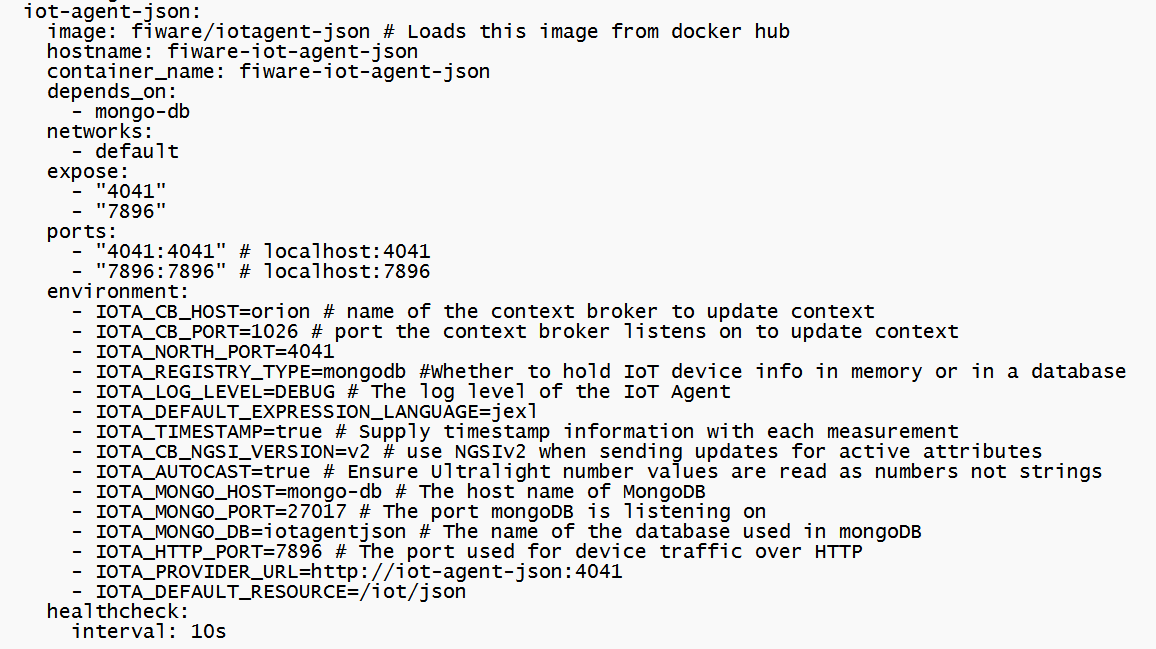
Configuração do container para o Orion Context Broker:



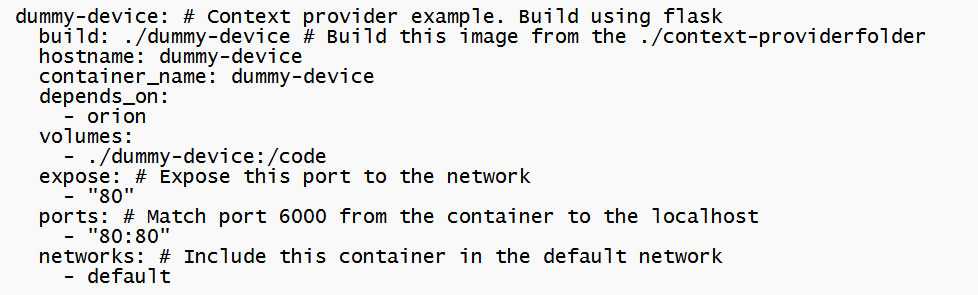
Configuração do container para o MongoDB:



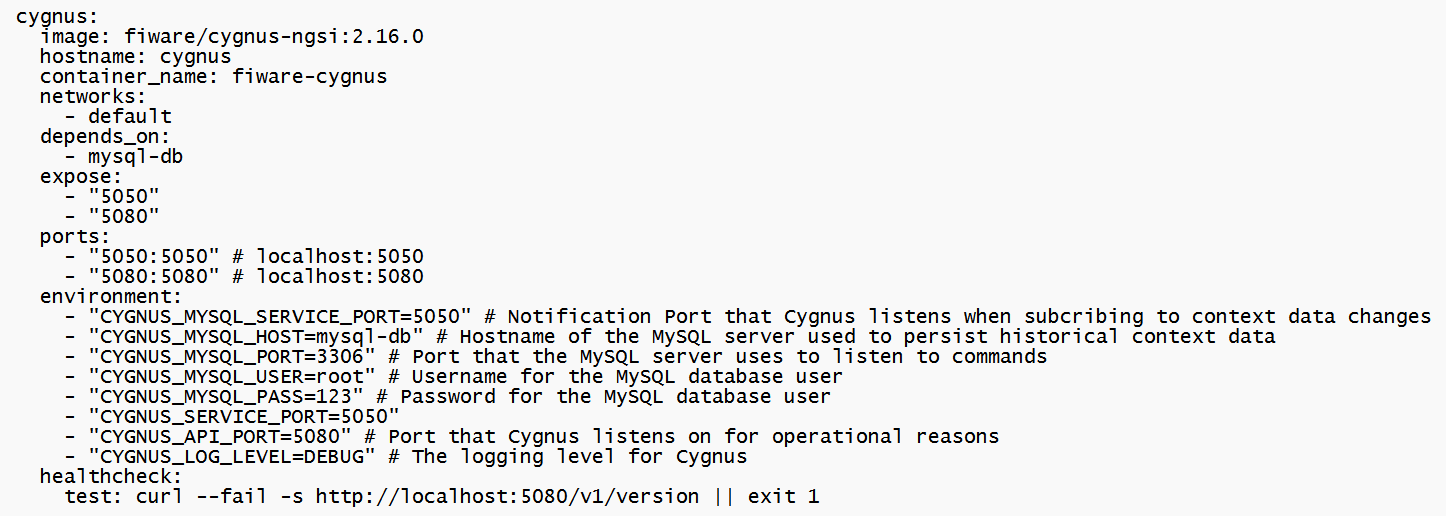
Configuração do container para o IoT Agent - Json:



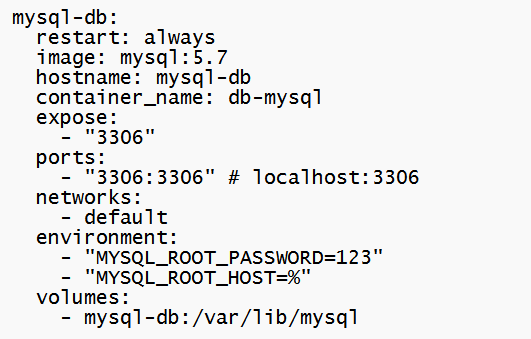
Configuração para o container do dummy-device:



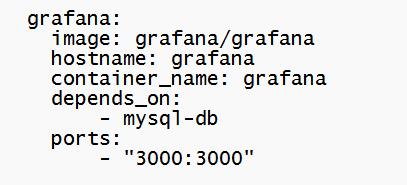
Configuração para o container do Cygnus:



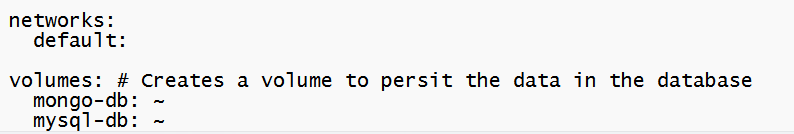
Configuração para o container do Mysql-db:

****

Configuração para o container do Grafana:



Configuração da Rede e Volumes para cada um dos bancos de dados:



Procedimento para iniciar os containers usando o comando único do Docker Compose e toda a estrutura do arquivo docker-compose.yml:

1. Entrar no diretório do trabalho onde o arquivo docker-compose.yml está copiado e executar o comando (no prompt da linha de comando):

***docker compose up -d***

Texto

Descrição gerada automaticamente

1. Verificar se os containers foram inicializados e executar o comando:

***docker compose ps -a***

Texto

Descrição gerada automaticamente

1. **Registros das entidades, dispositivos, serviços e assinaturas**

Usando o Postman, realizamos as chamadas das APIs para criação das entidades, serviços, dispositivos e assinaturas.

* Criação da entidade ETA (Estação de Tratamento de Água)

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

* Criação de um serviço

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

* Providenciar um dispositivo

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

* Início do envio dos dados – dummy device

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

* Obter o dispositivo já criado

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Os dados dos parâmetros mostrados no resultado acima ao obter o disposto são gerados aleatoriamente através do programa app.py

* Registro de uma assinatura do Cygnus

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

1. **Consulta dos registros dos 9 parâmetros da qualidade da água no MySQL**

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente com confiança baixa

1. **Criação dos containers blockchain**

Container CARTESI que roda o Algoritmo de Machine Learning do programa water.py

Texto

Descrição gerada automaticamente

1. Código Python do programa water.py que roda o algoritmo ML dentro da CARTESI:

<https://github.com/AdrianoBusson/Projeto-Water/blob/main/water.py>

1. Container do front-end que envia comandos para a blockchain

<https://github.com/cartesi/rollups-examples/tree/80be07fb68ba7048223c72f3f9fe1b2042ec9ae7/frontend-console>

1. Código Python do programa interface.py entre MYSQL e o blockchain – processa os dados do MYSQL e monta o comando que é enviado ao front-end.

<https://github.com/AdrianoBusson/Projeto-Water/blob/main/interface.py>

1. Tela do computador rodando todos os containers:

1 - Ferramenta HardHat – Emulador de blockchain Layer 1 – Ethereum

2 - Máquina CARTESI – Blockchain Layer 2

3 - Ferramenta Front-end – Envio de comandos para Blockchain Layer 1

4 - Containers FIWARE

5 – Programa Python interface.py – Leitura de dados MYSQL e envio

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

1. **Grafana**

A interface do Grafana pode ser acessada pelo link <http://localhost:3000>

Dashboard montado para o monitoramento:

Interface gráfica do usuário

Descrição gerada automaticamente

Segue abaixo o código SQL utilizado no GRAFANA para trazer os valores do MYSQL:

Cada display do dashboard possui este códio variando o @@@-Parâmetro-@@@ de acordo com o parâmetro que se deseja obter.

SELECT UNIX\_TIMESTAMP(recvTime) as "time",

CAST(attrValue as decimal) as "value"

FROM `urn\_ngsi-ld\_Device\_001\_Device`

WHERE attrName = "@@@-Parâmetro-@@"

ORDER BY "time"

# **Cenários de testes**

Seguem abaixo exemplos de dados recolhidos pelo programa interface.py, que traz os últimos 9 parâmetros, faz a montagem e roda um comando yarn no diretório da ferramenta front-end, que executa o envio para a blockchain.

* Caso de teste exibindo água portável

yarn start input send --payload "7.68878591609137,209.18562534531952,29568.535852018093,9.817335635498171,294.34773223445876,355.6986132389772,13.918768617655758,71.68153964307832,3.9186731557714785"

Resultado do cálculo: 1

* Caso de teste exibindo água não potável

yarn start input send --payload "8.316765884214679,214.37339408562252,22018.417440775294,8.05933237743854,356.88613564305666,363.2665161642437,18.436524495493302,100.34167436508008,4.628770536837084"

Resultado do cálculo: 0

# **Conclusão**

A elaboração desse trabalho prático foi desafiadora. Enfrentamos algumas dificuldades na instalação das ferramentas Docker e WSL 2.0 para rodarem na máquina local, devido a algumas incompatibilidades da versão do sistema operacional e alguns outros requisitos de hardware.

Outro desafio foi concluir todos de tutoriais necessários para termos segurança e avançarmos na elaboração do trabalho. Mas ao final conseguimos mostrar uma aplicação IoT utilizando alguns dos principais componentes do FIWARE.

O nosso trabalho mostrou a implementação de uma ideia ainda muito inicial diante do que imaginamos ser necessário para um monitoramento completo, robusto e confiável da qualidade da água. Mas através da experiência desse trabalho, conseguimos perceber que a Internet das Coisas pode contribuir muito com soluções inteligentes e inovadoras para melhorar os indicadores do saneamento básico no nosso país.