



Università degli Studi di Napoli Federico II
Scuola Politecnica e delle Scienze di Base
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Corso di Laurea in Informatica
A.A 2017/2018

Schema di massima relazione

Primo Progetto di Tecnologie Web

Inquinamento

Professoressa:
Anna Corazza

Gruppo Capone

Studenti:
Andrea Capone
N86001676

Gianluca De Lucia
N86001826

Per ogni riferimento
email: gianluca.delucia.94@gmail.com

Indice

CAPITOLO 1 : Proposal

- I. Richiesta
- II. Proposta di soluzione
- III. Stackholder: categorie di utenti
- IV. Casi d'uso
- V. MockUp
- VI. Site Map
- VII. Metodi di session tracking
- VIII. Proposta di strumenti da utilizzare

CAPITOLO 2 : Relazione di fine progetto

- I. Introduzione
- II. Modifiche alla proposta di soluzione
- III. Discussione e motivazioni alle tecnologie utilizzate
- IV. Modifiche alla sitemap
- V. Discussione generale sul sistema implementato
- VI. Funzionalità implementate con breve descrizione
- VII. Tecniche di session tracking utilizzate
- VIII. Discussione finale

Richiesta

Si vuole implementare un sistema per il controllo di dieci inquinanti atmosferici, le cui misurazioni sono generate ad intervalli regolari di 6 ore da sensori disposti sul territorio. Si consiglia di generare dei valori random per modellarli. Sulla base di questi valori, l'utente potrà richiedere medie temporali per specifiche località di suo interesse, o medie su territori più ampi. Inoltre, potranno venir generati degli allarmi quando i valori di inquinamento superano delle soglie prefissate per un certo intervallo di tempo. L'utente può inoltre configurare un profilo con informazioni sulla regione geografica a cui è interessato.

Proposta di soluzione

L'utente accede al sistema, specifica la località su cui vuole analizzare le statistiche. Egli può calcolare la media temporale dei valori di inquinamento sulla sua località o su un territorio più vasto: città e regioni. I sensori ogni 6 ore inviano i valori degli inquinanti sulla località su cui si trovano e vengono memorizzati. Se i valori degli inquinanti superano una certa soglia in un lasso di tempo da stabilire, viene mandato un allarme. I sensori vengono simulati attraverso delle funzioni che generano dei valori casuali da 1 a 100 per inviare una percentuale più o meno realistica rispetto alla quantità di aria pulita presente.

L'utente inoltre potrà decidere di registrarsi, inserendo uno username e una password, che gli garantiranno l'accesso alla piattaforma. Durante la registrazione gli verrà chiesto di scegliere una località preferita in modo da memorizzarla e visualizzarla direttamente dopo il primo accesso. Inoltre quest'ultimo può decidere successivamente di cambiare la località da memorizzare all'accesso.

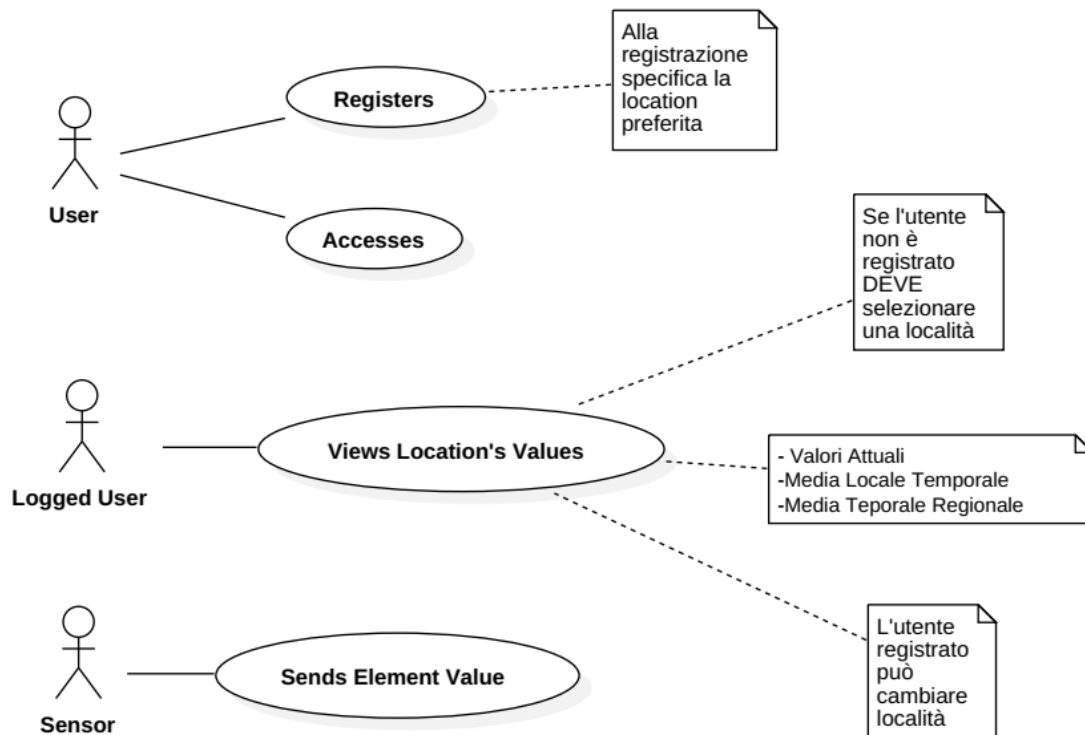
Per inquinante atmosferico si intende una molecola che altera lo stato dell'aria modificandone la struttura. I principali sono:

- Biossido di zolfo SO₂
- Biossido di azoto NO₂
- Particolato Atmosferico PM
- Benzene C₆H₆
- Idricarboni Policiclici Aromatici IPA
- Piombo Pb
- Nichel Ni
- Cadmio Cd
- Ozono O₃
- Monossido di Carbonio CO

Stackholder: categorie di utenti

L'applicazione web potrebbe risultare utile per tutti i ricercatori (chimici, biologi etc.) che lavorano presso un istituto di ricerca geologica o sugli inquinanti atmosferici come ad esempio L'IRC (Istituto di Ricerca sulla Combustione) che studia anche l'impatto ambientale da combustione, L'ENEA le cui attività di ricerca includono l'osservazione del sistema climatico a scala regionale, oppure CNR (Centro Nazionale di Ricerca). L'applicazione web, grazie alla sua facilità d'uso e un aspetto user-friendly, potrà essere facilmente utilizzata anche da qualsiasi utente inesperto interessato all'argomento.

Casi d'uso



MockUp

1)

HOME PAGE

Quale località ti interessa maggiormente?

Regioni

Città

GO

ACCEDI ORA!

2)

Sia nel caso l'utente decida di iscriversi, che di accedere subito, gli verrà chiesto di scegliere una località. Nel primo caso verrà anche memorizzata per i futuri login facilitati.



Scegli un'altra zona

GO

Elementi

Biossido di zolfo SO₂
Biossido di azoto NO₂
Benzene C₆H₆
Piombo PB
Nichel Ni
...

30%
20%
12%
47%
3%
...

3)

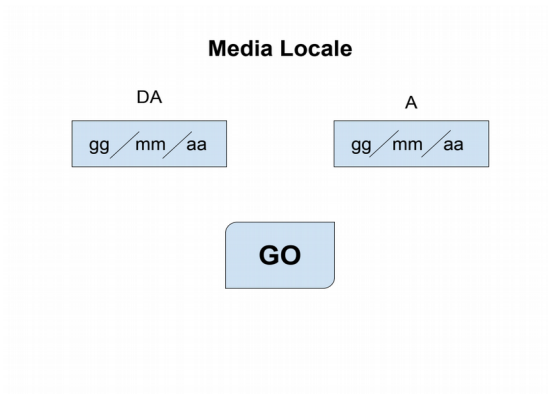
L'utente qui può decidere di cambiare località e

Media
Locale

Media
Regionale

tornare al mockup 2, o di visionare la media dei valori inquinanti rispetto alla località o regione andando rispettivamente ai mockup 4 e 5. Se si cambia zona l'utente registrato sceglie se impostarla di default.

4/5)



Media Locale

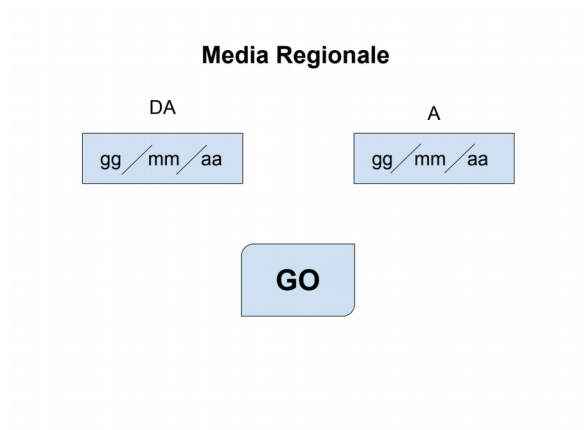
DA

gg/mm/aa

A

gg/mm/aa

GO



Media Regionale

DA

gg/mm/aa

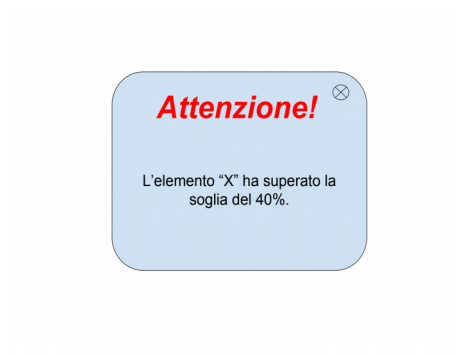
A

gg/mm/aa

GO

L'utente verrà rimandato al mockup 3.

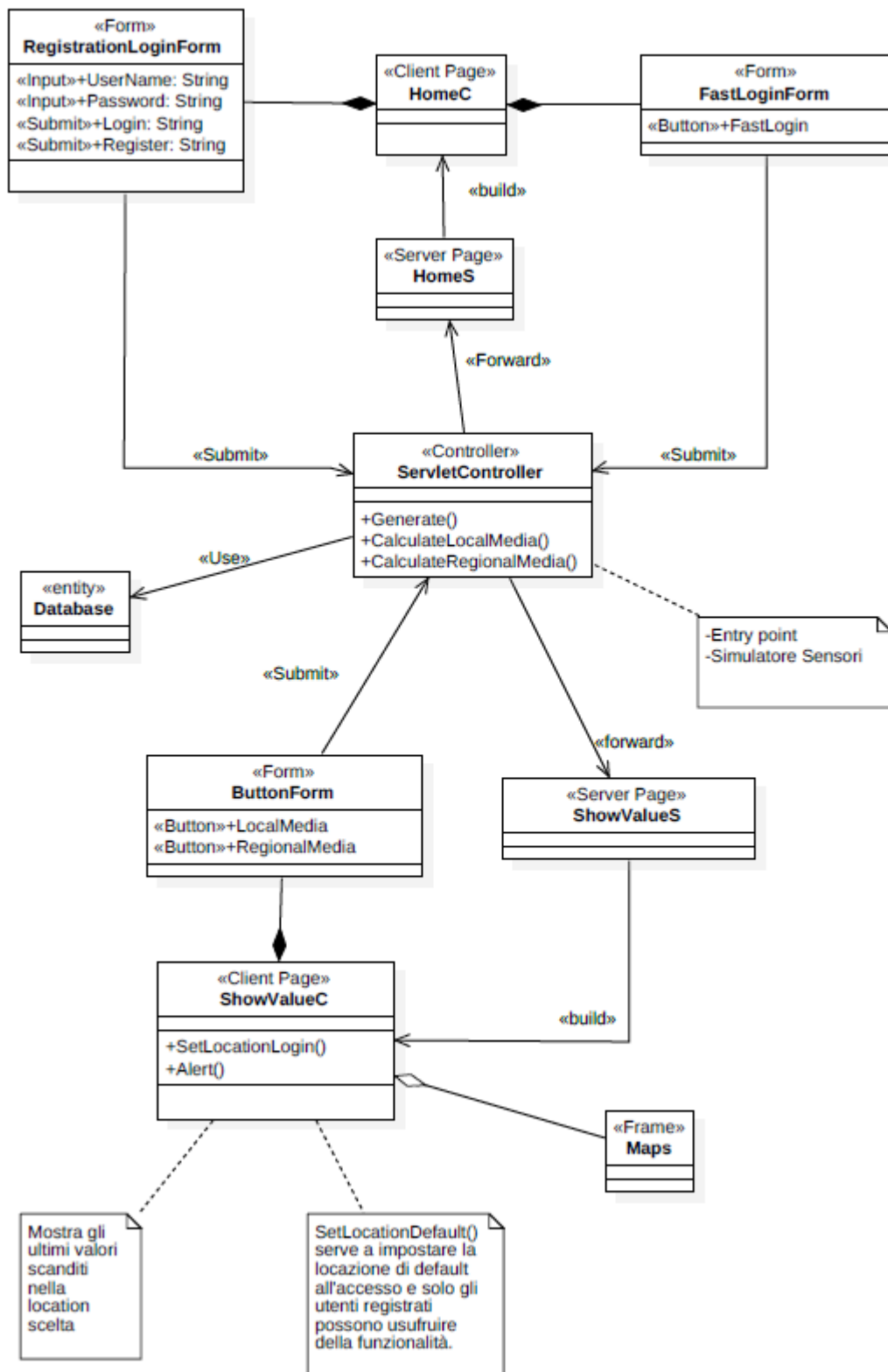
6) Pop-Up che avvisa se un elemento ha superato una determinata soglia.



Attenzione! ✕

L'elemento "X" ha superato la soglia del 40%.

Site map



Metodi di session tracking

1. Autenticazione dell'utente.
2. Bisogna conservare i valori degli inquinanti delle ultime 6 ore.
3. **Tracciare la località dell'utente registrato o quella che desidera cambiare.**

Per i punti 1 e 3 abbiamo deciso di utilizzare il local storage perchè più adatto a memorizzare in modo persistente i dati riguardo la località di default e conservare i valori degli inquinanti.

Per il secondo punto utilizzeremo il session tracking tramite autenticazione

Proposta di strumenti da utilizzare

- Oracle: tecnologia molto utilizzata e conosciuta (esame di Basi di Dati)
- HTML: struttura dell'applicazione
- CSS: layout grafico
- Javascript: gestione eventi (es. alert), storage, etc.
- Servlet e JSP: programmazione lato server e ottime per il design di MVC Web; Servlet per il controller, JSP per la View
- Google Maps: Per incorporare la mappa e visualizzare la località scelta.

Capitolo 2

Introduzione

Questa applicazione web è stata sviluppata dagli studenti della facoltà di Informatica presso l'università degli studi di Napoli Federico II Gianluca De Lucia e Andrea Capone, riuniti nel team "Atom Byte Corporation".

Di seguito sono presentate la proposta finale di sviluppo, con le relative modifiche, screenshot di esecuzione, informazioni relative alle tecnologie utilizzate e presentazione del codice dei principali metodi utilizzati.

Modifiche alla proposta di soluzione

L'utente, prima di accedere al sistema con le sue credenziali, mantiene salvata la località che ha specificato alla registrazione. Se durante una sessione cambia la località, viene memorizzata e sostituita alla precedente. Un utente ha anche la possibilità di effettuare un fast login senza registrarsi ma la sua località non viene salvata nella base di dati.

I sensori vengono simulati tramite uno "scheduling event" gestito dalla base di dati, la quale genera random i valori degli inquinanti ogni ora (la scelta di generarli ogni ora è solo a scopo dimostrativo).

Questi valori generati random sono dei numeri reali compresi tra 0 e 1 che corrispondono a delle percentuali leggermente più realistiche. Se almeno uno dei valori degli inquinanti supera la soglia di 0.9, lato client viene generato un alert che indica quali inquinanti raggiungono quest'ultima.

Discussione e motivazioni alle tecnologie utilizzate

Tecnologie utilizzate:

1) Java: abbiamo deciso di utilizzare java per quanto riguarda lo sviluppo lato server e per l'interazione con la base di dati perché abbiamo familiarità con il linguaggio, è sicuro ed è molto semplice da implementare. Abbiamo implementato una classe chiamata DAO che interroga il database, due classi servlet per usarle come controller (una per ogni pagina web) e gestire i dati che prende dalla DAO per passarli alla JSP in base alla richiesta http, per mandare dei messaggi (showMessageDialog()) e per gestire l'autenticazione dell'utente.

Abbiamo utilizzato poi una JSP per ogni pagina web per la visualizzazione. (Modello MVC).

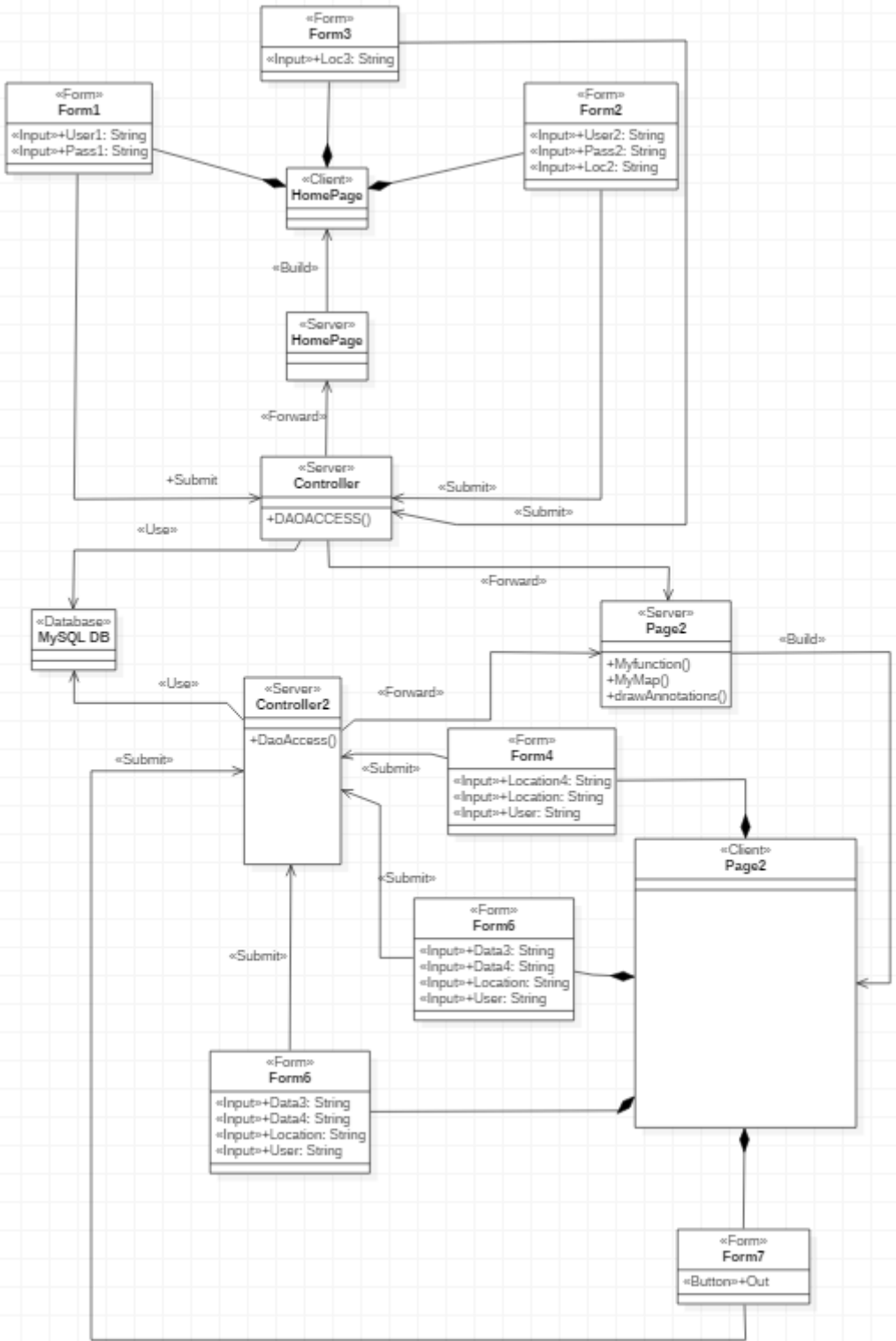
2) HTML5 e CSS: le seguenti tecnologie ci sono risultate utili per quanto riguarda la costruzione e il rendering del documento. HTML5 in particolare ha anche introdotto il local storage che abbiamo utilizzato nel progetto per salvare alcune informazioni.

3) Javascript: abbiamo deciso di utilizzare javascript come linguaggio di scripting per gestire eventi quali gli alert dei sensori, la visualizzazione della mappa con i marker e il grafico che mostra i valori degli inquinanti. E' stato possibile implementare la mappa e il grafico tramite delle API javascript fornite da google.

4) Apache Tomcat: Web server che implementa le specifiche della JSP e per eseguire l'applicativo web.

5) MYSQL: Software per la gestione delle basi di dati in sostituzione ad Oracle per problemi di connessione sul sistema operativo.

Modifiche alla sitemap



Discussione generale sul sistema implementato

Il sistema funziona nel seguente modo: L'utente accede al sistema e può scegliere di autenticarsi, registrarsi o fare un login veloce.

Se l'utente si registra, le credenziali vengono salvate nel database tramite il controller. Se si accede con credenziali ne viene tenuta traccia e viene controllato se sono inserite nel modo corretto. Il controller fa anche in modo tale che se si vuole accedere senza credenziali viene generata una pagina con un messaggio (http 401). Se l'utente decide di fare un fast login, non viene salvato nulla riguardo la sessione, ma solo la località che ha scelto.

Il primo controller inoltra la richiesta alla seconda JSP che viene gestita dal controller2, quest'ultimo può accedere al database e fornisce le informazioni riguardo gli ultimi valori degli inquinanti e la località dell'utente mostrata su una mappa.

L'utente a questo punto, può scegliere di calcolare medie temporali su una località o su una regione, oppure può cambiare location o fare il logout.


La seconda JSP fa in modo tale che tramite uno script manda un alert nel caso in cui i valori visualizzati, superano la soglia di 0.9.

Home

Log InRegistrati

oppure

Accedi Ora



Statistiche Inquinamento

Questo è un sistema per il controllo di inquinanti atmosferici, le cui misurazioni sono generate ad intervalli regolari di 6 ore da sensori disposti sul territorio. Sono a vostra disposizione medie temporali per specifiche località di proprio interesse, o medie su territori più ampi. I nostri sensori sono distribuiti sull'intero territorio nazionale, disposti sulle principali province italiane.

©Copyright della AtomByte Corporation di Gianluca De Lucia e Andrea Capone

Sensori:



Località: **Napoli**

Cambia LocalitàEsci

Benvenuto Guest !

Calcola:

Media LocaleMedia Regionale

Ultimo aggiornamento alle ore 16:41

Statistiche inquinanti sulla località:

| N.º | Inquinante | (%) | Max/Min |
|-----|--|--------|---------|
| 1. | Biossido di Zolfo (SO ₂) | 0.713% | inf/inf |
| 2. | Biossido di Azoto (NO ₂) | 0.001% | inf/inf |
| 3. | Particolato Atmosferico (PM) | 0.864% | inf/inf |
| 4. | Benzene (C ₆ H ₆) | 0.316% | inf/inf |
| 5. | Idrocarburi Policiclici Aromatici (IPA) | 0.991% | inf/inf |
| 6. | Piombo (Pb) | 0.006% | inf/inf |
| 7. | Nichel (Ni) | 0.057% | inf/inf |
| 8. | Cadmio (Cd) | 0.267% | inf/inf |
| 9. | Ozono (O ₃) | 0.162% | inf/inf |
| 10. | Monossido di Carbonio (CO) | 0.011% | inf/inf |

Tasso di inquinamento



Percentuale (%)

Elementi

Funzionalità implementate con breve descrizione

Lista dei metodi principali di ogni punto del progetto:

Funzioni e Script Javascript:

drawAnnotations(): è una funzione di google maps che permette di mostrare i valori degli inquinanti sul grafico.

myfuncion(): è una funzione che genera un alert con l'elenco dei valori che superano la soglia.

myMap(): è una funzione di google maps che permette di mostrare le mappe di google e posizionarvi i marker data latitudine e longitudine.

Abbiamo utilizzato uno script javascript per salvare in local storage delle stringhe contenenti il nome dell'utente e la location che ha scelto.

Metodi Servlet:

doPost(): per entrambe le servlet per elaborare le diverse richieste http

Eventi e Script MYSQL:

Genera: per generare i valori casuali periodicamente.

Incrementa: inserisce un numero progressivo nella tabella elemscanner del database, la quale contiene i valori degli inquinanti. Questo numero serve per distinguere le diverse scansioni dei sensori (quella con il numero più alto corrisponde all'ultima).

Insert, Create, Delete, Event: Script per creare, riempire ed eliminare il database.

Metodi DAO:

insertUser: Per inserire un utente alla registrazione;

changeLoc: Per cambiare la location all'utente che sta in quella sessione;

getLastValues: Per prendere i valori dell'ultima scansione sulla località della sessione;

getMin e getMax: servono per prelevare rispettivamente min e max di ogni inquinante scansionato in un intervallo temporale;

calculateLocalMedia e calculateRegionalMedia: sono due funzioni che calcolano media locale e media regionale in un intervallo temporale;

getLatitude e getLongitude: Sono due funzioni che restituiscono le coordinate del marker.

Tecniche di session tracking utilizzate

1) Local storage: Abbiamo utilizzato il local storage per memorizzare la location dell'utente sulla pagina.

2) Session Tracking per autenticazione: abbiamo implementato il session tracking mediante autenticazione con le servlet per impedire accessi senza credenziali o non autorizzati.

3) Session Tracking tramite campi nascosti dove passiamo nelle form della pagina 2, mediante degli input nascosti, l'utente attuale e la location su cui si trova:

```
<input id ="Slocation" name="l" value="<%= request.getAttribute("location") %>" type ="hidden">
<input id ="Slocation" name="u" value="<%= request.getAttribute("User") %>" type ="hidden">
```

Discussione finale

Questa applicazione risulta essere un prototipo basato su valori generati random all'interno del nostro DataBase, pronto nel caso ad essere venduto.

Il tempo di sviluppo è stato di circa un mese, grazie al valido sussidio degli appunti della professoressa Anna Corazza che hanno reso possibile una facile e mirata ricerca sul web per i linguaggi e gli strumenti tutti utilizzati.

Tra le principali fonti si riportano:

- <http://wpage.unina.it/anna.corazza/>

- <https://www.w3.org/>

- <https://scholar.google.it/>

Si consiglia uso di Google Chrome per l'utilizzo dell'applicazione.

Per avviare in localhost utilizzare Netbeans insieme a TomCat e al file .jar per la connessione al Database.

Come DataBase utilizzare Mysql, creare un database TECWEB e popolarlo attraverso gli script in allegato nella cartella Database. Poi creare un nuovo utente (username: tecweb; password: tecweb).