Relatório Técnico

1. Plano de Trabalho

- O plano de trabalho foi iniciado seguindo o cronograma. Houveram alguns desvios devido a algumas atividades pessoais e profissionais que se fizeram necessárias durante as etapas de desenvolvimento. Devido a estes contratempos, foi necessária a extensão do prazo por mais dois dias, porém sem comprometer o desenvolvimento e entregas da solução.

2. Tecnologias Utilizadas

1. Java 8

2. Spring Boot

3. Spring Cloud Gateway

4. Postgresql

5. Junit 5

6. Mockito

7. Docker Desktop 2.3.0.5 (48029) - Ambiente

8. Swagger 2.9.2 - Documentação de API

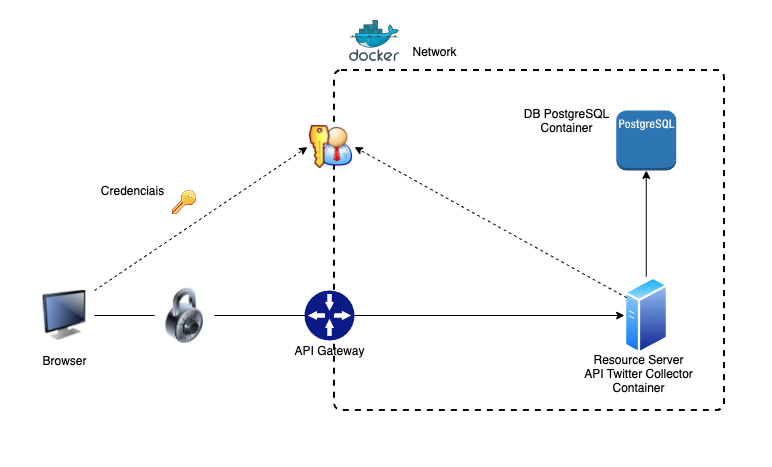
9. IntelliJ IDEA 2020.1.4 Community Edition - Desenvolvimento das API's

10. Draw.IO (https://app.diagrams.net) - Confecção dos diagramas

3. Requisitos

1. Possuir o Docker instalado na máquina
2. Possuir o Postman ou algum aplicativo para realizar requisições HTTP

3. Diagrama da Arquitetura



4. Diagrama da Implantação da Solução

1 – Baixar o Código do repositório github

2 – Executar o commando docker-compose up -d para instalar as imagens e subir as aplicações

7. Repositórios

* <https://github.com/gustavopomponi/twitterposts>

6. Referências

1 - Spring Cloud Gateway

\* https://spring.io/blog/2019/08/16/securing-services-with-spring-cloud-gateway

\* https://spring.io/blog/2019/07/01/hiding-services-runtime-discovery-with-spring-cloud-gateway

\* https://spring.io/blog/2019/06/18/getting-started-with-spring-cloud-gateway

2 - Spring Social Twitter

\* https://medium.com/@volmar.oliveira.jr/spring-social-twitter-adding-integration-in-a-spring-boot-project-dd4a66c57764

3 - Swagger

\* <https://medium.com/@volmar.oliveira.jr/swagger-versioning-springboot-restapi-1f6ceb3d58c>