# IOT – Metric Viewer

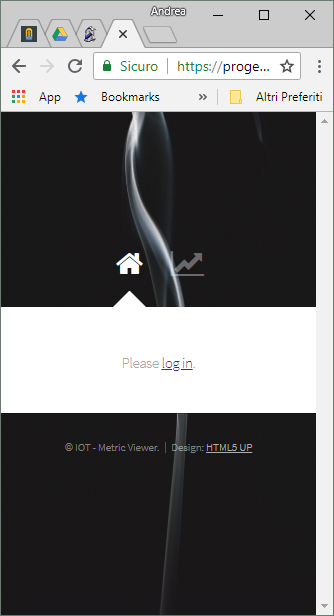
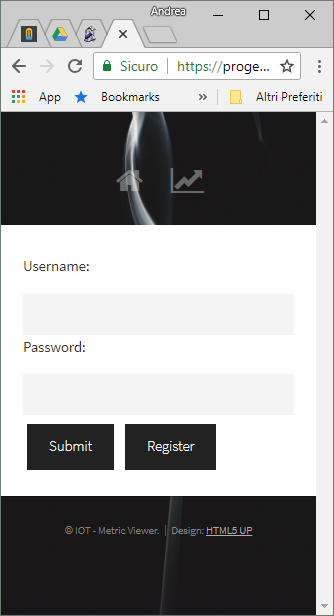
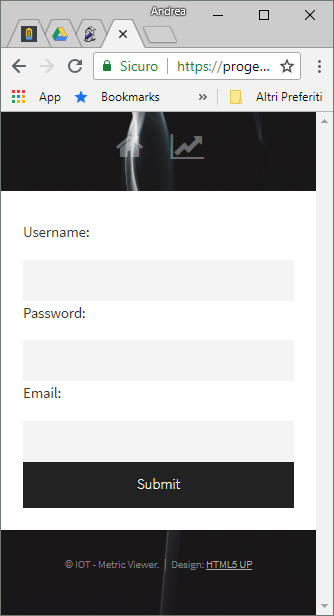
Autore: Andrea Cattaneo  
Matricola: 909227  
Data: 24/06/2018  
Versione del documento: 1.0

## OBIETTIVO DEL SOFTWARE

L’obiettivo del software realizzato è quello di mettere a disposizione dell’utente un’interfaccia web per l’accesso ai dati prodotti da un macchinario industriale. Questi dati sono metriche che rappresentano grandezze fisiche misurate sulla macchina stessa. Si tratta nello specifico dei valori di coppia, corrente, velocità, temperatura che vengono misurati su un motore elettrico asincrono. Non avendo a disposizione dei macchinari che misurino questi valori, si è realizzato un applicativo console che, ad intervalli di tempo costante, invia dei documenti contenenti le misure ad un database Mongodb.  
I messaggi visualizzati sul portale saranno solo quelli relativi all’utente loggato. In ogni messaggio spedito dal macchinario è infatti presente un campo “username” che permette di distinguere le metriche associate ad un utente.

## UTILIZZO

L’utilizzo del portale è molto semplice. L’utente non deve fare altro che collegarsi al sito web:

<https://progetto-web-unimi.herokuapp.com>

Compare un messaggio che invita l’utente ad effettuare il login. Premendo sul link compare un form dove è possibile effettuare il login oppure passare alla registrazione.   
La registrazione di un nuovo utente avviene tramite l’apposita schermata. Una volta registrato un nuovo utente non saranno disponibili dati associati ad esso, quindi il grafico sarà vuoto.

A questo punto va avviato l’applicativo console utilizzando il comando: “node metric-publisher.js”. L’applicativo richiede a quale utente si vuole associare i dati che verranno generati. Inserire quindi il proprio username.  
Una volta avviato, il software inizierà ad inviare documenti al database mongodb. Nel sito web sarà possibile vedere popolarsi il grafico popolarsi ed aggiornarsi automaticamente.

Compare un messaggio che chiede all’utente di loggarsi nel portale.

Premendo su link “log in” appare una finestra che permette all’utente di loggarsi o registrarsi:

## Valutazione delle parole

Per determinare se una parola trovata nel dizionario corrisponde alla parola inserita, si valuta la quantità di caratteri corrispondenti tra l’inizio di una parola e l’inizio dell’altra. Esse devono iniziare nello stesso modo. Si valuta il valore di caratteri coincidenti come punteggio della parola. La quantità di caratteri corrispondenti deve essere almeno del 70%. La parola più lunga tra la cercata e la trovata rappresenta il 100%.

Il motivo per cui vengono valutati solo i caratteri identici nella parte iniziale della parola è che con buona probabilità due parole che hanno inizio uguale hanno la stessa radice e quindi un significato simile.

Se la parola trovata coincide con i caratteri della stringa cercata per l’intera lunghezza della parola cercata, ma ha una lunghezza maggiore (la cercata è inferiore del 70% della trovata), la parola viene scartata. Il motivo è che essendo la parola trovata molto più lunga della cercata, potrebbe avere significato completamente diverso.

Ogni parola inserita viene confrontata con tutte le parole dei dizionari. La parola con corrispondenza percentuale maggiore è quella che viene considerata come “parola trovata” (a patto che superi la soglia minima del 70%).

Una parola trovata nel dizionario delle parole neutre o una parola non trovata assume valore zero. Al contrario una parola trovata nei dizionari positivo o negativo, assume appunto un segno positivo o negativo rispettivamente. In questo caso il modulo del punteggio è il numero di caratteri corrispondenti tra la parola cercata e la parola trovata.

Una parola corrispondente molto lunga vale più punti di una parola molto corta. Il motivo di questa scelta è che se una parola è più lunga tendenzialmente ha una peso maggiore, ad esempio quando stiamo analizzando dei superlativi.

*Bello < Bellissimo  
Brutto < Bruttissimo*

Oltre ai superlativi, anche normali aggettivi tendono ad essere più lunghi se hanno un significato di maggior peso:

*Bello < Fantastico  
Brutto < Schifoso < Orripilante*

## Valutazione della frase

La valutazione complessiva della frase viene ottenuta sommando la valutazione di ogni singola parola ed ottenendo così un punteggio finale positivo o negativo a seconda del risultato.  
In questa valutazione hanno molto peso le parole espressamente positive o negative anche se come quantità sono decisamente inferiori. Questo perché alle parole neutre che vanno per la maggiore è stato attribuito un punteggio zero.

## DIZIONARI

Per ottenere performance migliori i dizionari vengono caricati in memoria in modo da essere confrontabili velocemente con le parole inserite. Il caricamento avviene all’avvio del software oppure a seguito della modifica dei dizionari. Il processo consiste nell’allocare in memoria un array di stringhe tramite la funzione “*malloc*”. Una volta effettuata l’allocazione si leggono i file txt dei dizionari e si popolano gli array con le stringhe di testo lette nei file. Alla modifica dei dizionari, prima di effettuare un nuovo caricamento in memoria, viene eseguita la “*free*” della memoria liberando così spazio dallo “*heap*” prima di occuparla nuovamente con un nuovo array di valori.

## PROGRAMMA

Il programma si avvia nel **main.c** che è il punto d’ingresso. Esso richiama il caricamento dei dizionari all’avvio e gestisce gli inserimenti dell’utente andando poi a richiamare le funzioni per la valutazione del testo, le funzioni per la modifica dei dizionari o la chiusura del programma.

## MODULI

Per rendere più leggibile il software è stato organizzato in moduli suddivisi per macro-funzionalità. Ogni modulo ha un file di interfaccia \*.h che espone solo le funzioni necessarie. Gli altri metodi e variabili non necessarie dall’esterno del modulo, sono incapsulate non esponendole nei file di interfaccia e semplificando così l’utilizzo del modulo stesso. Di seguito vengono elencati ed illustrati i principali moduli:

**common** Contiene funzionalità comuni che possono essere utilizzate in qualsiasi parte del programma.

**fileAccess** Modulo di accesso ai file. L’implementazione contiene funzioni di lettura / scrittura da file di testo esterni.

**sentiment** Modulo per la valutazione del sentiment di una frase. Espone la funzione “*LaunchSentimentAnalysis*” che permette di effettuare la valutazione del sentiment di una frase. In questo modulo viene anche definita la struttura “*WordEvaluation”* che è utilizzata per salvare una parola trovata, il numero di caratteri corrispondenti e la percentuale di corrispondenza.

**vocabulary** Modulo contenente la gestione dei vocabolari e la loro esposizione alle altre parti del programma.   
Le funzionalità disponibili sono:  
-modifica del contenuto dei vocabolari tramite la funzione “*LaunchVocabularyEditing*”. Esso riceve in ingresso la frase (*statement*) e la stringa di split tra parola da modificare ed il suo nuovo attributo (*vocabEditSplitter*).  
-verifica se una parola è contenuta nel dizionario delle negazioni tramite “*NegationVocabularyContains”*   
-caricamento in memoria dei dizionari tramite funzione “*LoadAllVocabularies*”  
-liberazione della memoria dedicata ai dizionari tramite funzione “*FreeAllVocabularies*”

## FONTI ESTERNE