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:** Obter chaves de acesso na página de desenvolvedor do *Twitter*;
- 3. **RF03:** Solicitar *tokens* de usuário final passando os *tokens* provisórios;
- 4. **RF04:** O sistema deve solicitar autenticação aos servidores *Twitter* passando as chaves e *tokens* de usuário final como parâmetros;
- 5. **RF05:** Cadastrar os dados resultantes da consulta: DATA, HORA, LATITUDE, LONGITUDE, dos últimos 03 meses, em um arquivo texto;
- 6. **RF06:** Agrupar os dados e rotular os aglomerados com base nas coordenadas obtidas para cada hora do dia;

- 7. **RF07:** Com base nos aglomerados, o sistema deve solicitar um ponto P₁ de origem, um ponto P₂ de destino, e calcular trajetórias de um ponto a outro para cada hora do dia nos últimos 03 meses:
- 8. **RF08:** O sistema deve sugerir uma trajetória para cada hora do dia, considerando os aglomerados de maior freqüência adjacentes a um ponto P em todas as trajetórias calculadas para o horário específico;
- 9. **RF09:** 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

- 10. **NF01:** Cadastrar conta no *Twitter*;
- 11. NF02: Registrar aplicativo 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

Aglomerado

- aglo: array
- rotulo: char
- +agrupar(lat, long): aglo +cadastrar aglo(aglo): void

Coordenada

- data: date
- hora: time
- lat: float
- long: float
- +consultar token(): void
- +cadastrar dados(data, hora, lat, long): void

Trajetoria

- trajetoria: array
- data: date
- hora: time
- +calcular_trajetoria(p1, p2): trajetoria +cadastrar trajetoria (trajetoria): void

Token

- key, key_secret, token, token secret: string
- chave_consulta: string
- +consultar(): string
- +consultar_key(key, key_secret): string
- +consultar_token(token): string
- +consultar_token_usu(token_usu,
- token_usu_secret): string +filtrar(chave consulta): json

Trajetoria_sugerida

- traj_sug: array
- data: date
- hora: time
- + calcular_ traj_ sugerida(p1, p2): traj_sug
- + cadastrar traj sugerida(traj sug): void
- + solicitarpontos(): void
- + apresentar trajetorira():void

Camada Controller

Control_Trajet

- +consultar token(): void
- +cadastrar_dados (data, hora, lat, long): void
- +agrupar (lat, long): aglo
- +cadastrar_aglo(aglo): void
- +consultar(): string
- +consultar key(key, key secret): string
- +consultar_token(token): string
- +consultar token usu(token usu, token usu secret): string
- +filtrar(chave consulta): json
- +calcular trajetoria(p1, p2): trajetoria
- +cadastrar_trajetoria (trajetoria): void
- +calcular traj sugerida(p1, p2): traj sug
- +cadastrar_traj_sugerida(traj_sug): void
- +carregartelaSolicitarpontos origem-destino(): void
- +solicitarpontos(): void
- +carregartelaApresentartrajetoriasugerida(): void
- +apresentar trajetorira():void

Camada View

Solicitar pontos origem-destino

- p1: coordinate
- p2: coordinate
- Calcular Trajetoria: Button

Apresentar trajetoria sugerida

- mapa: map
- linha: poligon