

**PROJETO FINAL FUNDAMENTOS DA PLATAFORMA JAVA E UML**

**Turma 32SCJ**

**Luiz Fillipe Ribeiro Navarro RM332867**

**Bruno Lopes RM332786**

**Wagner Bina dos Santos RM332061**

Sumário

[1. Introdução 5](#_Toc535334047)

[2. Objetivo 5](#_Toc535334048)

[3.1 Componentes 6](#_Toc535334056)

[3.2 Bibliotecas 6](#_Toc535334058)

[3.3 Frameworks 6](#_Toc535334060)

[4. Descrição de Funcionabilidades 6](#_Toc535334062)

[3.1. Método CompareTo 8](#_Toc535334084)

[3.2. Método setNomeAutor 8](#_Toc535334086)

[3.3. Método setData 9](#_Toc535334088)

[3.4. Método Mensagem 9](#_Toc535334090)

[5. Diagramas 9](#_Toc535334092)

[5.1 Diagrama de Classe 9](#_Toc535334093)

[5.2 Diagrama de Sequência 10](#_Toc535334095)

[6. Referências 11](#_Toc535334097)

1. Introdução

Atualmente usuários de redes sociais contribuem para entendimento de informações massivas que são postados e atualizados continuamente. Tais informações podem servir de subsídio para melhor conhecimento e refinamento de diversos fatos que contribui para tomada de decisões rápidas e eficazes.

1. Objetivo

Distante deste cenário, o objetivo do projeto é construir uma aplicação Java que seja capaz de se integrar a API do Twitter para buscar as seguintes informações de uma determinada hashtag.

1. Quantidade por dia de tweets da última semana.
2. Quantidade por dia de retweets da última semana.
3. Quantidade por dia de favoritações da última semana.
4. Ordenar os tweets pelo nome do autor, e exibir o primeiro nome e o último nome.
5. Ordenar os tweets por data, e exibir a data mais recente e a menos recente.
6. Componentes, bibliotecas, frameworks utilizados
   1. Componentes

Twitter4j.Query, twitter4j.QueryResult, twitter4j.Status, twitter4j.Twitter, javaUtil.ArrayList, javaUtil.Collections, javaUtil.GregorianCalendar, javaText.SimpleDateFormat,

* 1. Bibliotecas

Twitter4j, javaUtil, javaText

* 1. Frameworks

Java e MVC

1. Descrição de Funcionabilidades

Classe Conexao

Fornece o método para obter o acesso a API Twitter. O método consiste em obter a sessão que é armazenada em cache perante a validação de chave e configurações do sistema.

Classe AccessToken

Fornece ao provedor o token de acesso autorizado para acessar recursos de serviços protegidos.

Classe TwitterFactory

Fornece a fábrica de sessões para a conexão com o Twitter.

Uma instância dessa classe é completamente thread-safe e pode ser reutilizada e usada simultaneamente.

Método AccessToken

Retorna uma instância autenticada do OAuth. APISecretKey. Fornece o acesso para consumir em twitter4j.properties ou propriedades do sistema.

Ao contrário do OAuthSupport.setOAuthAccessToken (twitter4j.auth.AccessToken), esse método da fábrica de sessões retorna potencialmente uma instância em cache.

Classe Twitter

Retorna uma instância associada à configuração vinculada a fábrica de sessão.

Classe SimpleDateFormat

Utilizada para formatar e analisar datas de uma maneira sensível ao código do idioma. Permite a formatação (data -> texto), análise (texto -> data) e normalização.

Classe Date

A classe Date foi utilizada para recuperar a data e hora do sistema.

Algumas curiosidades da API:

Antes do JDK 1.1, a classe Date tinha duas funções adicionais. Permitiu a interpretação de datas como ano, mês, dia, hora, minuto e segundo valores. Também permitiu a formatação e análise de strings de data. Infelizmente, a API para essas funções não era passível de internacionalização. A partir do JDK 1.1, a classe Calendar deve ser usada para converter datas e campos de tempo e a classe DateFormat deve ser usada para formatar e analisar strings de data. Os métodos correspondentes em Data foram reprovados.

Embora a classe Date se destine a refletir a hora universal coordenada (UTC), ela pode não fazer exatamente isso, dependendo do ambiente host da Java Virtual Machine. Quase todos os sistemas operacionais modernos pressupõem que 1 dia = 24 × 60 × 60 = 86400 segundos em todos os casos. Na UTC, no entanto, cerca de uma vez a cada ano ou dois, há um segundo extra, chamado de "segundo bissexto". O segundo bissexto é sempre adicionado como o último segundo do dia, e sempre em 31 de dezembro ou 30 de junho. Por exemplo, o último minuto do ano de 1995 foi de 61 segundos de duração, graças a um salto de segundo. A maioria dos relógios de computador são precisos o suficiente para serem capazes de refletir a distinção bissexto.

Classe Mensagem

Foi desenvolvida com métodos para monitorar os eventos twetts, retwetts, favoritações, formatar e ordenar os twetss de acordo com a data, mais recentes, menos recentes, nome do autor e o padrão de exibição.

* 1. Método CompareTo

Orderna primeiro por data, em ordem decrescente, caso a data seja igual, ordena por nome.

* 1. Método setNomeAutor

Insere o nome do autor e ordena com o primeiro nome e depois o ultimo nome.

* 1. Método setData

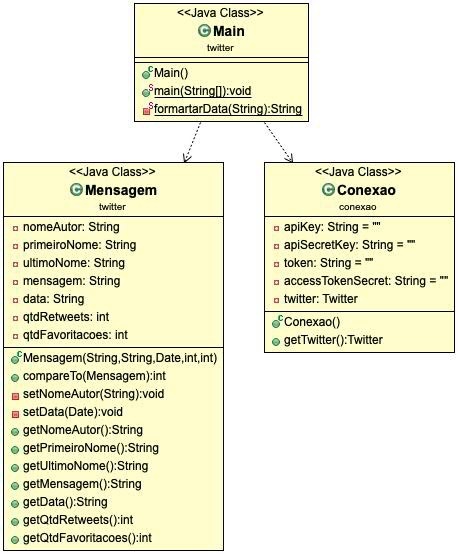
Insere a data de postagem do twitt.

* 1. Método Mensagem

Método construtor que recupera o nome do autor, a mensagem, data de publicação, quantidade de retwetts e a quantidade de favoritações.

1. Diagramas
   1. Diagrama de Classe

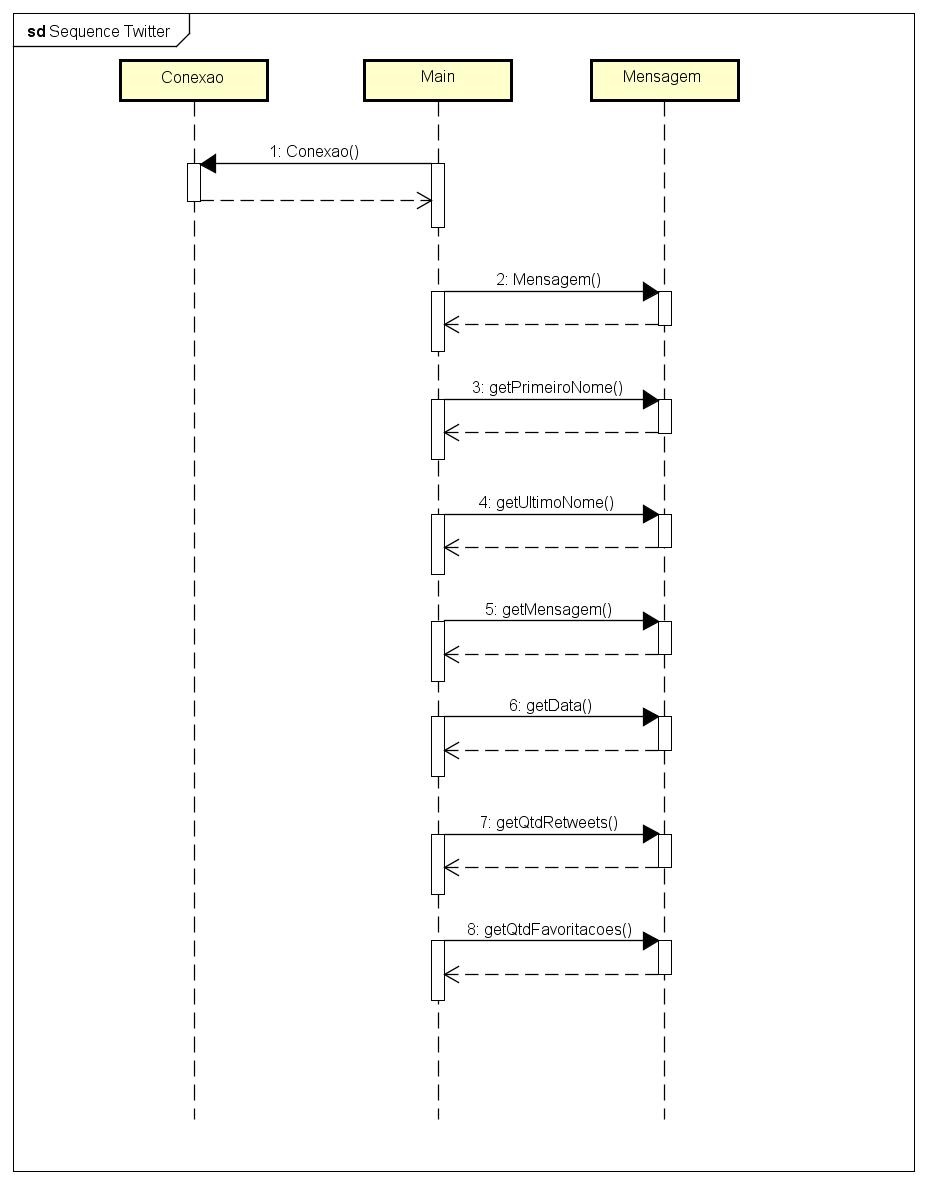
Figura 1 – Diagrama de Classe



Fonte Própria(2019, p. 6)

* 1. Diagrama de Sequência

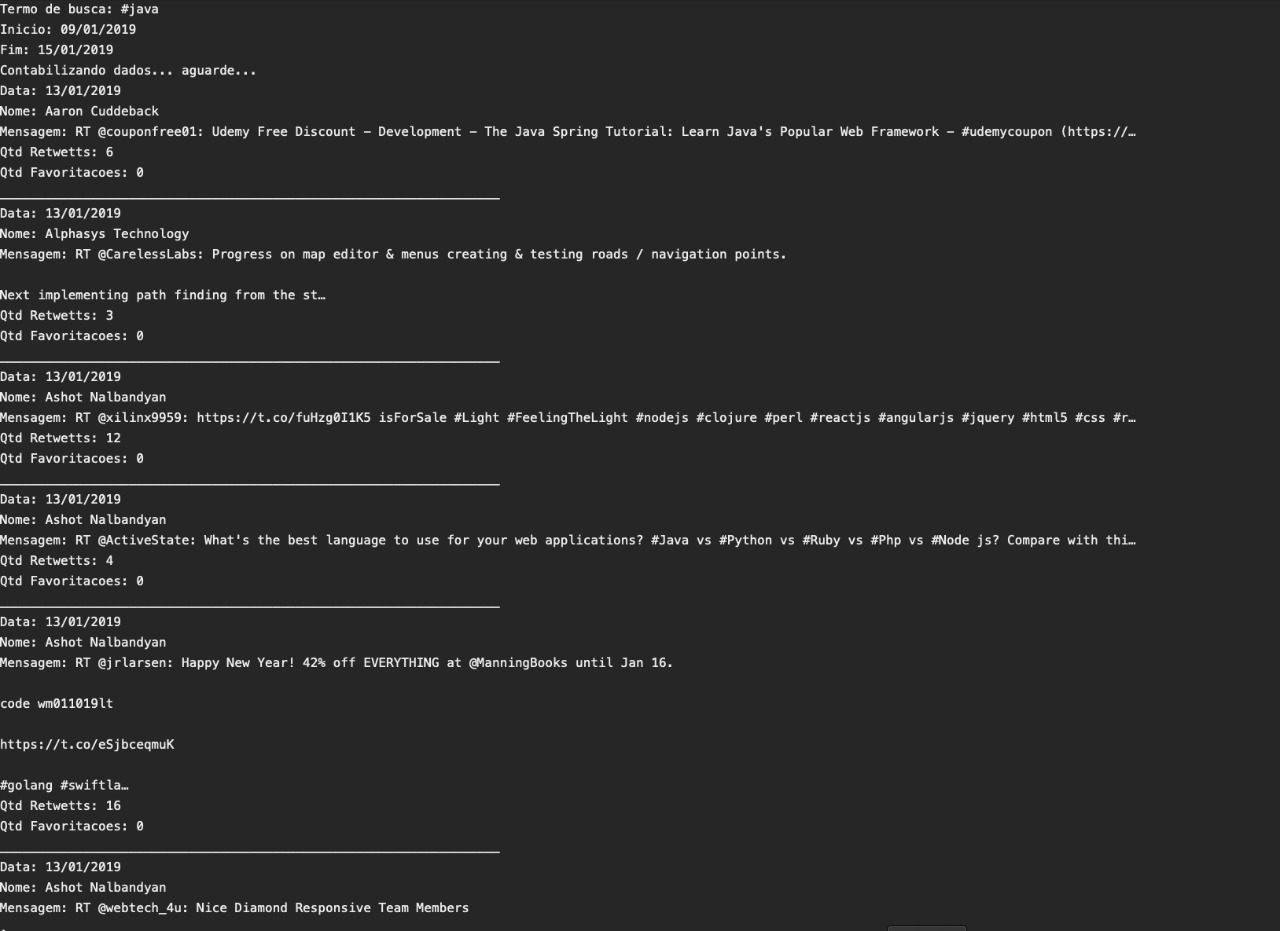
Figura 2 – Diagrama de Sequência



Fonte Própria(2019, p 7)

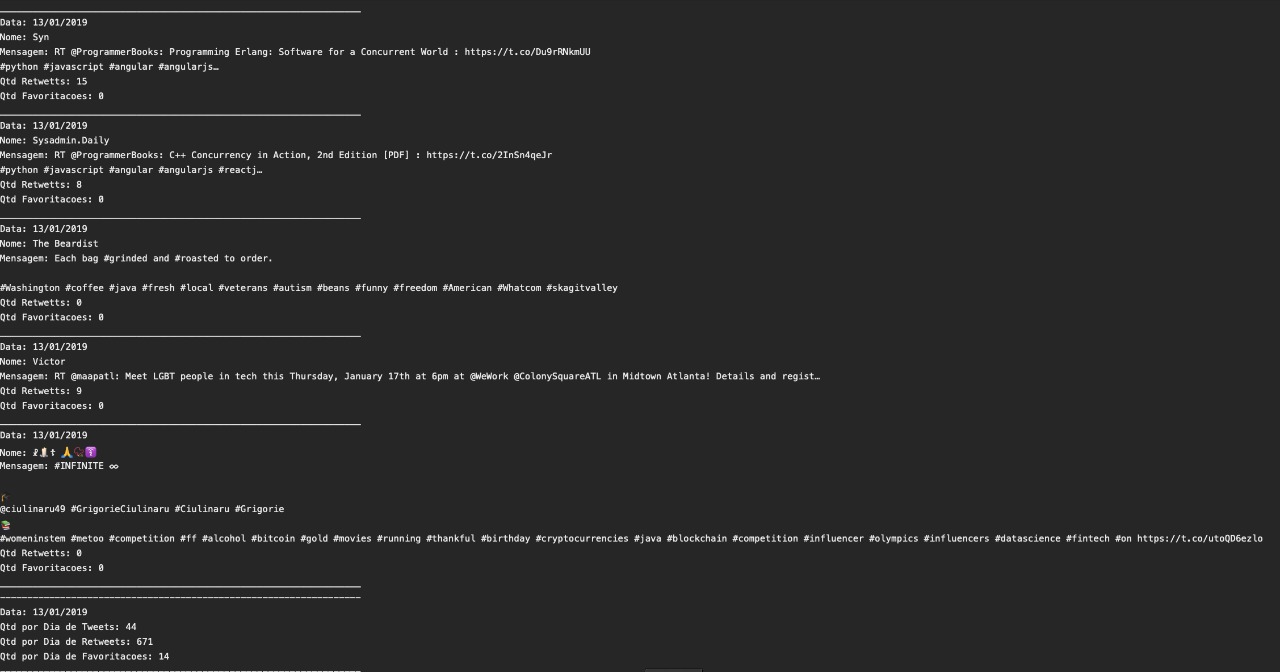
1. Prints de Tela

Figura 3 - Print de Tela 1



Fonte Própria(2019, p. 8)

Figura 4 - Print de Tela 2



Fonte Própria(2019, p. 9)

1. Referências

<http://twitter4j.org/en/index.html>

<https://docs.oracle.com/javase/6/docs/api/java>