Fundamentos Java e UML

FIAP 27SCJ

**Projeto Final de Avaliação**

Rodrigo Martinez da Rocha <rodrigogrohl@gmail.com> RM48236

FIAP 27SCJ

Fundamentos Java e UML

Ferramentas, API’s e aplicações Utilizadas

- JDK 8

- Eclipse Kepler

- Twitter4J

- JUnity 4.11

- Astah Community Edition

- Microsoft Word 2010

## Objetivo (notas do Professsor)

“A proposta deste projeto é aplicar os conhecimentos adquiridos e revisados em todo seu conteúdo, abrangendo conceitos de modelagem de classes, diagramas controle de exceção, coleções, streams, IO, threads, dentre outros.”

“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.”

Sumário

[Objetivo (notas do Professsor) 1](#_Toc452453161)

[1. Solução Definida 2](#_Toc452453162)

[2. Modelo Proposto 2](#_Toc452453163)

[3. Diagrama de Classes 3](#_Toc452453164)

[4. Diagrama de Sequência 3](#_Toc452453165)

[5. Protótipo E Execução 4](#_Toc452453166)

[6. Source 5](#_Toc452453167)

[7. Conclusão 5](#_Toc452453168)

# Solução Definida

Dado contexto, a solução escolhida por motivos de performance e quantidade de consultas foi:

Carregar **toda a lista de Tweets** de forma massiva em uma única lista e, após isso iterar os dados localmente catalogando as informações solicitadas:

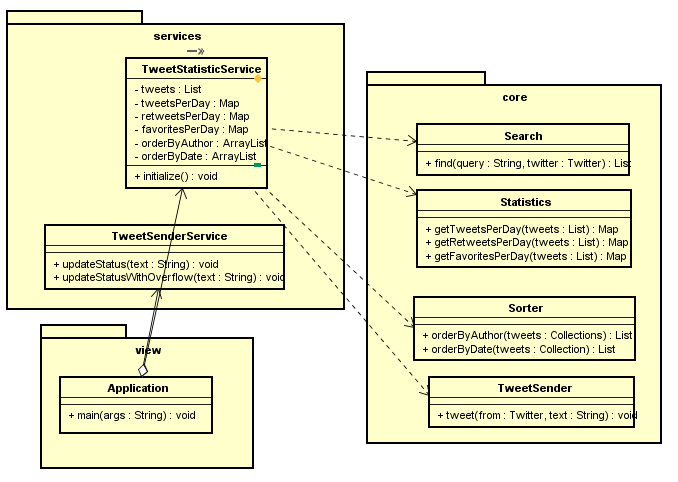
Esta abordagem foi selecionada pois a API do Twitter não permite buscas com domínio reduzido. Exemplo: “Buscar **todos os Usuários** que Postaram a Hastag xyz”.

É necessário retornar “**Todos os Tweets** que possuem a Hastag xyz” e, desta forma chegar aos seus respectivos autores.

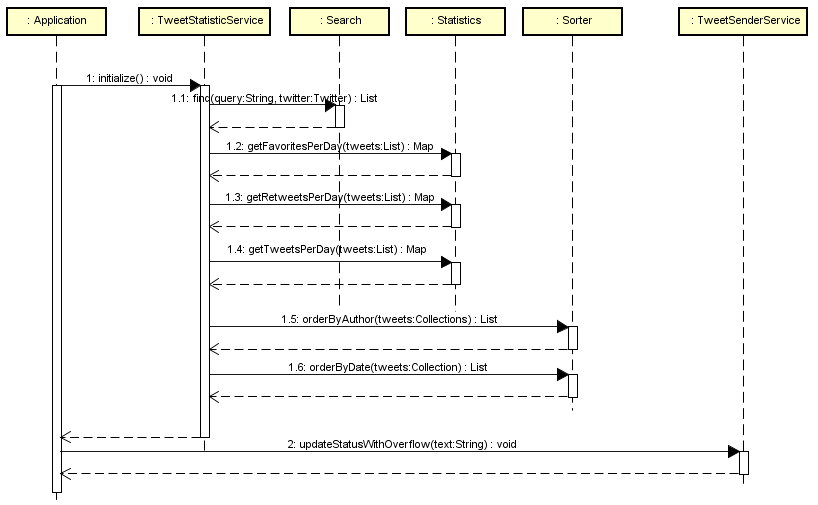
# Modelo Proposto

Um projeto (Eclipse Project), em três camadas distintas por empacotamento, sendo elas:

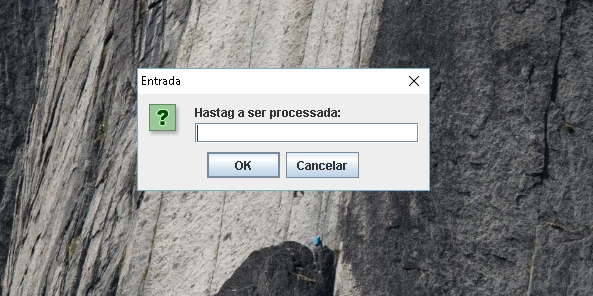
# Diagrama de Classes



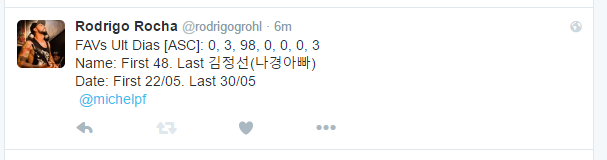
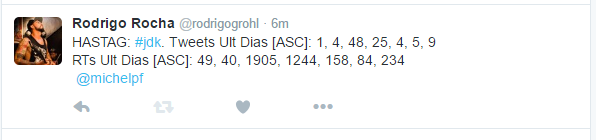
# Diagrama de Sequência



# Protótipo E Execução



**Modelo Tweet** (UpdateStatus) via API, respeitando os limites de caracteres da plataforma:

**Hastag** xyz;  
 Quantidade **Tweets** últimos dias (Ascendente): d -6, d -5, d -4, d -3, d -2, d -1, hoje.  
 Quantidade **Retweets**: d -6, d -5, d -4, d -3, d -2, d -1, hoje.  
 Quantidade **Favoritos**: d -6, d -5, d -4, d -3, d -2, d -1, hoje.  
 Primeiro e últimos **nomes** por ordenação alfabética;  
 Primeira e última **data** dos tweets;  
  


# Source

Esta atividade foi disponibilizada **no Portal Do Aluno FIAP** e também no **Github**:

<https://github.com/rodrigogrohl/fiap-rrocha/tree/master/27SCJ-FundJava-Final>

**Log de execução**: 27SCJ-FundJava-Final\documentacao\log-app.txt

**Diagramas UML**: /FundamentosJava-Final/documentacao/Especificacao-UML.asta

# Conclusão

A tecnologia deve sempre ser usada de forma eficaz, atendendo um objetivo proposto, soluções devem ser simples e utilizar outras soluções. Reinventar a roda nunca é uma opção!

Entender o cenário, avaliar as opções e ferramentas disponíveis, formam a base de qualquer solução efetiva.

Particularmente, agradeço o professor e efetividade em acrescentar pontos importantes no contexto da matéria, novidades da SDK, cases, experiências e cenários reais.