Laboratorio Interdisciplinare A

Progetto “Emotional Maps”

Validità: giugno 2019 – febbraio 2020

Data pubblicazione: 17 aprile 2019 Ultima modifica: 17 aprile 2019
Autore: Davide Tosi.

Versione: 2.0

1. Il Problema, Dati e Operazioni

Figura 1. Mappa emozionale di Milano e relativi POI

Siamo nella realtà della startup Facedoor srl. La nuova app da inserire sul mercato nel 2019 prevede la creazione di mappe emozionali, dove ogni Punto di Interesse (POI) della città, caratterizzato dalle sue coordinate latitudine/longitudine, ha associato uno stato di “gradimento” (stato emozionale) delle persone che in un dato istante vivono quel luogo.

Gli stati emozionali vengono rilevati automaticamente dai selfie e dalle immagini dei volti delle persone tramite riconoscimento facciale automatico.

Ad oggi, l’app è in grado di gestire esclusivamente la mappa di Milano e i 3 POI riportati in Figura 1. Il riconoscimento facciale è in grado di rilevare le 5 emozioni degli utenti, come riportato in Tabella 1.

Il programma da realizzare, avvalendosi di opportune strutture dati, dovrà supportare la gestione degli utenti dell’app (come login/logout, spostamenti, ecc…) ed effettuare il calcolo dello stato emozionale di ogni POI, relativo ad una specifica finestra temporale.

Stato Emozionale	Sigla	Emoji
Arrabbiato	A	
Felice	F	
Sorpreso	S	
Triste	T	
Neutro	N	

Tabella 1. Stati emozionali rilevati dal Riconoscimento facciale dell'app

Nello specifico, è utile considerare che gli utenti dell'app possono iscriversi / disiscriversi dall'app, possono effettuare il login / logout all'app essendo quindi utenti attivi o non attivi, possono muoversi all'interno della mappa, modificando quindi le loro coordinate geografiche (la posizione è sempre identificata tramite le coordinate lat/long) e hanno uno stato emozionale associato. L'app deve quindi essere in grado di capire dove l'utente si trova e su quale POI contribuisce emozionalmente.

Gli eventi interpretabili dal programma sono contenuti in un file di testo strutturato in modo tale che ogni riga rappresenti un evento di un utente in uno specifico giorno dell'anno. Gli eventi sono codificati come stringhe con formato: *stato_registrazione (IN/OUT) + stato_utente (LOGIN/LOGOUT) + timestamp (ggmmaaaa) + userID (5 caratteri alfanumerici) + coordinate_geografiche (lat/long) + sigla_stato_emotivo.*
Gli eventi contenuti nei file compaiono nell'ordine in cui si sono effettivamente verificati. Possono verificarsi anche più eventi, per lo stesso utente, nello stesso POI e con uguale timestamp, a patto che lo stato emozionale dell'utente sia differente.

L'esempio 1 mostra un possibile file *eventi.txt* con 5 eventi. Nell'esempio 1, abbiamo 3 utenti (identificati dai loro rispettivi userID) nelle prossimità del POI1; l'utente *df55s* che esegue il *logout* dall'app in data 11 aprile 2019 in prossimità del POI2, e l'utente *dr23a* che si disiscrive dall'app in prossimità del POI3.

Esempio 1. Eventi nel periodo 08/04/2019, 11/04/2019

```
IN LOGIN 09042019 ac11b 45.463,9.188 A
IN LOGIN 10042019 bc78x 45.465,9.191 F
IN LOGIN 10042019 df55s 45.464,9.190 A
IN LOGOUT 11042019 df55s 45.474,9.173 T
OUT LOGOUT 11042019 dr23a 45.458,9.181 N
… …
```

La mappa emozionale di ogni POI, in un dato intervallo di tempo, deve essere calcolata come percentuale tra il numero di occorrenze di ogni stato emotivo sul totale degli eventi. La mappa emozionale deve essere calcolata sia per gli utenti attivi (LOGIN) sia per tutti gli utenti (i.e., attivi, non attivi, registrati e non più registrati).

Riprendendo il file *eventi.txt* dell'Esempio 1, avremo quindi che, nella finestra temporale 08/04/2019 – 11/04/2019, la mappa emozionale degli utenti attivi sarà:
POI1 - 67% A, 33% F, 0% S, 0% T, 0% N
POI2 - 0% A, 0% F, 0% S, 0% T, 0% N
POI3 - 0% A, 0% F, 0% S, 0% T, 0% N

La mappa emozionale di tutti gli utenti nel file *eventi.txt* sarà invece:
POI1 - 67% A, 33% F, 0% S, 0% T, 0% N
POI2 - 0% A, 0% F, 0% S, 100% T, 0% N
POI3 - 0% A, 0% F, 0% S, 0% T, 100% N

Le operazioni che dovranno quindi essere supportate dal programma sono le seguenti:

- ***import(eventi)***: Importa in una opportuna struttura dati gli eventi contenuti nel file di nome `eventi`. Se il file di nome `eventi` non esiste non viene eseguita alcuna operazione. Gli eventi definiti in formato non conforme alla specifica riportata, o specificanti valori che non rispettano i vincoli precedentemente introdotti non dovranno essere interpretati.
- ***create_map (intervallo)***: Calcola le due mappe emozionali per i tre POI riferite alla finestra temporale `intervallo`, secondo le modalità di calcolo specificate nel paragrafo precedente. Il parametro `intervallo` deve essere nella forma: `datainizio-datafine` espressi come `ggmmaaaa`. Es.: `01042019-10042019`

2. Specifiche di Implementazione

Le modalità di esecuzione del programma e il formato dell'output prodotto devono rispettare rigorosamente quanto descritto nel seguito:

- **Input:** il programma deve leggere il file di testo *comandi.txt*, il cui nome è specificato sulla linea di comando. Il file *comandi.txt* contiene una sequenza di comandi, uno per ogni riga del file, che corrispondono alle operazioni di import del file *eventi.txt* e di calcolo delle mappe emozionali come descritto in precedenza. Esempio di specifica del formato del file *comandi.txt*:

```
import(eventi1.txt)
import(eventi2.txt)
import(eventi3.txt)
create_map(01042019-10042019)
import(eventi4.txt)
create_map(10042019-19042019)
```

- **Output:** il programma deve visualizzare sullo standard output il risultato dell'operazione *create_map*.

3. Sviluppo e Consegna del Progetto

I progetti che non rispettino la struttura descritta nel seguito o che non siano consegnati secondo le modalità descritte non verranno valutati.

- Il progetto deve essere sviluppato in team di al massimo 4 studenti
- Il progetto deve essere sviluppato in linguaggio Java (versione 8 o successive)
- Tutte le classi sviluppate dal team per la soluzione del progetto devono essere collocate in un package di nome *soluzione*. È necessario che il package *soluzione* contenga almeno 3 classi
- Il metodo *main* per l'esecuzione dell'applicazione deve essere contenuto in una classe di nome *EmotionalMaps* del package *soluzione*
- L'intestazione del file *EmotionalMaps.java* deve contenere nome, cognome e numero di matricola degli autori del progetto
- Le modalità di esecuzione del programma e il formato dell'output devono rispettare rigorosamente le caratteristiche specificate nei punti precedenti
- I progetti dovranno essere adeguatamente commentati mediante l'utilizzo dei commenti di documentazione javadoc
- La soluzione del progetto deve essere corredata da un manuale utente e manuale tecnico in formato .pdf in cui vengano evidenziate le modalità di esecuzione e funzionamento della soluzione a livello utente (manuale utente). Nel manuale tecnico dovranno essere motivate le scelte tecniche, tecnologiche, algoritmiche e delle strutture dati utilizzate, e le scelte implementative adottate. All'inizio della documentazione devono essere indicati nome, cognome e numero di matricola degli autori del progetto
- I progetti dovranno essere consegnati, di volta in volta, secondo le modalità e le tempistiche descritte sulla pagina dell'insegnamento su e-learning, in un unico file zippato (.zip) contenente:
    - i file manuale_utente.pdf + manuale_tecnico.pdf;
    - la directory *src/* contenente il codice sorgente del progetto;
    - la directory *bin/* contenente il codice eseguibile del progetto.

Quindi il contenuto del file .zip sarà del tipo: *manuale_utente.pdf + manuale_tecnico.pdf + src/soluzione/EmotionalMaps.java + src/soluzione/Classe1.java + ... + bin/soluzione/EmotionalMaps.class + bin/soluzione/Classe1.class + …*

- Il nome del file .zip deve coincidere con il numero di matricola del Project Manager del team
- Il progetto deve essere inviato una sola volta dal Project Manager del team

4. Criteri di Valutazione

L'attività di progetto coinvolge gli insegnamenti del primo anno del corso di laurea, in particolare i corsi di Algoritmi e Strutture Dati e di Programmazione. L'utilizzo di strutture dati e algoritmi appropriati costituisce quindi un elemento determinante nella valutazione del progetto.
Oltre agli aspetti algoritmici verrà inoltre valutata la qualità del codice: struttura del progetto, adeguatezza dei costrutti utilizzati e dei commenti e la qualità della documentazione prodotta a supporto.
I progetti vengono valutati dalla commissione d'esame, i progetti che superano la valutazione devono essere discussi dagli studenti di fronte alla commissione durante le date d'appello pianificate.
Durante la discussione gli studenti dovranno fornire chiarimenti sulle scelte effettuate e sul codice presentato. In particolare, dovranno giustificare la scelta delle strutture dati utilizzate nel progetto e dovranno spiegare nel dettaglio gli algoritmi implementati, argomentando opportunamente l'efficienza delle soluzioni proposte.
Ogni studente deve essere in grado di giustificare le scelte e commentare il codice.