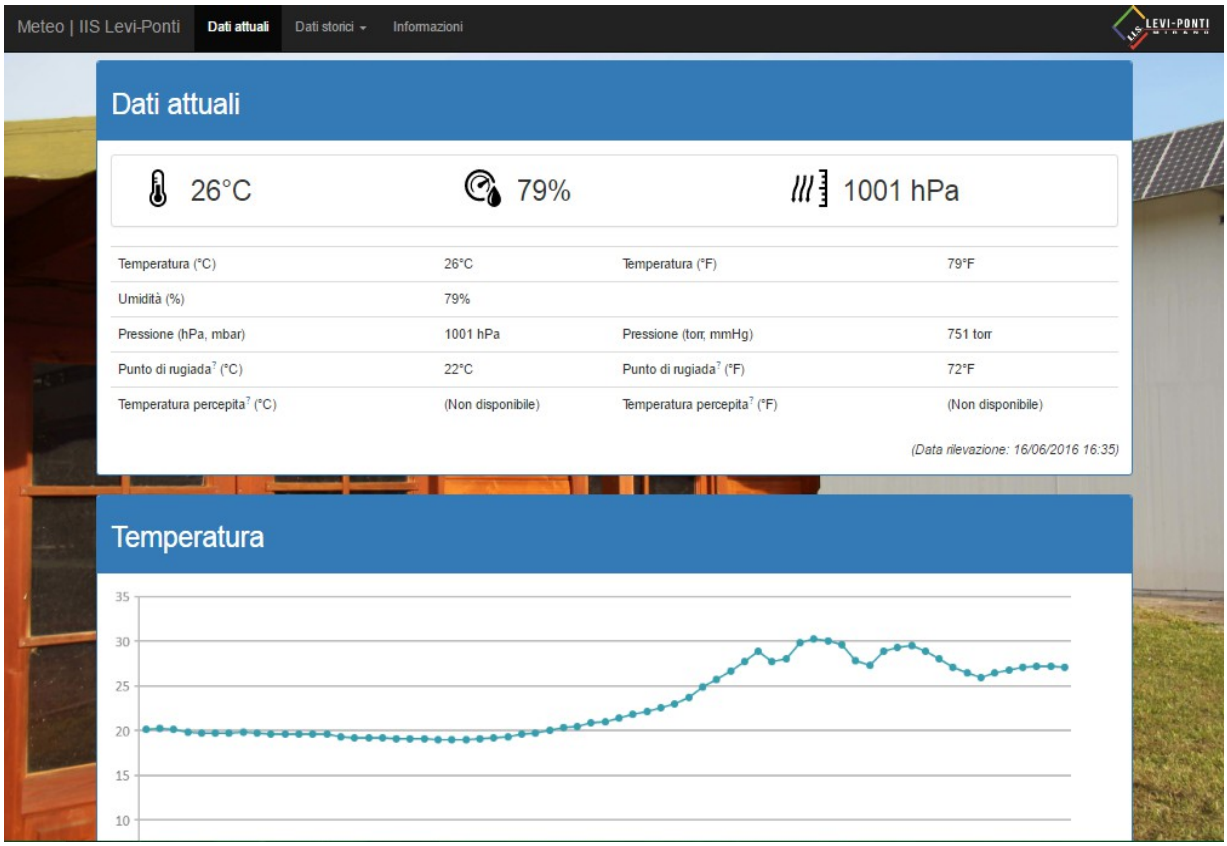
STAZIONE METEOROLOGICA

Documento di progettazione



Samuele Casarin

Classe 5^BIN

A.S. 2015/2016

Istituto di Istruzione Superiore P. LEVI-G.PONTI

Indice generale

1 OBIETTIVO....4
1.1 Sommario....4
1.2 Descrizione....4
2 PROGETTAZIONE DEL SISTEMA....5
2.1 Struttura logica....5
2.2 Web UI....5
2.3 Web Server e Web Files....6
2.4 Services....6
2.5 Data Fetching Service....6
2.6 Data Insertion Service....7
2.7 Notification Service....7
2.8 Historical Data....7
2.9 Current Data....7
2.10 Weather Station....8
2.11 Notification Mail Server....8
3 PROGETTAZIONE DEL DATABASE....9
3.1 Modello concettuale....9
3.2 Modello logico....9
3.3 Passaggio dei dati attuali a dati storici....10
3.4 La vista mlp_last_data....12
4 PROGETTAZIONE DELL'INTERFACCIA WEB....13
4.1 Prototipo....13
4.2 Dati attuali....14
4.3 Dati storici....14
4.4 Aspetti grafici....14
4.5 Aggiornamento dei dati....15
4.6 Collaudo....16
5 PROGETTAZIONE DEI SERVIZI....17
5.1 Data Fetching Service....17
5.2 Current Data Fetching Service....17
5.3 Historical Data Fetching Service....19
5.4 Data Insertion Service....20
5.5 Notification Service....21
6 PROGETTAZIONE DELLA STAZIONE METEOROLOGICA....23
6.1 Componenti hardware/software....23
6.2 Modello del circuito....24
6.3 Fase di rilevazione....25
6.4 Collaudo e installazione....26
7 POSSIBLE DEVELOPMENTS....27
8 FONTI....28

1 OBIETTIVO

In questa sezione è descritto l'obiettivo principale del progetto, che comprende tra l'altro i prodotti finali da realizzare e lo scopo del progetto.

1.1 Sommario

L'Istituto di Istruzione Superiore Levi-Ponti, a scopo informativo, propone la realizzazione di una stazione meteorologica semplificata per il campionamento di alcuni tra i principali tipi di dati relativi al campo della meteorologia (temperatura, umidità e pressione atmosferica) e la realizzazione di un'interfaccia web, accessibile via Internet, per la visualizzazione dei dati in tempo reale e il loro storico.

1.2 Descrizione

Ogni minuto, la stazione meteorologica effettua il campionamento dei dati di temperatura, umidità e pressione e li inserisce nella base di dati.

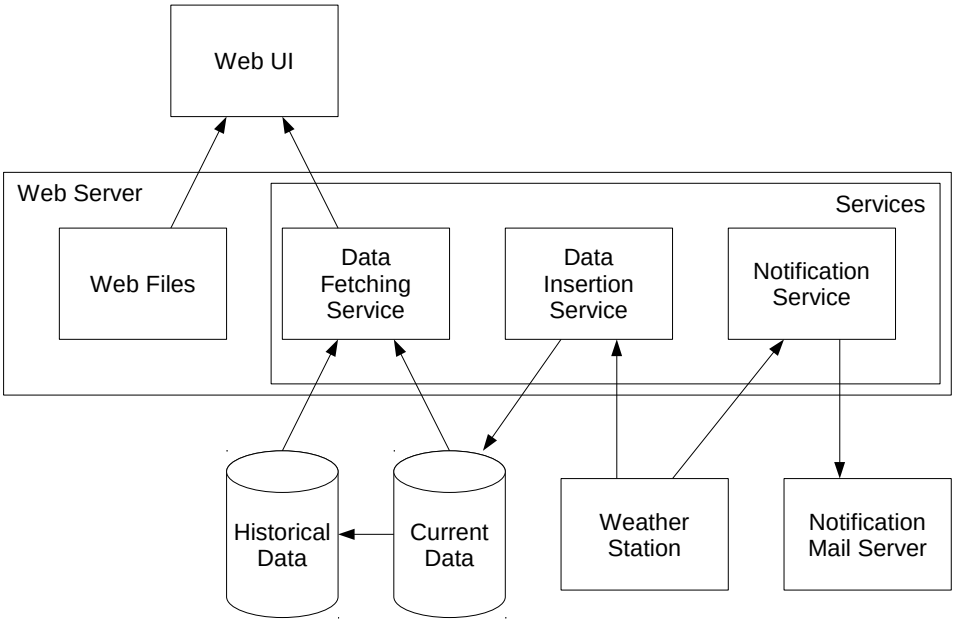
L'interfaccia web, di libero accesso via Internet, dà la possibilità di consultare due diverse sezioni:

- *Dati attuali*, che evidenzia gli ultimi dati rilevati e altre informazioni derivanti da tali dati;
- *Dati storici*, a sua volta divisa in tre sottosezioni corrispondenti ai tre tipi di dati rilevati, *Temperatura*, *Umidità* e *Pressione*, nelle quali sono rappresentati i rispettivi grafici dell'andamento in un dato intervallo di tempo, inizialmente giornaliero, selezionabile dall'utente.

2 PROGETTAZIONE DEL SISTEMA

In questo paragrafo sono riportate la struttura logica e la selezione dei componenti hardware/software impiegati per la realizzazione dei vari strati del sistema.

2.1 Struttura logica



2.2 Web UI

L'interfaccia web visibile al pubblico (Web UI) consiste in un documento HTML che, dopo il caricamento iniziale e successivamente ad intervalli di tempo definiti, effettua delle richieste di aggiornamento dei dati da visualizzare al Web Server.

La pagina si suppone essere compatibile e accessibile con tutti i browser, comprese le release per dispositivi mobili, che supportano una versione recente di JavaScript.

2.3 Web Server e Web Files

Il Web Server è la parte del sistema nella quale è allocato il sito web che comprende, in particolare, le risorse necessarie alla composizione della Web UI (Web Files).

La scelta di utilizzo di un hosting a pagamento, anziché di un server interno, è derivata principalmente dalle seguenti constatazioni:

- disponibilità del servizio da parte dell'Istituto;
- manutenzione hardware affidata al fornitore;
- continuità del servizio;
- possibilità di registrazione di un nome di dominio.

L'unico svantaggio della scelta potrebbe rilevarsi nel fatto che i servizi inclusi in un hosting sono limitati dal fornitore, ma in questo caso è stato ritenuto un limite accettabile, poiché gli unici servizi necessari alla realizzazione del progetto sono un server web, un database e una casella di posta elettronica, solitamente inclusi in un hosting gratuito[1].

2.4 Services

I Services sono una sottosezione del sito web che fornisce i servizi utilizzati per permettere l'interazione con il database e con la stazione meteorologica.

È da notare che tali servizi non sono dei veri e propri web services, ma una collezione di script PHP che si comportano come tali. In questo modo, il web server riesce a fornire in modo semplice tutte le funzionalità che il sistema richiede senza la necessità di ulteriori installazioni hardware/software, dato che la richiesta del servizio e lo stato della risposta sono incluse nel protocollo HTTP, garantendo così una facile portabilità dell'applicazione[2].

Inoltre, il loro utilizzo assicura un maggior controllo dei dati in transito.

2.5 Data Fetching Service

Si tratta del servizio per il recupero dei dati dal database, sia attuali che storici; viene consumato dalla Web UI per aggiornare i dati da visualizzare.

2.6 Data Insertion Service

È il servizio per l'inserimento dei dati acquisiti dalla stazione meteorologica nel database.

Si occupa anche di richiamare, contestualmente all'inserimento, la procedura del database per la gestione del passaggio dei dati attuali a dati storici (vedi sezione 3.3).

2.7 Notification Service

È il servizio per l'invio delle email di notifica dello stato della stazione meteorologica; tali messaggi informano l'amministratore riguardo un riavvio del sistema della stazione, un errore verificatosi a uno dei suoi componenti e la sua avvenuta risoluzione.

2.8 Historical Data

Si tratta della tabella del database che archivia i dati acquisiti in passato dalla stazione meteorologica.

È stato scelto di utilizzare un database MySQL primariamente per il fatto che l'hosting fornisce tale servizio.

Sono state prese alcune misure per garantire la conservazione delle prestazioni anche dopo anni di costanti inserimenti (vedi sezioni 3.3 e 3.4).

2.9 Current Data

È la tabella del database che contiene i dati acquisiti di recente dalla stazione meteorologica, inseriti attraverso il Data Insertion Service.

2.10 Weather Station

Si tratta della stazione meteorologica, collocata all'interno dell'Istituto, a cui si collegano i sensori per l'acquisizione dei dati meteorologici, posti all'esterno dell'Istituto, esposti all'atmosfera, ma protetti da eventuali agenti danneggianti. Ogni minuto inserisce i dati da essa acquisiti nel database attraverso il Data Insertion Service e invia una email di notifica in caso di particolari eventi, tra cui errori persistenti da parte dei componenti, appoggiandosi al Notification Service.

Considerato il basso costo, il basso consumo e la semplicità di programmazione, è stato scelto di utilizzare come hardware della stazione meteorologica Arduino Uno, una scheda elettronica di piccole dimensioni, utile per creare rapidamente prototipi per scopi hobbistici, didattici e professionali[3]. La scelta degli altri componenti verrà descritta in dettaglio nella sezione 6.l.

2.11 Notification Mail Server

Trattasi del server di spedizione e ricezione delle email di notifica dello stato della stazione meteorologica.

È stato scelto di utilizzare un sistema di notifica via email perché l'hosting fornisce tale servizio e, in più, la funzionalità IDLE (RFC 2177) permette agli utenti email IMAP di ricevere immediatamente cambiamenti della casella di posta elettronica senza la necessità di intraprendere azioni, come cliccare su un pulsante di aggiornamento o far intervenire il client email ripetutamente per verificare la presenza di nuovi messaggi sul server[4].

3 PROGETTAZIONE DEL DATABASE

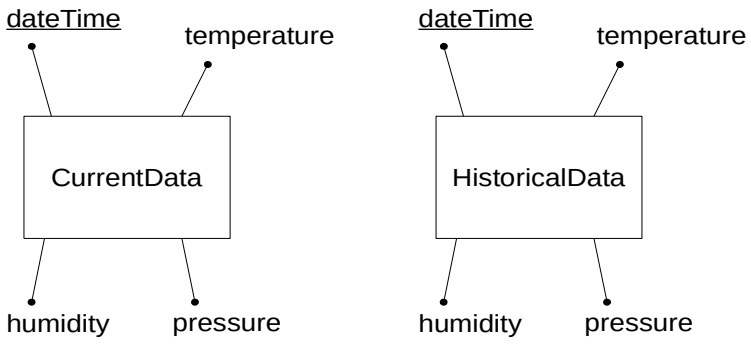
Nella seguente sezione sono riportati i modelli concettuale e logico della base di dati, la procedura che consente il passaggio dei dati attuali a dati storici e le misure adottate per la conservazione delle prestazioni.

3.1 Modello concettuale

La base di dati deve collezionare i dati delle rilevazioni recenti e, dopo un periodo prestabilito, archiviare questi ultimi come rilevazioni storiche, in modo che sia possibile costruire un grafico dell'andamento per ogni tipo di dati.

Si possono, quindi, individuare due entità, CurrentData e HistoricalData, come già previsto dalla progettazione del sistema nella sezione 2, non associate in alcun modo.

Le due entità hanno uguali attributi, la chiave primaria è formata dalla data e dall'ora della rilevazione, poiché non possono esistere due rilevazioni con dati diversi nello stesso istante.



3.2 Modello logico

CurrentData (<u>dateTime</u>, temperature, humidity, pressure)

HistoricalData (<u>dateTime</u>, temperature, humidity, pressure)

CurrentData	
PK	dateTime: DATETIME
	temperature: DECIMAL(4, 2) humidity: DECIMAL(5, 2) pressure: DECIMAL(6, 2)

HistoricalData	
PK	dateTime: DATETIME
	temperature: DECIMAL(4, 2) humidity: DECIMAL(5, 2) pressure: DECIMAL(6, 2)

Ogni componente della stazione meteorologica effettua delle misurazioni con precisione centesimale, quindi con due cifre decimali[5][6].

Il numero massimo delle cifre necessarie per rappresentare un qualsiasi valore intero della temperatura è due, per l'umidità è tre (il massimo teorico è 100%), per la pressione è quattro (la più bassa pressione atmosferica misurata è 870 hPa, mentre la più alta pressione atmosferica misurata è 1085.6 hPa[7]; in ogni caso, si trattano di casi eccezionali).

3.3 Passaggio dei dati attuali a dati storici

Una delle misure più importanti di conservazione delle prestazioni del database consiste nella separazione tra dati attuali e dati storici. In questo modo, infatti, la selezione delle due tipologie di dati diventa pressoché istantanea, in particolare su un server computer, progettato per operare su grandi quantità di dati.

Inizialmente era stato progettato di non separare le due tipologie di dati, ma mantenere tutte le rilevazioni provenienti dalla stazione meteorologica in un'unica tabella divisa in varie partizioni, in modo tale che, nel momento della selezione, venissero considerate solamente le partizioni necessarie all'ottenimento del risultato; sussisteva, però, il problema che le partizioni dovevano essere definite manualmente e vi era un limite al numero delle partizioni, quindi è stato necessario pervenire ad una soluzione alternativa.

Normalmente, la stazione meteorologica inserisce i dati che rileva una volta al minuto; poiché da un minuto all'altro tali dati non subiscono variazioni vertiginose, è stato deciso di inserire nella tabella dei dati storici ogni raggruppamento dei dati attuali “per quarto d'ora” (cioè per i minuti 0-14, 15-29, 30-44 e 45-59). In questo modo, il numero di rilevazioni presenti nella tabella dei dati storici è 15 volte inferiore rispetto al mantenimento di tutte le rilevazioni per minuto e, di conseguenza, la velocità di selezione delle rilevazioni storiche aumenta di 15 volte.

Per effettuare tale passaggio con raggruppamento si è ricorsi alla definizione di una Stored Procedure, più sotto riportata in forma di pseudocodice.

NB: Il prefisso “mlp” (acronimo di Meteo Levi-Ponti) è stato aggiunto al nome di tutte le tabelle e delle altre funzionalità nel caso non fosse possibile crearle in un database ad-hoc.

```
PROCEDURA `mlp_commit`
INIZIO
	R1 <= Seleziona da `mlp_current_data` tutte le rilevazioni che non
	appartengono all'ultimo quarto d'ora raggruppandole per quarto d'ora e
	calcolando la media di temperatura, umidità e pressione per ogni
	raggruppamento.
	PER OGNI RIGA r IN R1
		Inserisci r in `mlp_historical_data` considerando il quarto d'ora
		successivo.
	RIPETI
	Elimina da `mlp_current_data` tutte le rilevazioni che non appartengono
	all'ultimo quarto d'ora.
FINE
```

Nella fase di inserimento si considera il quarto d'ora successivo a quello approssimato per difetto perché, in questo modo, è più evidente che il valore rappresentato sul grafico corrisponde al risultato delle variazioni intervenute sui dati nell'ultimo quarto d'ora.

Qui sotto è riportata, in particolare, la corrispondente traduzione in linguaggio MySQL della selezione:

```
SELECT FROM_UNIXTIME(UNIX_TIMESTAMP(`dateTime`) - UNIX_TIMESTAMP(`dateTime`) %
900) `dateTimeDivision`, AVG(`temperature`), AVG(`humidity`), AVG(`pressure`)
FROM `mlp_current_data`
WHERE TIMESTAMPDIFF(MINUTE, FROM_UNIXTIME(UNIX_TIMESTAMP(`dateTime`) -
UNIX_TIMESTAMP(`dateTime`) % 900), NOW()) >= 15
GROUP BY `dateTimeDivision`;
```

NB: Durante i trasferimenti e le selezioni, per evitare conflittualità tra i formati delle date, queste sono codificate nel timestamp di UNIX, cioè in un numero intero che indica i secondi passati in un dato istante a partire dalla mezzanotte del 1° gennaio 1970[8].

3.4 La vista *mlp_last_data*

Un altro accorgimento volto a conservare la capacità del database di selezionare i dati attuali in un tempo accettabile è stato quello di utilizzare la vista *mlp_last_data*.

Tale vista riporta gli ultimi dati inseriti da *mlp_current_data* e *mlp_historical_data* e si aggiorna ogni qualvolta si effettua una modifica in una delle due tabelle. In questo modo, non sarà necessario che il Data Fetching Service analizzi continuamente le notevoli quantità di dati (si parla di inserimenti costanti protratti per un lungo periodo di tempo) delle due tabelle per ottenere i dati attuali, che in questo modo corrisponderanno sempre all'ultima rilevazione inserita nelle due tabelle (il Data Fetching Service selezionerà il valore massimo tra la data della riga riportata da *mlp_current_data* e quella riportata da *mlp_historical_data*; quest'ultimo caso, più raro del primo, normalmente si verifica nel momento in cui non vi sono dati presenti nella prima tabella in seguito ad un'operazione di passaggio dei dati).

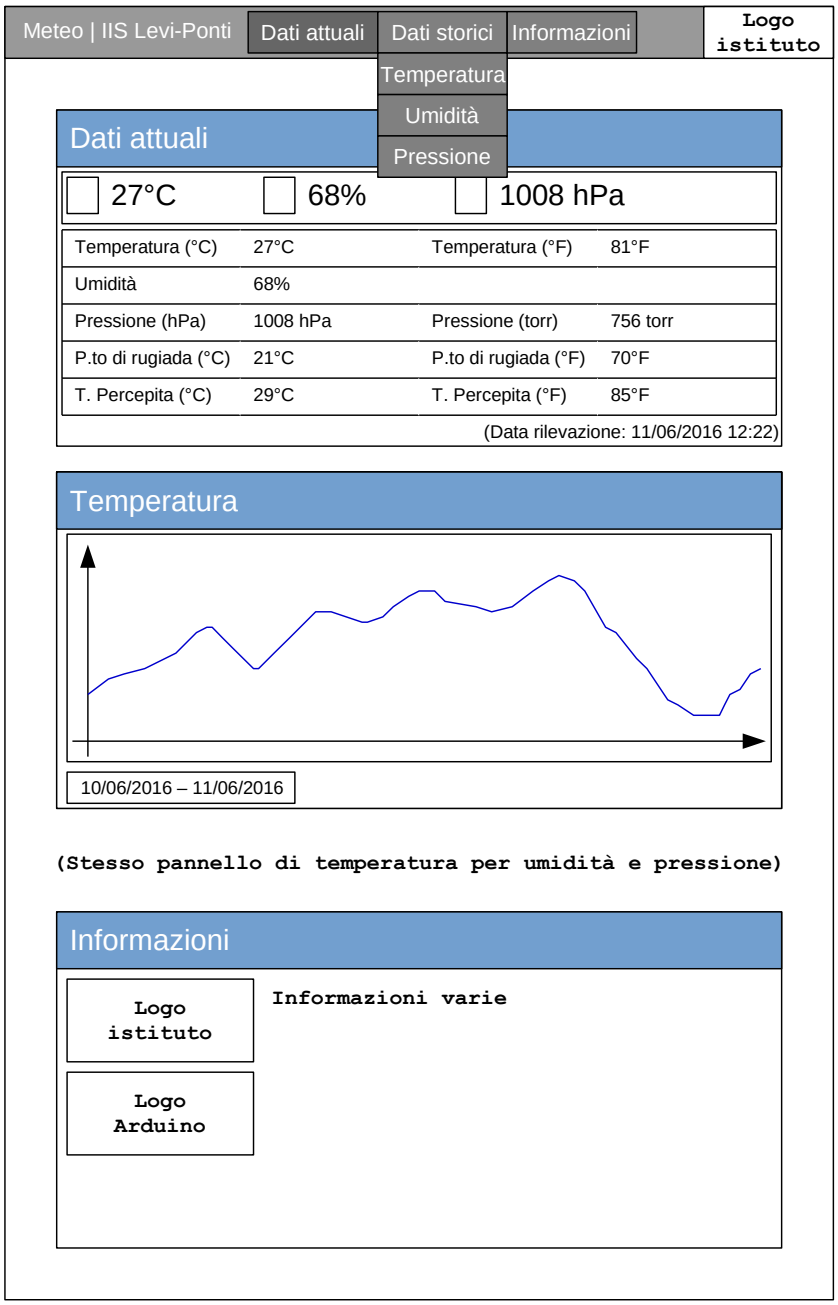
Il linguaggio MySQL la vista viene definita nel seguente modo:

```
CREATE VIEW `mlp_last_data` AS
SELECT *
FROM `mlp_data`
WHERE `dateTime` = (
       SELECT MAX(`dateTime`)
       FROM `mlp_data`
)
UNION
SELECT *
FROM `mlp_arduino_data`
WHERE `dateTime` = (
       SELECT MAX(`dateTime`)
       FROM `mlp_arduino_data`
);
```

4 PROGETTAZIONE DELL'INTERFACCIA WEB

Nella seguente sezione sono riportati il prototipo dell'interfaccia web, la tecnica adottata per l'aggiornamento programmato del suo contenuto e la procedura di collaudo.

4.1 Prototipo



4.2 *Dati attuali*

L'interfaccia web, come citato nella sezione 1.2, si suddivide fondamentalmente di due parti.

La prima, denominata *Dati attuali*, è composta da un unico pannello, nel quale sono evidenziati gli ultimi dati rilevati e una tabella contenente altre informazioni derivanti da essi, tra cui la conversione dell'unità di misura dei dati in un'altra unità di misura, anche differente da quella adottata nel Sistema Internazionale, e dati ricavabili dalla relazione di dipendenza tra almeno due dei dati rilevati.

Questi aspetti verranno trattati in dettaglio nella sezione 5.2.

4.3 *Dati storici*

Successivamente, vi sono tre pannelli posti in sequenza al cui interno vi è un grafico che traccia l'andamento nel tempo delle tre tipologie di dati considerate, *Temperatura*, *Umidità* e *Pressione*.

Azionando un pulsante posto sotto il grafico, l'utente può selezionare da un calendario l'intervallo delle date di cui il grafico deve tracciare l'andamento dei dati.

Inizialmente, l'intervallo considerato è quello del giorno corrente.

Dato che, se venisse selezionato un intervallo di tempo troppo esteso, il grafico diventerebbe troppo fitto di punti, è stato deciso di limitare la selezione ad un intervallo massimo di sette giorni (intercorrenti dal giorno della prima rilevazione al giorno dell'ultima rilevazione effettuata).

Infine, se l'intervallo selezionato coincide con la data dell'ultimo giorno nel quale sono disponibili dei dati (come l'intervallo iniziale) ed esistono nuove rilevazioni effettuate in una data successiva alla suddetta, l'intervallo verrà automaticamente aggiornato a quest'ultima data.

4.4 Aspetti grafici

La pagina è stata progettata per essere accessibile e facilmente utilizzabile da chiunque (user-friendly), in modo che rispetti il più possibile le WCAG[9] nella loro versione più recente (2.0).

Al fine di ottenere un'interfaccia grafica accattivante, ma allo stesso tempo semplice e veloce da realizzare, è stato deciso di utilizzare il framework *Bootstrap*[10], che fornisce una serie di soluzioni efficaci per rendere il sito responsivo, cioè dotato della capacità di adattarsi graficamente in modo automatico al dispositivo con la quale viene visualizzato[11].

In particolare, la barra di navigazione è stata realizzata con il componente *navbar* con la funzionalità aggiunta *scrollspy* per aggiornare automaticamente il collegamento selezionato nella lista di navigazione in base alla posizione di scorrimento della pagina.

Il calendario è stato ottenuto per mezzo di un componente esterno, *Date Range Picker*[12], che utilizza *Bootstrap* come base grafica e la libreria *Moment.JS*[13] per il parsing delle date, e, infine, il grafico è stato ottenuto con un ulteriore componente esterno, *CanvasJS*[14].

Per ridurre il carico di lavoro del server che ospita il sito, tutti i fogli di stile e gli script dei componenti aggiuntivi sono scaricati dal browser da varie CDN ("Content Delivery Network", o CDN, viene definito come un sistema di computer collegati in rete attraverso Internet, che collaborano in maniera trasparente per distribuire contenuti[15]).

4.5 Aggiornamento dei dati

L'interfaccia deve aggiornare periodicamente in modo automatico i dati che visualizza al fine di non presentare dati attuali che, dopo un certo tempo, diventano passati.

Per realizzare questa funzionalità è stato fatto ricorso alla tecnica di sviluppo software nota con l'acronimo AJAX (Asyncronous Javascript And XML). Lo sviluppo di applicazioni HTML con AJAX si basa su uno scambio di dati in background fra web browser e server, che consente l'aggiornamento dinamico di una pagina web senza l'esplicito ricaricamento da parte dell'utente[16].

L'aggiornamento viene effettuato, in primo luogo, dopo il caricamento della pagina, in seguito ogni 5 minuti; in caso di errore nell'elaborazione della richiesta, un nuovo tentativo viene effettuato automaticamente un minuto dopo.

Quando l'utente seleziona da uno dei calendari un nuovo intervallo di date, l'aggiornamento dei dati viene effettuato in maniera analoga e il tempo di attesa per un nuovo aggiornamento ritorna ad essere 5 minuti.

Se uno dei grafici sta visualizzando i dati rilevati lo stesso giorno dell'ultima rilevazione disponibile, nel momento in cui diventano disponibili altri dati rilevati in un giorno successivo, il grafico effettuerà automaticamente il passaggio alla visualizzazione dei dati rilevati in quest'ultimo giorno. Nella normale funzionalità, il passaggio avviene tra un giorno e il suo successivo.

A livello tecnico, l'aggiornamento deve rispettare i vincoli imposti dal servizio di recupero dati, che comprendono i dati richiesti in input, quelli restituiti in output e il formato nel quale sono organizzati. Tali vincoli sono descritti in dettaglio nella sezione 5.

4.6 Collaudo

Prima di effettuare il caricamento dei file nell'hosting, è stato deciso di effettuare il collaudo dell'interfaccia web utilizzando un servizio minimale installato in locale, *XAMPP*, che ha consentito di effettuare tutti i collaudi necessari senza l'utilizzo del servizio di hosting.

Per collaudare l'interfaccia web, è necessario che il Data Fetching Service sia pronto e funzionante e che il database contenga almeno dei dati di prova.

I test che devono essere effettuati consistono nel controllare che i dati, sia attuali che storici, vengano recuperati attraverso il Data Fetching Service con successo e correttamente presentati nell'interfaccia nei seguenti casi:

- al caricamento della pagina;
- dopo ogni intervallo di tempo*;
- dopo un errore*;
- quando l'utente cambia un intervallo di date in uno dei calendari (solo per i dati storici).

* Per questi test, è stato diminuito il tempo di ogni intervallo a 30 secondi al fine di ridurre i tempi di osservazione.

5 PROGETTAZIONE DEI SERVIZI

In questa sezione sono riportati i servizi del sistema, i vincoli da loro richiesti per il corretto utilizzo e la procedura di collaudo per ognuno di essi.

5.1 Data Fetching Service

Come già definito nella sezione 2, il Data Fetching Service è il servizio per il recupero dei dati dal database che viene consumato dalla Web UI per aggiornare i dati da visualizzare.

Si divide logicamente in due servizi sottostanti: Current Data Fetching Service e Historical Data Fetching Service.

5.2 Current Data Fetching Service

Si tratta del servizio per il recupero dei dati attuali.

Metodo HTTP	GET, POST
URL	/ajax/getCurrentData.php
Formato dati di input	-
Formato dati di output	JSON

Dati di input

Nessun dato di input richiesto.

Dati di output

timestamp	Intero	Il timestamp di UNIX dell'istante della rilevazione
temperature	Stringa	La temperatura in gradi Celsius all'istante della rilevazione con l'unità di misura specificata
humidity	Stringa	La percentuale di umidità relativa...
pressure	Stringa	La pressione in ettopascal (alias millibar)...
temperature1	Stringa	Come current-temperature

temperature2	Stringa	La temperatura in gradi Fahrenheit all'istante della rilevazione con l'unità di misura specificata
humidity1	Stringa	Come current-humidity
pressure1	Stringa	Come current-pressure
pressure2	Stringa	La pressione in torr (alias millimetri di mercurio) all'istante della rilevazione con l'unità di misura specificata
dewPoint1	Stringa	Il punto di rugiada in gradi Celsius... Il punto di rugiada è la temperatura alla quale, a pressione costante, l'aria diventa satura di vapore acqueo. Esso indica a che temperatura deve essere portata l'aria per far condensare in rugiada il vapore d'acqua in essa presente, senza alcun cambiamento di pressione[17]. Tale informazione viene calcolata in funzione della temperatura e dell'umidità relativa con la seguente formula (approssimazione di Magnus-Tetens): $T_d = \frac{(237{,}7 \cdot T)}{(17{,}271 - T)}$ dove $T = \frac{(17{,}271 \cdot T_C)}{(237{,}7 + T_C)} + \log(RH \cdot 0{,}01)$ e dove, a sua volta, T_C è la temperatura in gradi Celsius e RH la percentuale di umidità relativa.
dewPoint2	Stringa	Il punto di rugiada in gradi Fahrenheit...
heatIndex1	Stringa	L'indice di calore (o temperatura percepita) in gradi Celsius... L'indice di calore è la sensazione di 'caldo' o di 'freddo' che viene avvertita[18]. Tale informazione viene calcolata in funzione della temperatura e dell'umidità relativa con la seguente formula: $HI = -42{,}379 + 2{,}04901523 \cdot T_F + 10{,}14333127 \cdot RH$ $- 0{,}22475541 \cdot T_F \cdot RH - 0{,}00683783 \cdot T_F^{\,2}$ $- 0{,}05481717 \cdot RH^2 + 0{,}00122874 \cdot T_F^{\,2} \cdot RH$ $+ 0{,}00085282 \cdot T_F \cdot RH^2 - 0{,}00000199 \cdot T_F^{\,2} \cdot RH^2$ dove T_F è la temperatura in gradi Fahrenheit e RH la percentuale di umidità relativa. Tuttavia, la formula è valida solo per $T_F \geq 80°F$ e $40\% \leq RH \leq 100\%$. Se i valori di input non soddisfano i suddetti vincoli, il valore di output sarà *null*.
heatIndex2	Stringa	L'indice di calore in gradi Fahrenheit...

Collaudo

Per collaudare questo servizio, è necessario controllare che i dati selezionati dal database siano effettivamente gli ultimi rilevati in assoluto e che le altre informazioni derivate vengano restituite in modo corretto, confrontandole attraverso l'utilizzo di servizi presenti in Internet.

5.3 Historical Data Fetching Service

Si tratta del servizio per il recupero dei dati storici.

Metodo HTTP	GET, POST
URL	/ajax/getHistoricalData.php
Formato dati di input	Query string
Formato dati di output	JSON

Dati di input

target	Stringa	**Obbligatorio**. Il tipo di dato di cui il servizio deve restituire la lista di rilevazioni. I possibili valori che può assumere sono: “temperature”, “humidity” o “pressure”.
startTimestamp	Intero	*Facoltativo*. Il timestamp di UNIX che indica il minimo istante temporale delle rilevazioni da considerare nella ricerca. Se non specificato, il valore predefinito corrisponde al timestamp di UNIX che indica la mezzanotte del giorno dell'ultima rilevazione disponibile.
endTimestamp	Intero	*Facoltativo*. Il timestamp di UNIX che indica il massimo istante temporale delle rilevazioni da considerare nella ricerca. Se non specificato, il valore predefinito corrisponde al timestamp di UNIX che indica l'ultimo secondo del giorno dell'ultima rilevazione disponibile.
lastTimestamp	Intero	*Facoltativo*. Il timestamp di UNIX dell'ultima rilevazione inviata all'interfaccia. Conoscendo questo dato, il servizio seleziona solamente i dati appartenenti all'intervallo di tempo specificato dai parametri startTimestamp e endTimestamp e acquisiti dopo l'istante temporale indicato da lastTimestamp. Il valore deve essere compreso tra i valori dei parametri startTimestamp e endTimestamp. Se non specificato, verranno selezionati tutti i dati appartenenti all'intervallo di tempo specificato dai parametri startTimestamp e endTimestamp.

Dati di output

startTimestamp	Intero	Il timestamp di UNIX che indica il minimo istante temporale delle rilevazioni considerato nella ricerca. *NB: Potrebbe non essere uguale a quello specificato nei parametri di input.*
endTimestamp	Intero	Il timestamp di UNIX che indica il massimo istante temporale delle rilevazioni considerato nella ricerca. *NB: Potrebbe non essere uguale a quello specificato nei parametri di input.*
minTimestamp	Intero	Il timestamp di UNIX che indica l'istante temporale della prima rilevazione disponibile.
maxTimestamp	Intero	Il timestamp di UNIX che indica l'istante temporale dell'ultima rilevazione disponibile.
maxDateTimestamp	Intero	Il timestamp di UNIX che indica la mezzanotte del giorno dell'ultima rilevazione disponibile. Questo dato è utile all'interfaccia per effettuare il passaggio automatico al giorno dell'ultima rilevazione disponibile (vedi sezione 4.5).
dataPoints	Lista	La lista di oggetti {x,y} che rappresentano i punti del grafico considerato, in cui x è il timestamp di UNIX che indica l'istante della rilevazione e y è il valore della corrispondente tipologia di dato.

Collaudo

Per collaudare questo servizio, è necessario effettuare controlli che comprendano casi particolari, sia corretti che errati, di valori di input dapprima per i parametri startTimestamp e endTimestamp e successivamente specificando anche il parametro lastTimestamp, verificando per ognuno di essi che le rilevazioni selezionate corrispondano effettivamente a quelle richieste.

5.4 Data Insertion Service

È il servizio per l'inserimento dei dati acquisiti dalla stazione meteorologica nel database.

Si occupa anche di richiamare, all'istante dell'inserimento, la procedura del database per la gestione del passaggio dei dati attuali a dati storici (vedi sezione 3.3).

Si tratta di un servizio privato, cioè che non deve essere accessibile al pubblico per motivi legati alla sicurezza; pertanto, deve essere protetto da una chiave di accesso.

Metodo HTTP	GET, POST
URL	/arduino/insert.php
Formato dati di input	Query string
Formato dati di output	-

Dati di input

key	Stringa	**Obbligatorio**. La chiave di accesso al servizio.
temperature	Reale	**Obbligatorio**. Il valore della temperatura rilevata in gradi Celsius.
humidity	Reale	**Obbligatorio**. Il valore percentuale dell'umidità relativa rilevata.
pressure	Reale	**Obbligatorio**. Il valore della pressione rilevata in ettopascal (alias millibar).

Dati di output

Nessun dato di output restituito.

Collaudo

Per effettuare il collaudo, è necessario accertarsi che i dati rilevati dalla stazione meteorologica vengano puntualmente inseriti nella tabella *mlp_current_data* del database in ogni fase di rilevazione della stazione.

Durante il test, è stato diminuito il tempo di durata della fase a 30 secondi al fine di ridurre i tempi di osservazione.

5.5 Notification Service

É il servizio per l'invio delle email di notifica dello stato della stazione meteorologica; i messaggi informano l'amministratore riguardo un riavvio del sistema della stazione, un errore verificatosi di uno dei suoi componenti e la sua avvenuta risoluzione.

Si tratta di un servizio privato, cioè che non deve essere accessibile dal pubblico per motivi legati alla sicurezza; pertanto, deve essere protetto da una chiave di accesso.

Metodo HTTP	GET, POST
URL	/arduino/notify.php
Formato dati di input	Query string
Formato dati di output	-

Dati di input

key	Stringa	**Obbligatorio**. La chiave di accesso al servizio.
statusId	Intero	**Obbligatorio**. Il numero identificativo del tipo di messaggio di notifica. I possibili valori che può assumere sono: 1: [INFO] Avvio del sistema 2: [ERRORE] Il componente DHT22 ha smesso di funzionare (codice {statusData}) 3: [INFO] Il componente DHT22 ha ripreso a funzionare 4: [ERRORE] Il componente BMP180 ha smesso di funzionare (codice {statusData}) 5: [INFO] Il componente BMP180 ha ripreso a funzionare
statusData	Intero	*Facoltativo*. Codice di stato del componente specificato nel messaggio di notifica.

Dati di output

Nessun dato di output restituito.

Collaudo

Un primo test consiste nell'invio di una email di prova, quindi procedere controllando che il messaggio venga formattato come richiesto dai parametri di input.

6 PROGETTAZIONE DELLA STAZIONE METEOROLOGICA

Nella seguente sezione sono riportati in dettaglio la selezione dei componenti hardware/software destinati alla realizzazione della stazione meteorologica, il modello del circuito, i passaggi compiuti durante una fase di rilevazione e la procedura di collaudo e installazione.

6.1 Componenti hardware/software

Come già detto nella sezione 2, è stato scelto di utilizzare come hardware della stazione meteorologica Arduino Uno Rev3 soprattutto per il suo basso costo, il suo basso consumo e la semplicità di programmazione in linguaggio C.

L'unico software utilizzato è l'IDE (Integrated Development Environment) di Arduino, che include anche il compilatore e il programmatore per tutte le versioni del microcontrollore.

Si è resa necessaria, inoltre, l'acquisizione di altri componenti che costituiscono il cuore della stazione meteorologica. Vengono elencati qui di seguito.

Ethernet Shield 2

La scheda di rete che permette ad Arduino di collegarsi a Internet per inviare i dati e i messaggi di notifica al web server[19].

Questo componente è stato scelto, oltre che per i medesimi motivi del microcontrollore, anche perché dal luogo scelto per l'installazione è possibile connettersi alla rete dell'Istituto scolastico in due modi differenti: via cavo Ethernet oppure via Wi-Fi. Dato, però, che una rete Wi-Fi potrebbe presentare problemi di interferenza e, in generale, una rete cablata è più veloce, è stato deciso di adottare la soluzione di connettività via Ethernet.

DHT22

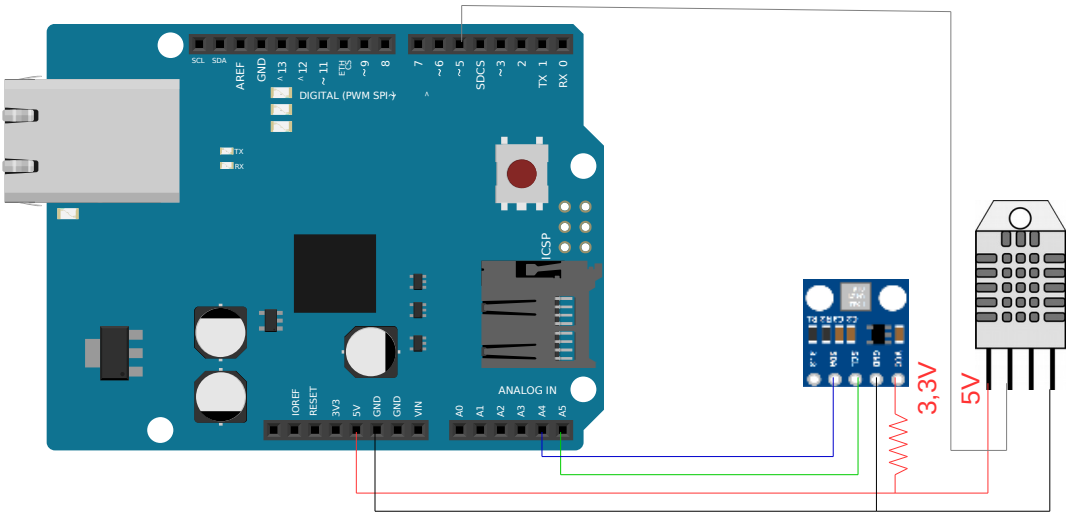
Un sensore di temperatura e umidità digitale a basso costo[5].

La varietà iniziale della scelta era di due componenti: il DHT11 oppure il DHT22. Alla fine è stato scelto il DHT22 perché più preciso e in grado di leggere anche le temperature sottozero, anche se leggermente più costoso (si trattano, comunque, di cifre molto contenute).

BMP180

L'ultimo componente aggiuntivo è il BMP180, un sensore di temperatura e pressione[6], utilizzato, però, solo per misurare quest'ultima.

6.2 Modello del circuito



Nella fase di realizzazione, i due componenti esterni DHT22 e BMP180 sono stati integrati in una scheda di materiale plastico con circuito stampato; è stata inclusa anche la resistenza all'ingresso di tensione del BMP180, che serve a trasformare l'unica tensione in ingresso di 5V nella tensione richiesta dal sensore, che altrimenti si danneggerebbe, di 3,3V (la capacità della resistenza è stata calcolata grazie alla formula del partitore di tensione[20]).

I due componenti utilizzano due sistemi differenti di trasmissione seriale: il componente DHT22 utilizza il protocollo 1-Wire, che richiede un'unica linea per i dati[21], mentre il componente BMP180 utilizza il protocollo I^2C, che utilizza una linea per la sincronizzazione (SCL) e una linea per i dati (SDA)[22].

6.3 Fase di rilevazione

Con fase di rilevazione si intende un periodo temporale eseguito ciclicamente nella quale la stazione meteorologica:

1. effettua le rilevazioni dai sensori;
2. inserisce i dati acquisiti nel database;
3. rimane in attesa fino alla fine del periodo.

È stata stabilita in un minuto la durata di una fase, di conseguenza nel database viene inserita una rilevazione per minuto. In una fase normale, un minuto è più che sufficiente per il completamento della stessa: le rilevazioni da parte dei vari componenti non avvengono in più di 5 secondi e altrettanto per l'inserimento nel database. Tuttavia, anche in una fase in cui si dovessero verificare degli errori, il tempo concesso è più che sufficiente. Questo perché, quando avviene un errore, viene riconosciuto immediatamente dal microcontrollore (per esempio, un collegamento interrotto) oppure per la scadenza di un timeout (come può accadere con un ritardo di trasmissione o ricezione di rete) e, quindi, la fase entra anticipatamente in stato di attesa.

Il consumo dei servizi del sistema destinati alla stazione meteorologica, cioè Data Insertion Service e Notification Service, viene gestito da una funzione della stazione che si occupa di stabilire una connessione con il server, inviare una richiesta HTTP e ricevere la relativa risposta.

Per quanto riguarda l'invio dei messaggi di notifica di errori, questi non vengono inviati immediatamente, dato che potrebbe trattarsi solamente di un problema temporaneo.

Quindi, è stato deciso di impostare due numeri massimi di errori consecutivi tollerati prima dell'invio della segnalazione: *il primo*, impostato a 5 errori consecutivi ammessi, riguarda la prima occorrenza del problema, mentre *il secondo*, impostato a 60 errori consecutivi ammessi (quindi, viene inviata una segnalazione all'ora), tutte le successive se il problema non è stato ancora risolto.

Una volta risolta l'anomalia in seguito alla prima segnalazione, alla fase successiva viene inviato un messaggio di notifica che conferma la rimessa in funzione della stazione meteorologica.

6.4 Collaudo e installazione

Per effettuare il collaudo della stazione meteorologica, è necessario effettuare dei controlli incrementali: il primo test in assoluto è quello per verificare il funzionamento di ogni singolo componente hardware, compreso Arduino, caricando dei programmi esemplificativi, quindi procedere con il controllo dei vari passaggi della fase, sia in via ordinaria che attraverso la simulazione di anomalie (come l'interruzione di un collegamento).

Il collaudo risulta positivo nel momento in cui la stazione meteorologica riesce ad inserire tutti i dati rilevati dai componenti nel database, che tali dati siano stati rilevati correttamente e che vengano ricevuti i messaggi di notifica nelle modalità desiderate al manifestarsi di ciascun evento (l'avvio del sistema, il verificarsi di un errore e la sua risoluzione).

Una volta caricato il programma definitivo, è possibile procedere all'installazione della stazione, cioè al suo piazzamento nel luogo prescelto.

Il microcontrollore deve essere piazzato in un luogo sicuro e riparato, mentre i sensori devono essere esposti all'atmosfera tanto da permettere la rilevazione, ma protetti da eventuali agenti danneggianti o che possono nuocere l'accuratezza della rilevazione stessa.

7 POSSIBLE DEVELOPMENTS

This project is not finished, but just started. This is an open-source project, that is a project in which everyone can modify and improve its source code.

There are some points that may catch some interest:

- **Guess the current meteorological conditions and forecast the future ones**. However, it is a very hard goal, because it requires high levels of knowledge in the field of the meteorology and more professional (and more expensive) instruments than those ones chosen for the project.
- **Develop a smart alarm clock mobile app**, which is a sort of alarm clock that rings some minutes before the hour which the user set when there are bad meteorological conditions. This is a perfect example of a new concept of information called the Internet of Things, or IoT[23].
- **Distribute the project to other schools**. In this way, it is possible to build a map of all the institutes which uses the application and so give it more support and more development possibilities.

However, there are also some small improvements that could be considered about the current version of the project:

- **Get more accurate detections**. Some detections of the weather station are really inaccurate; for instance, at night the percentual of humidity reaches 99.9%, a very improbable value. With a deep study of the issue (and maybe the adoption of more precise components), it is possible get more accurate detections.
- **Get data transfer secure**. When the weather station has to insert data into the database, it asks the Data Insertion Service to perform that request. However, an intruder could intercept one of the messages and use it to break the system. Of course, the service key which is used to impede that everyone could access directly to the service is necessary, but it can't protect the service from an attack like this, that is called “man-in-the-middle attack”[24]. The best solution is encrypt the communication to make sure there are no unexpected eavesdroppers between the weather station and the server.
- **Promote social networking**. Today we talk about Web 2.0. This is a new concept of the Internet in which users can contribute to add web contents and not to remain only visitors[25]. One of the most widespread subcategories is the one of social networks. Then, it is not a bad idea that anyone can comment on the project, giving suggestions to improve it or, simply, leaving a personal opinion.

8 FONTI

[1] https://en.wikipedia.org/wiki/Internet_hosting_service

[2] https://en.wikipedia.org/wiki/Web_service

[3] https://en.wikipedia.org/wiki/Arduino

[4] https://en.wikipedia.org/wiki/IMAP_IDLE

[5] https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf (datasheet DHT22)

[6] https://cdn-shop.adafruit.com/datasheets/BST-BMP180-DS000-09.pdf (datasheet BMP180)

[7] https://en.wikipedia.org/wiki/Atmospheric_pressure

[8] https://en.wikipedia.org/wiki/Unix_time

[9] https://www.w3.org/TR/WCAG20/ (Web Content Accessibility Guidelines 2.0 fornito da W3C)

[10] https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)

http://getbootstrap.com/ (documentazione di Bootstrap)

[11] https://en.wikipedia.org/wiki/Responsive_web_design

[12] http://www.daterangepicker.com/ (documentazione del componente Date Range Picker)

[13] http://momentjs.com/ (documentazione della libreria Moment.js)

[14] http://canvasjs.com/ (documentazione del componente CanvasJS)

[15] https://en.wikipedia.org/wiki/Content_delivery_network

[16] https://en.wikipedia.org/wiki/Ajax_(programming)

[17] https://en.wikipedia.org/wiki/Dew_point

[18] https://en.wikipedia.org/wiki/Heat_index

[19] http://download.arduino.org/products/ETHERNETSHIELD2/arduino-Ethernet-Shield2-V2-sch.pdf (datasheet Arduino Ethernet Shield 2)

[20] https://en.wikipedia.org/wiki/Voltage_divider

[21] https://en.wikipedia.org/wiki/1-Wire

[22] https://en.wikipedia.org/wiki/I%C2%B2C

[23] https://en.wikipedia.org/wiki/Internet_of_things

[24] https://en.wikipedia.org/wiki/Man-in-the-middle_attack

[25] https://en.wikipedia.org/wiki/Web_2.0