# Relatório Técnico

**Teste para Engenheiro de Software - Banco Itaú, 2Q/2016.**

**Adriano Rodrigo Guerreiro Laranjeira**

**adriano@argl.eng.br**

**Destinatários:**

**Fabiola Marchiori, Noemi Ushima, Wilson Neto.**

## Plano de trabalho

Plano entregue em 04/08/2016. Segue abaixo a análise sobre a lista de atividades:

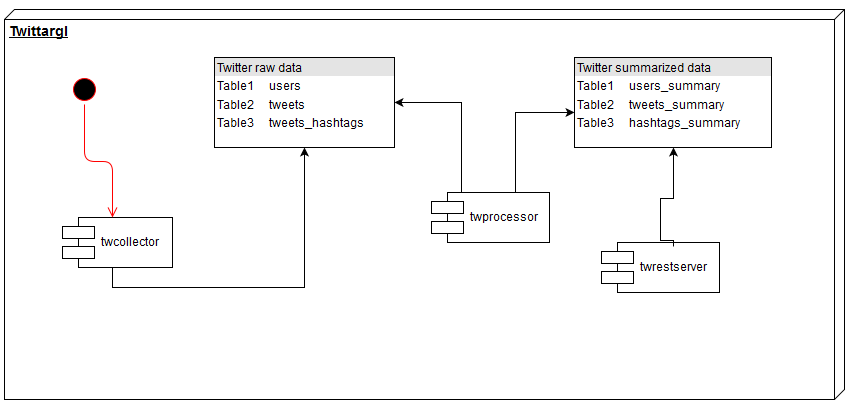
|  |  |
| --- | --- |
| Atividade Planejada | Atividade Executada |
| Elaborar um plano de trabalho para o teste. | Entregue conforme combinado em 04/08/2016. |
| Preparar um ambiente (VM ou Docker) com a seguinte lista: Apache Spark, Cassandra, Java8, MySQL, PHP, Python, Springboot, VertX. | Na lista planejada faltou o Node.JS, adicionado durante o trabalho. |
| Realizar coletas de teste com Python gravando os dados no Cassandra. Os parâmetros de OAuth do Twitter devem ser passados por parâmetro. | Executado conforme planejado. |
| Criar uma primeira modelagem em Cassandra para atender aos requisitos do teste. | Executado conforme planejado. |
| Gerar um script no Crontab que alimente continuamente a base Cassandra com Tweets. O script deve armazenar os dados de OAuth do Twitter. | Um script Python é executado pelo crontab a cada 5 minutos. Os parâmetros de OAuth estão fixos no script Python. |
| Criar uma app Java (VertX? Spring? POJO?) que possa ser passada ao Spark para gerar a visão sumarizada dos dados (relacional? Cassandra?); Pode ser necessário refinar a base de dados Cassandra e o desenvolvimento. | A app foi escrita em Scala. Os dados são sumarizados e gravados no próprio Cassandra, por isso a base MySQL não é necessária. |
| Criar uma API REST (PHP) para consumir os dados sumarizados. | O teste recomenda a utilização do Node.JS. |
| Criar um script bash que realize as chamadas via cURL, chamando as URLs e mostrando o resultado (STDOUT? Arquivo?). | Executado conforme planejado. |
| Adequar todo o trabalho dentro de um container Docker. | Não finalizado |
| Realizar testes de execução da solução já no container a ser entregue; Provável necessidade de adequações à solução. | Esta atividade não foi executada. |
| Elaborar o relatório técnico da solução. | Executado conforme planejado. |
| Disponibilizar artefatos no Github e DockerHub. | VM Virtualbox disponível em http://www.argl.eng.br/vmtwittargl.zip. Este link é temporário até que o link no DockerHub seja válido. |

## Tecnologias utilizadas

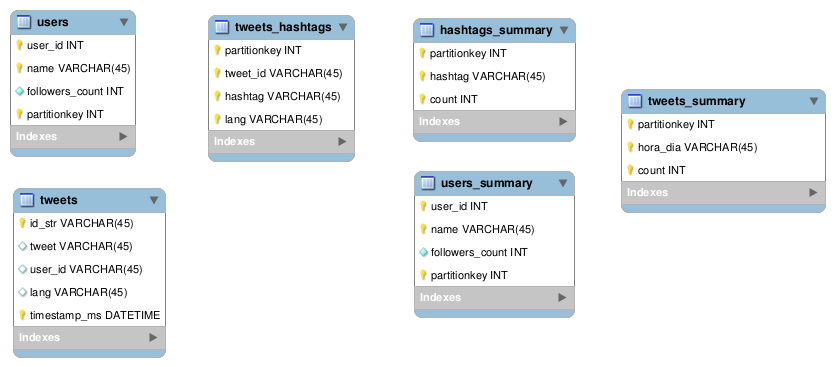
Segue a lista de todas as tecnologias utilizadas:

1. Ambiente virtualizado com sistema operacional GNU/Linux CentOS 7, com SELinux desabilitado, atualizado até a data de 07/08/2016;
2. Repositórios EPEL e DataStax adicionados ao SO;
3. Python 2.7, com os módulos *Tweepy* e *Cassandra*;
4. Java JDK versão 1.8u102 instalado no ambiente virtual;
5. Java JDK versão 1.7.0\_79 instalado no host de desenvolvimento;
6. Scala versão 2.10.1;
7. Apache Spark versão 2.0.0;
8. Cassandra versão 2.2;
9. API pública do Twitter (api.twitter.com);
10. IDE IntelliJ para construir o script Scala;
11. Shell script para automatizar a execução de tarefas rotineiras;
12. Crontab do sistema operacional para agendar a execução de tarefas.

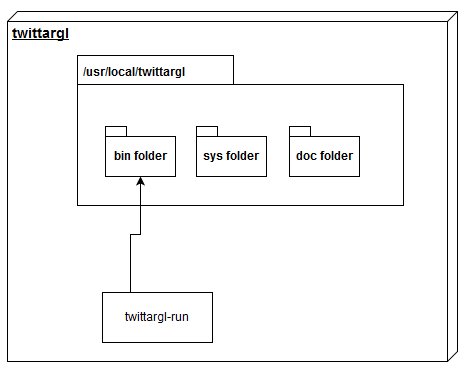
## Diagrama de arquitetura



## Modelagem da base de dados



## Diagrama de implantação da solução



## Referências GitHub & hub.docker.com

|  |  |
| --- | --- |
| **Github** | **DockerHub** |
| Url: http://github.com/arglbr/twittargl | https://hub.docker.com/r/arglbr/twittargl/ |

## Referências utilizadas

* http://www.xwebhosting.org/halo/knowledgebase.php?action=displayarticle&id=67
* https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/
* http://www.planetcassandra.org/blog/installing-the-cassandra-spark-oss-stack/
* https://github.com/datastax/spark-cassandra-connector
* https://spark-packages.org/package/datastax/spark-cassandra-connector
* https://chongyaorobin.wordpress.com/2015/07/16/step-by-step-of-how-to-configure-apache-spark-to-connect-with-cassandra/
* https://github.com/phalodi/spark-cassandra/blob/master/build.sbt
* https://github.com/datastax/spark-cassandra-connector/blob/master/doc/2\_loading.md
* http://homepage.cs.latrobe.edu.au/zhe/ZhenHeSparkRDDAPIExamples.html#sortBy

## Demais itens

* A versão do IntelliJ utilizada apresenta bugs para configurar a IDE de forma a remover as assinaturas contidas dentro um JAR, por isso o código Scala depende de um workaround para funcionar: deve-se copiar executar o script “remove\_signatures.sh” no mesmo diretório onde se encontra o arquivo sparkcassandra2.jar, antes de realizar o deploy desta aplicação.

## Informações DockerHub

* URL: https://hub.docker.com/r/arglbr/twittargl/