**FIAP**

**MBA EM DESENVOLVIMENTO DE APLICAÇÕES JAVA – SOA E INTERNET DAS COISAS**

**Twitter Small Analytics**

**#java8**

**FUNDAMENTOS DA TECNOLOGIA JAVA E UML**

**FIAP**

**MBA EM DESENVOLVIMENTO DE APLICAÇÕES JAVA – SOA E INTERNET DAS COISAS**

**Twitter Small Analytics**

**#java8**

HELENA STRADA FRANCO DE SOUZA – RM31321

JULIO OLIVEIRA DA SILVA – RM31524

VINICIUS RIGO – RM31667

**FUNDAMENTOS DA TECNOLOGIA JAVA E UML**

Sumário

[Considerações Iniciais 4](#_Toc469042074)

[Manual De Utilização 5](#_Toc469042075)

[Importar Projeto no Eclipse 5](#_Toc469042076)

[Configurar Chaves de Acesso 5](#_Toc469042077)

[Componentes, Bibliotecas e Frameworks 7](#_Toc469042078)

[Pacotes, classes e métodos 8](#_Toc469042079)

[Diagrama de classes 11](#_Toc469042080)

[Diagrama de sequência 12](#_Toc469042081)

[Capturas de tela e aplicação 18](#_Toc469042082)

[GitHub 18](#_Toc469042083)

# Considerações Iniciais

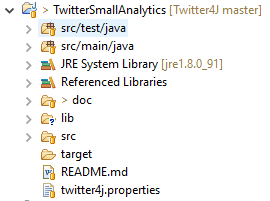
O Twitter4J é uma biblioteca não oficial que permite realizar chamadas para a API do Twitter através da linguagem Java, podendo realizar a contagem e o monitoramento de tweets em tempo real.

# Manual De Utilização

## Importar Projeto no Eclipse

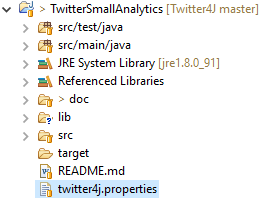
Após realizar o download do projeto disponível no GitHub, e descompactar o zip em sua pasta de preferência, realize os passos a seguir para importar o projeto no eclipse.

1. Abrir o eclipse;
2. Clicar em File > Import;
3. Escolha General > Projects from Folder or Archive:
   1. Clique em ‘Next’;
4. Clique em ‘Directory’:
   1. Selecione a pasta desejada;
5. Clique em Finish.



## Configurar Chaves de Acesso

Para utilizar o Twitter Small Analytics é necessário configurar as chaves de acesso no arquivo twitter4j.properties.



debug=false

oauth.consumerKey=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

oauth.consumerSecret=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

oauth.accessToken=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

oauth.accessTokenSecret=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Com a chave e o token de acesso configurados é necessário criar uma instância da classe *TwitterSmallAnalytics.java* passando como parâmetro a hashtag que desejar, para que a contagem dos tweets ocorra. Após isso deve-se escolher uma estratégia de contagem de dados dos tweets definidos na classe enum *Action.java*.

# Componentes, Bibliotecas e Frameworks

O sistema foi desenvolvido utilizando uma integração com a API Twitter4J.

Como dissemos, o Twitter4J permite chamadas à API do Twitter por meio da linguagem Java. Com o Twitter4J, você pode facilmente integrar seu aplicativo Java com os serviços do Twitter. O Twitter4J é uma biblioteca não-oficial.

Características:

* Funciona em qualquer plataforma Java na versão 5 ou posterior;
* Plataformas Android e Google App Engine incluídas;
* Dependência zero: não requer jar’s adicionais;
* Suporte a OAuth integrado;
* Suporte a gzip;
* Compatível com a versão 1.1 do Twitter API.

Essa API viabiliza a integração com o Twitter e assim explorar suas diversas funcionalidades objetivando a coleta, busca e análise de dados, como também formas de interação automatizadas bastante eficazes.

Essa API é composta por diversas bibliotecas nas quais utilizamos:

* twitter4j-async-4.0.4.jar;
* twitter4j-core-4.0.4.jar;
* twitter4j-media-support-4.0.4.jar;
* twitter4j-stream-4.0.4.jar.

Com a API Twitter4J é possível realizar qualquer tipo de operação que poderia ser efetuada manualmente pelo usuário no site do Twitter, dentre elas: criar *tweets*, buscar por assuntos com critérios específicos, enviar e visualizar mensagens diretas, entre outras funcionalidades.

Estas funcionalidades permitirão automatizar diversas operações e tornar mais ágil a comunicação com os usuários do Twitter. Por exemplo, é possível desenvolver um sistema que identifique mensagens de reclamação, por meio da busca de temas específicos, e que automaticamente responda estas mensagens solicitando mais informações ou fornecendo algum tipo de feedback. Também é possível que seja estabelecido uma interface com os sistemas próprios da empresa para facilitar e tornar ainda mais ágil este tipo de interação.

# Pacotes, classes e métodos

Além da estrutura explicada abaixo, o código fonte também está constituído de comentários.

A disposição dos itens a seguir, possuem as seguintes características:

**Caminho/do/pacote**

Definição do pacote

NomeDaClasse.java

Definição da classe

*Métodos*

Definição dos métodos

**src/test/java/scj28/java/atividadefinal**

Pacote que contém a classe principal para acesso à API do twitter.

TwitterSmallAnalyticsTestDrive.java

Classe de teste que executa os principais métodos de sistema para acesso a API do twitter.

**src/main/java/scj28/java/atividadefinal**

Pacote que contém todas as classes da aplicação

TwitterSmallAnalytics.java

Classe principal do sistema responsável por executar as consultas à base do Twitter.

*TwitterSmallAnalytics*

O construtor é responsável por definir qual hashtag a busca irá se basear.

*executeAction(Action action)*

Executa a ação passada como parâmetro

*executeAction(Action action, TipoDeOrdenacao type)*

Executa a ação de forma ordenada de acordo os parâmetros passados.

**src/main/java/scj28/java/atividadefinal/enumeration**

Pacote que contém as classes Enum.

Action.java

Classe enum que define os tipos de ordenação disponíveis no sistema.

TipoDeOrdenacao.java

Classe enum que define os tipos de ordenação disponíveis do sistema (Por autor ou data do tweet).

**src/main/java/scj28/java/atividadefinal/strategy**

Pacote onde ficam definidas as estratégias que mudam o comportamento do sistema em tempo de execução

Actionable.java

Interface que define a ação que será executada utilizando o padrão de projetos strategy.

Favorite.java

Ação que busca a quantidade de favoritações por dia na última semana.

*buscaTweets*

Realiza a busca das favoritações e retorna a quantidade.

Organizer.java

Ação que busca e imprime os tweets ordenados pelo nome do autor.

*imprimeDatas*

Imprime a data mais antiga e a mais recente da lista de tweets.

*imprimeNomes*

Imprime o primeiro e o último nome da lista de tweets.

*buscaTweets*

Realiza a busca dos tweets e retorna uma lista.

Retweet.java

Ação que busca a quantidade de retweets da última semana.

*buscaTweets*

Realiza a busca dos retweets e retorna a quantidade.

Tweet.java

Ação que busca a quantidade de tweets da última semana.

*buscaTweets*

Realiza a busca dos tweets e retorna a quantidade

Tweeter.java

Ação realiza um tweet referenciando o professor Mechel.

*buscaTweets*

Realiza um tweet referenciando o professor Mechel com a data e hora.

**scj28.java.atividadefinal.vo**

Pacote onde são colocados os Vos.

TweetVO.java

POJO criado para tragefar os dados necessários para executar as ações seguindo o padrão Value Object.

# Diagrama de classes

Figura 1 - Diagrama de Classes

# Diagrama de sequência

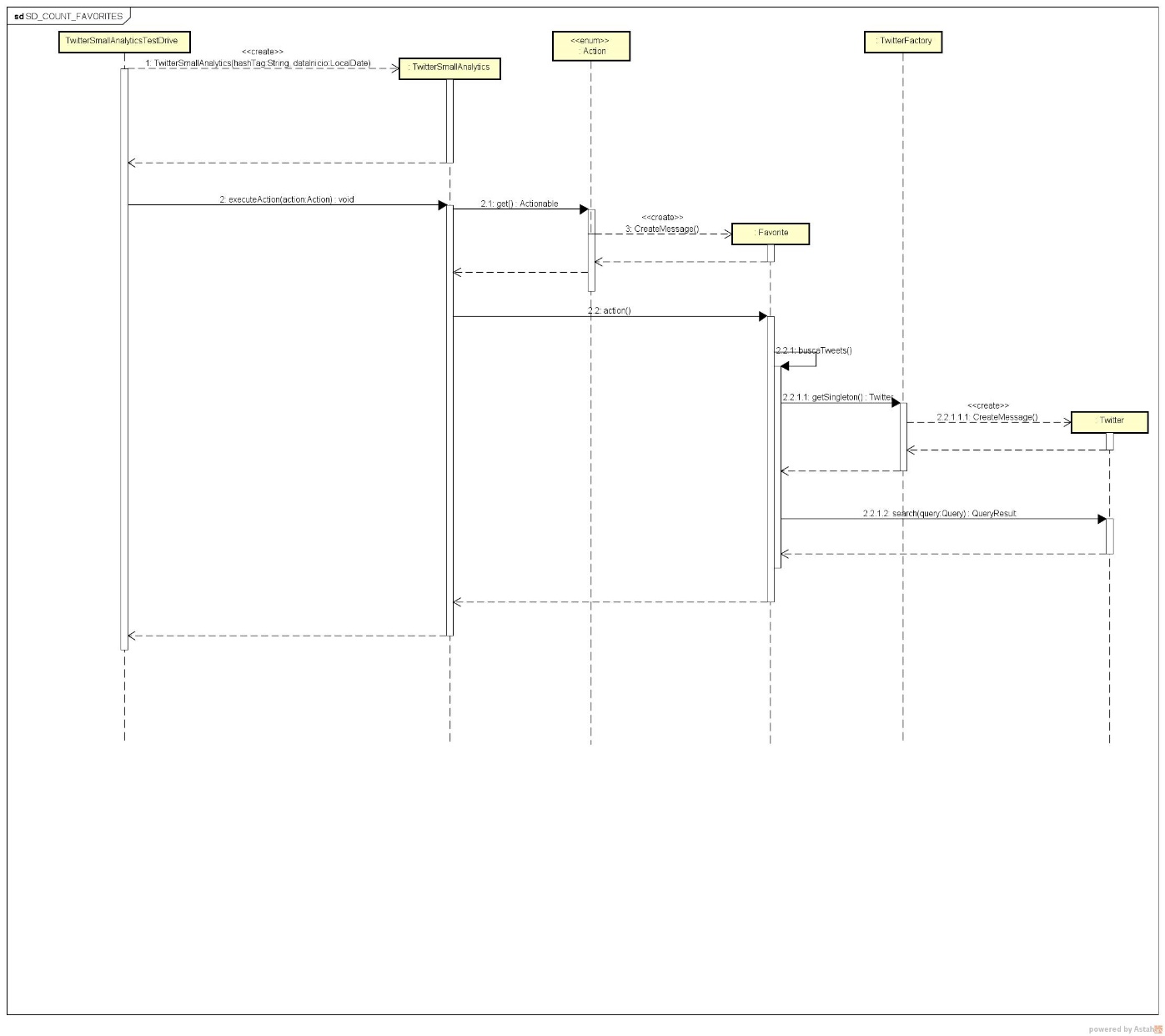


Figura 2 - Diagrama de Sequência - Contar Favoritos

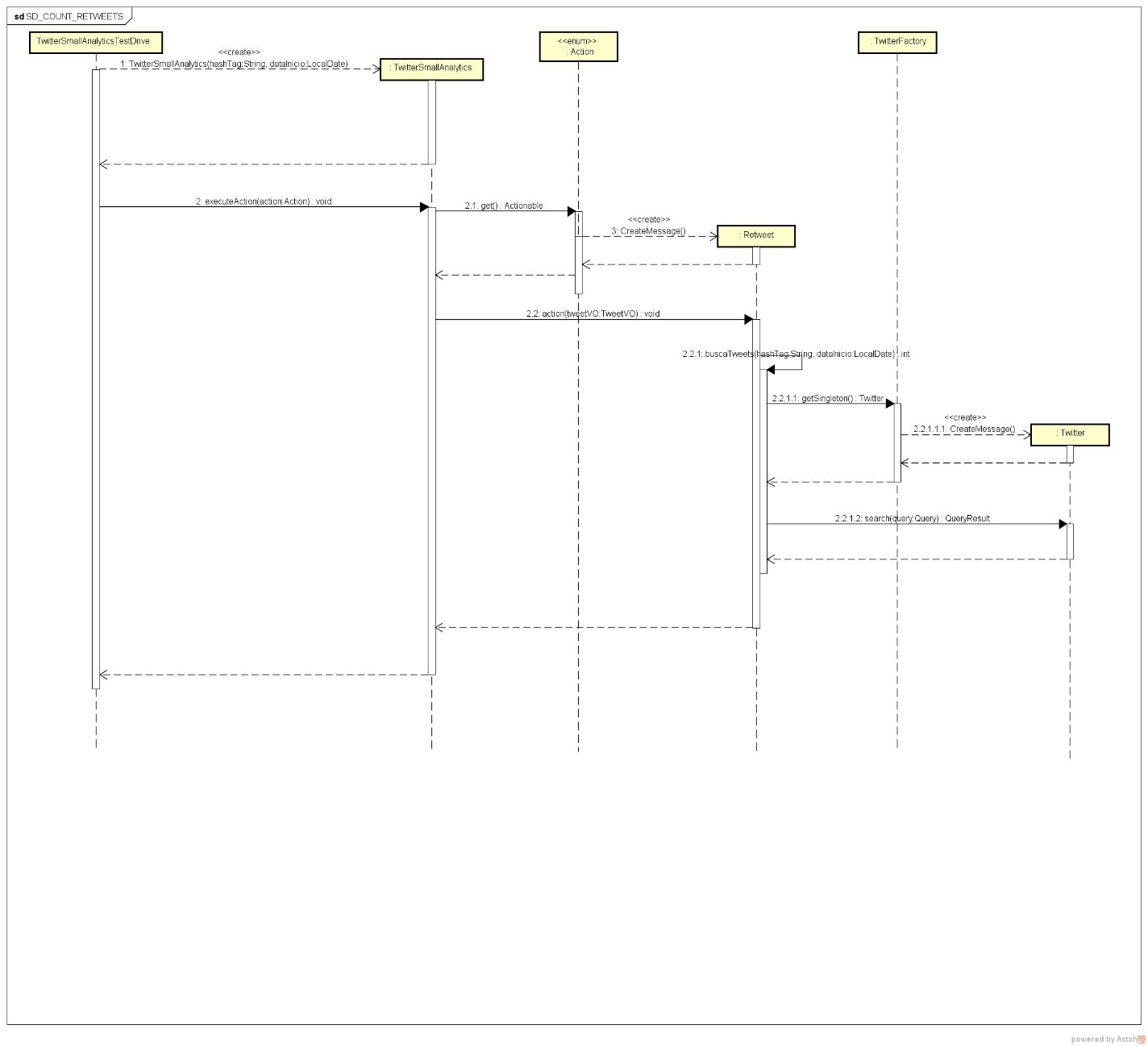


Figura 3 - Diagrama de Sequências - Contar Retweets

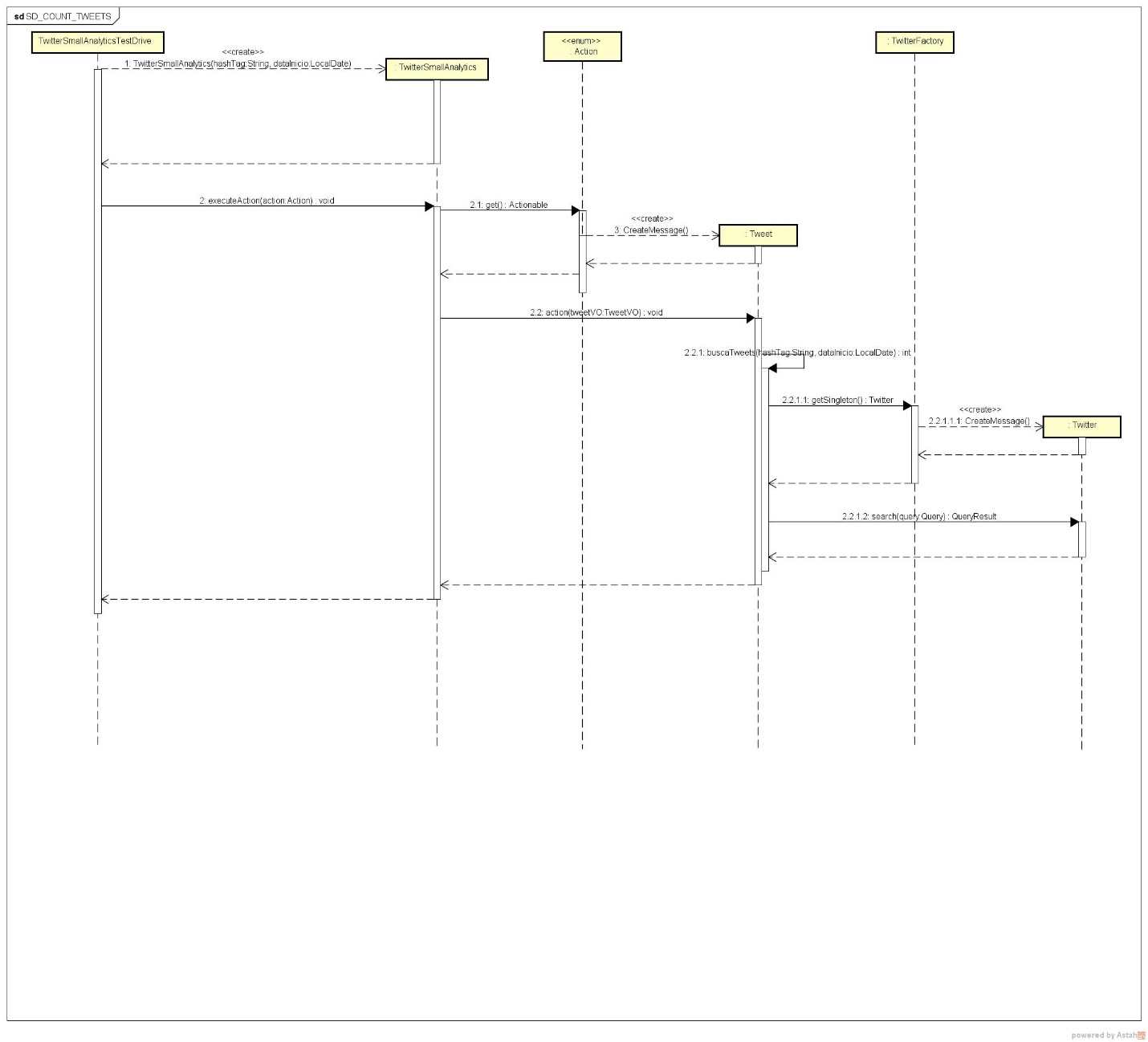


Figura 4 - Diagrama de Sequência - Contar Tweets

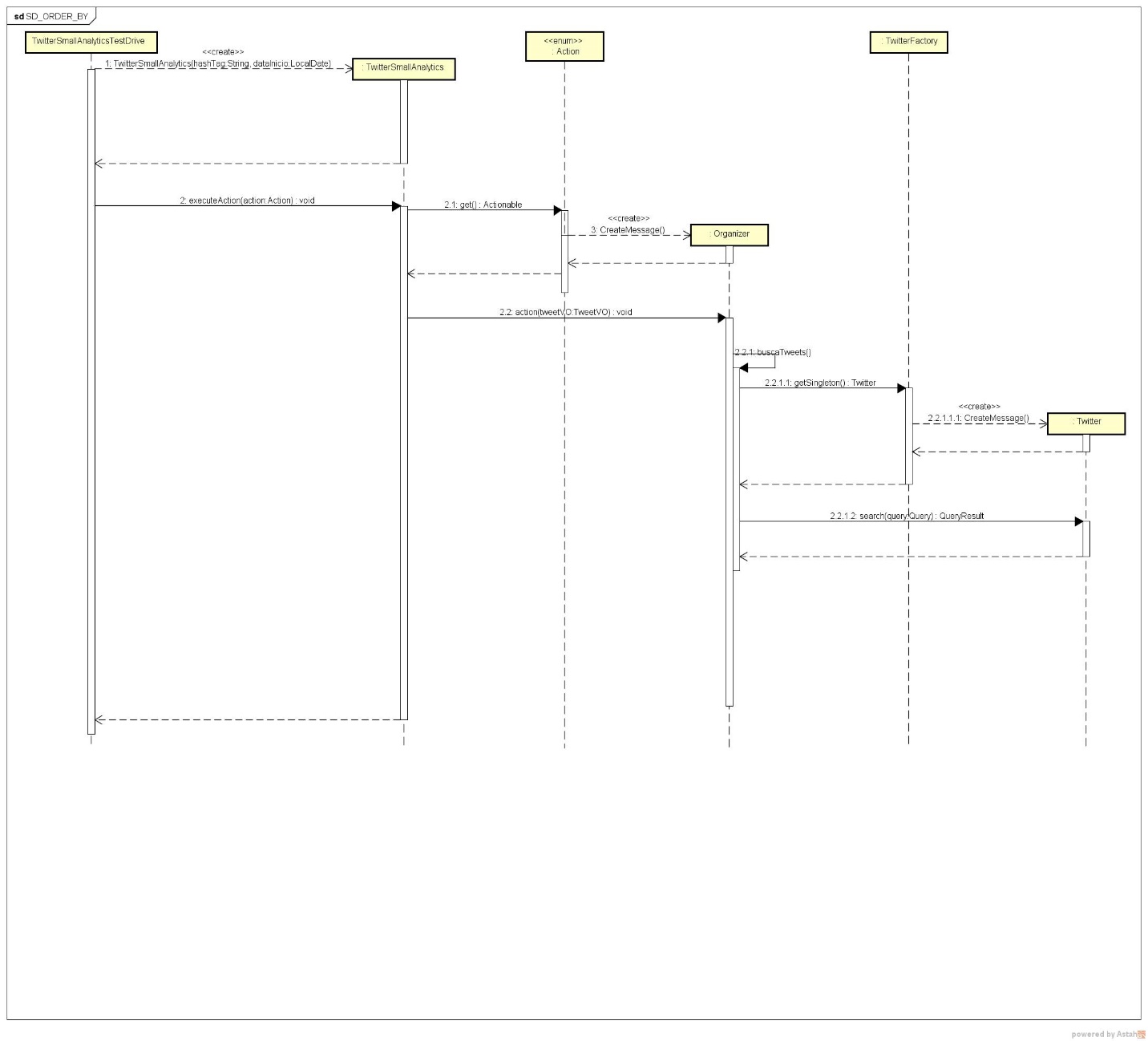


Figura 5 - Diagrama de Sequência - Ordenar

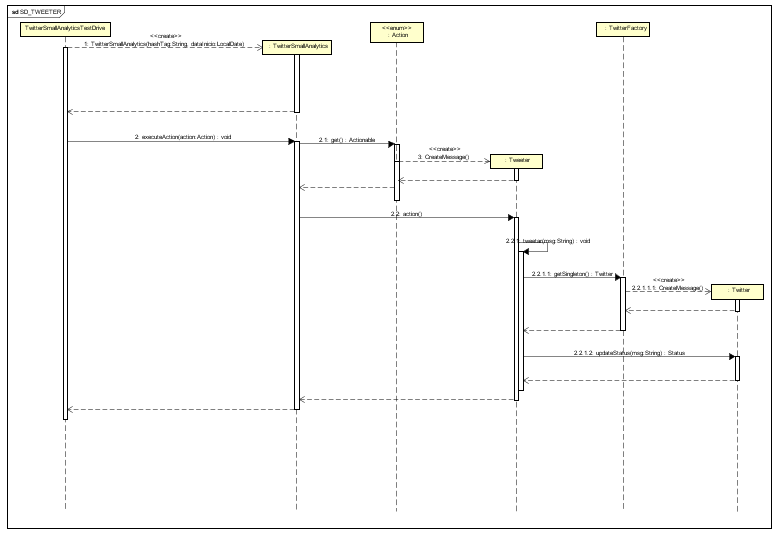


Figura 6 - Diagrama de Sequência - Tweeter

# Capturas de tela e aplicação

Esta versão de saída foi executada no dia 07 de novembro de 2016 com a hashtag **#java8**.

Data Inicial - 2016-12-05 | Data Final - 2016-12-12

Quantidade por dia de tweets da última semana:

2016-12-05 30 tweets.

2016-12-06 15 tweets.

2016-12-07 15 tweets.

2016-12-08 30 tweets.

2016-12-09 15 tweets.

2016-12-10 15 tweets.

2016-12-11 0 tweets.

<===========================================>

Quantidade por dia de retweets da última semana:

2016-12-05 156 retweets.

2016-12-06 12 retweets.

2016-12-07 12 retweets.

2016-12-08 42 retweets.

2016-12-09 5 retweets.

2016-12-10 4 retweets.

2016-12-11 0 retweets.

<===========================================>

Quantidade por dia de favoritações da última semana:

2016-12-05 57 favoritações.

2016-12-06 8 favoritações.

2016-12-07 8 favoritações.

2016-12-08 14 favoritações.

2016-12-09 8 favoritações.

2016-12-10 4 favoritações.

2016-12-11 0 favoritações.

<===========================================>

Primeiro Nome: @ADAMLDAVIS

Ultimo Nome: @ZDAVEP

<===========================================>

Data mais antiga: 04/12/2016 08:30

Data mais recente: 09/12/2016 21:43

<===========================================>

Tweet postado com sucesso! [12/12/2016 18:01:55 - Trabalho final de Fundamentos Java - 28SCJ - @michelpf].

<===========================================>

# GitHub

O código fonte da aplicação está disponível em: <https://github.com/hstrada/Twitter4J>.