

UNIVERSIDADE FEDERAL DO MARANHÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO (PPGCC)
PROGRAMA DE PÓS-GRADUAÇÃO DOUTORADO EM CIÊNCIA DA
COMPUTAÇÃO (DCCMAPI)

Disciplina: Engenharia de *Software*

Aluno: Wellington Luis Mineiro França

Prof: Dr. Davi Viana

Data: 20/07/2020

PROJETO DA DISCIPLINA ENGENHARIA DE SOFTWARE – DEFINIÇÃO DE
REQUISITOS E ARQUITETURA – TÉCNICAS ANÁLISE DOCUMENTAL e
PROTOTIPAGEM

1. INTRODUÇÃO

Elicitação de requisitos funcionais e não funcionais, além da definição e modelagem da arquitetura do *software* - produto do projeto de tese de doutorado – capaz de recuperar dados de geolocalização (latitude, longitude) de usuários da rede social *Twitter* no município de São Luís/MA, a partir de um termo encontrado em seus *twits*, como atividade da disciplina Engenharia de *Software*.

2. REGISTRO DE DOCUMENTOS

2.1 Documentos analisados

- Documentação *online* da *Twitter* API – Interface de programação que permite amplo acesso aos dados públicos do *Twitter* que os próprios usuários escolheram compartilhar com o mundo [1].

3. REQUISITOS

3.1 Funcionais

1. **RF01:** Obter *tokens* provisórios de usuário;
2. **RF02:** Solicitar *tokens* de usuário final passando os *tokens* provisórios;
3. **RF04:** O sistema deve solicitar autenticação aos servidores *Twitter* passando as chaves e *tokens* de usuário final como parâmetros;
4. **RF05:** Cadastrar os dados resultantes da consulta: DATA, HORA, LATITUDE, LONGITUDE, dos últimos 03 meses, em um arquivo texto;
5. **RF06:** Agrupar os dados e rotular os aglomerados com base nas coordenadas obtidas para cada hora do dia;
6. **RF08:** Com base nos aglomerados, o sistema deve solicitar um ponto P_1 de origem, um ponto P_2 de destino, e calcular trajetórias de um ponto a outro para cada hora do dia nos últimos 03 meses;

7. **RF12:** O sistema deve sugerir uma trajetória para cada hora do dia, considerando os aglomerados de maior frequência adjacentes a um ponto P em todas as trajetórias calculadas para o horário específico;
8. **RF13:** Exibir a trajetória sugerida em um mapa da cidade de São Luís, representando-a visualmente como uma linha contínua sobre aglomerados específicos;

3.1 Não Funcionais

9. **NF01:** Cadastrar conta no *Twitter*;
10. **NF02:** Registrar aplicativo na página de desenvolvedor do *Twitter*;
11. **NF03:** Obter chaves de acesso na página de desenvolvedor do *Twitter*;

5. SIGLAS E ACRÔNIMOS

API – Interface de Programação de Aplicação

Referências

- [1] Sobre as APIs do Twitter. Disponível em <<https://help.twitter.com/pt/rules-and-policies/twitter-api>>. Acesso em 19 jul 2020.

6. ARQUITETURA A SER UTILIZADA

- MVC

7. MODELAGEM – DIAGRAMA DE CLASSES

Camada Modell

Coordenada
- data: date - hora: time - latitude: float - longitude: float
+cadastrar(data, hora, latitude, longitude): void

Aglomerado
- aglomerado: array
+agrupar(latitude, longitude): void +rotular(aglomerado): void +calcular_trajetorias(p1, p2): void +trajetoria_otima(): void

Twitter
- key, key_secret, token, token_secret: string - chave_consulta: string
+consultar_key(key, key_secret): string +consultar_token(key, key_secret, token, token_secret): string +filtrar(chave_consulta): json

Camada Controller

Calcular trajetórias
+consultar_token(): void +cadastrar(): void +agrupar(): void +rotular(): void +carregartelaSolicitarpontos(): void +Solicitarpontos(): void +calcular_trajetorias(): void

Sugerir trajetória
+ trajetoria_otima(): void +carregartelaTrajetoriasugerida(): void

Camada View

Solicitar pontos origem-destino
- p1: coordinate - p2: coordinate - calcular: Button

Apresentar trajetória sugerida
- mapa: map - linha: line