

UNIVERSITA' POLITECNICA DELLE MARCHE

FACOLTA' DI INGEGNERIA

Dipartimento di ingegneria dell'informazione

Corso di Laurea in Ingegneria Informatica e dell'Automazione



Tesi di laurea

Gestione in cloud di batterie al litio

Cloud-Based Battery Management for Lithium-Ion Batteries

Relatore

Prof. Massimo Conti

Candidato

Giorgio Olivieri

Anno Accademico 2021-2022

"La logica ti porta da A a B, l'immaginazione ti porta ovunque"

Albert Einstein

INDICE

1	Introduzione.....	1
2	Introduzione al mondo delle applicazioni	3
2.1	Contesto generale.....	3
2.2	Principali sistemi operativi.....	5
2.3	Tipologie di applicazioni e loro sviluppo.....	9
2.3.1	App native.....	9
2.3.2	Web app.....	10
2.3.3	App ibride	17
2.3.4	Progressive web app (PWA)	20
3	Internet of Things (IOT)	21
3.1	Definizione.....	21
3.2	Come nasce Internet of Things.....	21
3.3	Architetture IOT.....	22
3.4	Ambiti Applicativi.....	23
3.5	Rischi, privacy e sicurezza.....	25
4	Piattaforme Cloud.....	26
4.1	Modelli di distribuzione dei servizi cloud.....	26
4.2	Tipologie di servizi offerti.....	28
4.3	Principali piattaforme.....	32
4.3.1	Google Cloud.....	33
4.3.2	Microsoft Azure.....	35
4.3.3	Amazon Web Services.....	37
4.3.4	ThingSpeak.....	40
4.3.5	nodeRed.....	41

5	Gestione dei dati	43
5.1	Interazione dei sensori iot con web app	43
5.2	MQTT	45
5.3	XMPP.....	48
5.4	CoAP.....	49
5.5	RESTful HTTP.....	50
5.6	Database Relazionali.....	51
6	Gestione intelligente delle batterie	53
6.1	Gestione della batteria.....	55
6.2	BMS attuali.....	57
6.3	BMS intelligente basato sul cloud.....	60
6.4	Considerazioni finali sui BMS.....	63
6.5	Ambiti applicativi principali.....	64
6.6	Normativa.....	67
7	Web App Sviluppata.....	72
7.1	Descrizione funzionalità.....	72
7.2	Dati utilizzati.....	76
7.3	Implementazione	80
7.3.1	Visualizzazione dei dati.....	80
7.3.2	Modello dei dati.....	84
7.3.3	Package utilizzati.....	86
7.3.4	Autenticazione e autorizzazione.....	88
7.3.5	Password criptate.....	90
7.3.6	Sanificazione di variabili.....	90
7.3.7	MySQL.....	91
7.3.8	Web Server Apache.....	92
	Conclusioni.....	94
	Ringraziamenti.....	95
	Bibliografia.....	96

1 INTRODUZIONE

Negli ultimi anni la tecnologia digitale ha rivestito un ruolo sempre più importante, tanto che gli storici definiscono gli ultimi decenni del Novecento a cavallo con il nuovo millennio come il periodo della terza rivoluzione industriale con al centro l'informatica. Oggi infatti tutto quello che è legato ad essa è diventata una componente fondamentale ed indispensabile per la vita quotidiana e ormai difficilmente potremmo rinunciare ad essa.

Anche il settore della mobilità si sta evolvendo velocemente con un'attenzione particolare alla sostenibilità ambientale. Infatti oggi si sente spesso parlare di auto elettriche o in generale di mobilità sostenibile per ridurre le emissioni di carbonio. L'avvento di auto elettriche è sicuramente legato alle tecnologie digitali avanzate per il loro sviluppo, in particolare per quanto riguarda il controllo, la gestione ed il monitoraggio delle batterie che esse sfruttano.

Secondo stime recenti, a causa della crisi climatica e delle normative europee sulla vendita di automobili, il settore dell'automotive elettrico registrerà un forte incremento di domanda nei prossimi anni e pertanto le batterie saranno al centro di questa rivoluzione della mobilità, che avrà anche un impatto notevole sulle politiche di approvvigionamento delle varie materie prime necessarie alla loro realizzazione. Il pacco batteria infatti rappresenta il principale componente di accumulo di energia per questa tipologia di autoveicoli e richiede un monitoraggio ed un controllo accurato e costante per garantire sicurezza e prestazioni elevate.

Attraverso questa tesi si presenta uno studio effettuato sulle varie tecnologie e piattaforme che si possono utilizzare per acquisire, trasmettere, conservare, analizzare, elaborare e visualizzare tutti i dati necessari al corretto monitoraggio delle batterie elettriche con il fine di garantirne un'efficienza sempre maggiore.

Nel capitolo 2 vengono descritte le varie tipologie di applicazioni grazie alle quali, tramite dispositivi come smartphone e computer, gli utenti possono visualizzare dati e informazioni relative al veicolo e alla batteria al fine di effettuare degli interventi di manutenzione, monitorare lo stato di alcune componenti essenziali o ottenere delle analisi come, ad esempio, quelle inerenti lo stile di guida.

Nel capitolo 3 viene descritto il fenomeno dell' Internet of Things (IoT) , un neologismo utilizzato nel mondo delle telecomunicazioni e dell'informatica per far riferimento all'estensione di internet al mondo degli oggetti reali e concreti, che acquisiscono una propria identità digitale in modo da poter comunicare con altri oggetti connessi alla rete e poter fornire servizi agli utenti di applicazioni ad essi connessi.

Nel capitolo 4 si descrivono le principali piattaforme cloud che gestiscono i dati raccolti da tutti i vari dispositivi che appartengono al mondo dell' Internet of Things. Si tratta di software estremamente potenti e capaci di eseguire la logica di elaborazione e lo storage di grandi quantità di dati.

Nel capitolo 5 si descrivono i principali protocolli di comunicazione che sfruttano i sensori e i dispositivi IoT per trasmettere i dati raccolti alle piattaforme cloud che poi li conservano ed elaborano al fine di effettuare importanti analisi.

Nel capitolo 6 si introducono alcuni degli aspetti principali sull'attuale normativa delle batterie elettriche riguardanti principalmente il corretto utilizzo, la produzione , il riciclo e lo smaltimento.

Infine nel capitolo 7 viene descritta una web app realizzata per monitorare lo stato di un veicolo elettrico da parte di una casa automobilistica e dei suoi clienti.

Keywords: Internet of Things, Cloud, Applicazioni, Batterie al litio,BMS

2 INTRODUZIONE AL MONDO DELLE APPLICAZIONI

In questo capitolo verrà illustrato il mondo delle applicazioni focalizzandosi sulla loro diffusione, sui principali dispositivi dove vengono utilizzate e sulle varie tipologie esistenti.

2.1 Contesto generale

Negli ultimi anni abbiamo assistito ad uno stravolgimento senza precedenti del mondo come lo conoscevamo e tra i fattori più evidenti che contraddistinguono questo cambiamento c'è sicuramente l'innovazione tecnologica. Non è la prima volta che la tecnologia consente "strappi" importanti nello sviluppo della civiltà umana ed infatti è ormai chiaro a tutti che anche Internet e tutto ciò che viaggia con esso è una grande rete che connette il mondo.

In questo caso l'accesso alla potenza tecnologica si accompagna alla sua diffusione e alla sua accessibilità distribuita, al punto che uno strumento tecnologico (il "Digitale") è anche un mezzo di comunicazione di massa interattivo (pensiamo ad esempio ai Social Networks): ne consegue che sono anche le singole persone e non solo le aziende o gli Stati ad avere il controllo e a governare l'utilizzo della tecnologia.

Ogni giorno la maggior parte della popolazione mondiale utilizza vari dispositivi tecnologici anche nelle più piccole e semplici azioni. Si potrebbe dire che è diventato quasi impossibile sopravvivere senza tecnologia in un mondo che si muove così velocemente e la sua importanza è ormai evidente in ogni campo sociale.

L'utilizzo di questi dispositivi sta diventando sempre più un aspetto "intrinseco" alla natura umana. In questa considerazione è anche interessante notare il particolare tipo di device utilizzato durante l'arco della giornata, come riportato nel seguente grafico.



Figura 1: device utilizzati quotidianamente [1]

Notiamo un dato molto importante, ovvero la prevalenza di dispositivi mobile (come smartphone, tablet, laptop e così via) che negli ultimi anni sono stati i dispositivi che hanno spopolato e prevalso sul mercato a scapito dei computer desktop che alle origini erano i veri ed incontrastati padroni della scena tecnologica. Questo risultato è anche legato alla diffusione dell'internet mobile perché questi dispositivi permettono di navigare in luoghi anche dove non arriva la fibra ottica, sono facilmente trasportabili per l'utente ed è sempre maggiore lo sviluppo di nuove applicazioni e funzionalità aggiuntive. Il computer desktop non è assolutamente caduto in disuso, ma il suo utilizzo è diventato molto più specifico rispetto al passato e spesso dedicato a lunghe ed approfondite sessioni di navigazione. A conferma di ciò si riporta un grafico che mostra la crescita del traffico web da dispositivi mobile negli ultimi anni.

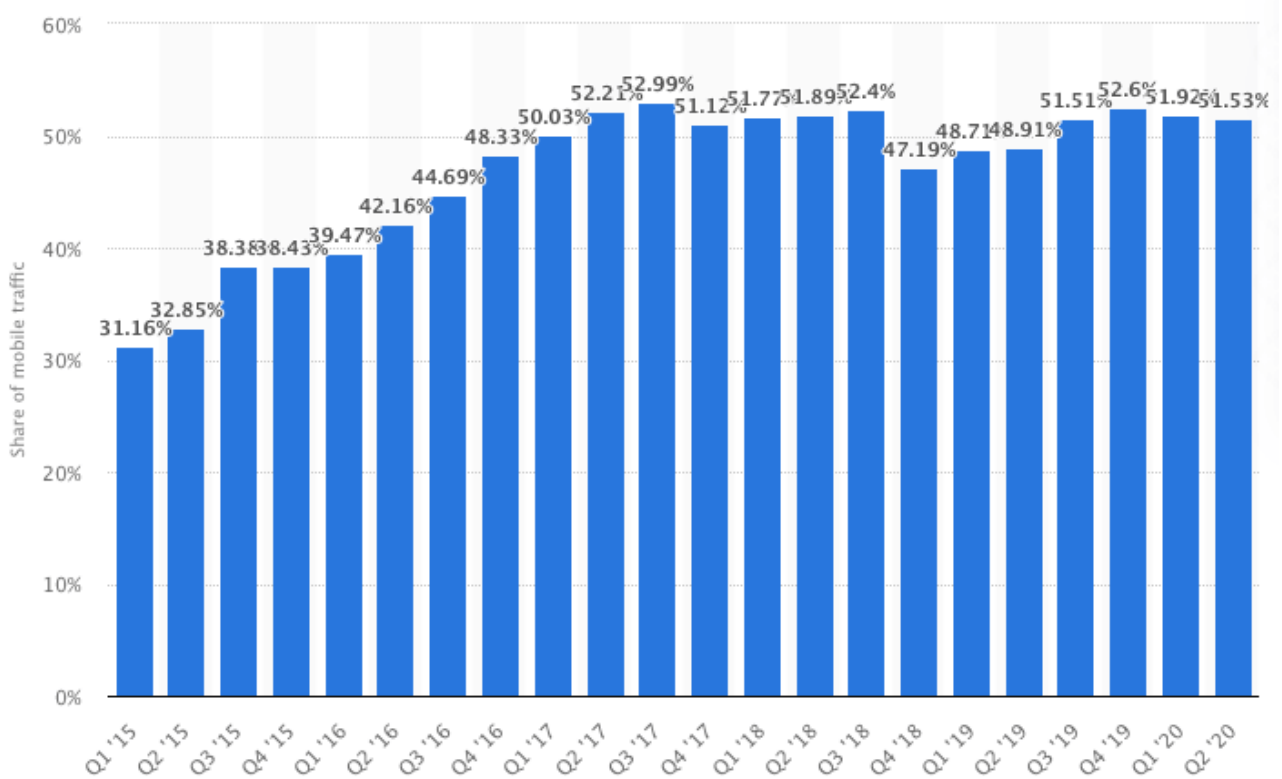


Figura 2: traffico web da dispositivi mobile [1]

2.2 Principali Sistemi Operativi

In generale per dispositivo mobile si intende un qualsiasi dispositivo dotato di comunicazione wireless in grado di accedere alle funzioni di rete. Da questa definizione si evince che sono innumerevoli i dispositivi che oggi possiamo considerare mobile.

In questo contesto specifico il significato dell'aggettivo mobile che più ci interessa è il concetto di mobilità relativamente alla mobilità del possessore del device. Di conseguenza per dispositivo mobile intendiamo un qualunque dispositivo elettronico che segue la mobilità dell'utente.

Questi dispositivi, proprio per la loro natura, rispetto a quelli non mobile, sono caratterizzati da minor capacità di elaborazione e processamento, memorie meno capienti, spesso sfruttano sistemi operativi ed applicazioni realizzate ad hoc, presentano limiti sul consumo di energia e sull'ampiezza di banda e soffrono di interferenze di trasmissione legate alle condizioni esterne.

I principali sistemi operativi per smartphone e tablet degli ultimi anni sono rappresentati nel seguente grafico:

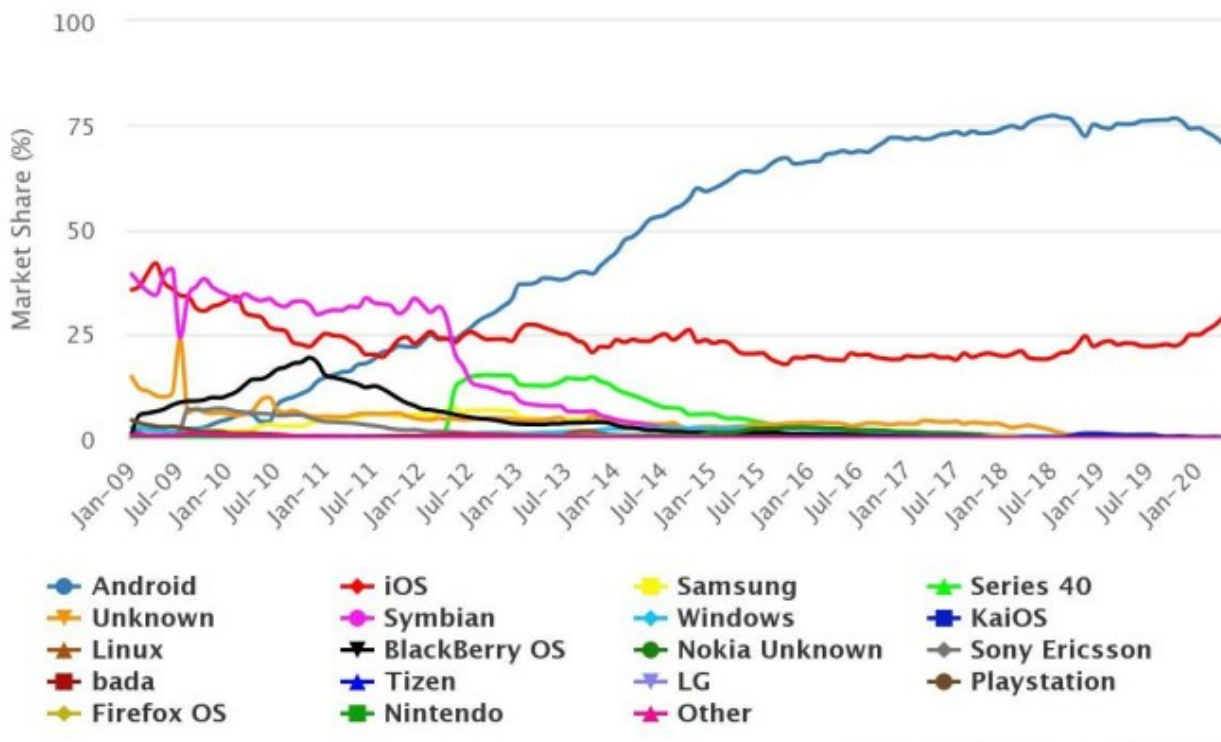


Figura 3:principali so smartphone e tablet [2]

I più diffusi sistemi operativi per smartphone e tablet sono Android e iOS:

a) Android



Figura 4: logo Android [3]

E' un sistema sviluppato da Google e progettato principalmente per sistemi embedded come smartphone e tablet, ma è dotato anche di interfacce utente specializzate per televisori, automobili, orologi da polso, occhiali e ancora molti altri dispositivi.

Presentato nel 2007 ed impiegato per la prima volta nel 2008 si basa sul Kernel Linux (è descritto successivamente). E' rilasciato in modalità open source ,pertanto chiunque voglia utilizzare tale sistema può farlo scaricando l'intero codice sorgente. I produttori di hardware tuttavia possono modificare alcune caratteristiche del sistema al fine di differenziare i propri prodotti da quelli degli altri marchi creando, di fatto, delle versioni "personalizzate" di Android e perfettamente "tarate" sulle caratteristiche dei propri prodotti. E' un sistema programmabile in Java e recentemente anche in Kotlin, un nuovo e recente linguaggio di programmazione sviluppato da JetBrains. [4]

b) iOS

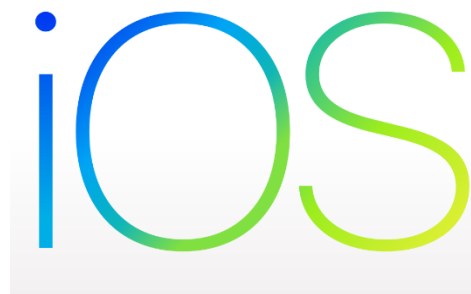


Figura 5: logo iOS [3]

E' stato progettato da Apple a partire da NeXTSTEP, un sistema operativo multitasking sviluppato negli anni '80 e presentato nel 2007 per il primo iPhone. Come Android è anch'esso fornito di interfacce utente specializzate per dispositivi come televisori, automobili, orologi da polso, ecc. E' un sistema molto più chiuso e controllato rispetto ad Android in quanto è disponibile solo su

hardware Apple e il suo sviluppo è possibile solo attraverso dispositivi Mac. A vantaggio di questo maggiore controllo si trovano applicazioni più controllate sugli store, il sistema presenta una maggiore stabilità ed una maggior ottimizzazione dei dispositivi su cui è installato. A differenza del precedente è programmabile in Objective-C e Swift.

I principali sistemi operativi per laptop degli ultimi anni sono rappresentati nel seguente grafico:

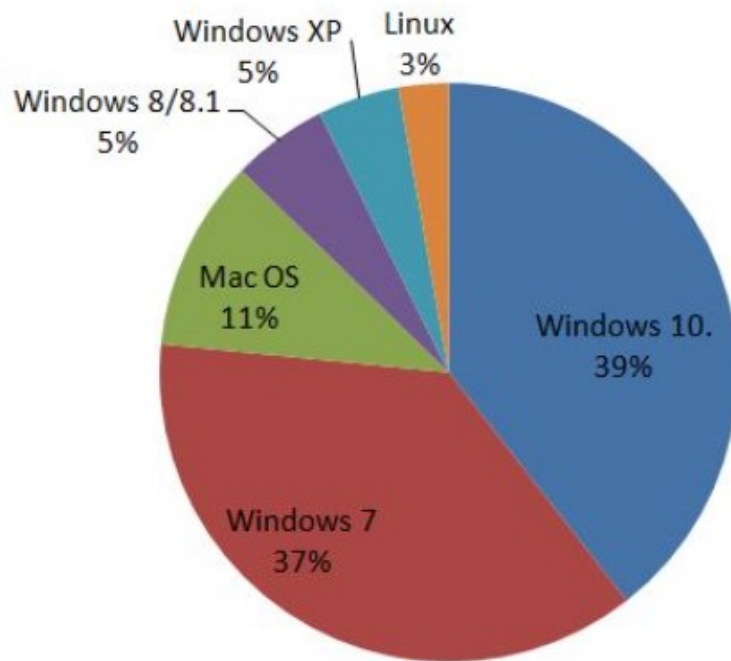


Figura 6 : Principali sistemi operativi per laptop nel mondo nel 2019 [5]

I più diffusi sistemi operativi per laptop sono Windows, macOS, Linux.

a) Windows

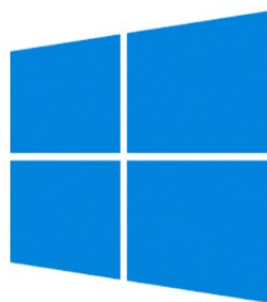


Figura 7: Logo Windows [3]

Microsoft Windows (comunemente indicato come Windows) è una famiglia di ambienti operativi e sistemi operativi prodotta da Microsoft Corporation a partire dal 1985, orientato principalmente a personal computer (detto anche laptop). Prende il nome dall'interfaccia di programmazione di un'applicazione a finestre ("windows" in lingua inglese) in quanto è un sistema che si basa sull'interfaccia grafica. A differenza dei sistemi a linea di comando quindi tutte le operazioni sono svolte con il mouse e ciò lo rende facile da usare anche ad utenti inesperti.

b) macOS



Figura 8: logo macOS [3]

macOS è il sistema operativo sviluppato da Apple per i computer Macintosh, ovvero una famiglia di computer della stessa azienda produttrice. È nato nel 2001 per combinare le note caratteristiche dell'interfaccia utente dell'originario con l'architettura di un sistema operativo di derivazione UNIX. Non è compatibile con i computer che non sono prodotti da Apple in quanto fa parte di un sistema chiuso, ovvero un insieme di software e hardware costruiti per essere compatibili al 100% e spesso prodotti dalla stessa azienda (come in questo caso)

c) Linux



Figura 9: logo Linux [3]

Linux è un sistema operativo che è disponibile gratuitamente in rete e viene sviluppato a livello mondiale da una community open source. Esistono, tuttavia, anche progetti proprietari, soprattutto in ambito commerciale. Come per Unix, anche di Linux esistono diverse versioni, che ampliano e completano il sistema operativo di base. Spesso vengono sviluppate distribuzioni Linux orientate a un determinato uso. Così troviamo, ad esempio, sistemi desktop o specifiche distribuzioni per server. La maggior parte delle distribuzioni ha attualmente un'interfaccia grafica (GUI) di facile comprensione. Nonostante ciò, Linux pone ancora numerose barriere d'accesso soprattutto ad utenti non esperti: infatti in molte distribuzioni Linux alcune installazioni possono essere eseguite solo da terminale. Non essendo popolare come i precedenti sistemi operativi, molti sviluppatori di software non si preoccupano di rendere i loro prodotti compatibili per Linux che vanta dunque una varietà di software inferiore. I sostenitori di Linux ne apprezzano piuttosto la stabilità e la flessibilità: mentre Microsoft ed Apple cercano il più possibile di nascondere le impostazioni rilevanti del proprio sistema operativo, Linux rende tutto visibile e pubblico, consentendo una gestione altamente flessibile del sistema.

2.3 Tipologie di applicazioni e loro sviluppo

Dal punto di vista tecnico esistono diverse tipologie di app, ciascuna con le proprie caratteristiche, vantaggi e svantaggi. Di seguito verranno elencate ed analizzate insieme ad alcuni tra i principali strumenti per il loro sviluppo.

2.3.1 App native

Le app native sono applicazioni sviluppate specificamente per un sistema operativo (ad esempio Android o iOS), per mezzo di uno specifico linguaggio di programmazione. Poiché è stata creata per girare proprio su quel dispositivo, e il relativo sistema operativo, ha la capacità di utilizzare hardware e software di quel dispositivo.

Vantaggi principali:

- Permettono di utilizzare le capacità del dispositivo al massimo
- Costruite tramite tool ed elementi grafici disponibili in un SDK
- Funzionano sempre, anche off-line in assenza di connettività

- Sono molto più veloci ed affidabili rispetto alle altre app garantendo una migliore User Experience
- Consentono l'uso di notifiche Push
- Sono app estremamente efficienti e facili da utilizzare lato utente perché sfruttano le caratteristiche distintive del design di ciascuna piattaforma

Svantaggi:

- Sviluppo, manutenzione ed aggiornamenti sono tipicamente più costosi
- Incompatibilità con piattaforme diverse
- Eventuali difficoltà per ricevere approvazione dagli app store

Per lo sviluppo di applicazioni native Android lo strumento più potente, efficace e diffuso è Android Studio, un ambiente di sviluppo integrato (IDE) specifico per la piattaforma, completamente gratuito e disponibile su Windows, Mac OS X e Linux. Ad oggi è diventato l'IDE primario di Google per lo sviluppo nativo di applicazioni Android. Nasce basandosi sul linguaggio Java ma nel tempo si è reso possibile anche lo sviluppo con Kotlin, un linguaggio più recente e interoperabile con Java. Con esso è possibile testare le proprie app, oltre che con un device fisico, anche tramite un emulatore messo a disposizione dallo stesso IDE. Invece per lo sviluppo nativo su iOS la Apple mette direttamente a disposizione due tools: Xcode e Swift Playgrounds. Fanno entrambi parte della categoria di SDK, sono gratuiti ma sono disponibili solo per hardware Apple. Come Android Studio presentano un emulatore con cui è possibile verificare il funzionamento di applicazioni simulando l'ambiente di lavoro di iOS.

2.3.2 Web app

Secondo una definizione tecnica si tratta di un programma applicativo memorizzato su un server remoto e che viene distribuito su Internet tramite browser. Le web app sono accessibili direttamente da browser e non presentano barriere di marca come nel caso delle app native che sono progettate per piattaforme specifiche.[6]

Vantaggi:

- funzionano su qualsiasi piattaforma

- non richiedono alcuno spazio di memoria per l'installazione o il download visto che sono caricate da un web server ed eseguite sul browser.
- possono essere realizzate con costi molto bassi e in poco tempo
- non è necessario ricordare agli utenti di aggiornarle
- non è necessario scaricarle o installarle

Svantaggi:

- non possono funzionare senza connessione
- si adattano meno a ciascun sistema operativo, non essendo disegnate su misura per nessuno in particolare
- tipicamente presentano prestazioni leggermente inferiori
- non è possibile effettuare il download dell'app sul proprio dispositivo perché non sono disponibili sugli store
- non è possibile inviare notifiche Push agli utenti

Per il fatto che non ci sono particolari barriere di marca esistono molti più framework per lo sviluppo di applicazioni web rispetto ai pochi che ci sono per le app native. Attualmente tra quelli più diffusi ci sono :

a) Spring

Spring è un framework open source per lo sviluppo di applicazioni su piattaforma Java. Il framework nasce nel 2002 con l'obiettivo di proporre un'alternativa leggera e flessibile allo standard EJB (Enterprise JavaBeans) per la realizzazione di applicazioni di livello enterprise. Per poterlo utilizzare c'è bisogno di installare una Java SE Development Kit (SDK). Il cuore del framework è costituito da

un Inversion of Control container che si occupa di gestire l'intero ciclo di vita degli oggetti presenti nel contesto applicativo, dalla configurazione, al reperimento delle dipendenze e creazione delle singole istanze , tramite un apposito Dependency Injection.

Nel corso degli anni il progetto core è stato poi affiancato da altri progetti satellite che vanno ad aggiungere utilità e funzionalità al framework, anche in termini d'integrazione con sistemi esterni all'applicazione.

Spring ,in generale, semplifica lo sviluppo di tutte le applicazioni Java, indipendentemente che si tratti di una applicazione offline, di un'applicazione web o anche di soluzioni native per desktop.

Per gli sviluppatori web che lavorano con Java, Spring offre sia la struttura necessaria che anche una semplificazione notevole del processo di lavoro. I benefici che porta l'affidarsi a Java per lo sviluppo delle proprie applicazioni web sono tra gli altri il fatto di essere un linguaggio multiplatforma e la capacità di integrazione dei software e dei processi delle aziende, così come della loro struttura semplice e chiara. Spring viene solitamente definito anche come Lightweight Framework per applicazioni Java, ovvero un framework leggero; questo perché non sono necessari adattamenti di alcun tipo al codice sorgente per poter approfittare dei numerosi vantaggi che esso ha da offrire. Tra le altre cose va anche considerata la realizzazione della gestione della transazionalità, delle interfacce e dell'accesso alla banca dati. La possibilità di condurre dei test sulle varie unità e sull'integrazione è un ulteriore motivo della popolarità del framework Spring. Tuttavia il merito maggior di Spring è senza dubbio l'abbandono dei componenti non standardizzati e specifici per le singole piattaforme, rinuncia che lo rende di diritto un framework portatile e indipendente dagli application server. Per questo motivo lo si può infatti utilizzare senza problemi come meta framework, in quanto possono essere integrate altri componenti esterni o addirittura interi framework aggiuntivi.

Tra i vantaggi del framework Spring:

- Flessibilità e modularità: libertà allo sviluppatore riguardo al percorso da seguire per arrivare al risultato desiderato in quanto ci sono pochi vincoli da rispettare .
- Elevata testabilità: grazie alla Dependency Injection, paradigma che permette di eliminare dal nostro codice dipendenze dirette tra le classi, possiamo ottenere un'elevata testabilità dei singoli componenti dell'applicazione.
- Community numerosa: gli sviluppatori che ruotano hanno adottato questo mondo sono tantissimi, aspetto che garantisce lo sviluppo di nuove funzionalità, testing e bug-fixing.

Aspetti negativi:

- Documentazione molto estesa e a volte dispersiva: volte risulta faticoso districarsi nella grande quantità d'informazioni disponibili
- Eccessiva flessibilità: la possibilità di implementare soluzioni in maniera differente a volte disorienta fino ad avere l'impressione di non aver utilizzato qualche componente nel modo giusto.

- Configurazione complessa: benché esistano differenti modi di configurare gli elementi del framework, servono spesso molte linee di codice per predisporre anche progetti molto semplici. [7]

b) Django

Si tratta di un framework open source e completamente gratuito, molto apprezzato per la sua intuitività e facile utilizzo a differenza di altre piattaforme che risultano più macchinose e meno efficienti. Il linguaggio di programmazione sul quale è basato è Python, ovvero uno dei più popolari e apprezzati tra gli sviluppatori, che risulta decisamente più interattivo di molti altri linguaggi, in quanto assume la forma di uno pseudo codice.

Sviluppare applicazioni web con Python e con Django comporta la scrittura di meno codice e, di conseguenza, minor possibilità di errori e minor sforzo considerando le numerose librerie disponibili per questo linguaggio, particolarmente apprezzato per la sua facilità di utilizzo e perché non necessita di molte risorse per eseguire tutte le parti necessarie alla progettazione di una web app.

Considerando che questo framework è basato proprio sul linguaggio Python è possibile integrare progetti di Data Science e Machine Learning al suo interno.

In termini di sicurezza, Django offre protezione nei confronti di attacchi contro siti o applicazioni web, come XSS, CSRF, SQL Injection e Clickjacking. Inoltre, Django può essere impiegato in qualsiasi tipologia di progetto, garantendo una buona scalabilità, che consente di amministrare in modo facile e veloce il traffico dell'app.

Il supporto della comunità è un altro fattore da non sottovalutare: grazie alla sua notevole diffusione infatti è facilmente possibile ricevere assistenza in caso di problemi o errori. Oltre a questi vantaggi puramente tecnici, come la sicurezza, la velocità e la scalabilità, è bene evidenziare come Django sia un framework già affermato e affidabile, quindi è sicuramente un'ottima scelta per coloro che intendono sviluppare applicazioni caratterizzate da stabilità nel lungo periodo. I vari componenti aggiuntivi integrabili nel framework consentono di ingrandire le capacità strutturali dell'applicazione web o del sito. Django ha una grande portabilità, infatti permette di creare applicazioni utilizzabili su diversi sistemi operativi (tra cui macOS, Linux, Windows).

Permette di passare da un database all'altro scrivendo solo una riga di codice. Inoltre possiede al suo interno strumenti di responsività, che fanno sì che le applicazioni progettate si adattino in modo immediato alle varie risoluzioni di schermo. [8]

c) Laravel



Figura 10: Logo Laravel [9]

Laravel è un framework PHP per applicazioni web rilasciato dal 2011 come framework open source. Un framework PHP fornisce un insieme di librerie di codice contenenti moduli pre-programmati che permettono all'utente di costruire applicazioni più velocemente. Offrono agli sviluppatori web una serie di benefici, tra cui uno sviluppo più rapido, una ridotta necessità di scrivere codice e una maggiore sicurezza. Aiutano anche gli sviluppatori principianti a costruire buone pratiche di programmazione, poiché richiedono un'organizzazione specifica del codice. Inoltre, i framework PHP richiedono in genere meno manutenzione delle applicazioni costruite da zero. Laravel, come altri framework PHP, si basa sul pattern Model-View-Controller (MVC) per l'architettura dove:

- model rappresenta il modello dei dati dell'applicazione
- view fornisce interazione con l'utente in quanto è la parte di interfaccia grafica
- controller è un condotto tra view e model in quanto in base alle richieste dell'utente recupera le informazioni necessarie dal model, le elabora e le restituisce per la visualizzazione in una view

MVC Architecture Pattern

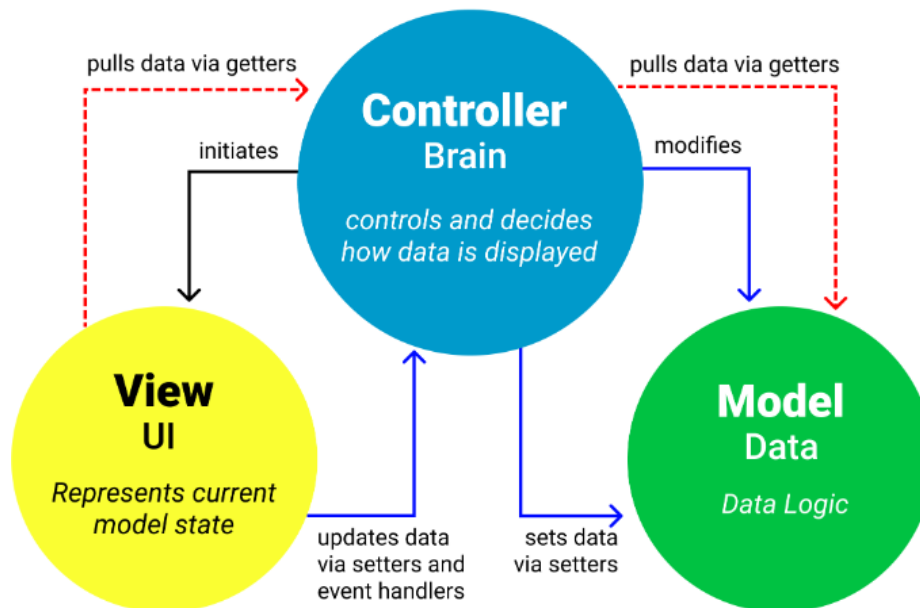


Figura 11: Architettura MVC [10]

Le componenti principali del framework sono:

- Blade costituisce il template engine e permette di sanificare variabili(funzionalità legata alla sicurezza), di utilizzare direttive di controllo e permette la definizione di layout che più views possono utilizzare
- Artisan è una interfaccia a linea di comando (CLI) parte integrante del framework che consente di attivare comandi che supportano lo sviluppo dell'applicazione
- Eloquent implementa l'ORM che permette una facile interazione con i database. Usando l'architettura MVC, i modelli corrispondono alle singole tabelle in un database. Con Eloquent, gli sviluppatori possono assicurarsi un recupero veloce, semplici inserimenti e aggiornamenti, e una definizione intuitiva delle relazioni. [9]

Le applicazioni PHP, in generale, hanno alcune vulnerabilità note. Tuttavia questo problema è facilmente superabile utilizzando Laravel in quanto ha meccanismi di autenticazione, autorizzazione, meccanismi di protezione per iniezione di codice e iniezione sql. Inoltre semplifica alcune delle attività più comuni delle web app come routing, sessioni e caching. Laravel offre anche una vasta raccolta di documentazione online, che offre un ottimo supporto agli sviluppatori,

e una vasta gamma di plugin per estendere le funzionalità di base facilmente installabili tramite dipendenze Composer (uno strumento di terze parti per la gestione delle dependency di PHP a livello di applicazione). È disponibile anche un'ampia varietà di risorse di apprendimento online dirette a tutti i livelli di conoscenza. Un altro vantaggio è la scalabilità. Laravel è altamente scalabile. Con il supporto integrato per sistemi di cache veloci e distribuiti, le applicazioni Laravel sono in grado di affrontare milioni di richieste. Per comprendere al meglio cosa può fare Laravel, è importante capire come tratta le richieste, cioè il ciclo di vita della richiesta. Come discusso in precedenza, Laravel è basato sull'architettura MVC, e risponde alle richieste degli utenti, che il controller utilizza per recuperare ed elaborare i dati dal modello per poi presentare le informazioni all'utente nella View. Una richiesta entra in un'applicazione Laravel attraverso il file `public/index.php`, che carica il resto del framework e recupera un'istanza dell'applicazione Laravel. Dopo aver recuperato l'istanza dell'applicazione, la richiesta viene indirizzata al kernel della console o al kernel HTTP. Tra gli altri compiti, il kernel definisce una selezione di bootstrapper che eseguono compiti che devono essere completati prima che la gestione delle richieste abbia luogo, così come definisce qualsiasi middleware che le richieste devono attraversare prima della gestione. Tra questi compiti di pre-gestione c'è il caricamento di qualsiasi fornitore di servizi richiesto per l'applicazione. Una volta che il bootstrapping è completo e i fornitori di servizi sono stati registrati, la richiesta passa al router, che poi indirizza la richiesta a una route o a un controller. Inoltre, il router esegue qualsiasi middleware richiesto specifico per le routes. Dopo che la richiesta passa attraverso tutti i middleware richiesti, l'esecuzione di una route o di un metodo controller restituisce una risposta che passa di nuovo alla View attraverso la catena.

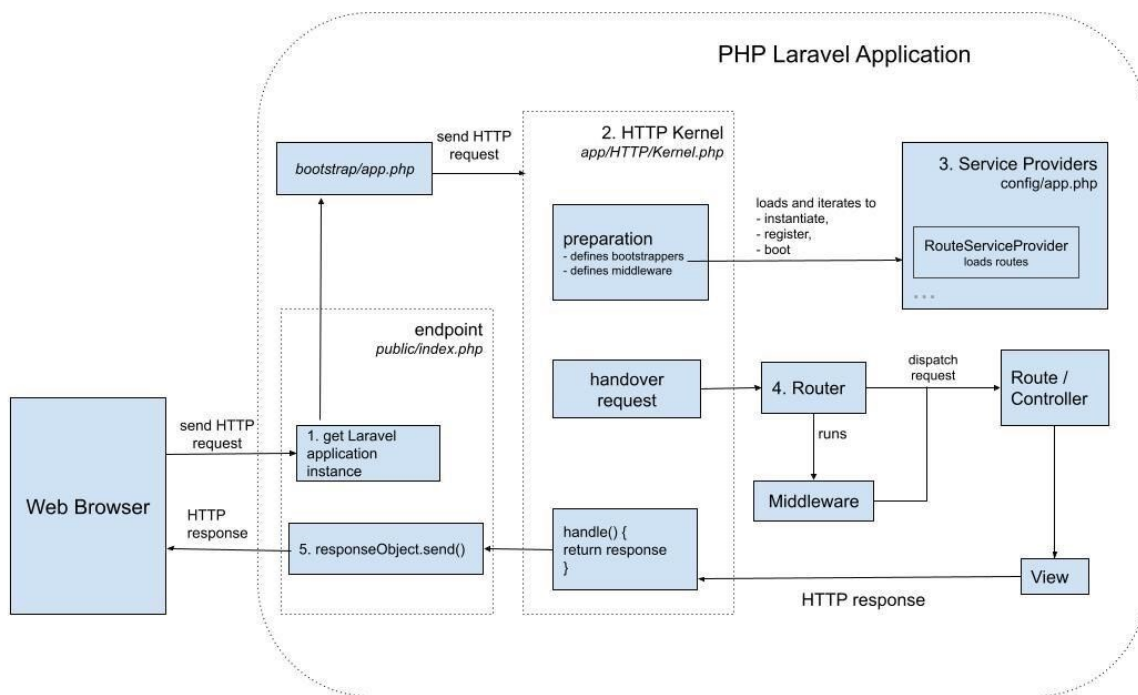


Figura 12: Workflow Laravel [9]

2.3.3 App ibride

Le app ibride sono l'anello mancante tra le app native e web app. Un'app ibrida infatti è un programma che combina gli elementi di applicazioni web e native, in cui l'applicazione è sviluppata usando i linguaggi di tecnologie web come HTML, CSS e JavaScript che sono condensati in un'applicazione nativa. Quindi funziona esattamente come una web app ma è installata su un dispositivo come un'app nativa. Le app ibride, con l'aiuto di interfacce del programma applicativo interno (API), mantengono il completo accesso alle funzioni del dispositivo mobile. Le app native possono operare offline in quanto vengono scaricate, ma le app ibride non hanno questa funzionalità. Un'app ibrida solitamente ha funzioni simili ad una web app essendo principalmente basata sullo stesso sistema. Le applicazioni web sono sviluppate per varie piattaforme e non sono installate sul dispositivo, ma accessibili tramite internet con l'utilizzo di un browser. Le app ibride invece sono generalmente associate al mobile computing, una tecnologia che abilita la trasmissione di dati tramite un computer o qualsiasi altro dispositivo connesso alla rete senza vincoli sulla posizione fisica dell'utente o delle apparecchiature coinvolte.

Vantaggi:

- Funzionano su diverse piattaforme
- Più rapide da sviluppare e da mantenere delle app native, in quanto il processo di sviluppo è unico
- Rappresentano un buon compromesso in termini di funzionalità tra app native e web

Svantaggi:

- Le performance sono inferiori rispetto alle app native
- Sono in genere meno stabili, perché sono adattate per funzionare su tutte le piattaforme
- Se richiedono l'uso di funzionalità native del dispositivo (come fotocamera, gps, giroscopio, ecc..) richiedono moduli in fase di sviluppo che rendono la complessità simile a quella per le app native

Tra i framework più diffusi per la creazione di app ibride vi sono Ionic e Flutter

a) Ionic

Ionic è dotato di un'interfaccia front-end che gestisce l'aspetto e le interazioni dell'interfaccia che l'applicazione necessita. Dato che Ionic è un framework HTML5, necessita di un wrapper nativo come Cordova per essere eseguito come un'applicazione nativa. Tramite a questi strumenti è possibile creare logica e interfaccia utente dell'applicazione sfruttando HTML5, CSS e JavaScript per accedere alla funzionalità della piattaforma mobile tramite apposite API e creare pacchetti per ciascuna piattaforma.

Abilitando quindi lo wrapping del codice CSS, HTML e JavaScript in base alla piattaforma del device in utilizzo, ne vengono estese le funzionalità in modo che queste possano lavorare direttamente con il device stesso, al pari di un'app nativa.

Le risultanti applicazioni pertanto saranno ibride in quanto non sono né vere applicazioni native (perché il rendering del layout viene fatto tramite le interfacce web invece che dal framework nativo della piattaforma), né vere applicazioni web (perché realizzate come applicazioni pronte per la distribuzione sugli app store e che hanno accesso alle API del device nativo).

Fornisce anche un'interfaccia a riga di comando (CLI) per creare progetti e consentire agli sviluppatori di aggiungere plugin e pacchetti front-end aggiuntivi. Ionic per la creazione di interfacce utente si può appoggiare a tre diversi "sotto-framework" che sono Angular, React e Vue. Sono simili tra loro ed infatti possiamo affermare che oggi quasi ogni applicazione di questo tipo utilizza uno di questi, ma la grande differenza è il fatto che Angular è un framework

completo e maggiormente supportato rispetto a React e Vue che sono essenzialmente una collezione di librerie. Inoltre utilizzando Ionic si può scoprire un aspetto che è un vero e proprio vantaggio per la programmazione mobile. Rispetto ad altri framework che danno la possibilità di fare il debug e il testing delle proprie app solo tramite un emulatore (un software che consente al computer host di comportarsi come un altro dispositivo) o solo connettendo un dispositivo fisico, con Ionic si vede che Chrome Developer Tools (ma anche altri browser) dà la possibilità di fare debug sia per le componenti visive sia per il codice JavaScript, fornendo quindi la possibilità di ispezionare ciascun elemento UI a run time e provare istantaneamente diverse modifiche.[11]

b) Flutter

Nel corso del 2018 il nuovo framework Flutter, sviluppato da Google è entrato nel panorama dello sviluppo delle app cross-platform. E' un progetto open source totalmente gratuito per la creazione di app native di alta qualità su iOS e Android in tempi rapidi e con il supporto alle interfacce native programmabile in Dart.

Flutter è diverso dalla maggior parte dei framework per lo sviluppo di app ibride poiché non usa WebView. E' infatti strutturato in un modo completamente nuovo, permettendo di creare applicazioni semplici e ad alte prestazioni venendo eseguito su ogni piattaforma in modo nativo usando la compilazione AOT (ahead-of-time), mentre durante la fase di sviluppo, per rendere il processo di test più veloce, viene utilizzata la compilazione JIT (just-in-time).

Nonostante condividano alcune filosofie comuni, sia Flutter che Ionic sono diversi l'uno dall'altro. Quando si crea un'applicazione su Ionic lo si fa sul "web" tramite gli standard e le tecnologie web aperte, ovvero CSS, HTML e JavaScript che abilitano un meccanismo che supporta l'esecuzione delle applicazioni Ionic su varie piattaforme incrociate con la stessa integrità e interfaccia utente. Quando Ionic garantisce un confine più ampio come gli standard Web aperti, Flutter presenta invece un ecosistema limitato e autonomo. JavaScript, che è un linguaggio di sviluppo web ampiamente accettato e popolare, non funziona su flutter e bisogna conoscere approfonditamente Dart per lavorare in Flutter. [12]

2.3.4 Progressive web app (PWA)

Più recentemente è stato coniato il termine Progressive web app per far riferimento ad un'altra tipologia di applicazione: particolari applicazioni web che se visualizzate da un dispositivo mobile si comportano come app native. Queste superano alcuni limiti delle web app pur condividendone i vantaggi e sfruttano le funzionalità dei browser moderni, come il Service Worker e il Web App Manifest. Il loro merito è quello di colmare il divario esistente tra browser ed app native. Proprio tramite il service worker, uno script che si trova *tra* il server e il browser, le PWA mettono a disposizione una serie di features aggiuntive rispetto ai normali siti web. Le funzionalità più interessanti introdotte da una PWA sono:

- Possibilità di aggiungere alla home del dispositivo l'icona della PWA, raggiungibile in questo modo attraverso una scorciatoia visto che non necessita dell'apertura del browser o della ricerca classica come nel caso delle web app "normali"
- Invio di notifiche push
- Fruibilità nella modalità offline grazie alla pre-memorizzazione delle risorse nella cache del client (ovvero il dispositivo mobile). Il Service Worker infatti decide come rispondere alle richieste dell'app scegliendo se scaricare una risorsa dal web oppure se recuperarne una versione precedentemente scaricata.
- Aggiornamenti automatici senza che l'utente debba compiere alcuna azione
- Indipendenza dagli store: a differenza delle app native, che devono soddisfare i requisiti spesso stringenti degli store per poter essere pubblicate, le PWA garantiscono maggiore libertà nello sviluppo.

3 INTERNET OF THINGS (IOT)

3.1 Definizione

IOT è un acronimo che sta per Internet of things, ovvero Internet delle cose. In termini molto semplici possiamo dire che l'Internet of things riguarda l'estensione della potenza di Internet oltre i computer a tutta una serie di altri processi e ambienti. È l'espressione utilizzata per definire l'intera rete delle apparecchiature e dei dispositivi connessi a Internet ed equipaggiati con un software che permette di scambiare dati con altri oggetti connessi. Gli obiettivi di questi dispositivi consistono in raccogliere dati e informazioni, dare comandi e controllare particolari attività. Le caratteristiche principali di un oggetto connesso alla rete sono la capacità di scambiare dati attraverso la rete stessa senza bisogno dell'intervento umano e avere un indirizzo IP che ne consenta l'identificazione univoca sulla rete.[13]

3.2 Come nasce Internet of Things

Nel 1998 i ricercatori dell'Auto-ID Center (un'organizzazione di ricerca globale, indipendente e no profit con sede presso il Massachusetts Institute of Technology) si misero alla ricerca di un nuovo modo per localizzare e identificare lo spostamento degli oggetti da un luogo fisico all'altro. Proprio i ricercatori di questa organizzazione capirono che l'identificazione automatica consentisse di elevare la qualità della gestione dei prodotti ad un livello più avanzato. Di Internet of Things si sentì parlare per la prima volta nel 1999 da parte di Kevin Ashton (cofondatore e direttore esecutivo di Auto-ID Center) e da allora si è assistito al grande sviluppo di Internet e alla connessione in rete di miliardi di dispositivi e persone in tutto il mondo.[14]

Non a caso negli ultimi anni sempre più delle grandi aziende che offrono servizi Cloud mettono a disposizione piattaforme per sviluppare l'IoT. Come certificato anche da Gartner (una società che si occupa di consulenza strategica, ricerca e analisi nel campo della tecnologia dell'informazione) sono sempre più i dispositivi connessi dalla nascita dell'IoT, inteso non solo come ideologia ma come reale interconnessione tra Internet, persone ed oggetti. [15]

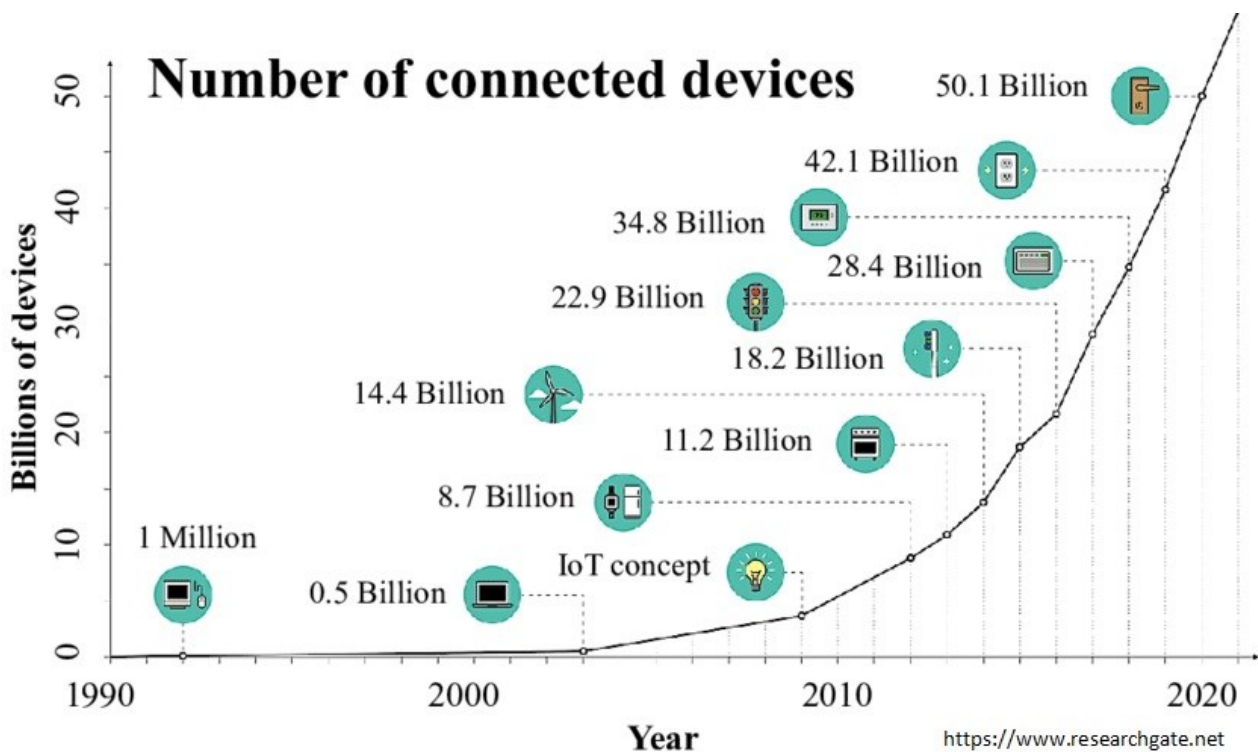


Figura 13 : Dispositivi connessi alla rete [16]

3.3 Architetture IOT

I dispositivi sono sempre più connessi ma, essendo dotati di risorse limitate, hanno continuamente bisogno di dialogare fra di loro in maniera sincronizzata, secondo protocolli ben precisi. Raccolgono dati mediante i sensori, li memorizzano in strutture dati, generalmente nel Cloud, dove vengono analizzati per ricavarne informazioni che successivamente vengono riportate nel mondo reale. Questi dati poi si trasformano in messaggi di notifiche push su app mobile, grafici su browser o anche Qr code. L'intero ecosistema di oggetti hardware e software che fa parte di un progetto IoT è basato su un'architettura composta da diversi elementi appositamente assemblati fra di loro per interagire correttamente.[13]

Il cloud , attraverso le sue caratteristiche più tipiche quali elasticità, scalabilità, potenza ed economicità, sta abilitando l'IoT.

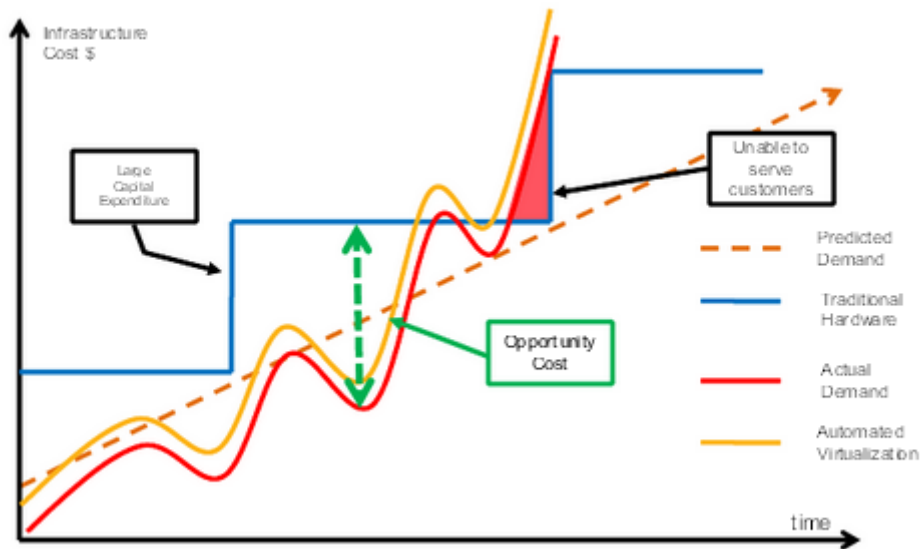


Figura 14 : scalabilità del cloud computing [17]

L'aspetto principale che rende molto interessante il collegamento tra cloud e IoT è senza dubbio l'enorme quantità di dati che l'IoT è in grado di generare, ed è proprio qui che l'IoT viene sostenuto dalla potenza e dalla flessibilità del cloud. Tra gli oggetti fisici e gli utenti è infatti presente una apposita piattaforma che ha il ruolo di comunicare con gli oggetti e con i software, gestire i dati, utilizzare algoritmi per estrarre informazioni rilevanti e mostrare i risultati ottenuti.[15]

3.4 Ambiti applicativi

Le applicazioni più interessanti dell' IoT, soprattutto in ambito aziendale, riguardano:

- Industria 4.0

Con questa espressione si intende un modello di produzione e gestione aziendale. Secondo la definizione che ne dà il Ministero italiano per lo sviluppo economico, gli elementi che caratterizzano il fenomeno sono connessione tra sistemi fisici e digitali, analisi complesse attraverso Big Data e adattamenti real-time al fine di rendere più flessibile la gestione del ciclo produttivo.

Questa espressione fa infatti riferimento alla quarta rivoluzione industriale con l'obiettivo di creare un connubio tra la produzione fisica e le operazioni con tecnologia digitale intelligente, apprendimento automatico e big data per creare un ecosistema innovativo. Non va intesa come un semplice investimento in nuove tecnologie e strumenti per migliorare l'efficienza produttiva, ma si tratta di un concetto rivoluzionario per il modo in cui l'intera azienda opera e cresce.

- Smart Logistics

E' il processo di adozione delle nuove tecnologie in tutte le operazioni che si svolgono all'interno e all'esterno del magazzino, dalla ricezione della merce nelle baie di carico e scarico fino alla distribuzione e consegna al cliente nelle condizioni accordate

- Smart Asset Management

Consiste nella gestione in remoto di asset di valore al fine di rilevare guasti e manomissioni, come pure la localizzazione e la tracciabilità .

- Smart Factory

Implementazione di nuovi metodi per il controllo e l'organizzazione della produzione grazie all'uso di macchine in grado di rilevare informazioni in tempo reale poter comunicare tra loro e prendere decisioni in autonomia senza bisogno dell'intervento umano. [15]

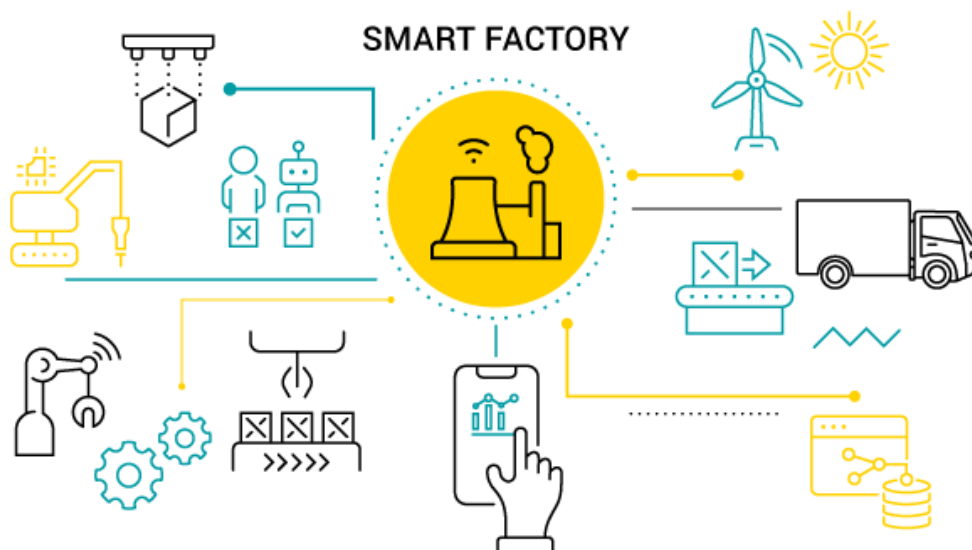


Figura 15 :Smart factor [18]

3.5 Rischi, privacy e sicurezza

Data la possibilità di connettere alla rete qualsiasi tipo di oggetto si è in presenza di una convergenza sempre più stretta tra la tecnologia e la quotidianità con lo scopo di semplificare azioni umane (tema affrontato anche nel capitolo 2).

A questo fine si rende spesso necessaria la raccolta di dati relativi agli utenti che utilizzano i vari dispositivi connessi alla rete con il fine di estrarre determinati contenuti ed informazioni. Questo aspetto deve portare a particolari considerazioni inerenti la privacy di chi utilizza oggetti appartenenti alla sfera IoT, sia riguardo il trattamento dei dati personali sia riguardo la sicurezza. Riguardo la privacy alcuni aspetti fondamentali da tenere in considerazione sono il consenso dell'utente sulla raccolta dei dati, la libertà da parte dell'utente di condividere o meno i suoi dati ed infine l'anonimato dell'utente da mantenere durante il raccoglimento dei suoi stessi dati.

Ovviamente è anche compito degli utenti capire come utilizzare al correttamente i servizi che vengono offerti e come questi possono rilevare dati sensibili. Per quanto riguarda la sicurezza i problemi derivano dal fatto che i servizi IoT si stanno sviluppando molto rapidamente, con il rischio che non siano considerate in maniera appropriata tutte le attenzioni inerenti la protezione dei dati perché si punta principalmente allo sviluppo delle performance, rendendo anche i sistemi e i servizi vulnerabili da attacchi informatici. Per ridurre i rischi andrebbe presa in forte considerazione la possibilità di limitare la quantità dei dati da raccogliere e ovviamente testare con assoluta attenzione i sistemi di sicurezza prima di lanciare prodotti sul mercato. [19]

4 PIATTAFORME CLOUD

4.1 Modelli di distribuzione dei servizi cloud

I servizi Cloud possono essere distribuiti in diversi modi in base alle necessità dell'utente come la gestione del trattamento di dati personali, il livello di performance richiesto, ecc...

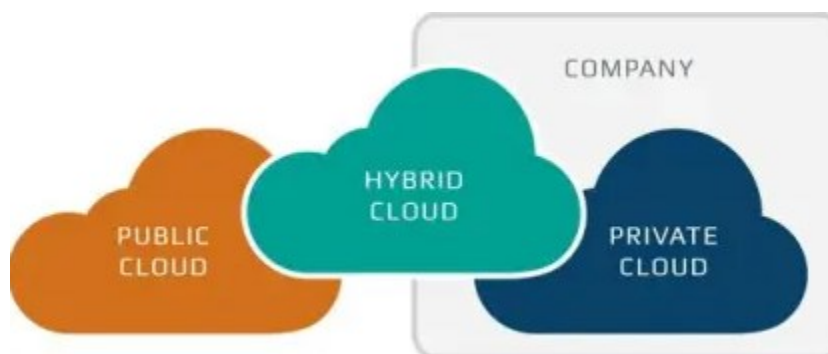


Figura 16 : modelli di distribuzione del cloud [20]

I modelli di distribuzione più comuni sono i seguenti:

a) Public Cloud

Quando il fornitore distribuisce e gestisce i propri servizi destinati al pubblico, il modello viene chiamato Cloud Pubblico. Generalmente, i più grandi fornitori di servizi Cloud pubblici come Amazon AWS, Microsoft e Google, sono proprietari dell'infrastruttura e offrono l'accesso solo attraverso Internet. I servizi offerti possono essere offerti gratuitamente, con risorse limitate, oppure a consumo (Pay-per-use o Pay-as-you-go). Gli utenti e i fornitori stipulano una sorta di contratto tra di loro, al cui interno vengono inserite le varie clausole legate al livello di qualità del servizio fornito e l'utente dovrà adattarsi al contratto esistente. Se questo non viene garantito, il fornitore sarà tenuto a pagare una penale all'utente come di solito stabilito. Inoltre, la sicurezza e la privacy dei dati degli utenti è gestita direttamente dal fornitore del servizio visto che i dati sono ospitati proprio sul Cloud del fornitore.

Uno degli ostacoli maggiori per i sistemi Public Cloud è la sicurezza e la privacy dei dati caricati sul Cloud. Infatti spesso è possibile che il cliente non sappia dove risiedono i propri dati né se vengano salvati in copie di backup e soprattutto nemmeno se utenti non autorizzati possono avere accesso a questi dati.

b) Private Cloud

Quando l'infrastruttura del sistema Cloud è accessibile solamente da una singola azienda oppure da un gruppo per il suo esclusivo uso interno vi sarà un modello di tipo Privato. Le risorse fisiche possono essere ospitate internamente all'organizzazione oppure anche esternamente. Il Cloud Privato può seguire le politiche di sicurezza dell'organizzazione perciò può fornire un maggiore livello di sicurezza sia per i dati che per le applicazioni rispetto al Cloud pubblico. Un altro vantaggio è la possibilità di fornire un maggior controllo sulle performance, visto che il controllo fisico dell'hardware è effettuato direttamente dall'organizzazione.

Il grande vantaggio offerto dal Private Cloud rispetto al Public Cloud è dato dalla sicurezza dei dati e dal rispetto della privacy su di essi. Il Private Cloud infatti è ospitato sempre nei data center dell'azienda che utilizza i servizi presenti su di esso per cui fornisce una maggiore sicurezza rispetto al Public Cloud.

c) Community Cloud

Quando l'infrastruttura e le risorse computazionali sono condivise tra diverse organizzazioni che condividono caratteristiche ed obiettivi comuni come requisiti di sicurezza e/o politiche aziendali, questo modello viene chiamato Community Cloud. Della gestione dell'infrastruttura si occupa una delle organizzazioni facente parte del gruppo di condivisione e i costi vengono ripartiti tra le varie organizzazioni secondo politiche prestabilite e condivise da tutti i membri dell'organizzazione.

d) Hybrid Cloud

Quando l'infrastruttura è ottenuta dalla composizione di più Cloud (public, private o community) si parla di modello ibrido quindi di Hybrid Cloud. E' fondamentale per questo modello differenziare al meglio i servizi che devono appartenere al private Cloud e quali al public Cloud. I soggetti, in questi casi, rimangono unici ma sono legati assieme per garantire una maggiore integrità dei dati. L'Hybrid Cloud offre il basso costo e i benefici nella scalabilità ed elasticità dei Public Cloud e offre la sicurezza e il controllo del Private Cloud: per le numerose caratteristiche positive ereditate dagli altri modelli è diventato uno dei modelli più in crescita sul mercato. Riesce a garantire una maggior

flessibilità alle aziende e viene spesso utilizzato per forzare il proprio carico di lavoro verso il Public Cloud per velocizzare il soddisfacimento delle richieste. [21]

4.2 Tipologie di servizi offerti

I vari modelli di distribuzione appena discussi possono essere utilizzati per fornire diverse tipologie di servizi.

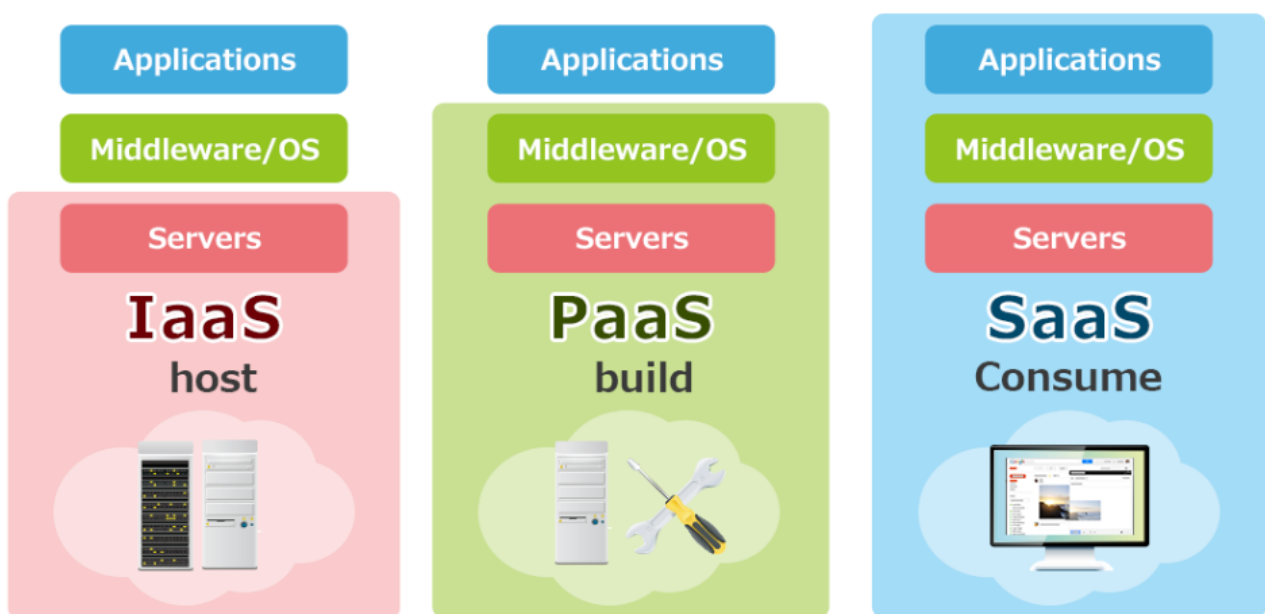


Figura 17 : Modelli di servizi cloud [22]

Di seguito le principali tipologie di servizi Cloud :

a) Software as a Service

SaaS è un modello di servizio che fornisce accesso al software e alle sue varie funzionalità attraverso un server web, quindi ospitate su un Cloud e raggiungibili attraverso internet. Le applicazioni SaaS hanno una gestione centralizzata grazie alla quale gli utenti non si devono preoccupare dell'installazione, della manutenzione, del supporto e dell'aggiornamento del software perché viene tutto gestito dal fornitore del servizio. Il servizio può essere distribuito in due modalità: multi-tenant se numerosi utenti possono usare lo stesso software

contemporaneamente o single-tenant se ogni utente ha il suo ambiente isolato per utilizzare il software.

La tipologia di servizi SaaS forniscono diversi vantaggi per le imprese:

- Abbassamento dei costi di servizio: non sono previsti costi di manutenzione o per aumentare l'hardware e la sicurezza perché la spesa principale è quella necessaria alla sottoscrizione del servizio ed è poi compito del singolo fornitore cloud preoccuparsi di tutte le altre voci di spesa per migliorare il servizio offerto
- Velocizzazione dei processi di business: grazie alla rapid elasticity la potenza computazionale e le potenzialità di memorizzazione possono scalare istantaneamente sia aumentando che diminuendo le risorse utilizzate
- Aumento della flessibilità: le applicazioni SaaS necessitano soltanto della connessione Internet per accedervi, pertanto le aziende possono abbattere i costi fissi legati alle strutture di ufficio favorendo modalità di lavoro alternative come lo smart working
- Miglioramento del servizio al cliente: applicazioni SaaS specializzate permettono alle aziende di sviluppare strumenti di misurazione del comportamento del cliente sui loro prodotti e servizi così da poter fare conclusioni su come migliorarli.

b) Platform as a Service

PaaS fornisce la piattaforma per implementare applicazioni software come un servizio all'utente. La piattaforma può includere linguaggi di programmazione, insiemi di librerie, funzionalità grafiche, ambienti di test, ecc... PaaS fornisce un metodo estremamente facile agli sviluppatori per creare e distribuire software su una infrastruttura Cloud. Inoltre i fornitori SaaS possono anche distribuire le proprie applicazioni sulle piattaforme da loro fornite. Il beneficio nell'utilizzo dei modelli PaaS per gli sviluppatori è che si possono focalizzare meglio sullo sviluppo dei loro software piuttosto che impegnare ore lavorative sulla manutenzione e l'aggiornamento della piattaforma stessa.

I vantaggi derivanti dall'utilizzo del modello PaaS:

- Velocizzazione di distribuzione e testing: grazie a fasi di testing veloci, ma estremamente sicuri ed affidabili, la distribuzione delle applicazioni risulterà più rapida

- Aggiornamenti automatici: il cliente non si deve preoccupare degli aggiornamenti del sistema Cloud visto che saranno fatti in automatico ed eseguiti da parte del Cloud provider
- Abbassamento costi: PaaS permette di includere tutto l'hardware in un unico luogo fisico così da ottenere benefici per il budget aziendale

Alcuni limiti invece sono costituiti da :

- Poca concorrenza: a causa del numero limitato di fornitori PaaS sul mercato il cliente si trova nella condizione di non poter acquistare servizi analoghi da un fornitore differente senza dover sostenere rilevanti costi e rischi per effettuare il passaggio
- Problemi di integrazione: l'integrazione dei servizi PaaS col resto del sistema potrebbe risultare estremamente complicata.
- Flessibilità limitata: le soluzioni PaaS non reggono il confronto coi modelli IaaS in termini di flessibilità non riuscendo ad allocare e disallocare numerose macchine virtuali facilmente.

c) Infrastructure-as-a-Service

IaaS fornisce le macchine fisiche e virtuali, il sistema operativo e tutte le varie risorse hardware come un servizio per gli utenti. In questa struttura diversi server possono lavorare in parallelo in maniera del tutto concorrente. Chi usufruisce di questa topologia di servizio è esentato da qualsiasi problema di manutenzione hardware. Le macchine virtuali sono un'astrazione dell'hardware, perciò permettono di fornire i servizi agli utenti in maniera più flessibile fornendo un ambiente isolato sotto il pieno controllo dell'utente.

I vantaggi derivanti dall'utilizzo del modello IaaS includono:

-Miglioramento continuo delle performance e dei prezzi: l'alta concorrenza presente nel mercato fa sì che si presenti un continuo abbassamento dei prezzi e un continuo aumento delle performance dei più grandi sistemi Cloud .

- Immediatezza delle risorse: il tempo che intercorre tra l'introduzione della richiesta al server e la sua risposta può essere ridotto notevolmente all'interno di un ambiente IaaS.

- Pagamento delle sole risorse effettivamente utilizzate: le aziende pagheranno solo per le risorse che realmente utilizzano così da non sprecare il proprio capitale di investimento

d) Hardware as a Service

HaaS viene utilizzato quando le risorse a disposizione dei computer non sono sufficienti per elaborare la grande quantità di dati richiesta. Con questo modello gli utenti usufruiscono delle risorse fisiche e virtuali messe a disposizione dalla piattaforma. E' attualmente la tipologia di servizio più difficile da realizzare, per via degli altissimi investimenti richiesti. Solitamente i provider di soluzioni HaaS forniscono servizi come il Load Balancing, lo storage Cloud e diversi altri strumenti per la gestione del networking all'interno della propria rete nell'infrastruttura Cloud.

e) Data as a Service

DaaS è una strategia di Cloud utilizzata per facilitare l'accesso ai dati: il servizio li rende disponibili all'utente come se fossero presenti nel disco locale del computer. DaaS elimina la ridondanza e riduce le spese ospitando tutti i dati in una singola locazione, rendendo possibile l'accesso ai dati a più utenti contemporaneamente. Potenziali problemi per questo modello possono essere la perdita dei dati e soprattutto la sicurezza dei dati stessi.[23]

4.3 Principali piattaforme

Una piattaforma IoT deve essere in grado di compiere delle analisi descrittive, diagnostiche, predittive e prescrittive sui dati raccolti al fine di elaborare relazioni tra questi, individuare cause di eventuali problemi, prevedere malfunzionamenti del sistema e stabilire quali azioni svolgere per evitare possibili problemi futuri.

Le principali abilità che una piattaforma IoT dovrebbe possedere sono:

- Connettività : la piattaforma rappresenta l'infrastruttura tramite la quale gli oggetti sono legati all'analisi delle loro funzioni
- Sicurezza : i dati devono essere protetti, ma bisogna prestare attenzione anche alla protezione della rete, alla gestione degli accessi e all'identificazione dei dispositivi
- Gestione : i dispositivi connessi devono essere monitorati dal punto di vista degli aggiornamenti e del comportamento
- Analisi : capacità che consente di passare dai dati alle informazioni, dalle quali dipendono le azioni da compiere[24]

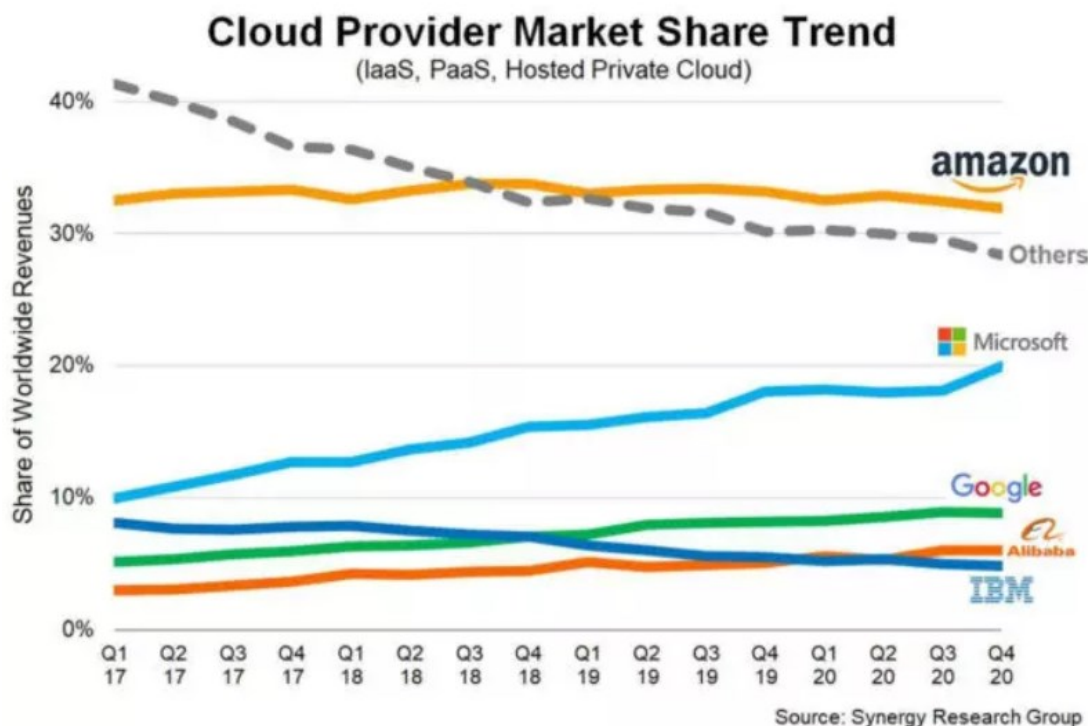


Figura 18: principali piattaforme cloud ultimi anni [25]

A questo punto è interessante conoscere alcune delle piattaforme attualmente più diffuse tra quelle proposte dai giganti del mondo tecnologico.

4.3.1 Google Cloud Platform



Figura 19 : Google Cloud Platform [26]

Google Cloud IoT Core è una piattaforma intelligente per l'IoT che consente di connettere e gestire in maniera facile e sicura i dispositivi IoT e i dati ad essi associati. Questi dati possono essere elaborati e analizzati in tempo reale, consentendo all'utente di intervenire in caso di bisogno. La piattaforma consente anche di eseguire degli aggiornamenti e supporta diversi sistemi operativi. Le componenti principali sono due: il device manager e il protocol bridge. Il device manager consente la connessione e la gestione in maniera sicura dei singoli dispositivi, stabilendone l'identità e impostando il meccanismo di autenticazione al momento della connessione. Il protocol bridge invece contiene tutti i protocolli necessari alla connessione di ogni dispositivo. Cloud IoT Core fa parte di un sistema più grande, il Google Cloud Platform, che offre diverse tipologie di servizi per la connessione dei dispositivi e la gestione e l'analisi dei dati.

Google Cloud Platform (GPC), offerto da Google, è una suite di servizi di cloud computing che gira sulla stessa infrastruttura che Google utilizza internamente per i suoi prodotti . È il classico esempio di cloud come parte di una più ampia suite di applicativi, con particolare riferimento alle

aziende. La forza della soluzione è infatti proprio quella di poter dialogare direttamente con servizi pensati per il mercato consumer.

I vantaggi dell'utilizzo di Google Cloud Platform sono molteplici, in particolare:

- il servizio, completamente scalabile, non necessita di investimenti iniziali (il costo addebitato corrisponde al reale utilizzo);
- l'infrastruttura a disposizione degli utenti è la stessa che utilizza Google per offrire i suoi servizi (Youtube, Gmail e rete di ricerca);
- l'infrastruttura poggia su una delle reti in fibra ottica tra le più avanzate, veloci e scalabili;
- i dati caricati nel sistema sono ridondati grazie a mirroring automatico tramite diversi dispositivi hardware e software che sono sempre all'avanguardia grazie ai costanti aggiornamenti

Le componenti principali sono Google App Engine, Google Compute Engine e Google Kubernetes Engine. La prima serve a gestire le applicazioni voluminose senza subire crash o down del server. Essendo una PaaS non serve nessuna configurazione se non il proprio codice personale di accesso. La seconda fornisce invece strumenti personalizzabili in base alle esigenze, indicata per chi ha un po' di padronanza in questo ambito. La terza, invece, è pensata per la gestione di cluster in grande scala, ideale per i veri esperti del settore.

Google ha creato una massiccia infrastruttura di data center progettati per scalabilità e tempi di risposta minimi con una capacità di calcolo enorme, che apre ai più svariati utilizzi in ambito aziendale. Con Virtual Private Cloud (VPC) è possibile connettere le proprie risorse o isolarle l'una dall'altra. Attraverso Cloud Delivery Network (CDN) il contenuto viene distribuito attraverso punti di presenza a livello edge e il servizio di Cloud Load Balancing può impedire il sovraccarico di una posizione bilanciando il traffico HTTP, TCP/SSL e UDP. Il Cloud Load Balancing di Google è lo stesso utilizzato dal motore di ricerca e gestisce fino a un milione di query al secondo. La parte di storage è composta da Cloud Datastore (per salvare dati non relazionali NoSQL), Cloud SQL (per gestire database relazionali MySQL) e Cloud Storage (un servizio di stoccaggio duraturo e altamente disponibile). In generale si tratta di una visione del cloud storage come parte di una più ampia suite di applicazioni web. In tal senso l'esempio concreto è certamente Google Drive, ormai confluito all'interno della suite Google One. Quest'ultima, oltre a Gmail e Calendar, offre anche Documenti (il corrispettivo di Word), Fogli (il corrispettivo di Excel), Presentazioni (il corrispettivo di Power Point), solo per citarne alcuni. Tutti servizi che sfruttano il cloud storage per un corretto

funzionamento, garantendo all'utente finale (privato o azienda che sia) una perfetta complementarità. Infine, Service Directory offre un unico posto in cui pubblicare, individuare e connettere tutti i servizi delle applicazioni, riducendo la complessità operativa e di gestione. Questo permette di migliorare la gestione dell'inventario su larga scala e ottenere informazioni sui servizi in tempo reale, sia che si disponga di pochi endpoint o di migliaia. Per quanto riguarda identità e sicurezza, entrambe esigenze essenziali nell'utilizzo del cloud, i servizi di Google si articolano in strumenti come Cloud Key Management per la gestione di chiavi di crittografia simmetriche e asimmetriche, nonché in servizi per l'individuazione, la classificazione e la protezione dei dati sensibili (Cloud Data Loss Prevention). Google Cloud Platform è dunque uno dei migliori servizi cloud attualmente disponibili sul mercato grazie al suo eccellente ecosistema digitale. Alcuni dei settori ed ambiti in cui Google Cloud Platform ha ottenuto un enorme successo sono i servizi finanziari, i social media, il gaming ed il retail. [26]

4.3.2 Microsoft Azure



Figura 20: Microsoft Azure [27]

Anche la Microsoft Corporation si è interessata al mondo dell'IoT e ha sviluppato la piattaforma cloud Azure che consente a sviluppatori e professionisti di gestire diversi tipi di servizi e soluzioni. Tramite Azure è possibile monitorare da remoto i dati raccolti dai device connessi e attivare avvisi e azioni automatiche. Inoltre vi è anche la funzionalità di manutenzione predittiva, per cui si può venire a conoscenza di potenziali problemi ancor prima che questi si verificano analizzando i flussi di dati raccolti da device e sensori. Le informazioni sui dati possono anche essere utilizzate come base per la creazione di app intelligenti. Nato inizialmente solo come servizio PaaS, si è evoluto

negli anni fornendo anche strumenti per la gestione dell'infrastruttura Cloud (IaaS). I componenti e i servizi possono essere raggruppati in tanti sottoservizi, dove i principali sono:

-Compute: fornisce un ambiente computazionale che consiste in un insieme di siti web, servizi Cloud e macchine virtuali che offrono un controllo totale sul server nel Cloud e lavorano per mantenere le richieste dell'utente. Presenta un modello scale-as-you-go grazie all'utilizzo di istanze condivise e riservate per una migliore distribuzione del carico di lavoro e per migliorare le performance. I servizi Cloud hanno il compito di gestire e distribuire applicazioni in cui Azure lascia all'utente il controllo di cui ha bisogno, compresa la gestione dell'affidabilità e i compiti relativi all'amministrazione. Il codice viene poi eseguito in macchine virtuali differenti da quelle che si possono creare con Azure Virtual Machines.

- Data Management: fornisce diversi modi per immagazzinare e gestire numerosi tipi di dati tramite appositi database che hanno tutte le caratteristiche chiave di un DBMS e possono essere acceduti tramite diverse tecnologie. Alcune delle caratteristiche più importanti sono il backup automatico e la possibilità di creare punti di ripristino.
- Networking: fornisce numerosi servizi che utilizzano reti dedicate attraverso il Cloud come Virtual Network o Express Route. Azure Virtual Network permette all'utente di avere a disposizione una rete privata nel Cloud in modo che diversi servizi Cloud possano comunicare con altri. Express Route utilizza una Virtual Network ma incanala le connessioni in una linea dedicata molto più veloce della rete Internet tradizionale, e può essere utilizzato in caso si necessiti di una larghezza di banda importante o per questioni di sicurezza.
- Identity & Access: fornisce servizi per l'identificazione, per aiutare a tracciare l'identità degli utenti di un'applicazione e per selezionare le azioni che un utente può compiere.
- Mobile: fornisce servizi di supporto per i clienti che lavorano con applicazioni per il mondo mobile per immagazzinare i dati nel Cloud, autenticare gli utenti e inviare notifiche push. Si possono sfruttare anche librerie per Android, iOS, HTML/JavaScript. Grazie all'efficiente meccanismo di scalabilità si può gestire facilmente il traffico in ascesa per l'applicazione.
- Backup: servizi di salvataggio dei dati con possibile ripristino in caso di esigenza, con la possibilità di lavorare anche con file criptati e/o compressi. E' capace di monitorare la sicurezza di un servizio così da poter automatizzare il processo di recupero in caso di errore critico.

- Big Compute & Big Data: fornisce servizi di gestione dei dati di grosse dimensioni tramite tecnologie Microsoft per poterli analizzare e produrre risultati di interesse sfruttando le macchine virtuali. Aiuta il processo di analisi su una enorme quantità di dati non gestibile da parte di un database relazionale o su dati non relazionali. Azure è anche in grado di disassociare o associare risorse collegate all'applicazione, come ad esempio il database associato.

L'elasticità e la scalabilità che Azure fornisce possono essere perfezionate sia manualmente che in maniera automatica. In pratica le esigenze di elasticità sono conseguite tramite l'aggiunta o la rimozione di istanze di macchine virtuali in base al carico di lavoro richiesto in un certo momento, secondo determinati criteri inerenti all'utilizzo di CPU e la disponibilità di altre risorse. Un'opzione molto utile ed attuata con ottimi risultati è quella di pianificare la scalabilità automatica solo in alcuni orari della giornata in modo da migliorare le performance e ridurre i costi senza la presenza di qualcuno che possa settare una scalabilità automatica.[27]

4.3.3 Amazon Web Services



Figura 21: Amazon Web Services [28]

Amazon AWS è un servizio web che fornisce una considerevole capacità computazionale come servizi di storage, database, rete e tantissimi altri servizi compreso quello DNS (Domain Name System) che lo rende un sistema con funzionalità sia IaaS sia PaaS. Inizialmente veniva utilizzata esclusivamente dalla stessa Amazon per i propri servizi interni fino al 2006, anno in cui l'azienda decise di distribuire i servizi anche all'esterno sotto forma di servizi a pagamento. Fornisce un controllo completo delle risorse e permette di lavorare direttamente sull'ambiente Amazon. Riduce il tempo richiesto per far partire le nuove istanze server, permette di scalare velocemente le capacità richieste, sia aumentandole che diminuendole al variare del carico di lavoro richiesto. Alcune delle fondamentali caratteristiche sono:

- Comportamento elastico: permette di aumentare o diminuire la capacità computazionale tramite pochi passaggi e in tempi brevi, fornendo anche un meccanismo automatico di scalabilità.
- Controllo completo: l'utente può interagire con ogni istanza, può stopparla e riavviarla in qualsiasi momento attraverso delle apposite API web.
- Servizio di Hosting flessibile: Amazon permette di configurare la quantità e la capacità di tutte le risorse scelte, compreso il sistema operativo da utilizzare (Linux o Windows Server).
- Affidabile: il servizio gira sull'infrastruttura di rete e nei data center di Amazon
- Improntato alla sicurezza: lavora in combinazione con Amazon VPC (virtual private cloud) riuscendo in questo modo a far girare le istanze su di una sezione logicamente isolata a cui è possibile associare un indirizzo IP o tenerla privata.
- Economico: include il modello Pay-per-use con tariffe orarie e senza contratti a lungo termine.

L'enorme potenzialità di calcolo per l'AWS proviene dalla importante catena di data center di cui Amazon dispone (infatti alcune stime parlano di oltre 1.5 milioni di server dislocati per tutto il mondo). Viene permesso all'utente di scegliere da quale regione vogliamo acquistare i servizi: la distanza dalla regione influirà notevolmente sulla latenza e sul tempo di risposta dei server e quindi sulla velocità dell'intero sistema Cloud. E' possibile raggruppare gli innumerevoli servizi offerti da Amazon nel seguente modo:

- Compute: vi fanno parte tutti quei servizi legati alle risorse di calcolo, che permettono la gestione di server virtuali. Comprende servizi come AutoScaling, Elastic Load Balancing, AWS Lambda, Container Service.
- Storage & Content delivery: riguarda servizi di memorizzazione di dati di grandi dimensioni, file multimediali, backup e recupero dei dati. Ne fanno parte servizi come S3, Glacier, Elastic Block Store, ecc...
- Database: riguarda la gestione di diverse tipologie di database di grandi dimensioni. Ad esempio Amazon RDS è compatibile con i database relazionali, Amazon Aurora compatibile con i database MYSQL e Amazon DynamoDB con database NOSQL.
- Analytics: servizi utilizzati per analizzare il carico di lavoro in tempo reale e per comunicare questi dati a tutti i servizi AWS.

- Security and Identity: comprende servizi in grado di fungere da amministratori del sistema, ossia in grado di controllare gli accessi, gestire allarmi di vario tipo e monitorare le risorse utilizzate.
- Networking: per tutto ciò che a che fare con la rete, come la funzione DNS per poter accedere facilmente alle istanze direttamente dall'esterno .
- Deployment & Management: servizi che offrono ambiente di esecuzione e sviluppo di applicazioni.
- Application Services: servizi preconfigurati senza bisogno di acquistare ulteriori istanze o storage.
- Mobile Services: servizi per sviluppo e testing applicazioni, per la notifica di eventi via messaggio.
- Enterprise Applications: servizi per le aziende, in particolare comprendono la virtualizzazione di ambienti desktop.
- Internet of Things: servizi che permettono di connettere facilmente dispositivi in modo che possano interagire tra di loro. Un grande vantaggio di questa piattaforma è che fornisce la possibilità di interagire con alcuni dispositivi anche quando questi sono offline. Questo perché il sistema memorizza lo stato più recente dell'oggetto, mettendo il dato a disposizione delle varie applicazioni che potranno consultarlo in qualsiasi momento. Le modifiche vengono poi passate al dispositivo nel momento in cui si ricollega al cloud. Nella pratica è come se si creassero delle versioni virtuali dei dispositivi nel cloud, che prendono il nome di "shadow" e che contengono appunto l'ultimo stato noto.

Un elemento fondamentale per l'infrastruttura di AWS è l'Amazon Elastic Compute Cloud (EC2), un servizio web con capacità computazionale dinamica in Cloud, quindi ridimensionabile secondo i bisogni effettivi dell'utente. Amazon per tutti i suoi servizi Cloud offre gratuitamente un servizio di Auto Scaling, servizio che migliora ulteriormente la disponibilità dell'applicazione e permette lo scaling automatico eseguendo tutte le istanze richieste dal cliente. In questo modo vengono mantenute le performance dell'applicazione anche in casi di notevole aumento dei carichi di lavoro, infatti la capacità sarà aumentata in caso di bisogno e decrementata durante i momenti di calma allo scopo di ridurre i costi. Aumentare la capacità per Amazon significa allocare gruppi di istanza EC2 con una specifica politica di scalabilità definita dall'utente relativamente alle macchine

virtuali. Il servizio di AutoScaling lavora insieme ad altri due servizi, uno per il bilanciamento del carico di lavoro (Elastic Load Balancing o ELB) ed uno per il monitoraggio delle risorse e il lancio degli allarmi (Amazon CloudWatch). L'integrazione con CloudWatch è di notevole importanza in quanto permette di impostare una scalabilità dinamica definendo logiche basate sul monitoraggio dell'utilizzo delle risorse. L'azione di controllo nella pratica consiste nell'invio di comandi al servizio di AutoScaling che eseguirà delle istruzioni sul gruppo di istanze EC2. L'integrazione con l'Elastic Load Balancer risulta importante per migliorare la tolleranza agli errori del sistema e si occupa principalmente di distribuire in maniera automatica il traffico in entrata sulle istanze EC2. Un'altra funzione di questo servizio è intervenire sulla sicurezza di rete: l'ELB si pone a metà tra la rete internet esterna e quella interna Amazon su cui girano le istanze, analizza il traffico e si occupa di cifrare e decifrare i dati in passaggio tra queste reti. Per poter applicare tutto ciò bisogna aggiungere istanze ai gruppi di sicurezza, questi ultimi fungono da firewall virtuale interno in grado di decidere se il traffico è consentito o meno in base a delle regole definite preventivamente. [28]

4.3.4 ThingSpeak



Figura22: logo ThingSpeak [29]

ThingSpeak è una piattaforma di analisi IoT di MathWorks che consente di raccogliere, visualizzare e analizzare flussi di dati in tempo reale nel cloud. ThingSpeak offre visualizzazioni real time dei dati pubblicati dai vari dispositivi o apparecchiature connesse alla rete, con l'ulteriore possibilità di eseguire codice MATLAB per fare analisi ed elaborazione online di tutti i dati non appena questi sono disponibili nella piattaforma.

Thingspeak accelera lo sviluppo di tutti quei sistemi che necessitano dell'analisi di dati per studiarne e/o compravare la fattibilità o validità, risultando particolarmente utile in tutti i progetti "proof of concept".

Con ThingSpeak è possibile memorizzare e analizzare i dati nel cloud senza configurare i server web, e si possono creare sofisticati avvisi e-mail basati su eventi che si attivano in base ai dati in arrivo dai dispositivi collegati.

I "canali" sono i contenitore dei dati provenienti dai dispositivi connessi e al suo interno vanno definite le caratteristiche di un progetto.

La visualizzazione di ogni canale può essere resa pubblica o privata; nel caso di un canale privato si ha la possibilità di condividere l'accesso ai dati del canale stesso attraverso apposite APIkeys: "write" per scrivere dati sul canale e "read" per consentire ad altri utenti di leggere i dati sul canale e le rispettive analisi effettuate su di essi.

Un aspetto importante di questa piattaforma è che fornisce una serie di servizi aggiuntivi, detti Thingspeak app, grazie ai quali si possono effettuare azioni e aggiuntive personalizzabili sui dati memorizzati.

Lo scambio di dati tra dispositivi e ThingSpeak può avvenire tramite apposita API Rest o MQTT grazie al fatto che la piattaforma rende disponibile un broker.[29]

4.3.5 nodeRed

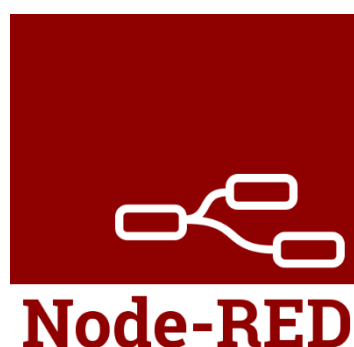


Figura 23 : logo nodeRed [30]

E' un tool sviluppato da IBM e rilasciato sotto licenza Apache 2.0 dedicato alla gestione dei flussi di dati. E' costruito su Node.js ,un Runtime di JavaScript , e pertanto si presta a lavorare con messaggi strutturati in JSON. Sfruttando il paradigma "low-code" consente di sviluppare

applicazioni molto velocemente, considerando anche il facile utilizzo grazie alla funzionalità drag and drop che consente di creare ed unire facilmente i nodi ed il supporto della community.

Infatti fornisce un editor basato sul browser che semplifica il collegamento dei flussi (cioè la connessione di più nodi) di dati utilizzando una gamma di nodi predefiniti e che sono descritti anche in HTML, pertanto si può personalizzare il layout della visualizzazione dei dati stessi.

I vari flussi che vengono programmati sono composti da "flow object", raggruppabili in "subflow", che si occupano della creazione e della gestione dei nodi che governa.

La distribuzione ed il mantenimento dei nodi sfrutta il Runtime di Node.js , convertendo in JSON tutti i vari nodi. Infatti nodered si basa sul protocollo MQTT , lo standard di riferimento della comunicazione per l'IoT, dove le informazioni tra i nodi di ogni flusso viaggiano sotto forma di messaggi alfa-numeric o di altro tipo (json appunto), contenuti in appositi blocchi.

Un programma nodeRed pertanto si sviluppa come un flusso di messaggi che attraversa una sequenza di nodi, ognuno capace di implementare una determinata elaborazione. I messaggi vengono passati in oggetti Javascript chiamati msg ,ognuno dotato di un proprio identificatore.

NodeRed inoltre offre la possibilità di creare nuovi nodi, ognuno dei quali rappresenta un microservizio, e di personalizzare ,tramite funzioni Javascript , il comportamento dei vari nodi già esistenti. Oltre al suo scarso impatto sulle risorse è capace di funzionare all'interno di contenitori Docker, quindi opera su diverse piattaforme.[30]

5 GESTIONE DEI DATI

5.1 Interazione dei sensori iot con web app

I sensori IoT sono piccoli pezzi hardware in grado di rilevare dati come temperatura, umidità, suono, e molto altro ancora. IoT è un insieme di tecnologie che comprende i sensori che registrano i dati prodotti, le applicazioni web che le interpretano per restituire informazioni e azioni intelligenti, le infrastrutture che stanno nel mezzo e che rendono possibile l'incontro del reale con il virtuale. Il passaggio dei dati dai sensori ad una applicazione web per la visualizzazione ovviamente non è immediato: non basta, cioè, prendere i sensori e collegarli ad un lettore intelligente per vederli pubblicati in internet. È infatti necessario che i dati siano inviati ad uno specifico broker che li archivia, gestisce la coda e la possibilità che un sensore non parli per assenza di connessione e tanti altri aspetti tecnici.[31]

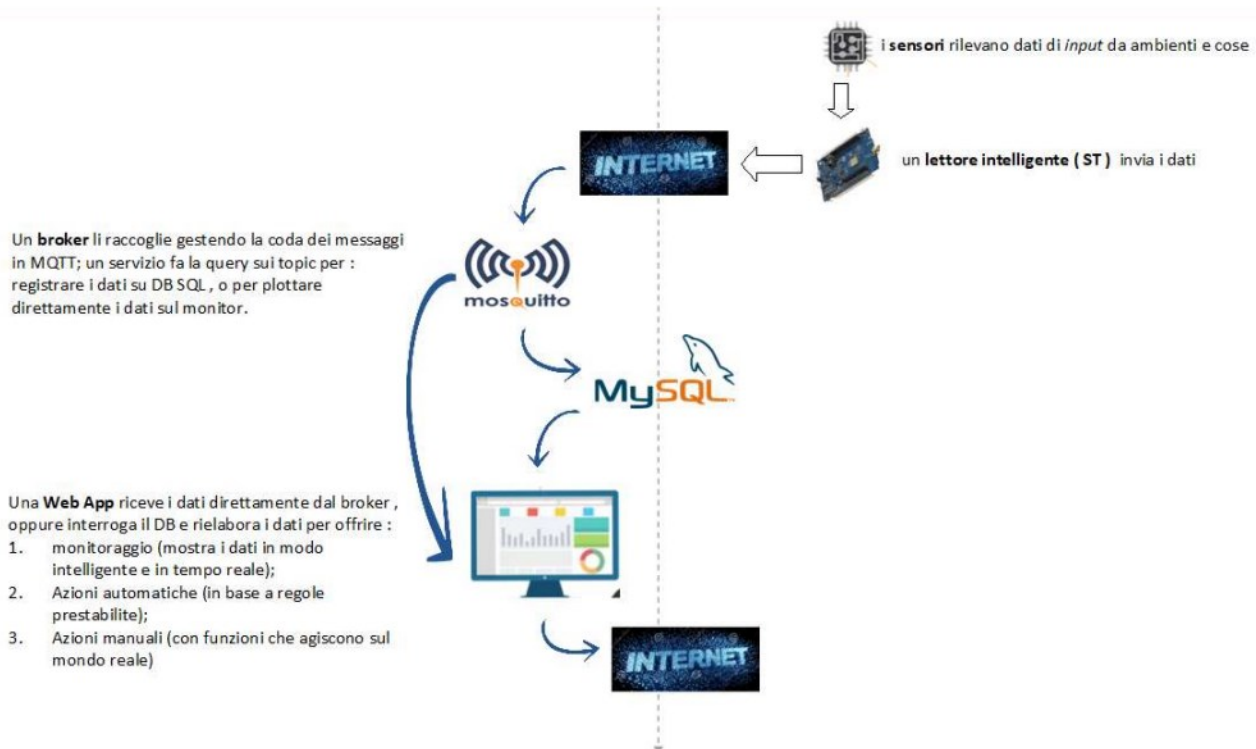


Figura 24: Funzionamento dell'IOT in una web app [31]

Esistono quindi problematiche specifiche legate alla gestione dei dati dell'IoT affinché questi passino correttamente dal mondo reale al web. Questo però non basta ad aggiungere le componenti che si occupano di intelligenza, elaborazione ed informazione, compito che spetta invece alle web app.

Prima dell'IoT i dati su cui lavoravano le web app erano generati dalle persone attraverso input manuali o da altre applicazioni spesso integrate. L'Internet of Things offre dunque una fonte di dati importantissima ed inedita perché si tratta di dati generati dalla realtà in modo automatico tramite appunto i sensori.

La struttura generica di un'applicazione di questo tipo e la relativa architettura software è mostrata di seguito:

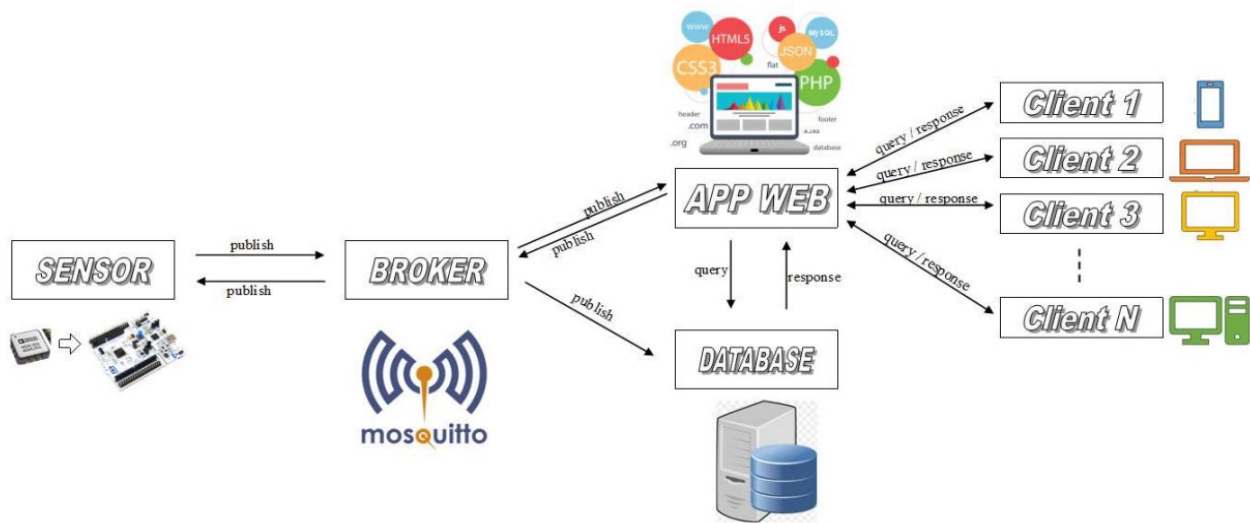


Figura 25 :Struttura interazione tra una web app e sensori IOT [31]

I livelli su cui si sviluppa sono:

- Presentazione (lato client): costituisce l'interfaccia utente dell'applicazione e corrisponde al browser web. Le principali tecnologie sono HTML e CSS.
- Applicazione (Servente Web): è il livello logico della web-app, la componente elaborativa. Tra le tecnologie più diffuse lato server troviamo PHP mentre lato client il linguaggio di scripting Javascript
- Dati (Servente RDBMS): consente di modellare e gestire il contenuto informativo dell'applicazione tramite un database
- Broker: è colui che riceve i dati dai sensori e li smista a tutti coloro che si sottoscrivono a determinati topic. [31]

Nelle sezioni successive saranno illustrati più nel dettaglio alcuni dei principali protocolli per lo scambio di dati utilizzati nell'IoT.

5.2 MQTT

MQTT (Message Queue Telemetry Transport) è un protocollo ISO Standard realizzato nel 1999 da Andy Stanford-Clark e Arlen Nipper. MQTT per scambiare informazioni sfrutta un meccanismo di pubblicazione e sottoscrizione (publish-subscribe) di messaggi tramite un apposito message broker, in grado di gestire migliaia di client contemporaneamente. Si tratta di un'ottima alternativa al modello client/server, dove ad instaurare una connessione diretta con il server è il client. In MQTT se un client vuole comunicare con un altro pubblica un messaggio su un certo argomento (detto topic) sul message broker. Un client MQTT è un qualsiasi dispositivo che esegue una libreria MQTT e si collega a un broker MQTT su una rete. Il message broker ha il compito di filtrare e distribuire le comunicazioni tra publisher e subscriber e tiene conto delle sessioni di tutti i client. [31]

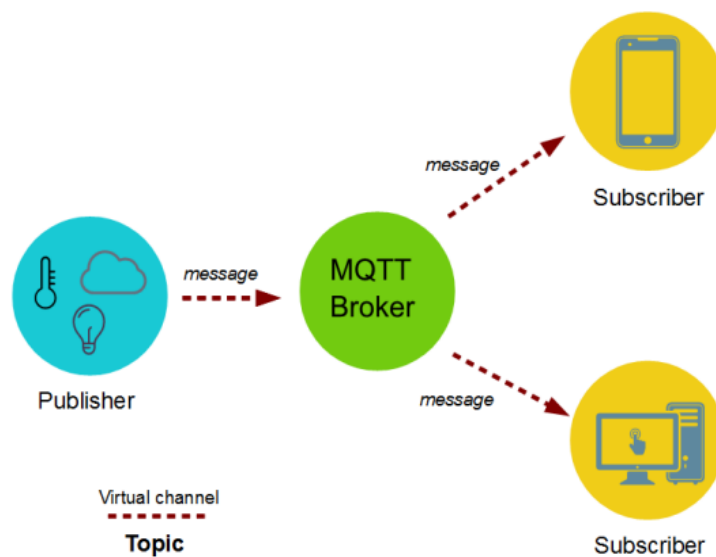


Figura 26: Processo MQTT [32]

Un'altra funzione del broker è l'autenticazione e l'autorizzazione dei client: bisogna prestare attenzione alla configurazione del broker, perché spesso è il componente connesso direttamente ad Internet e quindi il più esposto ad attacchi informatici. Ogni client può iscriversi a molteplici topics e ogni volta che viene pubblicato un nuovo messaggio in un determinato topic, il message broker lo distribuisce a tutti i client iscritti a quel determinato topic. Il Topic è una stringa UTF-8 con struttura simile a quella di una directory e i livelli dei topics, sono separati da uno slash (/). Per

la trasmissione di messaggi tramite MQTT, è necessario conoscere IP, porta del broker e i parametri necessari per l'autenticazione.

Prendendo come esempio un sistema di monitoraggio di parametri ambientali, per inviare i dati rilevati del sensore, il sistema dovrà:

1. Instaurare una connessione con l'MQTT broker;
2. Pubblicare in un determinato topic le rilevazioni effettuate

Per accedere in tempo reale alle rilevazioni che vengono trasmesse dal sistema di monitoraggio, si dovrà:

1. Instaurare una connessione con l'MQTT broker;
2. Sottoscrivere al topic sul quale vengono pubblicate le rilevazioni ed aspettare la ricezione dei dati.

I parametri da inviare al topic di pubblicazione devono essere in formato JSON (JavaScript Object Notation). È un semplice formato per lo scambio di dati, che risulta facile da scrivere per le persone ed altrettanto facile per le macchine analizzarne la sintassi. JSON è un formato di testo completamente indipendente dal linguaggio di programmazione, ma utilizza convenzioni conosciute dai programmatori di molti linguaggi. Questa caratteristica fa di JSON il linguaggio ideale per lo scambio di dati per MQTT. JSON è basato su due strutture:

- Un insieme di coppie nome/valore;
- Un elenco ordinato di valori.

Queste sono strutture di dati universali e tutti i linguaggi di programmazione moderni li supportano in entrambe le forme. [33]

Il broker è la parte fondamentale nel protocollo MQTT e per collegarsi, nel caso in cui sia abilitata l'autenticazione, è necessario fornire le credenziali di accesso. Nella configurazione base è invece sufficiente conoscere indirizzo IP e porta TCP. Per iniziare la connessione con il broker, il Client dovrà mandare, prima di tutto una richiesta Connect Command, dopodiché, il broker risponderà con lo stato della richiesta. Una volta stabilita, la connessione, se andata a buon fine, rimarrà aperta finché non ci sarà una richiesta di disconnessione o un problema di comunicazione. Ogni Client che si collega al broker avrà un ClientId per essere identificato univocamente tra i Client del broker. Un publisher senza nemmeno un subscriber non ha senso, in quanto, non ci sarebbe

nessun subscriber pronto a ricevere il messaggio, quindi è logico attendersi che ci sia almeno un Client che faccia la sottoscrizione per quel topic. Per effettuare la richiesta di sottoscrizione al topic, bisognerà mandare il messaggio Subscribe Request. Per notificare una conferma dell'avvenuta sottoscrizione, il broker risponderà con Subscribe ACK, che conterrà lo stato della richiesta con il livello di QoS. Da questo momento il subscriber riceverà tutti i messaggi spediti su quel topic da qualunque publisher connesso al broker. È inoltre possibile fare una richiesta di Unsubscribe in cui si specifica da quali dei topic si vuole cancellare la propria sottoscrizione. Una volta stabilita la connessione, il publisher potrà spedire messaggi. Ogni pacchetto conterrà: PacketId, topicName, QoS, RetainFlag, Dupflag e Payload. In particolare:

- TopicName è una stringa del tipo Topic/Sottotopic2/Sottotopic3, con lo slash che indica la sottocategoria del Topic, così da avere una gerarchia.
- QoS è un numero che indica il livello di qualità del servizio del messaggio e definisce fino a che punto il broker o il client si spingerà per garantire la ricezione di un messaggio.
- Retainflag definisce se il messaggio dovrà essere salvato dal broker, come l'ultimo messaggio per quel topic.
- Payload è il contenuto del messaggio
- PacketId è un identificativo univoco dei messaggi per Client, in modo da identificare i vari messaggi di un Client con QoS maggiore di 0.
- Dupflag per identificare i messaggi duplicati, cioè i messaggi che sono stati rispediti in quanto non è stata ricevuta la conferma di ricezione.

Una volta spedito il messaggio dal publisher, e ricevuto dal broker, sarà compito di quest'ultimo consegnarlo ai subscriber. Il publisher, in base al QoS scelto, potrà solo sapere se il messaggio è giunto al broker, ma non saprà se il messaggio è arrivato a tutti i subscriber e a quanti. I Client hanno la possibilità di scegliere un livello di affidabilità di consegna del messaggio: possono scegliere di non dare importanza alla consegna, di essere sicuri che quel messaggio arrivi a destinazione, eventualmente inviando copie dello stesso, oppure, di volere che arrivi esattamente una sola volta. Ci sono tre livelli di affidabilità (QoS), dal più basso partendo da zero, e più veloce, al più alto, ma più lento:

- (0) Invio solamente una volta (At most once)
QoS=0 è il livello più basso in cui non si hanno garanzie sulla ricezione, il messaggio non verrà memorizzato o ritrasmesso dal sender.

- (1) Invio almeno una volta (At least once)

QoS=1 con questo livello viene garantita la ricezione del messaggio. Il sender mantiene in memoria il messaggio finché non riceverà una conferma di ricezione, e se non la riceve entro un determinato tempo, il messaggio dato per perso verrebbe rispedito. È quindi possibile che lo stesso messaggio venga spedito più di una volta, e consegnato due volte, nel caso in cui il primo non fosse andato perso, ma ci fosse stato solo un ritardo. In questo livello si usa il campo packetId così da identificare i pacchetti e sapere quali sono stati ricevuti. In caso scadesse il timeout, il sender spedirebbe nuovamente il messaggio e questo verrebbe contrassegnato con il Dupflag, anche se in QoS=1 non è utilizzato.

- (2) Ricevuto esattamente una volta (Exactly once)

I messaggi inviati a QoS=2 vengono sempre consegnati esattamente una volta. È la modalità di trasferimento più affidabile e ma anche più lenta in una rete MQTT. I messaggi vengono archiviati localmente presso il mittente e il destinatario finché non vengono elaborati. Tra il mittente e il destinatario vengono eseguite almeno due coppie di trasmissioni prima che un messaggio venga eliminato dal mittente. Il messaggio può essere elaborato dal destinatario dopo che la prima o la seconda trasmissione è stata completata, purché non lo elabori più di una volta. La prima coppia di trasmissioni comprende il mittente che invia il messaggio e ottiene la conferma che il destinatario lo ha memorizzato. Se il mittente non riceve la conferma, i messaggi vengono ritrasmessi fino a quando non viene ricevuto un riconoscimento. Nella seconda coppia di trasmissioni, il mittente informa il destinatario che può completare l'elaborazione del messaggio inviando un messaggio detto "pubrel" che verrà inviato ripetutamente fino a quando non ne viene confermata la ricezione e a quel punto il mittente cancella il messaggio.

5.3 XMPP

XMPP sta per Extensible Messaging and Presence Protocol, un protocollo per la comunicazione in tempo reale che utilizza XML (Extensible Markup Language) come linguaggio base per scambiare le informazioni. Il funzionamento di XMPP si basa sullo scambio di informazioni fra un client e un server. Questo significa che, prima di poter usare XMPP per creare una connessione, è necessario installare un client XMPP gratuito. Il client consente lo scambio di messaggi, la visualizzazione degli

aggiornamenti di stato degli utenti e l'uso di chat e chiamate di gruppo comunicando con i provider di messaggistica ed e-mail che supportano il protocollo XMPP, consentendo anche di inviare appositi messaggi perché i server XMPP funzionano in modo simile ai server di posta elettronica e fungono da interfaccia di comunicazione per lo scambio di messaggi fra utenti che usano tecnologie con protocolli simili a XMPP .

Si tratta di un protocollo aperto, quindi chiunque lo può implementare e personalizzare secondo le proprie esigenze. Caratteristica unica di XMPP è quella che viene chiamata "transport" e che offre la possibilità, tramite opportuni gateway presenti su Internet, di interconnettere servizi differenti.

Con XMPP sul client non serve avere un software per ogni tipo di servizio, ma basta che esistano i gateway per quel servizio. In altre parole un client che supporta XMPP può collegarsi a tutti i servizi che supportano XMPP anche senza avere un software specifico per quel servizio.

Le applicazioni che si possono implementare usando i servizi messi a disposizione da XMPP riguardano principalmente chat, instant messaging, sistema di geolocalizzazione, sessioni peer-to-peer, crittografia, autenticazione e ovviamente il cloud computing.

Un altro aspetto di particolare rilevanza è il fatto che esistono client per tutti i principali sistemi operativi ed è compatibile con moltissimi linguaggi di programmazione.[34]

5.4 CoAP

CoAP (Constrained Application Protocol) è un protocollo aperto e leggero per dispositivi IoT. Uno degli obiettivi principali di CoAP è la realizzazione di un protocollo WEB adatto alle esigenze di dispositivi con risorse limitate a livello computazionale ed energetico.

Infatti è molto simile al linguaggio HTTP, ma è stato appositamente riprogettato per soddisfare i requisiti dei dispositivi alimentati a batteria che presentano risorse hardware e software limitate. Per questo motivo CoAP utilizza pacchetti di dati molto piccoli, risulta più semplice rispetto a HTTP e i messaggi scambiati presentano una dimensione inferiore. CoAP è progettato per essere facilmente convertito in HTTP per un'integrazione semplificata con il Web, entrambi si basano infatti su un modello client/server: il client invia una richiesta al server e il server restituisce una risposta con eventuale rappresentazione della risorsa.

La sicurezza delle comunicazioni è fornita dallo standard Datagram Transport Layer Security (DTLS), che funziona su UDP e fornisce un alto livello di sicurezza. Inoltre, a differenza di HTTP,

proprio in virtù del fatto che sfrutta UDP, lo scambio di messaggi in CoAP avviene in modo asincrono.

L'architettura CoAP è divisa in due livelli: uno sui messaggi ed uno sull'interazione "richiesta/risposta" che utilizza i metodi tipici dell' HTTP. Il livello Messaggio fornisce affidabilità per i messaggi digitati confermabili assicurando il controllo end-to-end, la ritrasmissione in caso di smarrimento, la conferma di ricezione del messaggio, un token per identificare ogni interazione client/server ed un'etichetta per individuare eventuali messaggi duplicati. Invece il livello richiesta/risposta sfrutta i metodi classici di HTTP, ovvero Get, Post, Put e Delete, per creare l'interazione client/server.[34]

5.5 RESTful HTTP

REST è l'acronimo di REpresentational State Transfer ed è uno stile architetturale per sistemi distribuiti, ovvero un'astrazione degli elementi che compongono l'architettura stessa, e rappresenta un sistema di trasmissione dati basato su HTTP. Adotta il principio della separazione dei concetti secondo cui si cerca di separare un sistema in moduli distinti in modo tale che ogni modulo svolga una certa funzionalità.

I sistemi REST proprio perchè si basano su HTTP non prevedono il concetto di sessione, ovvero sono stateless, e sfruttano il modello di interazione client/server. Il vincolo Client-Server richiede che il server offra una o più funzionalità e ascolti le richieste di possibili client. Un client invoca il servizio messo a disposizione dal server inviando il corrispondente messaggio di richiesta e il servizio lato server respinge la richiesta o esegue l'attività richiesta prima di inviare un messaggio di risposta al client. La gestione delle eccezioni è delegata al client.

La comunicazione tra client e server deve essere senza stato tra le richieste :ciò significa che ogni richiesta da parte di un client dovrebbe contenere tutte le informazioni necessarie per il servizio per comprendere il significato della richiesta. Tutti i dati sullo stato della sessione dovrebbero quindi essere restituiti al consumatore del servizio alla fine di ciascuna richiesta. Pertanto le interazioni passate non hanno influenza sulle interazioni future. In questo modo il sistema può scalare molto più facilmente e sicuramente sarà molto più semplice perché elimina alla radice il problema della gestione e sincronizzazione delle sessioni utente.

Il funzionamento prevede una struttura degli URL ben definita che identifica univocamente una risorsa o un insieme di risorse e l'utilizzo dei metodi http specifici per diverse tipologie di azioni che si possono effettuare sui dati.

Mentre il termine REST viene utilizzato per indicare uno stile architetturale con delle caratteristiche e principi, il termine RESTful viene in genere utilizzato per fare riferimento a tutti quei servizi Web che implementano l'architettura REST.

Affinch' un'API sia considerata RESTful deve quindi rispettare i principali criteri dell'architettura REST indicati di seguito:

- architettura client-server
- comunicazione client-server di tipo stateless
- dati memorizzabili nella cache che ottimizzano le interazioni client-server
- interfaccia uniforme in modo che le informazioni vengano trasferite in un formato standard sfruttando risorse univocamente identificabili e manipolabili dal cliente.[34]

5.6 Database relazionali

Molto spesso i dati raccolti da sensori e dispositivi IoT vengono poi salvati e conservati su un database relazionale, che è un tipo di database di archiviazione che fornisce accesso a data points correlati tra loro. I database relazionali sono basati sul modello relazionale, un modo intuitivo e diretto di rappresentare i dati nelle tabelle. In un questo tipo di database ogni riga della tabella è un record con un ID univoco chiamato chiave. Le colonne della tabella contengono gli attributi dei dati e ogni record di solito ha un valore per ogni attributo, rendendo facile stabilire delle relazioni tra i vari dati. Il modello relazionale implica che le strutture logiche dei dati (tabelle di dati, viste e indici) siano separate dalle strutture di storage fisico. Grazie a questa separazione gli amministratori di database possono gestire lo storage fisico dei dati senza compromettere l'accesso a tali dati come struttura logica. Ad esempio, la ridenominazione di un file di database non implica la ridenominazione delle tabelle archiviate al suo interno. La distinzione tra logico e fisico si applica anche alle operations di database, che sono azioni esplicitamente definite che consentono alle applicazioni di manipolare i dati e le strutture del database. Le operazioni logiche

consentono a un'applicazione di specificare il contenuto di cui ha bisogno, mentre le operazioni fisiche determinano come accedere a tali dati e quindi eseguono l'attività. Per garantire che i dati siano sempre accurati e accessibili i database relazionali seguono regole di integrità specifiche. Nei primi anni dell'esistenza dei database ogni applicazione archiviava dati nella propria struttura unica. Qualora fosse stato necessario creare applicazioni per utilizzare tali dati, gli sviluppatori avevano bisogno di conoscere numerosi dettagli sulla struttura dei dati specifica, al fine di trovare i dati necessari. Queste strutture dati erano inefficienti, difficili da gestire e da ottimizzare per offrire buone performance delle applicazioni. Il modello di database relazionale è stato progettato per risolvere il problema di più strutture di dati arbitrarie. Il modello dati relazionale ha fornito una modalità standard di rappresentare e interrogare i dati che potrebbero essere utilizzati da qualsiasi applicazione. Fin dall'inizio, gli sviluppatori hanno riconosciuto che il punto di forza del modello di database relazionale è l'uso delle tabelle, ovvero un modo intuitivo, efficiente e flessibile per archiviare informazioni strutturate e accedervi. Con il passare del tempo, è emerso un altro punto di forza del modello relazionale, poiché gli sviluppatori hanno iniziato a utilizzare il linguaggio SQL (Structured Query Language) per scrivere dati in un database ed eseguirvi query. Basato sull'algebra relazionale, SQL offre un linguaggio matematico internamente coerente che semplifica il miglioramento delle performance di tutte le query di database. Un database relazionale può essere preso in considerazione per reperire qualsiasi informazione in cui i data points si correlano tra di loro e devono essere gestiti in modo sicuro, regolamentato e coerente. I database relazionali sono utilizzati sin dagli anni '70 e i vantaggi offerti continuano a renderli anche oggi il modello di database più accettato a livello globale. Il modello relazionale è anche il migliore modo per mantenere la coerenza dei dati tra applicazioni e copie di database. Nell'ambito di questo genere di coerenza dei dati, i database relazionali consentono di ottenere risultati eccellenti, assicurando che più istanze di un database contengano sempre gli stessi dati. Per altri tipi di database è difficile mantenere questo livello di coerenza in modo rapido con grandi quantità di dati. [35], [36]

6 GESTIONE INTELLIGENTE DELLE BATTERIE AL LITIO

Il termine “batteria” indica quell’involucro contenente diverse celle energetiche che sono le unità elettrochimiche alla base della batteria, delle volte viene impropriamente utilizzato per indicare la singola cella elementare che effettivamente compone la batteria. Queste possono essere collegate elettricamente fra loro per raggiungere le caratteristiche desiderate in termini di tensione, corrente erogabile o capacità. Una batteria agli ioni di litio è composta da molte celle singole, ognuna delle

quali è costruita allo stesso modo e contiene le seguenti componenti:

-Elettrodi positivi: il catodo della batteria agli ioni di litio è composto da un ossido metallico di litio che può contenere una quantità variabile di nichel, manganese e cobalto. Gli ossidi metallici vengono denominati anche metalli di transizione.

-Elettrodi negativi: l’anodo è per lo più realizzato in grafite.

-Elettrolita: per far sì che gli ioni di litio possano muoversi nella cella, e svolgere quindi la funzione di portatore di carica, l’elettrolita presente nella batteria deve essere priva d’acqua

-Separatore: il separatore delle batterie agli ioni di litio gestisce e assicura le reazioni elettrochimiche all’interno della batteria isolando i due elettrodi l’uno dall’altro, impedendo così il verificarsi di cortocircuiti interni. Allo stesso tempo fa sì che, solamente gli ioni di litio, possano passare da una parte all’altra e muoversi quindi tra l’elettrodo negativo e positivo, rendendo il sistema permeabile. Inoltre, il separatore garantisce lo scambio di gas nelle celle chiuse della batteria al litio. [37]

Il funzionamento delle batterie agli ioni di litio, si basa principalmente sul movimento costante del litio ionizzato tra gli elettrodi. Il flusso di ioni di litio bilancia il flusso di corrente esterno durante le fasi di caricamento e scaricamento, in questo modo gli elettrodi rimangono elettricamente neutri.

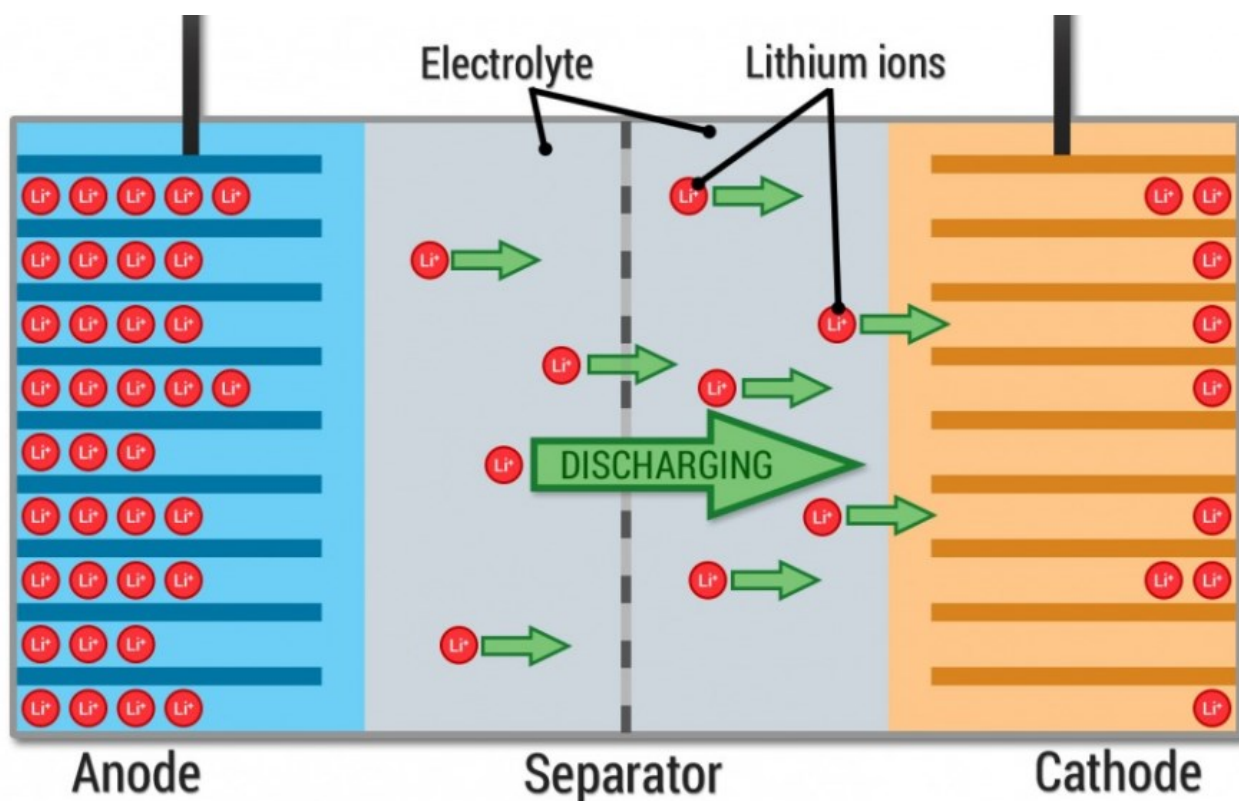


Figura 27: funzionamento batteria (dettaglio fase scarica) [38]

Se la batteria si scarica, ovvero la sua energia viene consumata da un dispositivo, ogni atomo di litio passa un elettrone agli elettrodi negativi. Con la ricarica attraverso il circuito elettrico esterno gli elettroni ritornano agli elettrodi positivi. Al contempo gli ioni di litio si muovono in egual misura dagli elettrodi negativi attraverso l'elettrolita e il separatore verso gli elettrodi positivi. Gli elettroni vengono inglobati negli elettrodi positivi, grazie ai cosiddetti ioni in metallo di transizione, fortemente ionizzati. Questo procedimento può essere differente in base al tipo di batteria. Diversamente dagli ioni di litio, questi ioni non sono mobili. Durante il caricamento delle celle degli accumulatori, gli atomi di litio non ionizzati si muovono dagli elettrodi positivi agli elettrodi negativi, attraverso il separatore. Qui vengono immagazzinati tra molecole di grafite.[37]

Il procedimento prende anche il nome di intercalazione; viene causato dal caricamento con corrente costante fino al raggiungimento della corrente nominale. Al raggiungimento della tensione di carica desiderata, la tensione nominale viene mantenuta, mentre il flusso di corrente diminuisce. Le celle sono i componenti di base della batteria: un certo numero di celle forma un modulo, e dall'unione di più moduli hanno origine le diverse batterie, la cui capacità totale è il risultato della somma di quelle individuali delle celle. Per valutare le prestazioni delle batterie bisogna considerare la sua capacità, ovvero la quantità di energia che può immagazzinare, il modo

e la rapidità con cui l'energia viene immagazzinata ed erogata ,tutte caratteristiche espresse da amperaggio (quantità di energia che può essere trasmessa in un secondo)e da voltaggio (corrisponde alla forza, dunque alla rapidità, con cui una quantità di energia apri a 1 Ampere viene trasmessa).Dunque, è facile comprendere come all'aumentare di voltaggio e amperaggio la batteria risulti più potente perché il flusso di corrente che entra o esce sarà maggiore per volume e intensità. Le sezioni successive descriveranno nel dettaglio gli aspetti legati alla gestione delle batterie agli ioni di litio, attualmente le più diffuse.

6.1 Gestione della batteria

Lo stoccaggio dell'energia svolge un ruolo importante nell'adozione di energia rinnovabile per aiutare a risolvere i problemi del cambiamento climatico. Le batterie agli ioni di litio (LIB) sono un'ottima soluzione per l'accumulo di energia grazie alle loro proprietà. Al fine di garantire la sicurezza e il funzionamento efficiente dei sistemi LIB, sono necessari sistemi di gestione della batteria (BMS). L'attuale design e funzionalità dei BMS soffre di alcuni inconvenienti critici, tra cui una bassa capacità computazionale e una limitata archiviazione dei dati. Recentemente, per ovviare a questi limiti, c'è stato un grande sforzo nella ricerca e nello sviluppo di BMS intelligenti utilizzando piattaforme cloud. Un BMS basato su cloud sarebbe in grado di risolvere i problemi di capacità computazionale e archiviazione dei dati negli attuali BMS, portando molti vantaggi per le future applicazioni delle batterie grazie anche ad algoritmi di batteria più accurati e affidabili consentendo lo sviluppo di altre complesse funzioni BMS. Poiché si prevede che la domanda di energia aumenterà in modo significativo in futuro, la necessità di energia rinnovabile, per ridurre l'impatto ambientale, è più alta che mai al giorno d'oggi come testimoniano i recenti avvenimenti legati all'approvvigionamento e alla sicurezza energetica. Con ciò, il ruolo dello stoccaggio di energia nella rete elettrica sta diventando sempre più importante per la società moderna sotto vari aspetti. Le batterie agli ioni di litio (LIB) hanno recentemente guadagnato un crescente interesse come eccellenti sistemi di accumulo di energia (ESS) grazie alla loro elevata densità di energia e potenza, alla lunga durata e alla bassa auto scarica. Negli ultimi anni infatti la capacità di stoccaggio dell'energia su larga scala è stata fornita annualmente dalle LIB sempre in quantità crescente. Le LIB sono utilizzate in molte applicazioni del mondo reale, come i veicoli elettrici (EV), i sistemi di accumulo di energia a batteria (BESS) e ovviamente anche per alimentare piccoli dispositivi portatili come laptop e smartphone. Il costante aumento dell'interesse verso le LIB ha

spinto la ricerca a concentrarsi sull'aumento della densità energetica e sulla riduzione dei costi. Nella maggior parte delle applicazioni del mondo reale il sistema di gestione della batteria (BMS) è un componente obbligatorio con lo scopo di monitorare la salute e la sicurezza della batteria. Il ruolo del BMS diventa più significativo nelle applicazioni che hanno un gran numero di celle di batteria come veicoli elettrici e centrali elettriche di accumulo di batterie. Il BMS è un dispositivo di monitoraggio della batteria, che combina hardware e software elettronici, monitora e intraprende azioni per proteggere la batteria da determinati usi o altre condizioni che potrebbero danneggiare o ridurre la vita delle celle. Originariamente, le funzioni principali del BMS erano la sicurezza e la protezione della batteria impostando soglie di tensione, corrente e temperatura operative, nonché il bilanciamento delle celle. Nel corso degli anni, sono state condotte ricerche sui BMS, con conseguenti ulteriori sviluppi nel miglioramento delle prestazioni, delle funzionalità e delle capacità. Attualmente i BMS aiutano la batteria a funzionare meglio e in modo più sicuro tramite il monitoraggio delle celle, la sicurezza e la protezione della batteria, la stima dello stato di carica (SOC), la stima dello stato di salute (SOH), il bilanciamento delle celle, la gestione termica e il controllo della carica. Poiché sussistono preoccupazioni per quanto riguarda la sicurezza, l'affidabilità e le prestazioni delle LIB, specialmente nei sistemi autonomi, un sistema di gestione della batteria ben progettato è fondamentale. Inoltre, i modelli di batterie sono utilizzati nel BMS per prevedere la tensione di lavoro, la potenza e la capacità energetica, stimare SOC e SOH, rilevare guasti e controllare il funzionamento della batteria. I BMS hanno un ruolo cruciale nel garantire prestazioni e sicurezza affidabili migliorando l'efficienza di utilizzo delle batterie ed evitando malfunzionamenti e guasti catastrofici. È importante disporre di modelli di batterie precisi e affidabili nel BMS. Nonostante i recenti lavori nello sviluppo di BMS migliori, ci sono ancora molti aspetti negativi del design attuale. Infatti l'attuale infrastruttura del BMS consente solo di utilizzare semplici modelli di batterie come i modelli di circuiti equivalenti o modelli elettrochimici semplificati, poiché ha funzionalità limitate nella capacità di calcolo e nella memorizzazione dei dati che impediscono l'adozione di algoritmi più complessi. Un altro svantaggio dell'attuale design BMS è la mancanza di flessibilità nel suo utilizzo, poiché i suoi algoritmi sono spesso programmati nel BMS e la modifica del firmware richiede un grande sforzo. Questi aspetti negativi del BMS sono principalmente causati dalla località delle applicazioni e possono essere eliminati da un'infrastruttura software e hardware del BMS che consente l'uso dell'Internet of Things e delle applicazioni cloud per sfruttare i vantaggi del cloud computing e del

cloud storage. E' quindi interessante andare a conoscere le differenze tra attuali BMS e quelli basati sul cloud che si stanno velocemente sviluppando.

6.2 BMS attuali

Di solito, il BMS per i sistemi di batterie agli ioni di litio ha diversi strati per celle, moduli e pacchetti. C'è l'unità "master" che è il principale dispositivo di controllo e calcolo, e ci sono molte unità "slave" che vengono utilizzate per monitorare la tensione, la corrente e la temperatura delle singole celle. Le unità "master" e "slave" sono costantemente in comunicazione tra loro. Le unità "slave" monitorano le celle e i moduli e raccolgono i dati attraverso sensori, quindi inviano i dati all'unità "master" per l'elaborazione e ulteriori azioni di controllo. Il BMS comunica anche con i contattori al fine di garantire la sicurezza e anche di regolare la corrente di ingresso e uscita per il sistema di batterie. L'attuale design BMS è offline e locale, il che significa che funziona senza connessione a Internet o ad alcun server. I dati vengono solitamente memorizzati localmente nell'unità "master", o talvolta in un'unità separata se i dati vengono scambiati in modo asincrono e i vecchi dati vengono eliminati regolarmente per fare spazio ai dati più recenti. Il firmware è anche programmato nel BMS, il che significa che il BMS deve essere specifico per l'applicazione e non può essere una soluzione valida per tutti i contesti applicativi. Le funzioni principali originali del BMS erano il monitoraggio e la protezione della batteria. La funzione di monitoraggio si riferisce alla misurazione di corrente, tensione e temperatura della batteria, mentre la funzione di protezione ha la responsabilità responsabile di portare il sistema a uno stato di sicurezza quando i valori misurati scendono al di sopra o al di sotto dei loro intervalli operativi di sicurezza. Il BMS attuale è relativamente più complesso e include funzioni come il monitoraggio delle celle, il bilanciamento delle celle, la sicurezza e la protezione della batteria, la stima dello stato e la gestione termica :

-Monitoraggio delle celle: questa funzione comporta l'acquisizione dei valori della corrente, della tensione e della temperatura di ciascuna cella del sistema. Le misurazioni devono raggiungere un certo grado di precisione, poiché i dati vengono spesso utilizzati per eseguire altre funzioni nel BMS, il quale è anche responsabile della comunicazione.

-Bilanciamento delle celle: lo squilibrio della capacità della cella può essere causato da incoerenza nella capacità e nella qualità del materiale durante il funzionamento. Il bilanciamento cellulare consiste nel ridistribuire l'energia e mantenere tutte le cellule a un livello SOC simile. Esistono due approcci di bilanciamento cellulare: bilanciamento attivo e passivo. Il bilanciamento attivo consiste nel trasferire l'energia in eccesso nelle celle SOC più alte in celle SOC inferiori fino a raggiungere lo stesso livello. Il bilanciamento passivo dissipa l'energia in eccesso delle celle SOC più elevate direttamente nel calore da rimuovere. La funzione di bilanciamento delle celle è necessaria perché la capacità e la durata della batteria saranno ridotte senza di essa.

-Sicurezza e protezione della batteria: una funzione principale del BMS è quella di garantire la sicurezza della batteria e proteggerla dal funzionamento in condizioni dannose e pericolose (causate spesso dalle caratteristiche chimiche) sia per la batteria che per gli utenti. La diagnosi dei guasti è una funzione significativa del BMS per garantire un funzionamento sicuro sfruttando appositi algoritmi di rilevamento del guasto, identificandone la posizione e il tipo al fine di eseguire le azioni necessarie per ridurre l'effetto del guasto. Il BMS imposta anche limiti di sicurezza per proteggere la batteria dal funzionamento oltre l'intervallo operativo sicuro di corrente, tensione e temperatura.

-Stima dello stato: si riferisce principalmente alla stima di SOC e SOH. Una stima accurata del SOC della batteria è necessaria perché consente una lunga durata della batteria, la prevenzione di guasti della batteria, un funzionamento efficiente e calcoli accurati del SOH e del bilanciamento delle celle. La stima della SOH è fondamentale nelle strategie di gestione dell'energia selezionate per prolungare la durata della batteria e organizzare in modo appropriato la sostituzione della batteria. Nonostante l'importanza della stima dello stato, i valori SOC e SOH non possono essere misurati direttamente dalla batteria ma tramite algoritmi che si basano sui dati misurabili

-Gestione termica: le prestazioni della batteria sono solitamente influenzate dalla sua temperatura a causa del suo effetto sul degrado e sulla resistenza interna. Il sistema di gestione termica della batteria (BTMS) può aiutare a ridurre la temperatura massima della batteria e le differenze di temperatura all'interno del pacco. Esistono tre classi di BTMS, tra cui attivo, passivo e ibrido. I BTMS attivi sono basati su aria, liquidi e termoelettrici, mentre i BTMS passivi sono basati su materiali a cambiamento di fase (PCM) e tubi di calore. I BTMS ibridi utilizzano combinazioni di

approcci attivi e passivi come PCM con circolazione d'aria, PCM con circolazione di liquidi e PCM con tubo di calore.

La figura sottostante illustra il progetto BMS corrente e alcune delle sue limitazioni.

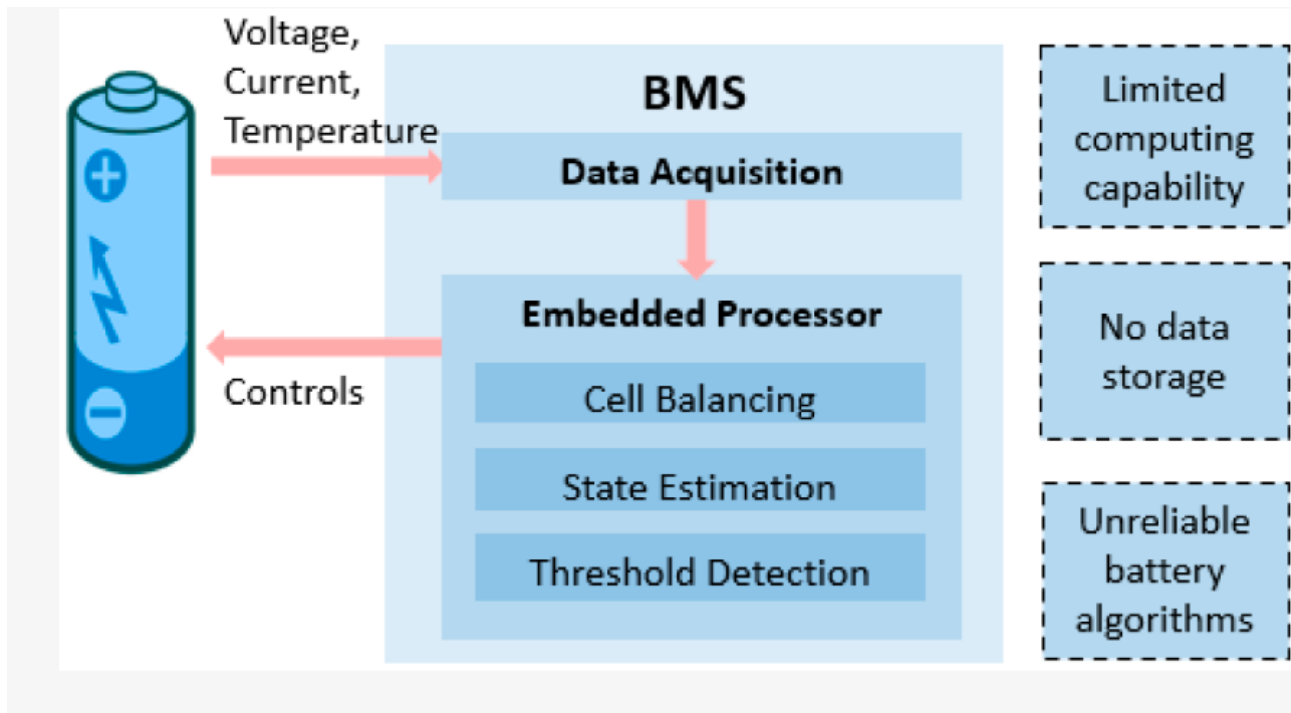


Figura 28 : Attuale design e funzioni del BMS [39]

Un grosso problema con il BMS offline è l'inaffidabilità degli algoritmi di stima dello stato. Sono stati proposti vari algoritmi di monitoraggio delle condizioni basati su modelli per stimare gli stati della batteria. Alcuni modelli di batterie comunemente usati nel BMS sono modelli elettrici, modelli elettrochimici, modelli termici e modelli accoppiati. Tuttavia, a causa dei complessi principi interni e delle condizioni di lavoro incerte, è difficile stabilire un modello di batteria in grado di rappresentare accuratamente le caratteristiche dinamiche della batteria. Pertanto, sono necessari algoritmi basati sui dati più complessi per stimare accuratamente SOC e SOH durante l'intera durata della batteria che tuttavia, nonostante il loro sviluppo sia realizzabile, non possono essere integrati nel BMS a causa della sua limitata capacità di calcolo e memorizzazione dei dati. Un altro problema importante nell'attuale BMS è la mancanza di algoritmi affidabili di diagnosi dei guasti in tempo reale. Alcuni guasti della batteria includono guasti del sensore, difetti di connessione delle celle, cortocircuiti, surriscaldamento e fuga termica. Questi guasti possono causare gravi problemi di prestazioni e sicurezza, e quindi devono essere identificati il prima

possibile. A causa delle limitazioni di elaborazione e archiviazione dei dati del BMS, non è possibile implementare algoritmi di diagnosi dei guasti in tempo reale più affidabili e accurati.[39]

6.3 BMS intelligente basato sul cloud

Gli algoritmi delle batterie sono stati studiati e sviluppati considerando i vincoli descritti nella sezione precedente e, quindi, nonostante i significativi sforzi di ricerca per migliorarli, non sono in grado di essere implementati praticamente per migliorare le prestazioni reali del BMS a causa della loro maggiore complessità. Questo aspetto ha ostacolato l'integrazione di sistemi LIB su larga scala e rallentato quindi anche l'adozione di energia rinnovabile. Il compromesso tra accuratezza e complessità è sempre stato un problema per gli sviluppatori BMS. Tuttavia, con la progettazione recentemente proposta di un BMS intelligente basato su cloud che utilizza i vantaggi del cloud computing e del cloud storage, questo problema può potenzialmente essere risolto. L'approccio proposto più recentemente ha utilizzato una piattaforma costituita da componenti IoT e infrastruttura cloud, per misurare e trasmettere senza problemi tutti i dati rilevanti della batteria al cloud.

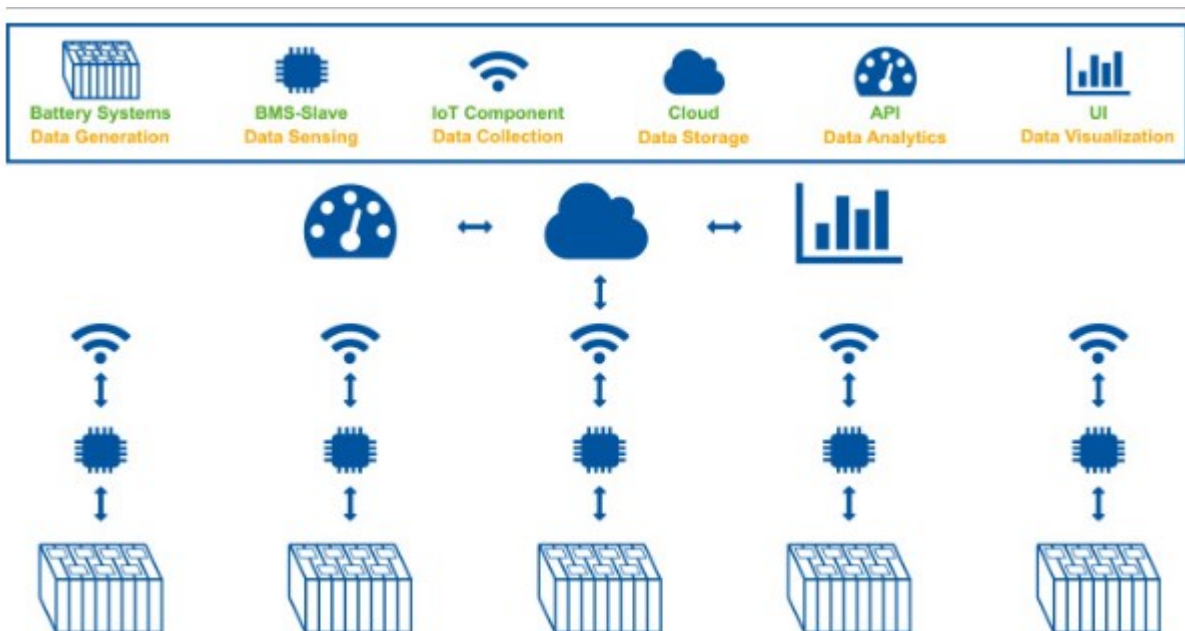


Figura 29 :Schema del BMS cloud e relativi sottosistemi associati [40]

I dati sono stati ulteriormente utilizzati per costruire un gemello digitale per il sistema di batterie, in cui gli algoritmi di diagnostica della batteria valutano i dati per fornire una migliore comprensione della carica e del livello di invecchiamento. Il BMS basato su cloud proposto ha consentito di raggiungere prestazioni della batteria visivamente migliori, ha memorizzato più dati della batteria per aiutare a sviluppare algoritmi più accurati e ha fornito supporto per lo sviluppo di strategie ottimali di controllo del sistema di batterie. Il BMS basato su cloud avrà un'elevata potenza di calcolo, capacità di archiviazione dei dati illimitate e grande affidabilità del sistema. Può eseguire funzioni di gestione della batteria, come monitoraggio, diagnostica, prognostica e ottimizzazione, in modo più accurato e affidabile.

In termini di hardware, la soluzione cloud ridurrà alcuni componenti necessari per l'elaborazione locale, con conseguente dispositivi più piccoli ma con una potenza di calcolo molto maggiore e archiviazione illimitata dei dati. In termini di software, il BMS basato su cloud sarà più efficiente per il controllo e l'ottimizzazione del sistema, avrà un migliore monitoraggio e visualizzazione dei dati ed eseguirà previsioni e diagnostica della batteria più accurate e affidabili.

Lo sviluppo del BMS intelligente basato su cloud consentirà potenzialmente un nuovo livello di controlli intelligenti verso la prossima generazione di tecnologie di stoccaggio dell'energia, aprendo la strada all'adozione di massa delle energie rinnovabili a livello globale. La progettazione BMS intelligente basata su cloud è descritta nella figura sottostante.

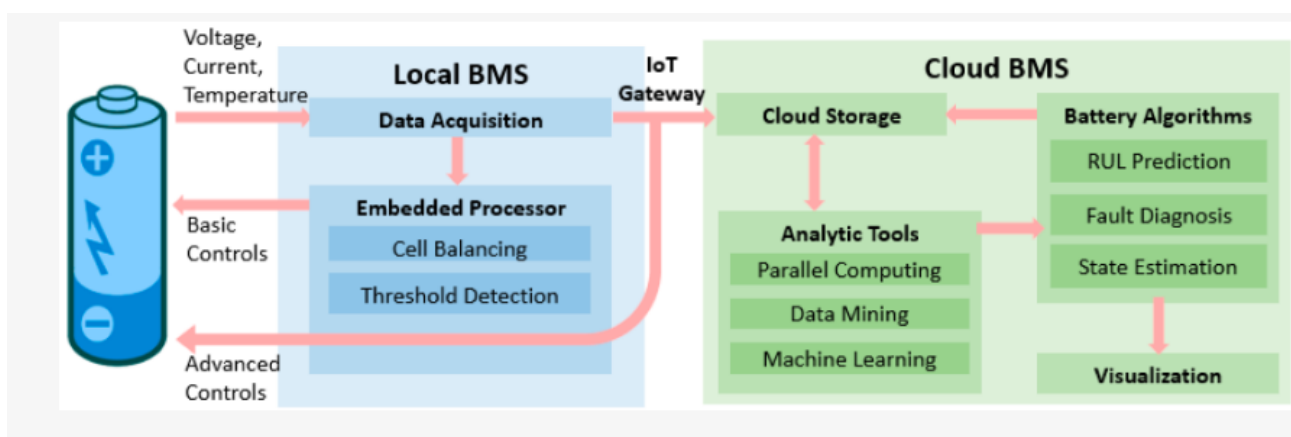


Figura 30: Design e funzioni del BMS basato sul cloud [39]

I metodi di data mining e machine learning, che richiedono grandi calcoli e memoria, sono stati implementati in vari campi, poiché i sistemi IoT e le piattaforme cloud (come quelle descritte nei capitoli precedenti) sono diventati progressivamente più avanzati e disponibili. I requisiti hardware e software per questo tipo di BMS sono più esigenti rispetto all'attuale design BMS poiché sono necessari alcuni componenti aggiuntivi per la connessione cloud. Il nuovo design ha ancora un BMS locale costituito dalle unità "slave". Queste unità eseguono l'acquisizione dei dati misurando la tensione, la corrente e la temperatura delle celle della batteria utilizzando sensori a diverse frequenze di campionamento. C'è anche un'unità "master" nel BMS locale, ma esegue solo funzioni di sicurezza di base come il bilanciamento delle celle e il rilevamento della soglia di guasto. Queste funzioni sono migliori localmente poiché devono essere eseguite rapidamente e in tempo reale. I componenti aggiuntivi per la connessione cloud includono un componente IoT, un'infrastruttura cloud, un'interfaccia di programmazione delle applicazioni (API) e un'interfaccia utente (UI). Per la componente IoT, una connessione Internet veloce e stabile è vitale per il trasferimento di dati in tempo reale tra il sistema di batterie e il cloud. I dati misurati della batteria possono essere inviati al componente IoT dalle unità "slave" utilizzando il protocollo CAN (Controller Area Network) e quindi inoltrati al cloud dal componente IoT con il protocollo MQTT (o altri già discussi nel capitolo 5) per garantire sicurezza e privacy. L'infrastruttura cloud di solito è costituita da un data logger e un database. Il data logger acquisisce una grande quantità di dati non strutturati o semi-strutturati dal sistema di batterie e consente un gateway sicuro e il trasferimento dei dati nel database cloud. Il database serve a memorizzare tutti i dati da utilizzare negli algoritmi avanzati di analisi dei dati della batteria. L'API, un altro componente del BMS basato su cloud, funge da ponte tra il database cloud e gli strumenti e gli algoritmi di analisi dei dati della batteria, utilizzando applicazioni di programmazione popolari come Matlab e Python. Il componente UI del BMS è necessario per consentire agli utenti di ispezionare e visualizzare il funzionamento e lo stato del pacco batteria in tempo reale, nonché di indagare sui dati operativi storici che possono aiutare gli utenti a pianificare la manutenzione e la riparazione. Questa è un'aggiunta significativa rispetto all'attuale design BMS che offre poca visualizzazione dei dati. Gli utenti possono essere informati dall'interfaccia utente non appena viene identificato un guasto del sistema di batterie tramite allarmi, che aiutano a ridurre gli effetti dannosi per le batterie, migliorando in definitiva la sicurezza e l'affidabilità del sistema di batterie e riducendo i costi di manutenzione. [39], [40]

6.4 Considerazioni finali sui BMS

Lo sviluppo di un BMS efficiente, affidabile e preciso svolge un ruolo molto significativo nella gestione e nel controllo sicuri ed efficaci della batteria. Il BMS intelligente basato su cloud è un passo nella giusta direzione per le future applicazioni delle batterie. Tuttavia, ci sono ancora alcuni aspetti da affrontare prima che possa diventare pratico, come una suddivisione logica delle caratteristiche e funzionalità BMS per utilizzare sia l'unità di elaborazione locale sia la piattaforma cloud. A livello locale ,tramite sensori collegati al sistema di batterie, si può agire su:

-trasmissione dei dati nel cloud dove vengono memorizzati per essere utilizzati per eseguire altre funzioni BMS

- bilanciamento delle celle che è necessario poiché l'incoerenza e la discrepanza tra le celle possono influire negativamente sulla capacità, sulla durata della confezione e su altre importanti prestazioni

-controllo della carica che impedisce al sistema di batterie di sovraccaricare e scaricare eccessivamente ed è un'altra funzione che può essere eseguita efficacemente con l'unità BMS locale

- gestione termica che spesso coinvolge fluidi di raffreddamento o materiali il cui flusso è controllato e regolato in base alla temperatura misurata, utilizzando algoritmi di controllo

- rilevamento dei guasti deve essere affidabile e fatto in tempo reale; pertanto, deve essere eseguita a livello locale perché la piattaforma cloud dipende in gran parte dalla connessione Internet che potrebbe non essere completamente affidabile in ogni momento e, quindi, è più sicuro avere la funzione di rilevamento dei guasti offline.

Alcune di queste funzioni, tuttavia, possono essere integrate con l'uso della piattaforma cloud con vantaggi di archiviazione dei dati e di maggiore capacità computazionale per gli algoritmi, ottenendo dei miglioramenti non indifferenti come per la previsione dei guasti: invece di rilevare i guasti nel momento in cui si verificano, tramite lo storico dei dati e l'apprendimento automatico la piattaforma può prevedere dove e quando potrebbe verificarsi un guasto nel sistema di batterie e notificare agli utenti azioni o comportamenti per evitare il guasto. Poiché si tratta di un concetto relativamente nuovo, ci dovrebbero essere più discussioni e sforzi di ricerca verso la suddivisione efficiente ed efficace delle funzioni locali e cloud per il BMS intelligente. Esistono alcuni inconvenienti associati alla soluzione basata su cloud per il BMS intelligente. In primo luogo, una connessione Internet veloce e affidabile è un componente obbligatorio del sistema. Tuttavia, ciò

potrebbe non essere fattibile in determinate località o per determinate applicazioni. Inoltre, anche per le comuni applicazioni della batteria, le interruzioni di Internet potrebbero influire sulle prestazioni del BMS intelligente. L'alimentazione di backup e la connessione Internet sono quindi necessarie per garantire il buon funzionamento dei sistemi di batterie. In secondo luogo, i costi sono anche una componente importante del BMS basato su cloud che deve essere studiato in modo più approfondito. I costi di gestione e manutenzione della piattaforma cloud dipendono in gran parte dalla quantità di dati e algoritmi utilizzati sul cloud. Pertanto, i ricercatori devono identificare i dati utili della batteria e la frequenza ottimale di raccolta dei dati al fine di ridurre al minimo i costi del funzionamento del cloud senza sacrificare l'accuratezza e l'affidabilità delle funzioni BMS.[39]

6.5 Ambiti applicativi principali

I maggiori contesti ed ambiti applicativi che attualmente sfruttano il cloud BMS riguardano principalmente l'automotive elettrico per il monitoraggio dello stato della batteria e per analisi sullo stile di guida ,applicazioni per monitorare il ciclo di vita delle batterie, applicazioni per l'energy storage domestico e applicazioni per le smart city (cioè una città intelligente che integra tecnologie digitali nelle proprie reti, servizi e infrastrutture per aumentare l'efficienza della loro amministrazione) che sfruttano le recenti smart grid, ovvero una rete intelligente che ottimizza la distribuzione dell'energia , decentralizza le centrali di produzione e minimizza sovraccarichi e variazioni della tensione elettrica.

Per creare un sistema che sia completamente in grado di operare efficientemente in questi ambiti si devono individuare i livelli su cui l' IoT va implementato e capire come questi possono interagire tra di loro. Quelli fondamentali riguardano:

-Device layer:

Livello costituito da sensori, attuatori, prese intelligenti, contatori intelligenti, ecc. capaci di rilevare e raccogliere tutti quei dati che poi sono utilizzati e gestiti dal microcontrollore del dispositivo stesso. Si occupa quindi di selezionare e raccogliere i dati fondamentali per il monitoraggio del dispositivo (come temperatura, corrente e tensione per le batterie) e della loro frequenza di campionamento in base al tipo di studio e applicazione che si intende realizzare.

Uno dei sensori più utilizzati nel mondo dell' IoT è l'ESP 8266 , ovvero un System on a Chip (SOC),cioè un chip con Wi-Fi integrato, sfruttato per il suo basso costo, i bassi consumi e un ampio range di temperatura nel quel può operare senza problemi. Il modulo Wi-Fi integrato può essere utilizzato, proprio per la sua natura, come punto di accesso e/o stazione, per ospitare un web server o per connettersi a Internet per recuperare o caricare dati sfruttando protocolli IoT come quelli descritti nel capitolo 5 .[41]

-Network Layer

Quando parliamo di IoT facciamo riferimento a un insieme di dispositivi interconnessi e ciascuno con una propria identità digitale. Questi, in generale, hanno bisogno di una rete per comunicare, trasmettere e scambiare informazioni. Questo livello si occupa proprio di connettere il dispositivo all'applicazione sfruttando tecnologie come Wi-Fi, reti cellulare, Rfid, power line communication (PLC), ecc.

Questo livello deve anche garantire riservatezza, sicurezza e protezione della rete e di conseguenza anche dei dati trasmessi tramite essa , l'interoperabilità dei dispositivi permettendo loro di connettersi attraverso una piattaforma comune, una latenza minima per garantire una comunicazione in tempo reale o quasi e scalabilità, cioè mantenere la rete efficiente nonostante un ampio numero di dispositivi e utenti contemporaneamente connessi.[17], [34]

-Cloud Management Layer

È il livello sui cui lavora il cloud per memorizzare e analizzare i dati raccolti per estrapolare informazioni, sfruttando tutti gli enormi vantaggi delle piattaforme cloud già discussi nel capitolo 4. Nel caso delle batterie questo livello, a partire da semplici dati come temperatura, tensione e corrente (che derivano dal device layer), è in grado, tramite appositi algoritmi e calcoli molto complessi, di ricavare la stima dello stato di carica e dello stato di salute, ovvero parametri fondamentali ma che non si possono misurare direttamente tramite i sensori. Questo livello offre anche servizi di management per gestire l'autenticazione e l'autorizzazione degli utenti: spesso nelle applicazioni reali utenti con ruoli differenti hanno accesso a dati diversi e possono svolgere azioni diverse su di essi . [34], [42]

-Application Layer

Questo livello si occupa di fornire vari servizi agli utenti finali dell'applicazione. Ovviamente questi servizi variano in funzione di ogni applicazione e del suo contesto di utilizzo. Queste applicazioni offrono all'utente la possibilità di visualizzare i vari dati su diversi grafici e/o tabelle a seconda della quantità e tipologia stessa dei dati, di visualizzare e/o scaricare lo storico dei dati relativo a un certo periodo, visualizzare dati in tempo reale, generare notifiche quando si verificano situazioni particolari e/o critiche che vengono rilevate dai sensori, configurare o modificare alcune caratteristiche del sensore o del dispositivo e gestire gli utenti che possono accedere a determinati dati. Come già descritto in precedenza, la maggior parte delle applicazioni di questo tipo oggi riguarda principalmente l'automotive elettrico, offrendo la possibilità di avere in tempo reale la posizione e velocità attuale, la velocità media, i km percorsi, il tragitto effettuato, il consumo medio, il tempo di ricarica, ecc. e lo storage di energia, quindi la possibilità di visualizzare tensione, corrente e potenza erogata, autonomia energetica attuale e residua, energia prodotta da fonti rinnovabili, energia totale utilizzata, media dell'energia prodotta e/o utilizzata in un certo periodo, ecc. [34], [43], [44]

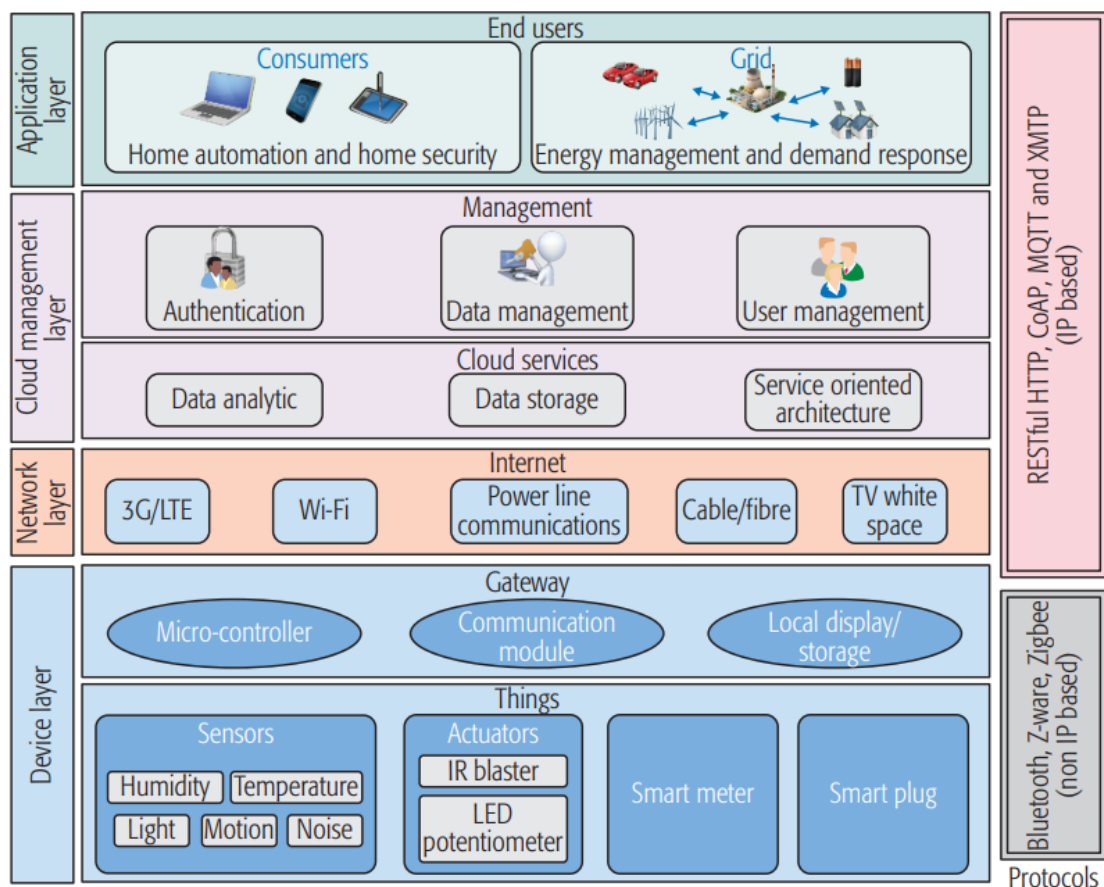


Figura 31 : principali layer IoT [34]

6.6 Normativa

La nascita e la diffusione delle batterie al litio risale ai primi anni 90'. Oggi questa tecnologia ha raggiunto una diffusione molto ampia ed è utilizzata sempre più frequentemente in molte applicazioni proprio perché in grado di racchiudere un'alta capacità energetica in un design compatto. L'utilizzo delle batterie agli ioni di litio oltre a presentare numerosi benefici comporta, però, anche dei rischi concreti legati alla sicurezza di cui bisogna tener conto. Con il crescente uso di batterie agli ioni di litio di media-alta potenza i pericoli legati allo stoccaggio e in particolare alla ricarica di queste batterie aumentano sia negli ambienti commerciali che in quelli privati. Ricaricare e stoccare questi sistemi senza metodi, regole e i dispositivi adeguati è fortemente sconsigliato. Le batterie al litio, come ogni prodotto, vengono fabbricate, messe in vendita e utilizzate per poi ritornare in mano al produttore[45], a un rigeneratore o a un'impresa che le ricicla e devono rispondere a molte normative per questioni di sicurezza: dal 2009 infatti sono considerate ufficialmente come merce pericolosa. Anche se la loro chimica è considerata una delle più sicure, le batterie agli ioni di litio comportano ancora rischi significativi se non maneggiate con cura. L'alta tensione di una batteria agli ioni di litio comporta rischi elettrici, come cortocircuito, folgorazione, scosse elettriche o ustioni, mentre il componente chimico all'interno della batteria (l'elettrolita vero e proprio) potrebbe fuoriuscire e causare intossicazione o corrosione. Le batterie agli ioni di litio sono anche soggette a fughe termiche. Infatti, se la temperatura supera una certa soglia, le celle iniziano a emettere gas caldi, il che aumenta ulteriormente la temperatura e alla fine può portare a combustione e incendio. Le principali normative e linee guida per le batterie agli ioni di litio sono contenute in tre documenti:

- Direttiva europea sulle batterie 2006/66/CE: si tratta di una direttiva UE che fornisce linee guida agli Stati membri in merito alla produzione e allo smaltimento delle batterie nell'UE. Il suo scopo è migliorare le prestazioni ambientali di batterie e accumulatori.

- Direttiva sulla sicurezza generale dei prodotti: La DSGP fornisce standard per la sicurezza dei prodotti per proteggere i consumatori da potenziali pericoli, tramite gli standard EN. La norma EN che si riferisce alle batterie agli ioni di litio è la EN 60086-4 ed è il punto di riferimento per specifiche e soluzioni tecniche in fase di progettazione del prodotto. Seguire gli standard EN non è obbligatorio, ma fortemente raccomandato.

- ADR: è un documento dell'ONU, adottato dall'Unione Europea, che regola il trasporto terrestre di merci pericolose. Seguire le regole ADR è obbligatorio per il trasporto di batterie agli ioni di litio. Tutte le batterie agli ioni di litio sono di Classe 9 e ottengono il numero ONU 3480.

I rischi e le regole variano ovviamente in base alle condizioni di una batteria e alla fase del suo ciclo di vita. Le batterie nuove sono le più stabili a meno che non ci siano difetti di fabbricazione. In questo caso i rischi sono relativamente bassi, ma è comunque necessaria cautela durante il trasporto e la movimentazione per evitare l'incorrere di eventuali rischi termici e meccanici minori, motivo per cui tutti i requisiti ADR, compresa l'etichettatura e l'imballaggio, devono essere sempre presi in considerazione. La Direttiva sulle batterie 2006/66/CE stabilisce che ogni produttore di batterie risponde al principio della responsabilità estesa e deve pensare al fine vita del proprio prodotto. Può adempiere a quest'obbligo da solo o lavorare con un'organizzazione che in grado di assicurare la compliance e la gestione di questa tipologia di rifiuto, come ad esempio, in Italia, Cobat. In base alla gerarchia dei rifiuti, prima del riciclo c'è il riutilizzo o la rigenerazione, il che significa che la batteria mantiene lo status di "prodotto", e non di "rifiuto", anche se il riciclo è ancora l'opzione più comune. [46] In caso di riutilizzo o rigenerazione, le batterie agli ioni di litio sono etichettate e imballate allo stesso modo delle nuove batterie agli ioni di litio. Quando una batteria giunge a fine vita e non può essere rigenerata o riutilizzata per uno scopo diverso, diventa ufficialmente un "rifiuto" e le sue specifiche ADR cambiano. Bisogna però distinguere il caso in cui essa sia danneggiata oppure ancora integra perché ovviamente cambiano le regole di trasporto e smaltimento e i pericoli connessi. Le batterie danneggiate infatti presentano il rischio maggiore poiché vengono trasportate in uno stato di potenziale e forte instabilità. Durante tutte le fasi di utilizzo si devono rispettare suggerimenti e regole specifiche, come, ad esempio, la conservazione in ambiente asciutto e a temperatura controllata, assicurarsi che i contatti non possano andare in cortocircuito, evitare sollecitazioni eccessive, tenerle lontane da altri materiali infiammabili o comunque pericolosi.

Anche la normativa italiana si è interessata a questo aspetto in vari ambiti, compreso ovviamente quello lavorativo. Il D.Lgs. 81/2008, che rappresenta il Testo Unico sulla Salute e Sicurezza sul Lavoro, è molto chiaro sugli obblighi previsti dalla legge. A fronte di un'accurata valutazione dei rischi specifici per il tipo di attività svolta, è dovere del datore di lavoro "l'eliminazione dei rischi e, ove ciò non sia possibile, la loro riduzione al minimo in relazione alle conoscenze acquisite in base

al progresso tecnico “. Inoltre, è obbligo del datore di lavoro “adottare le misure necessarie ai fini della prevenzione incendi e dell’evacuazione dei luoghi di lavoro (...) Tali misure devono essere adeguate alla natura dell’attività, alle dimensioni dell’azienda o dell’unità produttiva, e al numero delle persone presenti”. Nel caso delle batterie al litio, è necessario quindi, che il datore di lavoro riconosca il pericolo intrinseco derivante dalla loro gestione, e prenda di conseguenza i provvedimenti adeguati per la tutela di operatori, beni aziendali e ambienti di lavoro. Un importante orientamento sui rischi è fornito dalle istruzioni per l’uso ed eventualmente da una “scheda informativa sulla sicurezza” del produttore della batteria. Questi documenti risultano quindi essenziali per un uso corretto e sicuro delle batterie.

Visto che il mercato delle batterie sta assumendo una dimensione sempre più strategica, il Parlamento europeo è al lavoro su nuove norme per affrontare le questioni ambientali, etiche e sociali. Nel dicembre del 2022 il Parlamento ed il Consiglio Europeo hanno raggiunto un accordo provvisorio per rivedere le norme sulle batterie per tener conto degli sviluppi tecnologici e delle sfide future. Una volta approvato formalmente da entrambe le istituzioni, le nuove regole entreranno direttamente in vigore. Questo accordo dovrebbe rendere le batterie più sostenibili, performanti e durevoli e porre più attenzione sull'intero ciclo di vita del prodotto dalla fase di progettazione fino alla fase di riciclo, in linea anche con l'adozione di un'economia circolare sempre più forte da parte dell'UE. Un tema importante di questo accordo è l'introduzione di una nuova categoria di batterie, ovvero quelle utilizzate per alimentare i "mezzi di trasporto leggeri" come biciclette o scooter elettrici, a causa del loro crescente utilizzo: questa nuova categoria andrebbe ad affiancarsi alle classi di batterie portatili già esistenti, come quelle automobilistiche e quelle industriali.

BATTERIE DISPONIBILI NELL'UE

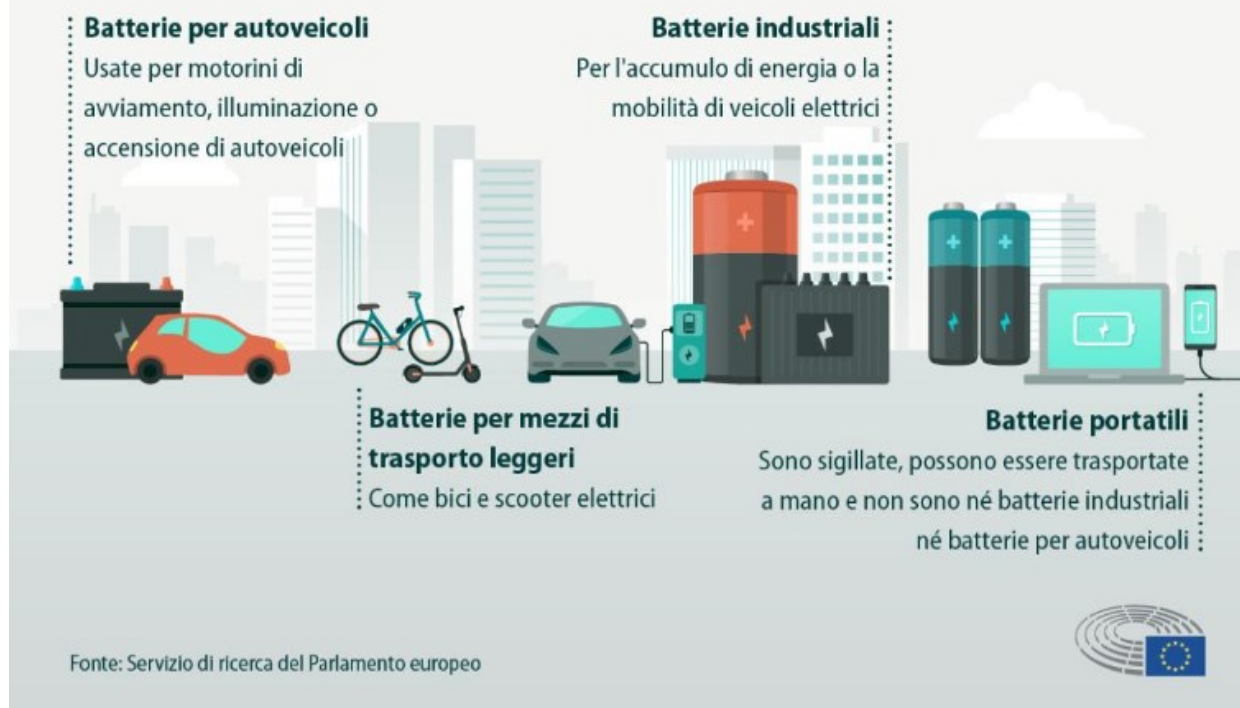


Figura 32 : batterie disponibili nell' UE [47]

Secondo le attuali previsioni entro il 2030 saranno almeno 30 milioni i veicoli elettrici a emissioni zero immessi sulle strade dell'UE. All'aumentare del numero di batterie deve corrispondere una crescita del tasso di riciclo delle stesse, la produzione di batterie non deve contribuire oltre un certo livello alle emissioni di anidride carbonica e l'estrazione delle materie prime necessarie alla loro produzione deve rispettare precisi vincoli ambientali e sociali per le aziende nell'intera filiera delle batterie. [48]La Commissione europea vuole individuare un'impronta di carbonio massima per tutte le batterie prodotte, vendute e utilizzate in Ue: in pratica le emissioni di CO2 associate alla costruzione di una batteria non deve superare un certo livello, che sarà stabilito da un successivo atto delegato. Oltre a tutelare l'ambiente è un incentivo per lo sviluppo di un mercato interno delle batterie.[49] Infatti, le batterie attuali sono prodotte principalmente nel continente asiatico con un mix energetico ad alta intensità carbonica e dopo l'entrata in vigore del nuovo Regolamento la maggior parte delle attuali batterie, in teoria, non sarà più a norma. Per avere batterie sostenibili, l'Unione europea dovrà quindi in gran parte produrle. Il problema è che per produrre batterie servono materie prime che l'Europa non ha a sufficienza o addirittura, nel caso del litio, non c'è neppure una miniera nelle nazioni dell'Ue. Questo spiega perché il riciclo in

un'ottica di economia circolare è un aspetto fondamentale da sostenere. [50] Il Regolamento lo fa introducendo target di riciclo per i materiali critici delle batterie a fine vita con lo scopo di ridurre la necessità di estrarre altri materiali vergini e fare in modo che l'Europa disponga del suo capitale di materie prime riducendo l'import da paesi esterni. Individuare delle soglie minime di materiale riciclato come fa l'Unione europea è l'unica strada percorribile, non solo per la sostenibilità ambientale, ma anche economica: la crescita a cui stiamo assistendo del prezzo delle materie prime potrebbe non solo avere effetti devastanti sull'economia europea, ma potrebbe persino decretare la morte sul nascere del nuovo mercato interno delle batterie.

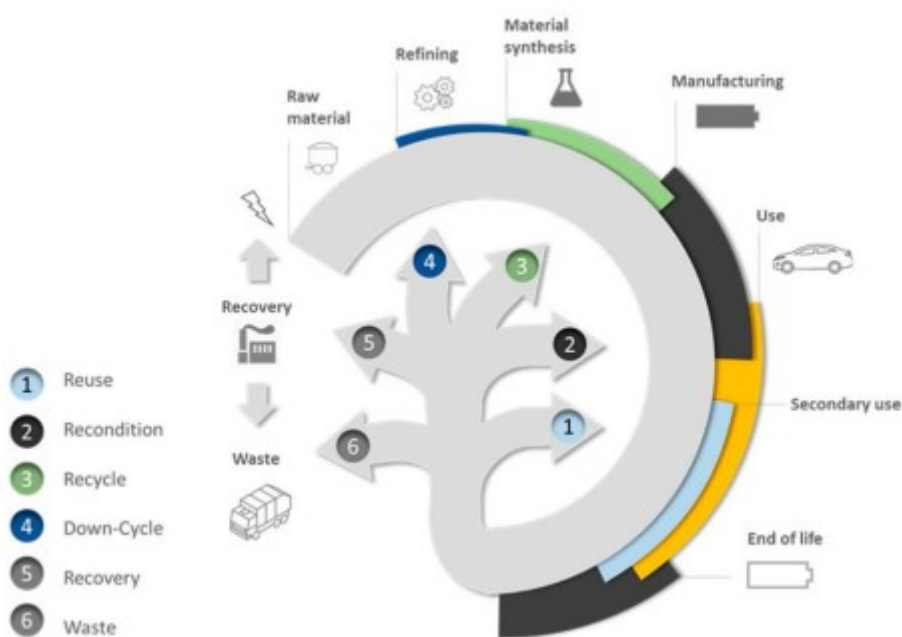


Figura 33 : flowchart del ciclo di vita delle batterie a litio proposto dalla EU[51]

C'è infatti da dire che il Regolamento batterie ha un'ottica decennale impostata sul lungo periodo, mentre la situazione attuale è piuttosto contingente: stiamo assistendo ad uno shock di mercato e uno squilibrio tra domanda e offerta, ma non si tratta di problemi strutturali destinati a durare per sempre perché a livello geologico non c'è penuria di materie prime e lo sviluppo tecnologico si sta dirigendo verso l'impiego di materiali più facili da trovare rispetto a quelli attuali. A tal proposito con le nuove regole per le batterie verranno inseriti obiettivi aggiornati e più stringenti per la loro raccolta.

7 WEB APP SVILUPPATA

La batteria, vero cuore delle auto elettriche, è l'elemento più importante, complesso e costoso di tutto il veicolo. Non rappresenta, infatti, soltanto la riserva energetica, ma un vero e proprio organo vitale con una tecnologia in costante evoluzione. La batteria è vista come una sorta di serbatoio in cui immagazzinare l'energia necessaria a far muovere l'auto, ma in realtà è molto più di questo. Anche se la spinta arriva dal motore, le prestazioni di quest'ultimo sono influenzate direttamente dalla batteria e dalla sua capacità di erogare l'energia. Quindi, si può dire che in realtà il vero "motore" dell'auto elettrica sia proprio la batteria. Per valutare le prestazioni delle batterie, e di conseguenza dell'auto, bisogna considerare la sua capacità, ovvero la quantità di energia che può immagazzinare, il modo e la rapidità con cui l'energia viene immagazzinata ed erogata al motore, tutte caratteristiche espresse da amperaggio (quantità di energia che può essere trasmessa in un secondo) e da voltaggio (corrisponde alla forza, dunque alla rapidità, con cui una quantità di energia pari a 1 Ampere viene trasmessa). Dunque, è facile comprendere come all'aumentare di voltaggio e amperaggio la batteria risulti più potente perché il flusso di corrente che entra o esce sarà maggiore per volume e intensità.

Lo scopo principale di questa web app è fornire un esempio del modello dei dati da associare al monitoraggio delle batterie di auto elettriche insieme ad un semplice meccanismo per la loro analisi, monitoraggio e visualizzazione.

Il codice di questa web app è scaricabile tramite il seguente link github:
<https://github.com/1092407/tesi>

Tra i vari file è presente anche "istruzioni.pdf" dove sono contenute istruzioni per testare la web app.

7.1 Descrizione funzionalità

Web app per una casa automobilistica di auto elettriche utile per registrare clienti, gestire il parco auto e per visualizzare dati sulle batterie montate sulle auto vendute.

Livelli di accesso al sito e funzionalità:

1) Parte pubblica

Qui ci sono informazioni generali come “chi siamo”, “dove siamo”, privacy, ecc. Chiunque può accedere alla parte pubblica, dove c’è anche la possibilità di effettuare il login e andare in una dei livelli successivi in base al proprio livello di utenza (casa automobilistica o cliente). Non c’è la possibilità di registrarsi in quanto la casa automobilistica è già pre-registrata nel database e i clienti possono fare login solo dopo che la casa automobilistica, nella sua area privata del sito, crea un profilo per ciascuno e fornisce loro le credenziali di accesso. In questo modo solo chi acquista un’auto può usare la web app per visualizzare i dati.

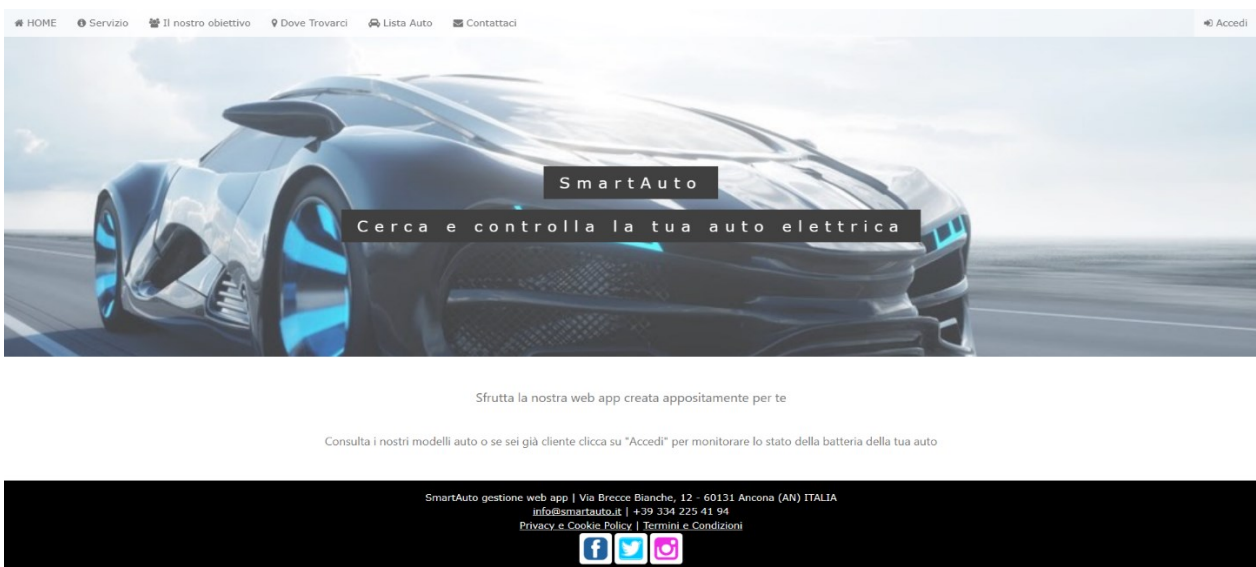


Figura 34: Homepage pubblica

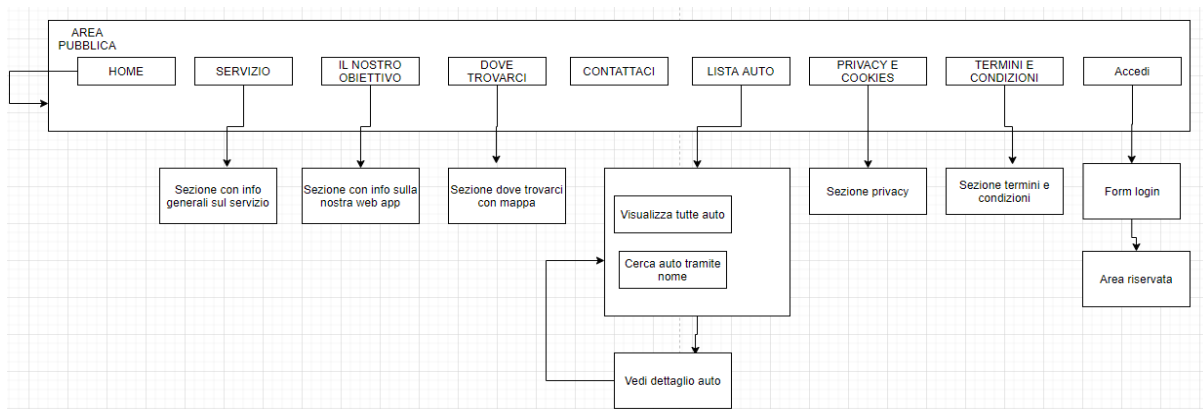


Figura 35: Schema dei link area pubblica

2) Livello casa automobilistica

Account pre-registrato nel database.

Crea il profilo ad ogni nuovo cliente che acquista un'auto, avendo anche la possibilità di modificare/eliminare dati relativi ad ogni cliente.

Gestisce una lista con tutte le auto che sono nel suo parco auto, avendo possibilità di inserire/modificare/eliminare dati relativi ad esse; queste sono anche visualizzabili nella parte pubblica per attirare nuovi potenziali clienti. Visualizza i dati sulle batterie delle auto dei clienti.



Figura 36: Homepage livello casa automobilistica

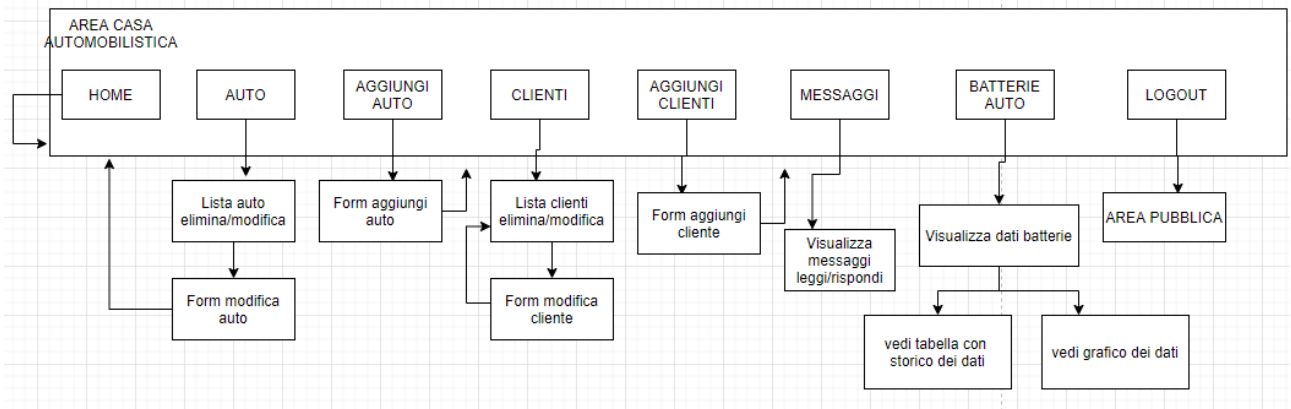


Figura 37: Schema dei link area casa automobilistica

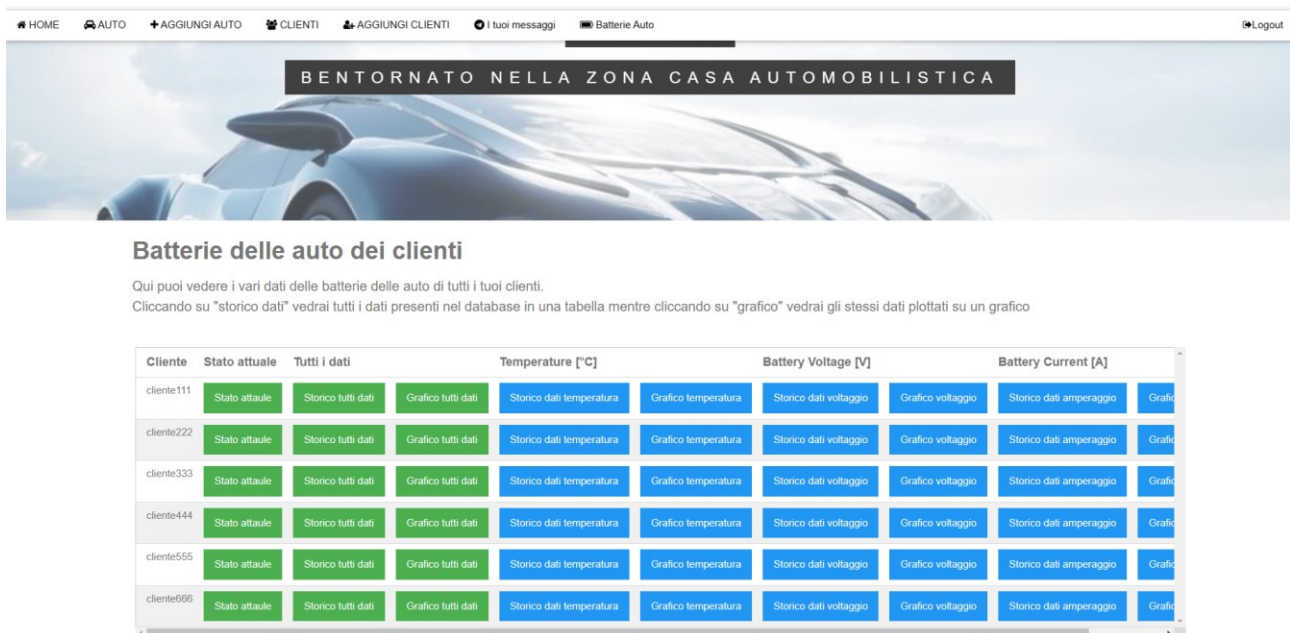


Figura 38: porzione della schermata dove la casa automobilistica può scegliere quali dati andare a visualizzare relativamente alla batteria di un certo cliente

3) Livello cliente

Ogni cliente può visualizzare dati relativi alla batteria della propria auto, ma questi sono meno dettagliati e specifici rispetto a quelli che può visualizzare il fornitore.

Ogni cliente ha una chat dedicata con la casa automobilistica attraverso la quale può richiedere assistenza o ulteriori informazioni sulla propria auto.

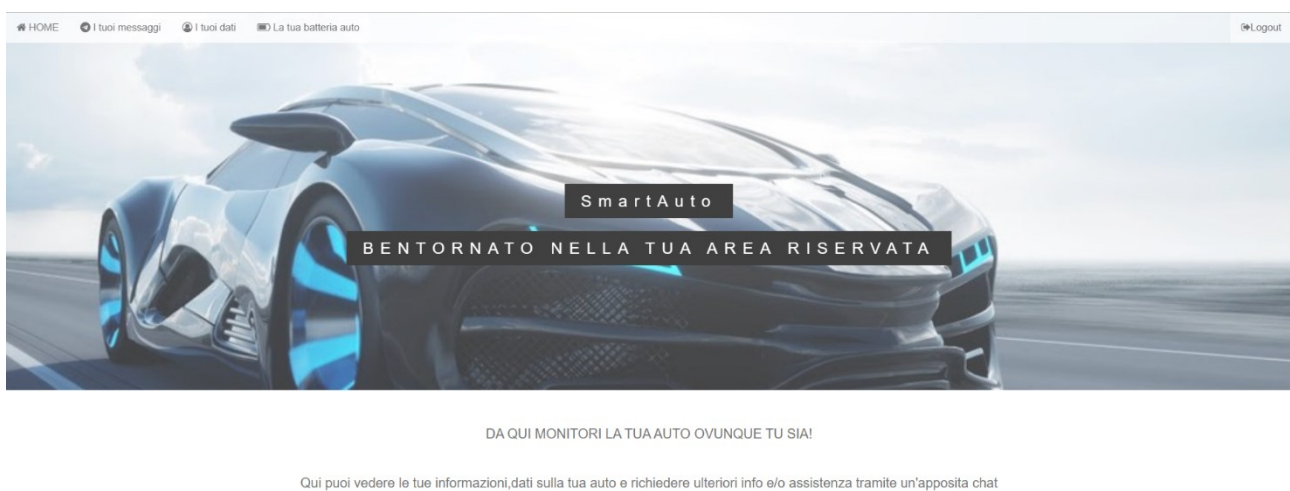


Figura 39 : Homepage livello cliente

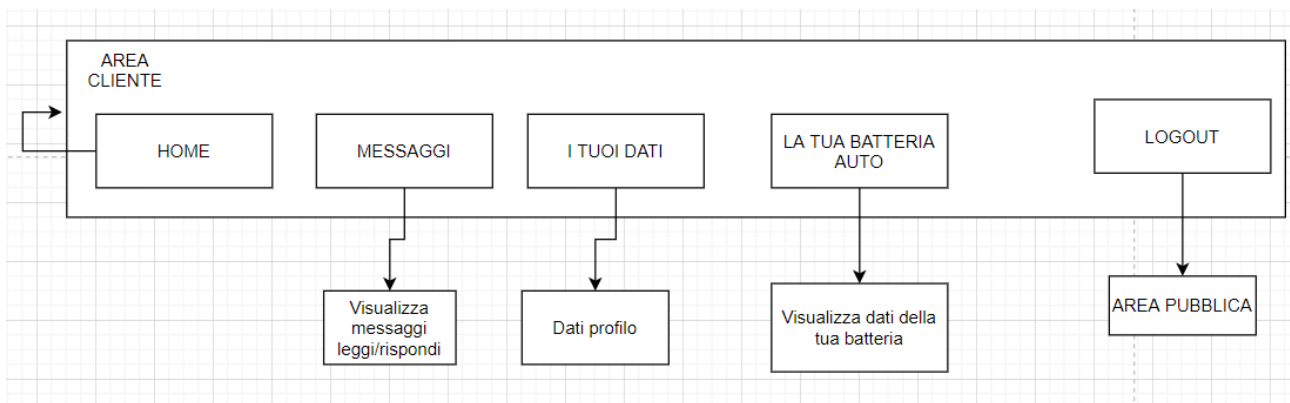


Figura 40: Schema dei link area cliente

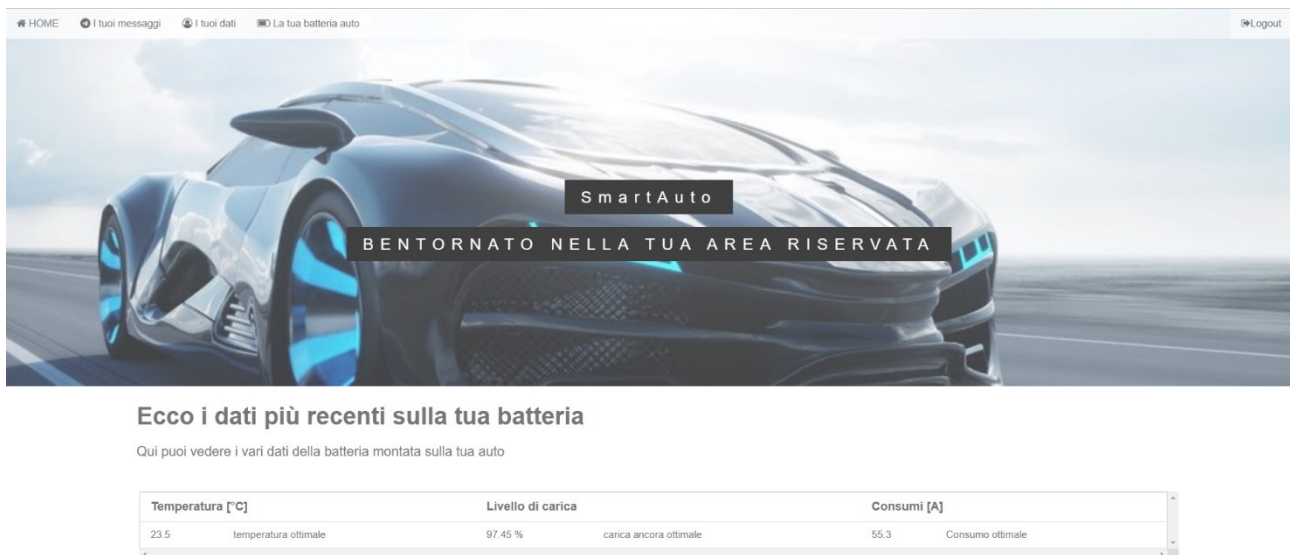


Figura 41: schermata dove un cliente può vedere lo stato attuale della batteria della sua auto

7.2 Dati utilizzati

In uno sviluppo futuro di questo progetto i dati della batteria saranno inviati in tempo reale in cloud e monitorati dall'utente. In questa tesi i dati sono presi dal dataset di dati reali disponibile online (<https://ieee-dataport.org/open-access/battery-and-heating-data-real-driving-cycles>). I dati

sono stati acquisiti nel 2020 da Matthias Steinstraeter, Johannes Buberger e Dimitar Trifonov della Technical University of Munich, Institute of Automotive Technology.

Questi dati sono stati inseriti nel database collegato alla web app tramite il meccanismo dei “seeders”, come meglio descritto nella sezione “Implementazione”.

In particolare questi dati sono stati registrati durante diversi viaggi di guida reale con un'automobile elettrica ed utilizzati durante uno studio per la convalidazione di un modello completo di veicolo costituito dal gruppo propulsore e dal circuito di riscaldamento dove, per ogni viaggio, ad intervalli regolari di 100 millisecondi, sono stati acquisiti dati ambientali (come temperatura ed altitudine), dati del veicolo (come velocità e acceleratore), dati sulla batteria (tensione, corrente, temperatura, SoC) e dati sul circuito di riscaldamento (come temperatura interna, potenza di riscaldamento, ecc.).

I dati si riferiscono a 70 percorsi di durata di circa 30 minuti per un percorso di circa 20 Km. Per ogni viaggio sono riassunte le seguenti informazioni generali:

- Trip #
- Date
- Route/Area
- Weather
- Battery Temperature (Start) [°C]
- Battery Temperature (End)
- Battery State of Charge (Start)
- Battery State of Charge (End)
- Ambient Temperature (Start) [°C]
- Target Cabin Temperature
- Distance [km] Duration [min]
- Fan

Durante ogni viaggio sono acquisiti i seguenti dati con tempo di campionamento di 100ms:

- Time [s]
- Velocity [km/h]
- Elevation [m]

- Throttle [%]
- Motor Torque [Nm]
- Longitudinal Acceleration [m/s²]
- Regenerative Braking Signal
- Battery Voltage [V]
- Battery Current [A]
- Battery Temperature [°C]
- max. Battery Temperature [°C]
- SoC [%]
- displayed SoC [%]
- min. SoC [%]
- max. SoC [%]
- Heating Power CAN [Kw]
- Heating Power LIN [W]
- Requested Heating Power [W]
- AirCon Power [Kw]
- Heater Signal Heater Voltage [V]
- Heater Current [A]
- Ambient Temperature [°C]
- Coolant Temperature Heatercore [°C]
- Requested Coolant Temperature [°C]
- Coolant Temperature Inlet [°C]
- Heat Exchanger Temperature [°C]
- Cabin Temperature Sensor [°C]

Inoltre sono acquisiti con tempo di campionamento di 100ms i seguenti dati:

- Environment
 - Slope [rad]
 - Wind Speed [m/s]
 - Ambient Temperature [C]
- Longitudinal Vehicle
 - Longitudinal Acceleration [m/s²]

- Speed [m/s]
- Position [m]
- Traction Force [N]
- Drive Power [W]
- Drive Train
 - Drive Torque [Nm]
 - Wheel Torque [Nm]
 - Drive Force Front [N]
 - Drive Force Rear [N]
 - Roll Resistance Coeff [-]
 - Mass Jred [kg]
 - Drive Current [A]
 - Engine Speed[1/min]
 - Wheel Rot Speed [rad/s]
 - max Charging Power [W]
 - EM Power [Kw]
- Battery
 - Pack Voltage [V]
 - Battery Current [A]
 - Power [Kw]
 - SOC [%]
 - Ageing SOH [-]
 - Mean Cell Temperature [C]
- Consumption
 - Power [Kw]
 - Energy [kWh]
 - [kWh/100km]
- High Voltage Heater
 - Cabin Temperature [C]
 - Heating Power [Kw]
 - Heating Power inclPeak [Kw]
 - Heatexchanger Inlet Coolant Temp [C]
 - Heatexchanger Outlet Air Temp [C]

- Auxiliaries
 - Seat Heating [W]
 - Brake Lights [W]
 - Rest Lights [W]
 - Window Lifters [W]
 - Infotainment [W]
 - Wipers [W]
 - Blower [W]
 - CU Sensors [W]
 - Auxiliaries Total [W]

7.3 Implementazione

Il framework utilizzato per lo sviluppo è stato "Laravel 6" in quanto è una versione LTS (Long Term Support) , cioè con affidabile supporto e con aggiornamenti relativi a sicurezza, manutenzione e funzionalità per un lungo periodo di tempo che la rendono una versione stabile a seguito dei molti test approfonditi che ha superato.

Altro strumento fondamentale è stato XAMPP. Si tratta di un software composta da Apache http Server, MySQL (o una sua fork chiamata MariaDB) e PHP che facilita l'installazione e la gestione degli strumenti più comuni per lo sviluppo di applicazioni web, raggruppandoli in un unico luogo, e particolarmente utile per il testing.

7.3.1 Visualizzazione dei dati

Oltre ad essere visualizzati tramite tabelle HTML, alcuni dati delle batterie sono visualizzabili tramite appositi grafici che sfruttano "Laravel Charts", un'apposita libreria di grafici creata per supportare Laravel nella visualizzazione di dati.

Time	Temperature [°C]	Battery Voltage [V]	Battery Current [A]	SOC [%]
2022-03-11 12:03:00	23.5	389.8	55.3	75
2022-03-11 12:02:00	23.6	391.3	45.8	75.3
2022-03-11 12:01:00	23.8	381.4	23.8	75.5
2022-03-11 12:00:00	22.5	387.7	27.8	75.7
2022-03-10 12:05:00	24.5	391.3	57.8	69.1
2022-03-10 12:04:00	23.5	381.4	19.8	69.2
2022-03-10 12:03:00	21.5	389.8	37.8	69.4
2022-03-10 12:02:00	22.5	376.2	17.8	69.6
2022-03-10 12:01:00	23.5	387.7	12.8	69.8
2022-03-10 12:00:00	24.5	391.3	21.8	69.9
2022-03-09 12:04:00	26.1	376.2	35.5	72.1
2022-03-09 12:03:00	26.5	382.3	37.8	72.3
2022-03-09 12:02:00	25.5	381.4	47.8	72.4
2022-03-09 12:01:00	22.7	387.7	23.8	72.6
2022-03-09 12:00:00	22.5	391.3	57.8	72.8
2022-03-08 12:03:00	22.7	376.2	37.8	83
2022-03-08 12:02:00	23.5	392.1	77.8	83.3
2022-03-08 12:01:00	23.6	389.8	57.8	83.5
2022-03-07 12:00:00	24.5	392.3	47.8	85.9

Figura 42: esempio di una tabella HTML per leggere lo storico dei dati presenti nel database

Laravel Charts è una libreria di grafici disponibile come pacchetto PHP in grado di generare un elevato numero di combinazioni di grafici. L'API di Chart è progettata per essere estensibile e personalizzabile, consentendo di utilizzare rapidamente qualsiasi opzione nella libreria JavaScript. Oggi i grafici sono per lo più realizzati in JavaScript, ma i dati che contengono molto probabilmente provengono da un database e Laravel è quello che li gestisce. Questa libreria offre la possibilità di non usare direttamente codice JavaScript perché crea una variabile PHP a cui viene associato un grafico, consentendone la personalizzazione. [52]

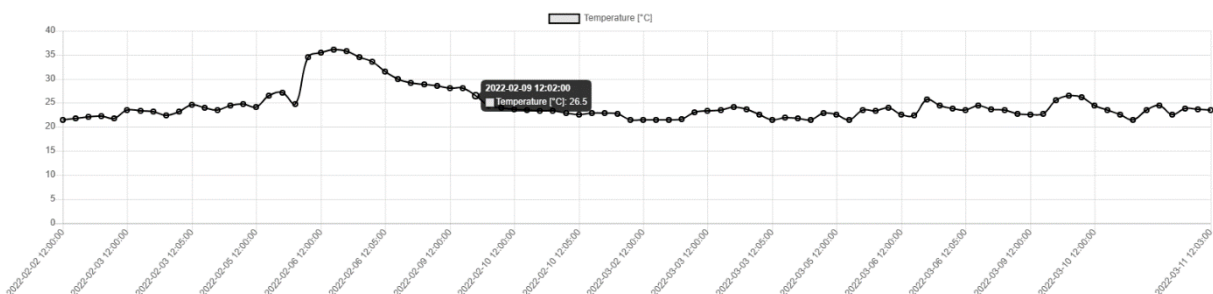


Figura 43: esempio di un grafico utilizzato nella web app e che legge dati sulla temperatura della batteria

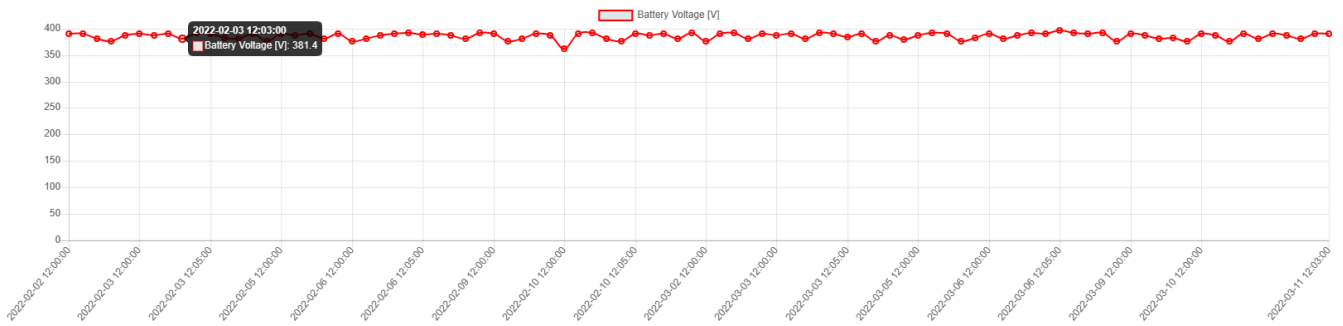


Figura 44: esempio di un grafico utilizzato nella web app e che legge dati sul voltaggio della batteria

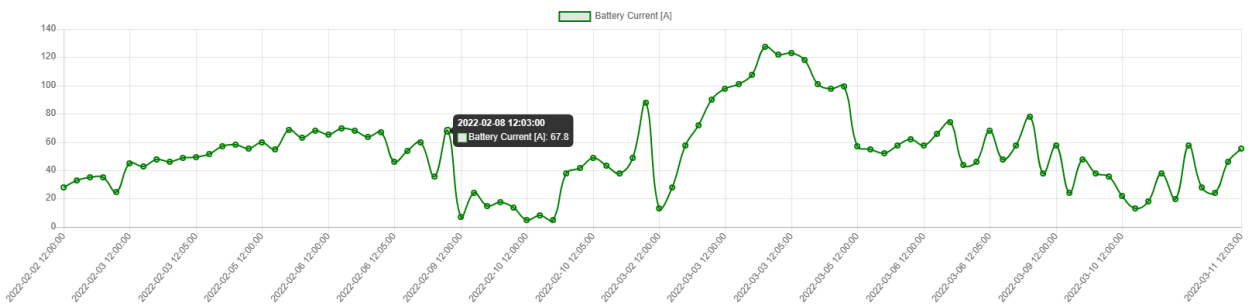


Figura 45: esempio di un grafico utilizzato nella web app e che legge dati sulla corrente della batteria

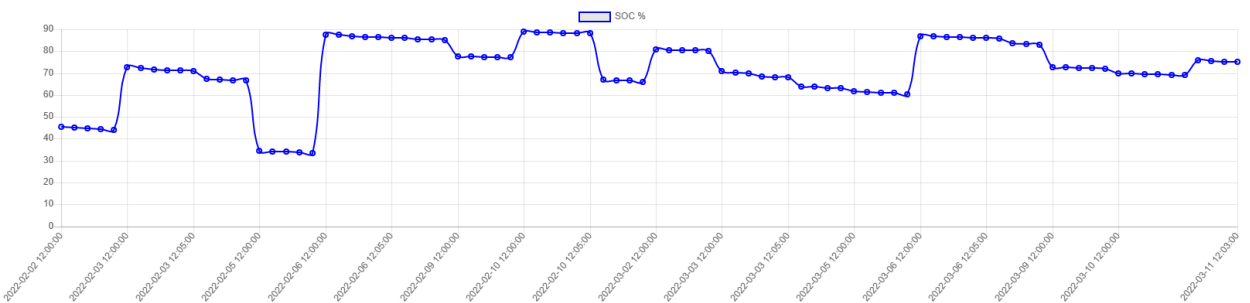


Figura 46: esempio di un grafico utilizzato nella web app e che legge dati sulla SOC della batteria

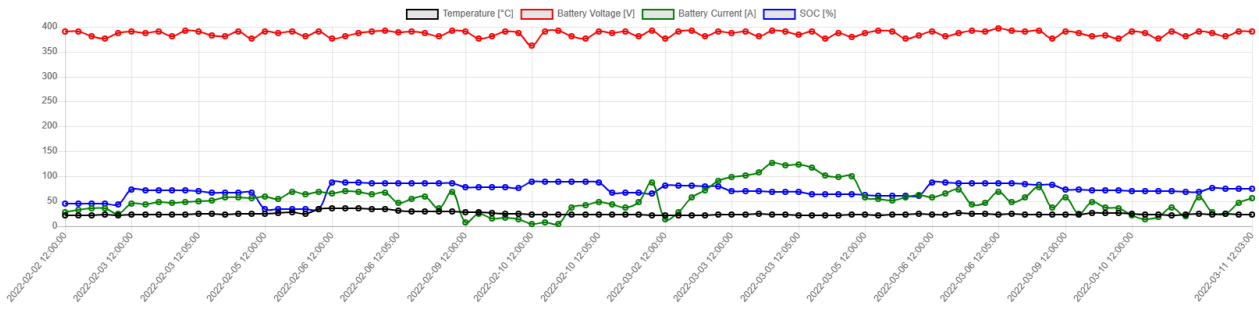


Figura 47: esempio di un grafico utilizzato nella web app e che legge contemporaneamente tutti i
della batteria

In particolare, per ogni grafico, viene riportato il tipo di dato che viene visualizzato, la data della rilevazione e il valore stesso del dato. Di seguito si riporta, come esempio, la funzione “ShowChartAll()” presente all’interno di “app/http/Controllers/CasaAutoController” che si occupa di recuperare i dati dal database e mandarli alla view dove vengono visualizzati attraverso un grafico.


```

//questa per vedere dati di tutte le misurazioni
public function ShowChartAll($cliente){
    $chart = new examplechart;
    $allid=Misurazioni::where("cliente",$cliente)->select("id")->orderBy("data", "asc")->get()->toArray(); //sono id delle misur
    $valoriTemp=[]; //una per ogni tipo di dato
    $valoriVolt=[];
    $valoriAmp=[];
    $valoriSoc=[]; //ora popolo i vettori con i dati del db per poi passarli al grafico

    for ($i=0;$i<count($allid);$i++){
        $app1=Misurazioni::where("id",$allid[$i])->value("temperatura"); //uso una variabile di appoggio
        $valoriTemp[$i]=$app1;
    }

    for ($i=0;$i<count($allid);$i++){
        $app2=Misurazioni::where("id",$allid[$i])->value("voltage");
        $valoriVolt[$i]=$app2;
    }

    for ($i=0;$i<count($allid);$i++){
        $app3=Misurazioni::where("id",$allid[$i])->value("amp");
        $valoriAmp[$i]=$app3;
    }

    for ($i=0;$i<count($allid);$i++){
        $app4=Misurazioni::where("id",$allid[$i])->value("soc");
        $valoriSoc[$i]=$app4;
    }

    $lab=[]; //questa per segnare le date sulle label:ne uso una sola perchè sono uguali per i vari tipi di dati che recupero
    for ($i=0;$i<count($allid);$i++){
        $appoggio=Misurazioni::where("id",$allid[$i])->value("data");
        $lab[$i]=$appoggio;
    }

    $chart->labels(array_values($lab));
    $chart->dataset('Temperature [°C]', 'line', array_values($valoriTemp))->options(['borderColor'=>'black','fill'=>'false']);
    $chart->dataset('Battery Voltage [V]', 'line', array_values($valoriVolt))->options(['borderColor'=>'red','fill'=>'false']);
    $chart->dataset('Battery Current [A]', 'line', array_values($valoriAmp))->options(['borderColor'=>'green','fill'=>'false']);
    $chart->dataset('SOC [%]', 'line', array_values($valoriSoc))->options(['borderColor'=>'blue','fill'=>'false']);
    return view('sample_view', compact('chart'));
}

```

Figura 48 : funzione che recupera i dati del database di una certa batteria e li usa per costruire il grafico

7.3.2 Modello dei dati

Laravel consente la gestione dell'evoluzione della struttura della base di dati e dei suoi contenuti durante lo sviluppo di un'applicazione attraverso Migrations e Seeding.

Migrations è un meccanismo di definizione della struttura del DB (tabelle relazioni tra di esse) che consente il versioning :

-Basato sulla definizione di classi che estendono la classe Laravel predefinita `Illuminate\Database\Migrations\Migration`

- Le classi consentono di definire lo schema delle tabelle utilizzando la FACADE Schema in due metodi pubblici: up() e down()
- I file che definiscono le classi sono memorizzati nella directory database/migrations

Seeding è un processo di popolamento delle tabelle del DB con valori utili per lo sviluppo ed il test dell'applicazione:

- Basato sulla definizione di classi che estendono la classe Laravel predefinita

Illuminate\Database\Seeder

- Ogni classe contiene un unico metodo pubblico, run(), che usa il Query Builder di Laravel per inserire valori nelle tabelle
- I file che definiscono i seeder sono memorizzati nella directory database/seeds

C'è la necessità di definire un meccanismo di interfaccia tra l'applicazione Laravel e le sorgenti di dati persistenti, in particolare lo scambio di dati con il DBMS. L'interfaccia si sviluppa su due livelli:

- Connessione dell'applicazione Laravel al DBMS tramite il meccanismo del PDO (PHP Data Object: consente astrazione dei dati in modo indipendente dal DBMS) attraverso i file “.env” e “config/database.php”

```
'connections' => [
    'mysql' => [
        'driver' => 'mysql',
        'url' => env('DATABASE_URL'),
        'host' => env('DB_HOST', '127.0.0.1'),
        'port' => env('DB_PORT', '3306'),
        'database' => env('DB_DATABASE', 'tesi'),
        'username' => env('DB_USERNAME', 'root'),
        'password' => env('DB_PASSWORD', ''),
        'unix_socket' => env('DB_SOCKET', ''),
        'charset' => 'utf8mb4',
        'collation' => 'utf8mb4_unicode_ci',
        'prefix' => '',
        'prefix_indexes' => true,
        'strict' => true,
        'engine' => null,
        'options' => extension_loaded('pdo_mysql') ? array_filter([
            PDO::MYSQL_ATTR_SSL_CA => env('MYSQL_ATTR_SSL_CA'),
        ]) : [],
    ],
],
```

Figura 49: dettaglio file “config/database”

```
APP_NAME=Laravel
APP_ENV=local
APP_KEY=base64:ZB0JX5iNWdu9MfZn2Pzu6PFza/dZ+ExvmIRqKoc/ToA=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack

DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=tesi
DB_USERNAME=root
DB_PASSWORD=

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120
```

Figura 50 : file “.env”

-Accesso alle funzioni del DBMS per la manipolazione dei dati del database e la mappatura degli stessi dati in strutture dati PHP. Ci sono diversi metodi e componenti, ma il più efficace è Eloquent, ovvero il componente di Laravel che implementa un ORM (Object Relational Mapping) :è una tecnica di programmazione che effettua il mapping tra i dati visti come relazioni dal DBMS e il mondo PHP che li vede come oggetti.

Tramite appositi comandi Artisan (l'interfaccia a linea di comando di Laravel), una volta stabilita la connessione tra applicazione e DBMS, è possibile creare le tabelle del database proprio come sono state definite le migrations e poi popolarle con i dati inseriti nel seeder. In questo modo si può cambiare velocemente il DBMS collegato alla web app perché tramite due semplici comandi le tabelle e i dati (presenti nelle migrations e nei seeders) possono essere "spostati" da un DBMS all'altro senza la necessità di riscrivere tutte le query ogni volta. [9]

All'interno dei files "database/migrations" e "database/seeds" sono visibili le tabelle ed i dati inseriti ed utilizzati per le fasi di sviluppo e test della web app sviluppata.

7.3.3 Package utilizzati

I package o plugin sono lo strumento primario per estendere le funzionalità di Laravel. Sono componenti software integrabili nel framework e sviluppate per fornire nuove funzionalità o estendere quelle esistenti

Si dividono in :

- Stand alone: progettati per l'uso di qualsiasi package PHP
- Specifici: nati per integrarsi con lo specifico workflow di Laravel e le sue componenti

Questi vengono installati tramite Composer, un *dependency manager* scritto in PHP che ci permette di installare e aggiornare le librerie esterne in modo molto semplice. Una volta definite le librerie che ci servono, Composer le scarica automaticamente e le inserisce nella cartella "vendor".

I package utilizzati in questa web app sono:

a)Laravel Collective

Package specifico che integra nel framework componenti:

- HTML : aiuta generare form e componenti HTML complesse con costrutti sintattici più semplici
- remote:per gestire le connessioni SSH all'interno dell'applicazione
- Annotation :per inserire annotazioni nel codice

```
{{ Form::open(array('route' => 'inserisciauto_post', 'files' => true, 'class' => 'contact-form')) }}
<div class="wrap-input">
  {{ Form::label('', '', ['class' => 'fa fa-picture-o']) }}
  {{ Form::label('foto', 'Foto', ['class' => 'label-input']) }}
  {{ Form::file('foto', ['class' => 'input', 'id' => 'foto']) }}
  @if ($errors->first('foto'))
  <ul class="errors">
    @foreach ($errors->get('foto') as $message)
    <li>{{ $message }}</li>
    @endforeach
  </ul>
  @endif
</div>
<div class="wrap-input">
  {{ Form::label('', '', ['class' => 'fa fa-id-card-o']) }}
  {{ Form::label('marca', 'Marca', ['class' => 'label-input']) }}
  {{ Form::text('marca', '', ['class' => 'input', 'id' => 'marca']) }}
  @if ($errors->first('marca'))
  <ul class="errors">
    @foreach ($errors->get('marca') as $message)
    <li>{{ $message }}</li>
    @endforeach
  </ul>
  @endif
</div>
<div class="wrap-input">
  {{ Form::label('', '', ['class' => 'fa fa-id-card-o']) }}
  {{ Form::label('modello', 'Modello', ['class' => 'label-input']) }}
  {{ Form::text('modello', '', ['class' => 'input', 'id' => 'surname']) }}
  @if ($errors->first('modello'))
  <ul class="errors">
    @foreach ($errors->get('modello') as $message)
    <li>{{ $message }}</li>
    @endforeach
  </ul>
  @endif
</div>
<div class="wrap-input">
  {{ Form::label('descrizione', 'Descrizione', ['class' => 'label-input']) }}
  {{ Form::textarea('descrizione', '', ['class' => 'input descrizione', 'id' => 'descrizione']) }}
  @if ($errors->first('descrizione'))
  <ul class="errors">
    @foreach ($errors->get('descrizione') as $message)
    <li>{{ $message }}</li>
    @endforeach
  </ul>
  @endif
</div>
<div class="container-form-btn">
  {{ Form::submit('Inserisci', ['class' => 'my-button']) }} </div> {{ Form::close() }}
```

Figura 51 : esempio di una form costruita sfruttando Laravel Collective

b) Laravel UI

Package molto diffuso perché tramite la componente “auth” vengono integrati nell’applicazione tutti gli elementi necessari per gestire il processo di autenticazione. Inoltre, mette a disposizione la “facade auth” (facade è un pattern che fornisce un’interfaccia semplice per accedere a sottosistemi molto complessi) che consente di accedere ad informazioni legate all’utente attualmente autenticato memorizzato nelle sessioni PHP e attiva il processo di logout

7.3.4 Autenticazione e autorizzazione

L’autenticazione è il processo che verifica l’identità di un utente, che si qualifica attraverso le sue credenziali (solitamente username e password). In Laravel questo processo si basa su una serie di componenti già configurate al momento della creazione di un nuovo progetto che aiutano a gestire:

- I driver di autenticazione, cioè le sorgenti di informazioni per l’autenticazione
- Il processo di autenticazione attraverso una serie di controller predefiniti e associati diverse fasi
- La persistenza dell’identità di un utente utilizzando le sessioni PHP (di default)

L’autorizzazione è un’attività che verifica se qualcuno (ad esempio un utente) ha il permesso di compiere un’azione su una risorsa. In Laravel ci sono diversi meccanismi possibili per gestire e definire le autorizzazioni. Quello utilizzato in questa web app sfrutta i Gates, ovvero funzioni che definiscono le condizioni per cui l’utente autenticato può eseguire una certa azione , definiti in “app/Providers/AuthServiceProvider”. Le regole così definite possono essere sfruttate utilizzando il “Middleware Authorize” (un software che funge da intermediario tra applicazioni, strumenti e database) in modo tale che il meccanismo di autorizzazione entri in azione prima dell’attivazione di una rotta o di un’azione di un controller.

```

public function boot()
{
    $this->registerPolicies();

    Gate::define('isCasaAuto', function($user){
        return $user->hasLivello('casaauto');
    });

    Gate::define('isCliente', function($user){
        return $user->hasLivello('cliente');
    });

    Gate::define('isLoggato', function($user){
        return isset($user);
    });
}

```

Figura 52 : porzione del file dove vengono definiti i Gates

Questi concetti sono stati utilizzati per garantire ai diversi tipi di utente (definiti in 6.1) l'accesso alle sole funzionalità dedicate in base al corrispettivo livello di utenza, evitando che un utente possa accedere a dati che non può visualizzare ed impedire che compia azioni a lui non concesse.

Infatti diverse rotte possono essere attivate con parametri passati nell' URL: questi in teoria possono essere modificati dall'utente per accedere a pagine e/o informazioni a cui normalmente non ha accesso. Il meccanismo dei Gates serve proprio ad evitare l'accesso non autorizzato perché prima dell'attivazione di una rotta o di un'azione del controller l'applicazione stessa, tramite il middleware, svolge dei controlli preventivi al fine di verificare se l'accesso a certe risorse è autorizzato oppure no.

403 | This action is unauthorized.

Figura 53 : messaggio mostrato dal browser se si tenta di effettuare azione non autorizzata

7.3.5 Password criptate

Per proteggere dati molto sensibili, come ad esempio le password degli utenti registrati, Laravel mette a disposizione la Facade “hash”. In questo modo se un utente con intenzioni malevole dovesse accedere al database dove sono contenuti i dati, quelli criptati sarebbero comunque protetti.

In questa web app la facade “hash” è stata appunto utilizzata per criptare le password degli utenti:

- all’interno della directory “database/seeds” per le password degli utenti inseriti tramite il seeding del database
- All’interno di “app/http/Controllers/CasaAutoController” nella funzione “storecliente()” quando la casa automobilistica , tramite apposita form, va a creare il profilo di un nuovo utente inserendo tutti i dati, password compresa

```
//serve per registrare un nuovo cliente
public function storecliente (NewClienteRequest $request){
    $cliente= new Users; //con la s sta sotto package mod
    $cliente->fill($request->validated()); //serve per la
    $appoggio1=$cliente['password']; // devo utilizz
    $appoggio2=Hash::make($appoggio1); // password affi
    $cliente['password']=$appoggio2; // per il nuov
    $cliente->save(); //salva dati nel db
```

Figura 54 : porzione della funzione “storecliente()”

7.3.6 Sanificazione di variabili

Laravel permette di “sanificare” i valori delle variabili PHP passate al template Blade applicando la funzione PHP “ htmlspecialchars” a fini di sicurezza: se la variabile passata contiene del malware (cioè del codice con intenzioni malevole) questa funzione è in grado di convertire i caratteri speciali di una stringa (quindi tutti quei caratteri che potenzialmente identificano delle istruzioni di codice da eseguire) nei corrispondenti codici HTML, evitando quindi che eventuale codice malware possa essere eseguito e causare danni.[9], [53]

La variabile \$prova per essere sanificata deve essere passata alla view con la seguente sintassi : {{ \$prova }}. Di seguito un esempio estrapolato proprio dal codice della web app.

```
@foreach($clienti as $cliente)
<tr>
  <td>{{ $cliente->name }}</td>
  <td>{{ $cliente->cognome }} </td>
  <td>{{ $cliente->auto }}</td>
  <td>{{ $cliente->targa }} </td>
  <td>{{ $cliente->datavendita }}</td>
  <td>{{ $cliente->username }} </td>
```

Figura 55 : dettaglio di una view dove la variabile \$clienti, utilizzata all'interno di un foreach, viene utilizzata solo dopo essere stata sanificata

7.3.7 MySQL

MySQL è un sistema open source di gestione di database relazionali SQL sviluppato e supportato da Oracle, utilizzato per archiviare i dati necessari all'applicazione web sviluppata. Tramite "phpMyAdmin", un'applicazione disponibile con Xampp, è possibile gestire il database MySQL utilizzato per gestire i dati della web app. Di seguito si riporta una porzione dell'app con le tabelle create.

Tabella	Azione	Righe	Tipo	Codifica caratteri	Dimensione
<input type="checkbox"/> auto	Mostra Struttura Cerca Inserisci Svuota Elimina	11	InnoDB	utf8mb4_unicode_ci	32.0 KiB
<input type="checkbox"/> messaggi	Mostra Struttura Cerca Inserisci Svuota Elimina	9	InnoDB	utf8mb4_unicode_ci	48.0 KiB
<input type="checkbox"/> migrations	Mostra Struttura Cerca Inserisci Svuota Elimina	6	InnoDB	utf8mb4_unicode_ci	16.0 KiB
<input type="checkbox"/> misurazioni	Mostra Struttura Cerca Inserisci Svuota Elimina	90	InnoDB	utf8mb4_unicode_ci	32.0 KiB
<input type="checkbox"/> users	Mostra Struttura Cerca Inserisci Svuota Elimina	6	InnoDB	utf8mb4_unicode_ci	64.0 KiB

Figura 56 : porzione della schermata di phpMyAdmin con le tabelle del database

In particolare la tabella:

- auto: serve per gestire le auto della casa automobilistica, visibili anche dall'area pubblica
- messaggi: per gestire i messaggi che i clienti possono scambiarsi con la casa automobilistica

- misurazioni: per gestire i dati sulle varie misurazioni che i sensori effettuano sulle auto vendute
- users: per gestire gli utenti in base al loro livello

7.3.8 Web Server Apache

Apache HTTP Server, o più comunemente Apache, è il nome di un server web libero sviluppato dalla Apache Software Foundation. È la piattaforma server Web modulare più diffusa, in grado di operare su una grande varietà di sistemi operativi. È un software che realizza le funzioni di trasporto delle informazioni, di internet e di collegamento, ed ha il vantaggio di offrire funzioni di controllo per la sicurezza come quelle effettuate da un proxy. Il suo compito è stabilire una connessione tra un server e i browser dei visitatori del sito web (Firefox, Google Chrome, Safari, ecc.) mentre inviano file avanti e indietro tra di loro (struttura client-server). Quando un visitatore vuole caricare una pagina sul tuo sito web, il suo browser invia una richiesta al server e Apache restituisce una risposta con tutti i file richiesti (testo, immagini, ecc.). Il server e il client comunicano tramite il protocollo HTTP e il server web Apache è responsabile della comunicazione fluida e sicura tra le due macchine.[54]

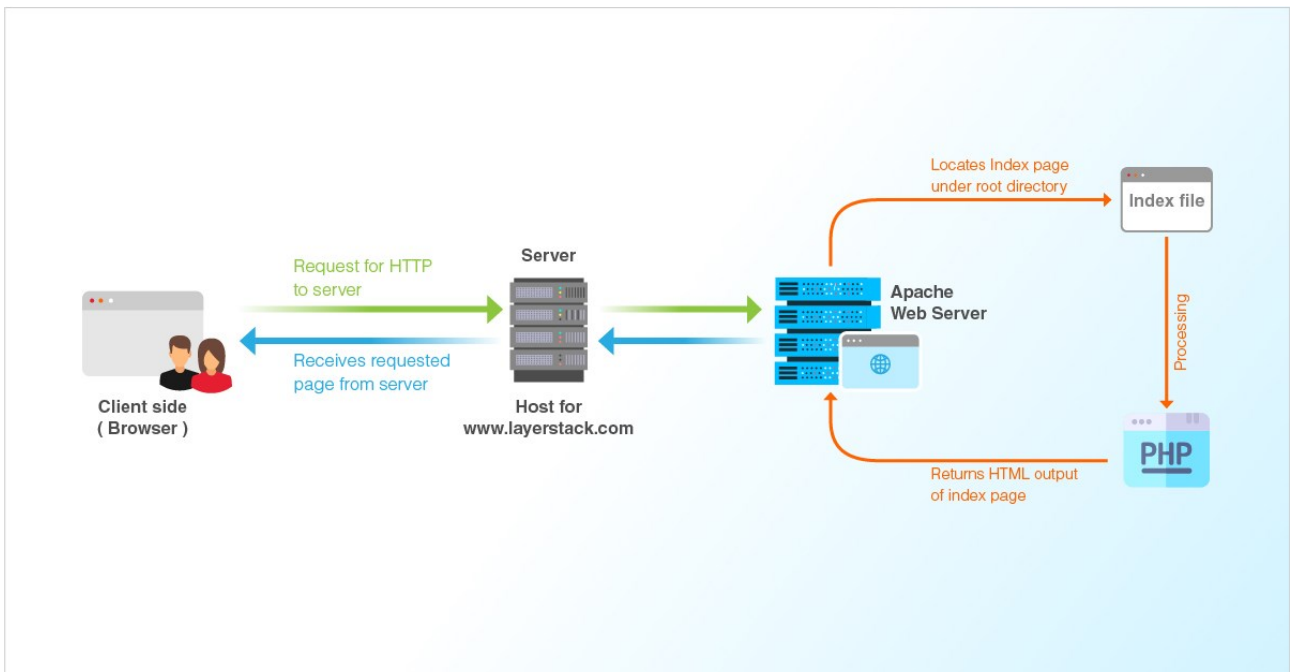


Figura 57 : Web server apache [54]

Apache è altamente personalizzabile grazie alla sua infrastruttura open source. Per questo motivo, gli sviluppatori web e gli utenti possono adattare il codice sorgente in base al tipo di sito web che stanno creando. Apache inoltre fornisce numerosi moduli che consentono agli amministratori del server di attivare e disattivare funzionalità aggiuntive. Il web server Apache dispone di moduli per la sicurezza, la memorizzazione nella cache, la riscrittura degli URL, l'autenticazione della password e altre funzionalità. Per configurare Apache si deve accedere al file “.htaccess”.

CONCLUSIONI

Questo studio si è posto l'obiettivo di definire uno stato dell'arte per la gestione delle batterie in cloud vista la sempre crescente rilevanza energetica e strategica che queste componenti stanno avendo. Dallo studio emerge infatti che le batterie al litio sono una fonte di energia sicura ed affidabile, ma queste necessitano di un accurato monitoraggio per garantirne la massima efficienza.

Per questo scopo gli attuali sistemi BMS, cioè quelli locali e non connessi alla rete, presentano delle limitazioni di elaborazione ed archiviazione che non consentono di sfruttare a pieno l'enorme potenziale delle batterie in quanto non sono in grado di supportare in modo efficace ed in tempo reale algoritmi per la rilevazione di guasti e/o situazioni critiche.

La rapida diffusione della tecnologia IoT che si è vista negli ultimi anni ha permesso alle principali aziende e società di telecomunicazioni, tecnologia ed energia a valutarne un impiego sempre maggiore nei loro campi di utilizzo specifici, ottenendo un continuo miglioramento delle prestazioni e riduzione dei consumi.

Il cloud computing è uno strumento che si sta rivelando essenziale per superare i limiti dei normali BMS grazie alle sue caratteristiche, ovvero grandi capacità di storage ed elaborazione di tutti i dati raccolti proprio dai sensori IoT.

Sfruttando il cloud computing e il paradigma dell'IoT è possibile implementare dei BMS intelligenti basati sul cloud superando le attuali limitazioni dei BMS locali. Nonostante questi sistemi siano ancora nelle loro fasi "embrionali" sono però già in grado di offrire soluzioni estremamente potenti ed efficaci, e risulta evidente che nei prossimi anni la ricerca e lo sviluppo consentirà di ottenere performance sempre più elevate.

L'implementazione di questi cloud BMS permette anche di fornire un sistema per la visualizzazione e l'analisi dei dati raccolti agli utenti di applicazioni che sono collegate ai sensori IoT. Queste applicazioni sono molteplici e ovviamente la loro struttura e complessità variano a seconda del tipo di dispositivo da monitorare (come, ad esempio, la batteria di un'auto elettrica o i pannelli fotovoltaici per la produzione di energia) e al tipo di utente collegato (produttore del dispositivo stesso o cliente).

RINGRAZIAMENTI

Desidero ringraziare il professor Massimo Conti, che mi ha guidato e che ha sostenuto il mio percorso di tesi, aiutandomi a valorizzare il mio lavoro, ad approfondire le giuste argomentazioni e accompagnandomi durante la stesura dell'elaborato. In particolare lo ringrazio anche per aver creduto fortemente nel programma UNIVPM "Doppia Carriera Studente-Atleta" che mi ha aiutato a conciliare la mia attività sportiva con lo studio.

Ringrazio tutta la mia famiglia e la mia fidanzata Ilaria per avermi costantemente supportato nella scelta di intraprendere questo percorso di studio e formazione.

Dato che non sempre è così ovvio, ringrazio anche tutti i professori che durante questi anni, con passione e dedizione, hanno saputo stimolare interesse e curiosità durante le loro lezioni spingendomi a dare sempre il massimo nel mio percorso.

Un ringraziamento speciale va anche a tutti i miei compagni di corso, in particolare i miei amici Giovanni, Filippo, Gioele ed Alessandro, per i momenti di confronto e supporto reciproco nei momenti di difficoltà ma anche per aver condiviso i momenti più belli dopo aver superato gli esami e per la collaborazione nell'aver svolto diversi progetti didattici durante questi anni.

BIBLIOGRAFIA

- [1] “Mobile marketing statistics compilation | Smart Insights.” <https://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- [2] “Dazeinfo:: Business News On Smartphone, Social Media, Ecommerce.” <https://dazeinfo.com/>
- [3] “Wikipedia.” <https://www.wikipedia.org/>
- [4] “Android Mobile App Developer Tools – Android Developers.” <https://developer.android.com/>
- [5] “Home Page – Gamma Ufficio Srl Aulla MS – Soluzioni Informatiche Professionali – Siti Web, E-Commerce, Oa Sistemi - Gamma Ufficio Srl.” <https://www.gammaufficio.eu/>
- [6] Q. I. Sarhan and I. S. Gawdan, “Web Applications and Web Services: A Comparative Study,” *Science Journal of University of Zakho*, vol. 6, no. 1, p. 35, Mar. 2018, doi: 10.25271/2018.6.1.375.
- [7] “Spring | Home.” <https://spring.io/>
- [8] “The web framework for perfectionists with deadlines | Django.” <https://www.djangoproject.com/>
- [9] “Laravel - The PHP Framework For Web Artisans.” <https://laravel.com/>
- [10] “The Model View Controller Pattern – MVC Architecture and Frameworks Explained.” <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>
- [11] “Ionic Framework - The Cross-Platform App Development Leader.” <https://ionicframework.com/>
- [12] “Flutter - Build apps for any screen.” <https://flutter.dev/>
- [13] H. C. Y. Chan, “Internet of Things Business Models,” *Journal of Service Science and Management*, vol. 08, no. 04, pp. 552–568, 2015, doi: 10.4236/jssm.2015.84056.
- [14] T. di Laurea Magistrale and T. Accademico Profssa Sara Monaci Tutor Aziendale Dott Luca Vignaroli Candidato Martina Puliafito, “Internet of Things e Transmedialità L’evoluzione della televisione connessa,” 2017.
- [15] R. Chmo Guzzonato Luca Correlatore Chmo Bonollo Giuliano Laureanda Drivas Mariagiulia, “IOT E BIG DATA: QUALE VALORE PER LE PMI?”.

- [16] “Gartner | Delivering Actionable, Objective Insight to Executives and Their Teams.”
<https://www.gartner.com/en>
- [17] Laura Sartorato "Cloud computing :pervasività della rete"
- [18] “What is Smart Factory? The Impact of Factory 4.0.”
<https://www.avsystem.com/blog/smart-factory/>
- [19] F. Ferrara Matr and C. Profssa Federica Brunetta, “L’ERA DELL’INTERNET OF THINGS E LE NUOVE SFIDE PER LA PROTEZIONE E VALORIZZAZIONE DELLA PROPRIETA’ INTELLETTUALE RELATORE Prof.ssa Maria Isabella Leone CANDIDATO”.
- [20] “Modelli di Distribuzione – Cloud Computing & Mobile App.”
<https://annaponti97.wordpress.com/2016/04/14/modelli-di-distribuzione-2/>
- [21] Tesi_Laura_Sartorato
- [22] “Cloud Computing: Differenza tra Cloud IAAS, PAAS e SAAS | Informatica e Ingegneria Online.” <https://vitolavecchia.altervista.org/cloud-computing-differenza-tra-cloud-iaas-paas-e-saas/>
- [23] Martina Puliafito Politecnico di Torino Internet of Things e Transmedialità L’evoluzione della televisione connessa,”
- [24] N. Roman Herbst, S. Kounev, and R. Reussner, “Elasticity in Cloud Computing: What It Is, and What It Is Not.”
- [25] “Synergy Research Group | Strategic Market Intelligence for Emerging IT & Cloud.”
<https://www.srgresearch.com/>
- [26] “Servizi di cloud computing | Google Cloud.” <https://cloud.google.com/?hl=it>
- [27] “Servizi di cloud computing | Microsoft Azure.” <https://azure.microsoft.com/it-it>
- [28] “Servizi di cloud computing – Amazon Web Services (AWS).”
<https://aws.amazon.com/it/>
- [29] “IoT Analytics - ThingSpeak Internet of Things.” <https://thingspeak.com/>
- [30] “Node-RED.” <https://nodered.org/>
- [31] R. Calero and F. Rivera, “PROGETTAZIONE E SVILUPPO DI UNA WEB APPLICATION WEB APP PER VISUALIZZAZIONE REAL-TIME DI DATI PROVENIENTI DA ACCELEROMETRO ADXL355 USANDO PROTOCOLLO MQTT,” 2019.
- [32] “Cos è come funziona MQTT broker automazione domotica...”
http://www.soluzionisemplici.com/cosa_e_MQTT_come_funziona.htm

- [33] UNIVERSITÀ POLITECNICA DELLE MARCHE FACOLTÀ DI INGEGNERIA
Corso di Laurea Specialistica in Ingegneria delle Telecomunicazioni , Attorresi
Giorgio “Studio e implementazione di un sensore NB-IoT per la rilevazione di dati
ambientali”
- [34] S. K. Viswanath *et al.*, “System design of the internet of things for residential smart
grid,” *IEEE Wirel Commun*, vol. 23, no. 5, pp. 90–98, Oct. 2016, doi:
10.1109/MWC.2016.7721747.
- [35] “Oracle | Cloud Applications and Cloud Platform.” <https://www.oracle.com/>
- [36] Paolo Atzeni, Stefano Ceri, and Piero Fraternali, *Basi di Dati*, V. McGraw-Hill
Education, 2018.
- [37] N. Nitta, F. Wu, J. T. Lee, and G. Yushin, “Li-ion battery materials: Present and
future,” *Materials Today*, vol. 18, no. 5. Elsevier B.V., pp. 252–264, Jun. 01, 2015.
doi: 10.1016/j.mattod.2014.10.040.
- [38] “Sicurezza delle batterie agli ioni di litio – Antincendio Italia.” <https://antincendio-italia.it/sicurezza-delle-batterie-agli-ioni-di-litio/>
- [39] M. K. Tran, S. Panchal, T. D. Khang, K. Panchal, R. Fraser, and M. Fowler,
“Concept Review of a Cloud-Based Smart Battery Management System for Lithium-
Ion Batteries: Feasibility, Logistics, and Functionality,” *Batteries*, vol. 8, no. 2. MDPI,
Feb. 01, 2022. doi: 10.3390/batteries8020019.
- [40] W. Li, M. Rentemeister, J. Badede, D. Jöst, D. Schulte, and D. U. Sauer, “Digital
twin for battery systems: Cloud battery management system with online state-of-
charge and state-of-health estimation,” *J Energy Storage*, vol. 30, Aug. 2020, doi:
10.1016/j.est.2020.101557.
- [41] S. Haldar, S. Mondal, A. Mondal, and R. Banerjee, “Battery Management System
Using State of Charge Estimation: An IOT Based Approach; Battery Management
System Using State of Charge Estimation: An IOT Based Approach,” 2020.
- [42] X. Li, H. Liu, W. Wang, Y. Zheng, H. Lv, and Z. Lv, “Big data analysis of the Internet
of Things in the digital twins of smart city based on deep learning,” *Future
Generation Computer Systems*, vol. 128, pp. 167–177, Mar. 2022, doi:
10.1016/j.future.2021.10.006.
- [43] Z. Liang, Z. Liang, Y. Zheng, B. Liang, and L. Zheng, “Data analysis and
visualization platform design for batteries using flask-based python web service,”
World Electric Vehicle Journal, vol. 12, no. 4, Dec. 2021, doi:
10.3390/wevj12040187.

- [44] K. Friansa, I. N. Haq, B. M. Santi, D. Kurniadi, E. Leksono, and B. Yulianto, "Development of Battery Monitoring System in Smart Microgrid Based on Internet of Things (IoT)," in *Procedia Engineering*, 2017, vol. 170, pp. 482–487. doi: 10.1016/j.proeng.2017.03.077.
- [45] C. Giosuè *et al.*, "An exploratory study of the policies and legislative perspectives on the end-of-life of lithium-ion batteries from the perspective of producer obligation," *Sustainability (Switzerland)*, vol. 13, no. 20, Oct. 2021, doi: 10.3390/su132011154.
- [46] Y. Zhao *et al.*, "A Review on Battery Market Trends, Second-Life Reuse, and Recycling," *Sustainable Chemistry*, vol. 2, no. 1, pp. 167–205, Mar. 2021, doi: 10.3390/suschem2010011.
- [47] "Servizio Ricerca del Parlamento europeo." <https://www.europarl.europa.eu/at-your-service/it/stay-informed/research-and-analysis>
- [48] Tsiropoulos I, Tarvydas D, and Lebedeva N, "Li-ion batteries for mobility and stationary storage applications Scenarios for costs and market growth," 2018. doi: 10.2760/87175.
- [49] "European Commission-Press release Green Deal: Sustainable batteries for a circular and climate neutral economy," 2020.
- [50] O. Velázquez-Martínez, J. Valio, A. Santasalo-Aarnio, M. Reuter, and R. Serna-Guerrero, "A critical review of lithium-ion battery recycling processes from a circular economy perspective," *Batteries*, vol. 5, no. 4. MDPI Multidisciplinary Digital Publishing Institute, Dec. 01, 2019. doi: 10.3390/batteries5040068.
- [51] A. Beaudet, F. Larouche, K. Amouzegar, P. Bouchard, and K. Zaghbi, "Key challenges and opportunities for recycling electric vehicle battery materials," *Sustainability (Switzerland)*, vol. 12, no. 14. MDPI, Jul. 01, 2020. doi: 10.3390/su12145837.
- [52] "Laravel Charts." <https://charts.erik.cat/>
- [53] "PHP: Hypertext Preprocessor." <https://www.php.net/>
- [54] "Welcome! - The Apache HTTP Server Project." <https://httpd.apache.org/>

