

**Alma Mater Studiorum – Università degli studi di Bologna**  
Campus di Cesena

---

SCUOLA DI SCIENZE

---

Corso di studi in Ingegneria e Scienze Informatiche

# **WEB APP PER LA GESTIONE DELLA PROPRIA AUTO ELETTRICA**

**Tesi di Laurea in:**  
Sistemi Multimediali

**Relatore:**  
Prof.ssa Paola Salomoni

**Presentata da:**  
Monica Gondolini

**Correlatore:**  
Dott.ssa Catia Prandi

**Sessione III**  
**Anno accademico 2014/2015**



---

# Indice

---

<b>INDICE.....</b>	<b>I</b>
<b>INTRODUZIONE.....</b>	<b>1</b>
<b>1 GESTIONE DELLA MOBILITÀ ELETTRICA.....</b>	<b>7</b>
1.1 MOBILITÀ ELETTRICA.....	7
1.1.1 <i>Definizione di auto elettrica.....</i>	7
1.1.2 <i>Efficienza dell'auto elettrica e vantaggi.....</i>	8
1.1.3 <i>Benefici dell'auto elettrica.....</i>	9
1.1.4 <i>Batterie dell'auto elettrica.....</i>	11
1.1.5 <i>Infrastrutture di ricarica .....</i>	12
1.2 ECO-DRIVING.....	15
1.2.1 <i>Concetto di Eco-driving.....</i>	15
1.2.2 <i>Regole d'oro dell'Eco-Driving.....</i>	16
1.2.3 <i>Regole d'argento dell'Eco-Driving.....</i>	19
1.3 APPLICAZIONI A SUPPORTO DELLA MOBILITÀ ELETTRICA.....	19
1.3.1 <i>Classificazione delle applicazioni a supporto della mobilità.....</i>	20
1.3.2 <i>Esempi di applicazioni per la mobilità elettrica.....</i>	22
<b>2 TECNOLOGIE DI PROGETTO.....</b>	<b>33</b>
2.1 SMARTPHONE E TABLET.....	33
2.2 WEB APP.....	34
2.2.1 <i>Applicazioni ibride come evoluzione delle Web App.....</i>	35
2.3 APACHE CORDOVA COME FRAMEWORK DI SVILUPPO.....	37
2.3.1 <i>Adobe Phonegap e Apache Cordova: le differenze.....</i>	39
2.4 HTML PER LA STRUTTURA DEI CONTENUTI.....	40
2.4.1 <i>Gestione dei documenti HTML da parte del Browser.....</i>	40
2.4.2 <i>Sintassi HTML.....</i>	41
2.4.3 <i>Struttura di un documento HTML.....</i>	42
2.4.4 <i>Novità introdotte da HTML5.....</i>	43
2.5 CSS PER LA PRESENTAZIONE DEI CONTENUTI HTML.....	46
2.6 JAVASCRIPT PER L'INTERATTIVITÀ UTENTE.....	48
2.6.1 <i>Aspetti strutturali .....</i>	49
2.6.2 <i>Sicurezza.....</i>	50
2.6.3 <i>Eventi.....</i>	50

2.6.4 Libreria <i>jQuery</i> e richieste <i>Ajax</i> .....	51
2.7 PHP PER L'ACCESSO AL DATABASE.....	53
2.7.1 <i>Caratteristiche di PHP</i> .....	54
2.7.2 <i>Sintassi</i> .....	55
<b>3 IMPLEMENTAZIONE.....</b>	<b>57</b>
3.1 INTRODUZIONE.....	57
3.2 DESCRIZIONE E MOTIVAZIONE DELLE SCELTE IMPLEMENTATIVE.....	58
3.3 ANDROID STUDIO COME IDE DI SVILUPPO.....	59
3.4 ADOBE PHONEGAP.....	60
3.4.1 <i>Struttura del progetto con Adobe Phonegap</i> .....	61
3.4.2 <i>Plugin di Adobe Phonegap</i> .....	63
3.5 LE INTERFACCE GRAFICHE DI MYECO CAR.....	64
3.6 JAVASCRIPT DI MYECO CAR.....	68
3.6.1 <i>ProgressBar.js API</i> .....	68
3.6.2 <i>Google Maps JavaScript API</i> .....	69
3.6.3 <i>Google Charts API</i> .....	70
3.6.4 <i>Collegamenti tra pagine</i> .....	71
3.6.5 <i>Connessione al database tramite richieste Ajax</i> .....	72
3.6.6 <i>Funzione responseMessage()</i> .....	73
3.7 WAMP SERVER PER LA SIMULAZIONE DATABASE.....	75
3.7.1 <i>PhpMyAdmin</i> .....	75
3.7.2 <i>Organizzazione delle pagine server-side</i> .....	75
3.7.3 <i>Query per inserimenti e richieste al database</i> .....	76
3.7.4 <i>Codifica JSON</i> .....	78
<b>CONCLUSIONI.....</b>	<b>79</b>
<b>BIBLIOGRAFIA.....</b>	<b>I</b>

# Introduzione

---

Uno dei più importanti settori dell'industria mondiale è senza dubbio quello automobilistico. L'automobile è sempre stata il mezzo di trasporto che ha rivoluzionato la vita dell'uomo. La sua eccellenza è data dalla sua continua evoluzione in base alle esigenze dell'uomo e grazie ai vantaggi che offre agli utenti durante gli spostamenti. Possedere un proprio veicolo infatti permette spostamenti più rilassanti rispetto ai mezzi pubblici per svariati motivi, tra i quali possiamo nominare principalmente le tempistiche: non doversi adattare ad orari prefissati dall'ente di trasporto pubblico, con notevole risparmio di tempo nel percorso da casa alla fermata più vicina, guida più veloce grazie all'assenza di fermate e poter raggiungere direttamente il luogo di destinazione senza dover cambiare linea o spostarsi a piedi. L'alimentazione di veicoli a benzina o Diesel era conveniente negli anni passati, in cui il prezzo al barile non era molto elevato. Tuttavia, l'innalzamento dei prezzi del petrolio e di conseguenza quello dei suoi derivati, ha reso pian piano l'utilizzo della propria auto molto dispendioso, soprattutto i piccoli spostamenti giornalieri.

Il petrolio non è altro che un combustibile fossile formato da una miscela di idrocarburi accumulata nel corso degli anni in giacimenti sotto la crosta terrestre, per cui presenta il problema di essere una fonte di energia non rinnovabile. A causa della sua limitata disponibilità a lungo termine, gli specialisti del settore automobilistico hanno iniziato a pensare a fonti di energie alternative e rinnovabili come nuovo metodo di alimentazione del motore. Altri problemi dei motori a benzina o diesel riguardano l'emissione di polveri nocive e altamente inquinanti come il PM10 e la generazione di CO2. Questi agenti inquinanti causano l'aumento del buco dell'ozono con conseguente diminuzione della protezione dei raggi UV che giungono sulla terra e aumento del riscaldamento globale, con gravi conseguenze sia sull'ambiente sulla salute della

popolazione mondiale. Attualmente, in numerosi paesi si sta diffondendo una certa sensibilizzazione per l'ecologia grazie alla presenza di normative sulla diminuzione dei gas serra. A causa dei problemi dovuti al costo del carburante sempre meno a portata degli utenti e ai problemi ecologici, si sta passando all'utilizzo di energie alternative ai combustibili fossili.

Una soluzione nel settore automobilistico ai problemi sopraelencati è la diffusione di veicoli elettrici (Electric Vehicles EV). Per EV si intende qualsiasi veicolo il cui movimento è azionato dall'energia elettrica. Questa tesi propone uno studio più specifico, concentrandosi esclusivamente sul concetto di auto elettrica come tipologia di veicolo elettrico.

L'energia elettrica è una fonte rinnovabile che nelle auto elettriche viene immagazzinata in batterie ricaricabili. Il motore ricava l'energia necessaria per il movimento dalle batterie situate. I vantaggi a livello ambientale dovuti all'adozione di questo motore sono notevoli nonostante questa soluzione non sia ancora del tutto ecologica per via del ciclo di produzione dell'energia elettrica per la ricarica delle batterie. Essendo una fonte di energia secondaria, l'energia elettrica viene prodotta da combustibili fossili. Tuttavia, l'inquinamento prodotto è comunque nettamente minore rispetto a quello generato dalle auto con motore a combustibile per cui la scelta elettrica rimane conveniente.

Le auto elettriche risultano molto vantaggiose, oltre che a livello ambientale, per il basso costo dell'energia. Confrontando i costi dell'energia elettrica e del carburante annui per lo stesso numero di chilometri percorsi, si ottiene un risparmio di circa 1000€. A fare i conti con questa cifra sorprendente vi sono le limitazioni legate all'autonomia dell'auto elettrica e la ancora scarsa densità di colonnine elettriche sul territorio. Questi fenomeni generano “range anxiety” nei guidatori, cioè la paura di esaurire l'energia durante il percorso e rimanere bloccati. Per far fronte a queste limitazioni il governo dovrebbe impegnarsi alla diffusione su tutto il territorio nazionale di una vasta rete di rifornimenti elettrici e gli utenti dovrebbero attenersi a tecniche di guida ecologiche per diminuire i consumi durante gli spostamenti. Questo concetto si chiama Eco-driving e contiene un insieme di suggerimenti per sfruttare al meglio l'energia cinetica generata durante il movimento, quindi diminuendo lo sfruttamento continui di energia elettrica.

A supporto della guida ecologica e alla mobilità elettrica sono state create applicazioni per dispositivi mobili, le quali vengono in aiuto agli utenti di auto elettriche nella gestione della propria vettura e nella ricerca delle colonnine elettriche più vicine.

La diffusione dell'auto elettrica si inserisce a poco a poco in un contesto europeo eco-compatibile. Le statistiche affermano che in Europa i paesi più interessati al veicolo elettrico sono Turchia (93%), Russia (87%), Portogallo (82%), Spagna e Italia (76%). Tra i paesi meno convinti all'adozione di veicoli elettrici si hanno Francia, Germania e Regno Unito. Tuttavia, le persone che realmente prenderebbero in considerazione l'acquisto di un'auto elettrica sono nettamente minori rispetto all'interesse. Ciò è dovuto alle limitazioni precedentemente accennate, al prezzo ancora elevato delle auto elettriche rispetto a quelle tradizionali e il costo delle batterie di ricambio. Nonostante il risparmio sia decisamente conveniente a lungo andare grazie al basso costo dell'energia elettrica, l'automobile elettrica viene ancora considerata un lusso inopportuno, come dimostrato dal successo dei veicoli a basso costo.

In questa tesi viene trattato il tema della mobilità elettrica essendo il contesto nel quale si è sviluppato il progetto di tesi. Verrà illustrata la definizione di auto elettrica e ne verranno spiegate le caratteristiche principali, definendone i benefici ambientali, i vantaggi e gli svantaggi. Si parlerà del concetto di Ecodriving da associare alla guida di un veicolo elettrico per ottimizzare i consumi di energia e diminuire il livello di “range anxiety” negli utenti. Inoltre, verranno elencate applicazioni già esistenti a favore delle auto elettriche.

In seguito verranno descritte le tecnologie adottate durante lo sviluppo del progetto. Principalmente verrà approfondito il concetto di Applicazione Ibrida e i framework di sviluppo associati. Si parlerà delle principali tecnologie web utilizzate: HTML per la struttura dei contenuti web, CSS per la presentazione dei contenuti, JavaScript per l'interattività utente e PHP per la comunicazione col server.

Infine verranno illustrate le decisioni implementative adottate per lo sviluppo del progetto. In particolare verrà spiegato il motivo di adozione di una scelta rispetto ad altre nella realizzazione del progetto stesso. Verrà descritto il framework utilizzato per la creazione della nostra applicazione ibrida, Adobe Phonegap, e tutte le API JavaScript

utilizzate, come jQuery API per facilitare la scrittura del codice, l'API di Google Maps per la geolocalizzazione e l'API di Google Charts per la realizzazione di grafici. Inoltre verranno spiegati i passi seguiti per la simulazione server tramite WAMPServer e codice PHP.

Questa tesi si pone come obiettivo lo sviluppo di un'applicazione per il supporto alla mobilità elettrica dedicata ai dispositivi mobili. Lo scopo essenziale della realizzazione dell'applicativo è la diminuzione della range anxiety dovuta alla guida di mezzi elettrici caratterizzati da scarsa autonomia, fornendo agli utenti informazioni riguardanti lo stato di carica della propria auto e una serie di grafici dei consumi per tenere sotto controllo la propria guida ecologica. Inoltre l'applicazione realizzata è stata pensata per l'utilizzo fuori casa, per questo motivo è stata necessaria la realizzazione di due caratteristiche per il controllo della propria auto a lunga distanza. Quest'ultime sono la localizzazione della propria auto nel caso sia stata parcheggiata in luoghi poco conosciuti e l'invio di comandi come l'accensione del condizionatore durante la carica dell'auto.

La tesi è strutturata in tre capitoli:

- *Gestione della mobilità elettrica*: in questo capitolo viene esposto il concetto di mobilità elettrica. Viene data una definizione di auto elettrica elencandone i vantaggi e gli svantaggi. Verranno approfondite le caratteristiche principali delle auto elettriche e dei loro consumi, come il rendimento dei motori elettrici, le batterie e i metodi di ricarica. Viene introdotto il concetto di Ecodriving e le sue regole fondamentali. Vengono offerti esempi di applicazioni a supporto della mobilità elettrica già esistenti in modo tale da rendere chiaro il quadro generale in cui si ambienta il nostro progetto.
- *Tecnologie di sviluppo*: questo capitolo parla delle specifiche di sviluppo del nostro progetto. Vengono analizzate le caratteristiche fondamentali di ogni tecnologia utilizzata, il loro ruolo all'interno del progetto.
- *Implementazione*: In quest'ultimo capitolo vengono esposte le scelte implementative per lo sviluppo dell'applicazione myEcoCar. Vengono descritte passo a passo le funzionalità della nostra applicazione,



indicando le varie API JavaScript utilizzate e descrivendone il proprio compito all'interno del progetto. Inoltre viene spiegato il processo adottato per la simulazione server.



# 1 Gestione della mobilità elettrica

---

In questo capitolo verrà introdotto il concetto di mobilità elettrica e ne verranno descritte le principali caratteristiche. Verrà inoltre descritto il concetto di Eco-driving e verranno classificate le applicazioni a supporto delle auto elettriche.

## 1.1 Mobilità elettrica

### 1.1.1 Definizione di auto elettrica

Per auto elettrica si intende un'automobile che viene azionata da uno o più motori elettrici che utilizzano come fonte energetica primaria l'energia elettrica, energia chimica che viene immagazzinata in un serbatoio energetico costituito da una o più batterie ricaricabili che la rendono disponibile al motore [WIK16a].

Le auto elettriche rientrano nella tipologia più generale di veicolo elettrico (EV, Electric Vehicle), il quale si riferisce a qualsiasi veicolo che fa uso di motori elettrici per la propulsione. Questi motori possono essere alimentati da batterie ricaricabili ma anche tramite reti aeree con fili sospesi (filobus), binari o fasce di alimentazione conduttrici per contatti striscianti laterali [WIK16b].

Possiamo riferirci alle auto elettriche anche con il termine “battery electric vehicle (BEV)” quando alimentate da batterie ricaricabili, mentre per quelle alimentate da fonti di energia diverse bisogna utilizzare una terminologia specifica. Per esempio, le auto alimentate tramite pannelli fotovoltaici sono chiamate “solar car”, quelle alimentate da generatori a benzina sono invece “auto ibride” [WIK16a].

Spesso si pensa all'auto elettrica come un'invenzione moderna. Contrariamente, la prima fu costruita prima dell'invenzione del motore a combustione, tra il 1832 e il

1839, da un imprenditore scozzese di nome Robert Anderson [GMH16]. Il suo prototipo utilizzava celle di energia non ricaricabili, in quanto la prima batteria ricaricabile adatta alla conservazione di quantità di energia elettrica per il moto di un'automobile, la batteria piombo-acido, fu inventata nel 1859 dal fisico francese Gaston Planté.

### **1.1.2 Efficienza dell'auto elettrica e vantaggi**

Diversamente dall'idea comune dovuta alla sua scarsa diffusione, l'efficienza e il rendimento del motore elettrico è maggiore rispetto a quella dei motori a combustione interna. I motori elettrici vantano un rendimento del 90%, per cui l'energia potenziale prodotta dalle batterie viene quasi completamente trasformata in energia meccanica essenziale per il moto, diversamente nei motori a benzina si ha un rendimento molto basso, tra il 25% e il 28%, mentre nei motori diesel il valore raggiunge il 40% , tenendo conto che nei motori a combustione interna la percentuale rimanente si disperde in calore. Inoltre questa tipologia di motore raggiunge il regime massimo di coppia in poco tempo mantenendolo costante e permette un'accelerazione e una decelerazione della velocità lineare grazie alla mancanza di frizione, per cui il motore risulta sempre collegato.

A livello di impatto ambientale i vantaggi sono notevoli: non viene prodotta nessuna emissione nociva, a differenza delle auto tradizionali che rilasciano sostanze altamente inquinanti. Non si può considerare ancora un mezzo totalmente ecologico per via del ciclo di produzione dell'energia elettrica per la ricarica delle batterie, la quale essendo una fonte di energia secondaria è spesso ricavata da combustibili fossili. Tuttavia, l'inquinamento prodotto da questo processo è decisamente minore rispetto a quello causato dai mezzi tradizionali. Per quanto riguarda la riduzione di CO<sub>2</sub> si hanno dei risultati significativi: l'auto elettrica rispetto al veicolo a propulsione termica produce fino al 46% di gas serra in meno [SCA16a].

Un altro problema a livello ecologico è lo smaltimento e il riciclaggio delle batterie al litio, le quali possono essere rigenerate fino all'80% della loro capacità iniziale seguendo un processo non esente da inquinamento.

Tra gli svantaggi rispetto alle auto tradizionali si ha l'elevato costo delle batterie. La durata del set si aggira sui 33000km per un costo di 10000€. A favore invece le minori spese di manutenzione, ad esempio il cambio dell'olio non necessario,

le parti meccaniche facilmente riparabili. Il guadagno si trova soprattutto nel basso costo dell'energia, si stima infatti una spesa di 1,50€ ogni 100km contro gli elevati costi della benzina e del diesel. L'auto elettrica è destinata principalmente ai brevi spostamenti quotidiani all'interno della città per via della sua bassa autonomia, la sua minore usura permette la triplicazione dei tempi di vita delle batterie.

Per quanto riguarda la sicurezza, i rischi legati all'incendio dopo uno scontro diminuiscono notevolmente, mentre si sostiene che aumenti il rischio per i pedoni e i ciclisti, che non sentirebbero l'arrivo dell'automobile per via della sua silenziosità [ALB13].

Attualmente le auto elettriche in Italia godono di altri vantaggi a livello urbano, come l'accesso in zone a traffico limitato (ZTL), agevolazioni sul bollo, minore costo dell'assicurazione (dal 30% al 50% sulla RC auto) e parcheggi gratuiti su strisce blu e strisce per i residenti [GAA16].

### 1.1.3 Benefici dell'auto elettrica

I benefici dell'auto elettrica possono essere classificati in tre categorie: ambientali, economici, energetici. L'obiettivo è sviluppare e diffondere la mobilità elettrica a livello culturale, sociale, economico, ambientale.

- *Benefici ambientali:* la diffusione dei veicoli elettrici svolge un ruolo importante per il nuovo concetto di mobilità sostenibile, che comporta evidenti benefici sia in termini di riduzione delle emissioni che di abbattimento di polveri sottili particolarmente nocive per la salute (PM10, PM2,5), con conseguenti miglioramenti a livello sociale ed economico. I vantaggi ambientali sono la premessa all'affermazione della mobilità elettrica, particolarmente se inserita in una più ampia visione del sistema energetico in un'ottica di:
  - *Uso razionale delle risorse:* come descritto nel paragrafo precedente, l'efficienza energetica dell'auto elettrica è di tre volte superiore rispetto a quella di un veicolo a combustione interna, con una vita media dieci volte superiore.
  - *Utilizzo di energia pulita:* l'adozione di veicoli elettrici per la mobilità nelle aree urbane migliorerebbe la qualità dell'aria, eliminando polveri sottili e composti tossici. Per risolvere il

problema della generazione di CO<sub>2</sub> durante la produzione di energia per la ricarica delle batterie bisognerebbe promuovere parallelamente l'utilizzo di fonti rinnovabili.

- *Stoccaggio di energia elettrica:* le batterie dei veicoli sono dei veri e propri accumulatori di energia, grazie a questo viene ovviato il problema dell'intermittenza delle fonti energetiche rinnovabili
- *Attuazione del modello delle smart grid:* una a volta a regime, entrerà in gioco il progetto delle smart grid che prevede la redistribuzione dell'energia on demand nei momenti di picco della richiesta, ottimizzando l'efficienza nella trasmissione. Questa soluzione è ancora del tutto teorica [ENE16]. Ad esempio, per avere gli stessi benefici ambientali ottenuti con la sostituzione di un veicolo Euro 3 con uno a batteria si devono sostituire:
  - 1.2 veicoli Euro 3 con veicoli ibridi
  - 3 veicoli Euro 3 con veicoli Euro 5
  - 1.2 – 2 veicoli Euro 3 con veicoli a celle di combustibile (Idrogeno)
- *Benefici economici:* per comprendere meglio i benefici economici possiamo confrontare il costo dell'energia di un'auto elettrica con il costo del carburante necessario per un altro tipo di macchina, ad esempio a gasolio, sulla base di una percorrenza di 15.000 km/anno:
  - Auto elettrica: (Costo energia ore notturne: 0.20 €/kwh)  
 $5.000 \text{ km/anno} \times 125 \text{ Wh/km} = 1875 \text{ kWh/anno}$   
pari a 375 € per percorrere 15.000 km in un anno
  - Auto a gasolio: (Costo gasolio: 1.5 €/l)  
 $15.000 \text{ km/anno} \times 0.06 \text{ litri/km} = 900 \text{ Litri /anno}$   
pari a 1350 € per percorrere 15.000 km in un anno.
- *Benefici energetici:* Se consideriamo tutti le fasi della produzione dell'energia elettrica fino alla resa del propulsore elettrico e assumendo che il gas naturale sarà probabilmente la fonte più utilizzata nei prossimi anni si può delineare un rendimento complessivo di [SCA16b]:
  - Veicoli a batteria: 35%
  - Veicoli ibridi plug-in: 28%

- Veicoli a celle di combustibile idrogeno: 24%

Se invece assumiamo che l'energia sia ricavata da fonti rinnovabili notiamo che il rendimento complessivo è sempre favorevole per la soluzione a batteria.

- Veicoli a batteria: 63%
- Veicoli a celle di combustibile idrogeno: 23%

### 1.1.4 Batterie dell'auto elettrica

Nel primo paragrafo abbiamo introdotto il concetto di batteria, inventata nella seconda metà del 1800, grazie alla quale è stata possibile la costruzione di auto elettriche ricaricabili. In questo paragrafo verranno introdotti i tipi di batteria moderni e le loro prestazioni.

Innanzitutto bisogna definire in cosa consiste un ciclo di carica e la capacità di scarica. Il ciclo di carica è definita come il tempo di vita di una batteria, corrisponde al numero di volte che la batteria viene scaricata completamente, fino allo zero. La batteria di un'auto elettrica non dovrebbe essere mai scaricata completamente perché compromette la durata nel tempo della batteria. La capacità di scarica, invece, è un importante indicatore della qualità di una batteria per auto. Più è alta la capacità di scarica, maggiore è la quantità di energia che una batteria può fornire, migliori sono le prestazioni del motore.

Le batterie per auto elettriche attualmente utilizzate sono costruite con tre tecnologie, ognuna con costi e prestazioni differenti [IDG16]:

- *Batterie al nickel-metal-idrato (NiMH)*: in via di estinzione, sono state le prime batterie utilizzate nelle automobili ibride di vecchia generazione, come alimentazione aggiuntiva al motore termico a benzina. Tra gli svantaggi troviamo la bassa autonomia che conferiscono all'automobile quando viaggia solo in elettrico: solo 3-4km.
- *Batterie piombo-gel (PbGel/Silicon)*: inizialmente solo al piombo, è la tecnologia più antica; è stata poi innovata con la tecnologia piombo-gel, contenenti gel di silicio, per migliorarne le prestazioni non renderla obsoleta. Questo tipo di batterie sono considerate sicure e affidabili, nei modelli più vecchi è necessario aggiungere acqua o acido periodicamente. La vita media delle batterie al piombo è di 300-400 cicli di scarica completi, tenendo conto che ogni ciclo di scarica corrisponde a circa 70km di

autonomia, si può approssimare la loro durata a 20km circa. Il costo per un'auto elettrica è di 400-500€. Per le batterie piombo-gel le prestazioni sono migliori.

- *Batterie al litio (Li)*: sono le batterie più moderne, stanno prendendo il posto delle batterie sopraelencate. Ne esistono di diverse versioni:
  - *Batterie al litio-polimero (LiPo)*: non utilizzate sui mezzi di trasporto perchè pericolose, in caso di urto possono incendiarsi ed esplodere. Vengono applicate nel mondo dell' Internet Of Things per alimentare i sensori di cui sono dotati gli oggetti (batterie “thin-film”).
  - *Batterie a ioni di litio (Li-Ion)*
  - *Batterie al litio-ferro-fosfato (LiFePO<sub>4</sub>)*: sono le migliori in circolazione, perdono solo il 5% della capacità di carica ogni anno.
  - *Batterie al litio-ferro-ittrio-fosfato (LiFeYPO<sub>4</sub>)*

### 1.1.5 Infrastrutture di ricarica

La mobilità elettrica è fortemente legata ai punti di ricarica: le batterie devono essere ricaricate periodicamente, solitamente tramite la rete elettrica. L'energia fornita alla rete può essere generata da diverse risorse come carbone, olio combustibile, gas naturale, energia idroelettrica, altre fonti rinnovabili o energia nucleare [WIK16a].

La ricarica è possibile anche mentre il veicolo è in movimento grazie al freno rigenerativo, un tipo di freno che recupera energia ricavandola da una parte di quell'energia che viene dissipata in calore durante il rallentamento del veicolo, oppure tramite celle fotovoltaiche [DTE11].

Le infrastrutture di ricarica possono essere di tipo domestico e di tipo pubblico. Questi punti di ricarica non sono altro che colonnine elettriche che permettono diversi tempi di ricarica in base alla potenza erogata (4 – 8 ore nel caso di ricarica lenta, 5 – 10 minuti nel caso di elevata potenza).

L'alimentazione da colonnina elettrica può avvenire in due modi:

- *Per via conduttiva*: si tratta di una presa di corrente più o meno normale, che attraverso un trasformatore e un raddrizzatore fornisce alla batteria l'energia necessaria alla ricarica. Il circuito di regolazione della carica risiede sulla colonnina. Molto efficiente, è presente un minor numero di componenti, il quale garantisce un minor costo dell'impianto.



- *Per via induttiva:* l'avvolgimento primario viene inserito in una fessura del veicolo in cui si accoppia con l'avvolgimento secondario. Non essendoci parti accessibili sotto tensione, si elimina il rischio di folgorazione. In questo caso il circuito di regolazione della carica risiede a bordo dell'automobile, con conseguente aumento dei costi e appesantimento del veicolo.

Una corretta programmazione della ricarica in ambiente domestico si ha quando si sfruttano le ore di bassa richiesta di energia da parte della rete, solitamente durante la notte, per disporre di energia a bassi costi e, in alcuni casi, tramite l'aiuto di fonti rinnovabili solari ed eoliche. La ricarica può essere fatta anche in aree pubbliche opportunamente attrezzate durante le soste diurne nel caso di carica lenta o mediamente rapida, mentre le infrastrutture per la ricarica ultra rapida saranno rese disponibili nelle stazioni di servizio.

Velocità di ricarica	kW	Connessione tipica	Percorrenza corrispondente a 10 minuti di ricarica	Tempo di ricarica completa
Lenta	≤ 3,7 kW	AC 230 V / 10-16 A	Meno di 2 km	Circa 8 ore
Semiveloce	3,7-22kW	AC 240 V / 80 A AC 400 V / 16-32 A	Fino a 15 km	Da 1 a 8 ore
Rapida	22-43 kW	AC 400 V / 32-63 A DC	Fino a 30 km	Da 30 min. a 1 ora
Ultrapida	≥ 43 kW	DC	Oltre 30 km	Fino a 30 min.

**Figura 1: Classificazione della potenza di ricarica in macrocategorie [SCA16c]**

I valori presentati nella tabella in Figura 1 sono indicativi, corrispondenti ad un consumo tipico di 125 – 180 Wh/km ed una capacità delle batterie di 30 kWh [SCA16c].

Secondo la normativa IEC 61851-1 sono ammessi quattro modi per la ricarica dei veicoli elettrici [SCA16d]:

- 1) *Ricarica in ambiente domestico, lenta (6 – 8 ore):* corrente massima 16A, ammessa solo negli ambienti privati, tramite presa domestica o presa industriale.
- 2) *Ricarica in ambiente domestico e pubblico, lenta (6 – 8 ore):* il dispositivo Control Box (sistema di sicurezza PWM) sul cavo di

alimentazione del veicolo garantisce la sicurezza durante la ricarica tramite prese domestiche o industriali 16A.

- 3) *Ricarica in ambiente domestico e pubblico, lenta (6 – 8 ore) o mediamente rapida (30 minuti – 1ora):* obbligatorio per gli ambienti pubblici, deve avvenire tramite uno specifico sistema di alimentazione dotato di connettori (sistema di sicurezza PWM), ricarica a 63A, 400V.
- 4) *Ricarica in ambiente pubblico, ultra rapida (5 – 10 minuti):* corrente continua fino a 200A – 400V. Il caricabatteria è esterno al veicolo.

Il modo 3 dispone di normative sui connettori di ricarica (IEC 62196-1 e 2), descritti nella seguente tabella (Figura 2).

Tipo 1	Tipo 2	Tipo 3a	Tipo 3c
 Presa tipo 1	 Presa tipo 2	 Presa tipo 3a	 Presa tipo 3c
<ul style="list-style-type: none"> <li>• Monofase 32A 250Vac</li> <li>• 2 contatti pilota</li> <li>• IPXXB</li> <li>• Connettore solo per lato veicolo</li> </ul>	<ul style="list-style-type: none"> <li>• Monofase 16A, 250Vac</li> <li>• Trifase 63A, 480Vac</li> <li>• 2 contatti pilota</li> <li>• IPXXB</li> <li>• Obbligo del sistema di ritenuta della spina nella presa</li> </ul>	<ul style="list-style-type: none"> <li>• Monofase 16A, 250Vac</li> <li>• 1 contatto pilota</li> <li>• IPXXD sulla presa</li> <li>• Sistema di ritenuta opzionale</li> </ul>	<ul style="list-style-type: none"> <li>• Monofase 16A,250Vac</li> <li>• trifase 63A, 480Vac</li> <li>• 2 contatti pilota</li> <li>• IPXXD su presa e spina</li> <li>• Sistema di ritenuta opzionale</li> </ul>

**Figura 2: Norme sui connettori di ricarica [WIK16c]**

Un'alternativa ai lunghi tempi di ricarica è quella di sostituire le batterie di accumulatori scarichi con altre già cariche. La sostituzione può avvenire nelle stazioni di servizio, rivendite di auto, grandi magazzini oppure parcheggi. Si tratta di batterie modulari adagiate in un doppio fondo sotto l'abitacolo, hanno una dimensione standard e prevedono maniglie, rotelline e un peso di 20 – 40kg. Il cambio avviene in pochi secondi, molto conveniente rispetto ai lunghi tempi di ricarica dell'automobile attraverso la corrente elettrica, l'operazione risulta però più costosa. Le batterie scariche vengono messe sotto carica in preparazione del prossimo cambio.

La presenza di una rete ben distribuita di colonnine di ricarica nel territorio e la possibilità di sostituire le batterie permetterebbero la riduzione della “range anxiety”, cioè la paura degli utenti di auto elettriche di non aver autonomia sufficiente per raggiungere destinazione e quindi rimanere bloccati durante il percorso. [SCH09]

## 1.2 Eco-driving

### 1.2.1 Concetto di Eco-driving

L'Ecodriving (Guida Ecologica) rappresenta una nuova cultura nello stile di guida che presta attenzione agli aspetti ecologici, economici e alla sicurezza. La guida ecologica propone un migliore uso delle avanzate tecnologie dei veicoli, migliorando allo stesso tempo la sicurezza sulla strada e contribuendo alla protezione dell'ambiente e alla riduzione dell'inquinamento. In sostanza il termine Ecodriving descrive un utilizzo efficiente dell'energia dei veicoli in modo da ridurre il consumo di carburante rispetto al normale consumo sulla stessa tratta.

Negli ultimi decenni le tecnologie e le macchine sono migliorate in poco tempo, mentre gli utenti hanno fatto fatica ad adattare il loro stile di guida, per questo tramite la Guida Ecologica si presenta una nuova cultura di guida basata sui nuovi motori, che ne sfrutti le tecnologie in maniera ottimale.

I benefici coprono diversi aspetti:

- *Sicurezza*: migliora la sicurezza sulla strada, aumentando l'abilità alla guida
- *Ambientale*: riduzione dell'emissione dei gas serra, diminuzione degli agenti inquinanti nell'aria, riduzione del rumore
- *Finanziaria*: risparmio benzina-denaro (5-15% a lungo termine), diminuzione dei costi di manutenzione del veicolo, riduzione dei costi legati ad incidenti
- *Sociale*: adattamento di una guida più responsabile, meno stress quando si guida, una guida più comoda per l'autista e i passeggeri

Inoltre, l'adozione di questo stile di guida permetterebbe agli utenti delle auto elettriche di guidare in modo più rilassato, la consapevolezza di un utilizzo intelligente delle risorse ridurrebbe il livello di “range anxiety”. [ECD16a]

### 1.2.2 Regole d'oro dell'Eco-Driving

Un buon punto iniziale per adottare una Guida Ecologia è porre attenzione alle regole presentate in questo paragrafo, il cui scopo è di ridurre le emissioni di CO<sub>2</sub>, agenti inquinanti dell'ambiente e utilizzare meno carburante. Il consumo del carburante aumenta con la guida a giri elevati del motore (high RPM), così come con l'accelerazione e frenata non controllate. Le seguenti regole sono state definite dal

consorzio ECOWILL con la partecipazione di collaboratori esterni nel 2011, seguendo le linee guida “Safety First”.

- **Anticipare il flusso del traffico:**

- *Osservare la strada il più avanti possibile e anticipare il flusso del traffico*
- *Agire invece di reagire - aumentare il raggio d'azione con una distanza adeguata per utilizzare lo slancio:* l'aumento sistematico della distanza tra un veicolo e l'altro aumenta la sicurezza stradale. Una distanza di sicurezza maggiore equivalente a circa 3 secondi dall'auto di fronte ottimizza le opzioni per bilanciare le fluttuazioni di velocità del flusso di traffico e che consente una guida costante a velocità costante, riducendo situazioni rischiose. Si tratta togliere il piede dall'acceleratore invece che frenare e disperdere l'energia cinetica accumulata, riducendo la necessità di accelerare troppo dopo una grande decelerazione.
- *Sfruttare la massimo lo slancio del veicolo:* consiste nell'utilizzo di energia cinetica del veicolo in modo più efficiente. L'obiettivo è lasciare andare la macchina a velocità costante invece di frenare e accelerare. Si divide in tre diverse tecniche di guida classificate all'interno di due categorie:

- (i) **utilizzo dello slancio con marcia inserita:**

- Tecnica di guida #1: Con marcia inserita, la velocità del veicolo si riduce per effetto frenante del motore per attrito meccanico. Questa decelerazione può diminuire inserendo la marcia giusta, la quale provoca un'accelerazione. Questa tecnica è utile per il risparmio di carburante, anche durante la guida a velocità superiori.

- (ii) **utilizzo dello slancio in folle:**

- Tecnica di guida #2: Utilizzare lo slancio dell'auto in folle consente un utilizzo più efficiente dell'energia cinetica del veicolo in quanto non c'è nessun effetto freno motore. Questa tecnica è utile nelle situazioni di avvicinamento a un ostacolo o un arresto identificato in precedenza, ad esempio un semaforo. Una distanza relativamente lunga può essere percorsa a velocità costante senza accelerazioni aggiuntive, con

consumo di carburante minimo. Per ragioni di sicurezza è necessario tenere la marcia inserita durante la guida in pendenza.

- Tecnica di guida #3: Sfruttare lo slancio dell'auto con marcia inserita e frizione innestata. Questa tecnica è consigliabile in situazioni in cui si presume che la corsa possa essere continuata presto nella stessa marcia, e l'effetto frenante del motore non è utile per una buona efficienza nel consumo del carburante.
- **Mantenere una velocità costante a basso RPM:**
  - *Guidare dolcemente a basso numero di giri con la marcia più alta possibile*: una guida fluida permette di risparmiare un sacco di carburante rispetto a una guida con frenate brusche o accelerazioni inutili, causanti stress da guida nell'utente e un aumento del rischio sulla sicurezza stradale. L'utilizzo di un cruise control, sistema elettronico che permette la regolazione automatica della velocità di un veicolo, è consigliabile per mantenere una guida fluida soprattutto in aree urbane ma anche in situazioni di traffico in città.
  - *Ricordare che la guida ad alta velocità o con elevato numero di giri aumenta in modo significativo il consumo di carburante*: durante la guida in città non vi è quasi alcuna possibilità di aumentare la velocità media o di guadagnare tempo con accelerazioni inadeguate e alti picchi di velocità. Anche sulle autostrade il tempo risparmiato è piuttosto basso e deve essere pagato un consumo maggiore di carburante. Aumentare la velocità significa aumentare il numero di giri del motore, di conseguenza aumenta anche la resistenza aerodinamica, causando uno spreco di combustibile e di denaro. Questo avviene anche rispettando i limiti di velocità (ad esempio 110-120 km orari).
- **Cambiare marcia in anticipo:**
  - *Passare alla marcia superiore a circa 2.000 giri*
  - *Considerare la situazione del traffico, le esigenze di sicurezza e le specifiche dei veicoli*: guidare con alti o medi RPM consuma sempre più carburante rispetto a una guida a bassi RPM a qualsiasi velocità. Pertanto, cambiare marcia anticipatamente è consigliabile, tenendo conto delle specifiche del veicolo e del traffico.

Linee guida per cambiare marcia:

- 1<sup>a</sup> marcia: accensione e guida per al massimo 5 metri.
- 2<sup>a</sup> marcia: 20kph
- 3<sup>a</sup> marcia: 30kph
- 4<sup>a</sup> marcia: 40kph
- 5<sup>a</sup> marcia: 50kph
- 6<sup>a</sup> marcia: 60+ kph

Durante un percorso in salita scegliere una marcia che non necessiti la pressione completa dell'acceleratore. Saltare delle marce durante un'accelerazione veloce può aiutare a salvare carburante [ECD16b].

- **Controllare la pressione degli pneumatici spesso, almeno una volta al mese, prima di guidare ad alta velocità:**
  - Mantenere gli pneumatici correttamente gonfi perchè la bassa pressione negli pneumatici è un rischio per la sicurezza e spreca carburante
- **Considerare che qualsiasi energia supplementaria costa carburante e denaro:**
  - Utilizzare aria condizionata e apparecchiature elettriche saggiamente e spegnerle se non necessarie.
  - L'energia elettrica viene prodotta da carburante extra bruciato in un motore a combustione, quindi anche il materiale elettrico è complice del consumo.
  - Evitare peso morto e resistenza aerodinamica.

### 1.2.3 Regole d'argento dell'Eco-Driving

Oltre alle regole d'oro che regolano l'utilizzo ottimale di una Guida Ecologica, è necessario tenere conto di altri aspetti per migliorare i benefici ambientali e finanziari [ECD16c].

- *Il risparmio di combustibile inizia con la scelta di un'auto a basse emissioni:* possono essere automobili elettriche come auto a carburante con basse emissioni di CO<sub>2</sub>. I veicoli a diesel devono essere equipaggiati con filtri speciali.
- *Evitare percorsi in macchina corti:* a freddo, i motori necessitano di più combustibile e non riescono a raggiungere la temperatura operativa ottimale, aumentando l'usura ed emettendo CO<sub>2</sub>.
- *Partire immediatamente dopo aver azionato il motore, non riscaldarlo facendolo girare al minimo*
- *Spegnere il motore quando la fermata dura più di 20 secondi*
- *Chiudere i finestrini quando si guida ad alte velocità per diminuire l'attrito aerodinamico e così anche il consumo di carburante*
- *Utilizzare oli a basso attrito e pneumatici a basso consumo energetico (etichettatura UE)*
- *Controllare l'auto regolarmente e portarla in manutenzione per mantenerla ecologica e anche a norma di sicurezza*
- *Prendere in considerazione mezzi di trasporto alternativi*

## 1.3 Applicazioni a supporto della mobilità elettrica

Negli ultimi anni, con la diffusione degli smartphone dei tablet, sono state implementate svariate applicazioni a supporto della mobilità in modo da facilitare gli spostamenti e diminuire il livello generale di ansia da guida. Una parte di queste applicazioni vengono in aiuto alla mobilità elettrica e alla guida ecologica, come gli indicatori di colonnine di ricarica vicine e servizi di monitoraggio dei consumi. In questo paragrafo andremo a classificare le varie applicazioni a disposizione degli utenti e ne daremo esempi.

### 1.3.1 Classificazione delle applicazioni a supporto della mobilità

Le applicazioni a supporto della mobilità hanno la funzione di semplificare gli spostamenti agli utenti di veicoli all'interno e fuori dai centri urbani. Queste applicazioni possono riguardare diverse tipologie:

- *Applicazioni di supporto al route planning:* Sono applicazioni che fanno uso di algoritmi per la ricerca di cammini minimi in un grafo, quali l'algoritmo di Dijkstra, Priority Queues, Bidirectional Search. Il loro compito infatti è trovare tutti i percorsi disponibili per arrivare da partenza a destinazione, in particolare viene indicato percorso ottimale in termini di tempo e chilometri e vengono offerte agli utenti informazioni stradali sul percorso. In quelle più approfondite vengono visualizzati anche il costo del carburante per i chilometri previsti, vengono fornite informazioni sulla località e permettono di memorizzare i viaggi effettuati. Oltre alle applicazioni per smartphone sono disponibili siti web per il route planning. Parlando di mobilità elettrica, le applicazioni di route planning permettono agli utenti di calcolare una stima della durata della batteria in base ai chilometri da percorrere, in genere si parla di un'autonomia di 150km. Anche questo è un metodo a favore della diminuzione di “range anxiety” a carico degli utenti di veicoli elettrici.
- *Applicazioni di visualizzazione parcheggi:* Sono applicazioni studiate per automobilisti che lavorano o effettuano spostamenti nelle grandi città, si dividono in base alla loro funzionalità in
  - *Ricerca parcheggi:* permettono la visualizzazione di tutti i parcheggi disponibili nella località inserita o attorno alla propria posizione, segnalando le indicazioni per arrivarvi, il numero di posti liberi, consultare gli orari di apertura e chiusura e le modalità e i prezzi di pagamento. Ad esempio, l'applicazione per Android “Parkopedia”.
  - *Segnalazione parcheggi:* rappresentano un vero e proprio social network per l'indicazione in tempo reale di parcheggi liberi indicati dagli utenti. Un chiaro esempio di questa applicazione è Social Parking.
  - *Timer per parcheggi:* sono applicazioni la cui funzionalità è ricordare quando il parchimetro sta per scadere. Ad esempio, l'applicazione Parktimer.



- *Applicazioni per trovare l'auto parcheggiata*: aiutano l'utente a ritrovare la propria auto dopo averla parcheggiata in un luogo o una località insolita. La loro funzione è basata sull'inserimento di fotografie del parcheggio, l'inserimento della via o, nei casi in cui sia presente un rilevatore GPS all'interno dell'auto, viene geolocalizzata la posizione.
- *Applicazioni di controllo velocità*: sono applicazioni che facilitano la localizzazione di autovelox o tratti autostradali controllati dal sistema “Tutor”, tenendo traccia della velocità media, suggerendo la massima velocità raggiungibile e avvertendo quando la velocità si fa troppo alta. Per esempio, DroidTutor, Autovelox!, iTutor.
- *Applicazioni salva traffico*: consentono di evitare ingorghi di traffico tramite la segnalazione di code o rallentamenti da parte degli utenti registrati, segnalando posti di blocco e autovelox. Ad esempio Waze e Traffico Autostrade Italiane.
- *Applicazioni sul costo del carburante*: sono applicazioni che indicano il prezzo giornaliero del carburante, comparano i prezzi dei distributori più vicini alla propria posizione e segnalano quello più economico. Esempio: PrezziBenzina.
- *Applicazioni per la gestione delle multe*: sono applicazioni che permettono di conoscere le scadenze riguardanti le multe ed eventuali informazioni sulla contravvenzione. iMulte è un perfetto esempio di queste applicazioni, la quale permette, grazie a un team di avvocati specializzati, di fare ricorso alla multa e fornire aiuto nell'analisi del verbale [ASS13].
- *Applicazioni per misura delle prestazioni dell'auto*: si tratta di applicazioni per misurare accelerazione, cavalli, forza di gravità sui lati e sui freni, inserendoli e memorizzandoli in una tabella (Dinolicious Classic) [COR13].
- *Applicazioni per la memorizzazione dei dati dei veicoli e della patente*: queste applicazioni aiutano l'utente a tenere sotto controllo il saldo punti patente e validità, dati relativi al veicolo, lo stato delle pratiche in corso presso la motorizzazione (iPatente). Altri servizi offerti sono la funzione di libretto di circolazione, la funzione di agenda sui controlli meccanici del veicolo (#OnTheRoad) e di gestione di assicurazione e bolli (Veicoli - Gestione auto e moto) [ASS13].

- *Applicazioni di supporto alla guida ecologica*: permettono di comprendere qual è l'impatto ambientale della propria guida, inserendo dati relativi all'auto e alle condizioni metereologiche (GreenMeter) [GRE16].

### 1.3.2 Esempi di applicazioni per la mobilità elettrica

Secondo uno studio condotto dal dipartimento dell'energia degli Stati Uniti, quasi il 70% di chi non possiede un veicolo elettrico non lo acquisterebbe a causa della “driver range anxiety”, il timore che il veicolo rimanga senza autonomia e opportunità di ricarica. Le applicazioni a supporto della mobilità elettrica vengono in aiuto per diminuire questo tipo di ansia e, di conseguenza, incentivare l'acquisto di veicoli elettrici. Riferendoci al paragrafo 1.3.1, possiamo stilare una classificazione più accurata riguardante le applicazioni per veicoli elettrici:

- *Route planning*: nonostante la mobilità elettrica sia una valida soluzione ecologica per la riduzione di emissioni nocive, è scarsamente considerata per i problemi ad essa legati, come la limitata autonomia, la scarsa densità delle stazioni di rifornimento e la lunga durata delle ricariche. La scarsa conoscenza della durata della batteria o delle stazioni di rifornimento genera ansia da guida e porta a modificare il percorso in base alle stazioni di ricarica. Per questo motivo le applicazioni di route planning per la mobilità elettrica vengono in aiuto agli utenti nel trovare le stazioni di ricarica vicine alla zona in cui ci si trova o raggiungerne un'altra durante il percorso e a monitorare lo stato della batteria. In alcuni casi si ha l'opportunità di aggiungere distributori e di andare a migliorare la copertura.
- *Eco-driving*: le limitazioni sull'autonomia e la ricarica dei veicoli elettrici hanno favorito lo sviluppo di un nuovo modello di guida chiamato Ecodriving. Questo sistema è in grado di massimizzare l'energia e l'efficienza di un veicolo elettrico attraverso lo stile di guida, non solo permette di minimizzare il consumo di energia, ma anche di recuperare più energia cinetica possibile per inerzia e frenata. Inoltre, i concetti di veicolo elettrico e guida ecologica devono tener conto dell'ottimizzazione dei modelli di mobilità, al fine di impedire ai conducenti di imboccare strade altamente congestionate e garantire se necessario punti di ricarica lungo il percorso. Per prendere conto dello scenario completo, sono necessari nuovi

algoritmi che considerino variabili di input provenienti da diversi sensori. In questo contesto, i dispositivi mobile (es. smartphones, tablets) sono piattaforme promettenti per la raccolta dei dati e per ottenere informazioni in tempo reale interfacciandosi con il veicolo elettrico. Questo approccio permette di ridurre il costo degli equipaggiamenti integrati e offre la fusione dei dati tra quelli provenienti dal veicolo e quelli percepiti dai sensori dello smartphone, includendo la capacità di connessione alla rete e i vantaggi della navigazione in Internet. Il termine per descrivere questo tipo di approccio è “bring your own device (BYOD)”. Esempi di applicazioni sviluppate e algoritmi [FCR13]:

- Araujo afferma che lo stile di guida ha impatti sull'efficienza dell'energia di almeno il 20%. Con il suo gruppo sviluppò un'applicazione mobile per aiutare i piloti a ridurre il consumo di carburante tramite l'utilizzo dei sensori presenti nello smartphone e lo stato del veicolo per suggerire nuovi comportamenti di guida agli utenti.
- Kamal propose un meccanismo di controllo della velocità che considera la pendenza della strada come input del sistema e anticipa l'accelerazione e la decelerazione in modo di ridurre il consumo di energia. Questo sistema richiede di conoscere in anticipo il percorso per ottenere la pendenza della strada e calcolare i controlli ottimali per il veicolo. Il risultato di questa applicazione era un risparmio del 9% del carburante rispetto ad altri algoritmi di controllo della velocità che non consideravano la pendenza.
- Barth propose un sistema Eco-driving dinamico che considera il traffico e le informazioni della strada in tempo reale.
- *Applicazioni per prenotazione di ricarica*: monitorano in tempo reale il livello di carica e lo stato della batteria e visualizza le stazioni di servizio lungo il percorso, permettendo di visualizzare se sono libere o occupate ed, eventualmente, prenotarle in modo sicuro. L'autenticazione presso la colonnina di ricarica può avvenire tramite la tecnologia “Near Field Communication (NFC)”. Questo sistema può contribuire a diminuire il livello di “range-anxiety” e aumentare l'interesse per i veicoli elettrici.

- *Siti web con mappe sulle stazioni di ricarica*: si tratta per lo più di mappe con interfaccia basata sul Web per la localizzazione delle stazioni di ricarica all'interno del territorio o tramite l'inserimento della città di interesse e un raggio di chilometri. Si può visualizzare il numero di colonnine elettriche e se sono disponibili, occupate, in manutenzione o pianificate. Inoltre quest'ultime possono essere filtrate per tipo di presa. Due tipici esempi di siti web per la ricerca di stazioni di ricarica nel territorio italiano sono:
  - **Enel Drive**: permette di inserire una località e visualizzare tutte le colonnine elettriche disponibili, occupate, in manutenzione o pianificate. Inoltre è possibile filtrare la ricerca per il tipo di presa. In Figura 3 possiamo vedere un esempio di ricerca di colonnine elettriche nella città di Cesena, filtrate per presa di tipo 3 [ENE16].

The screenshot shows the Enel Drive website interface. At the top, there's a navigation bar with 'Enel Drive / Home'. The main area features a map of Cesena with a red pin and a search radius. To the right, a 'Dashboard' section displays three statistics: 215.611 ricariche effettuate, 1.263.655 kWh erogati, and 1.308.255 Kg risparmiati. Below the dashboard is a search bar with 'cesena' entered and a 'cerca' button. A distance filter is set to 100m. The 'Filtra per presa' section shows 'Type 3a' selected among other options. At the bottom, a table lists search results with columns for Stato, Accessibilità, Indirizzo, Distanza, Asset Provider, and Serial number.

Stato	Accessibilità	Indirizzo	Distanza	Asset Provider	Serial number
Disponibile	Libera	CESENA - Via Jemolo, snc	826m	Enel Distribuzione	13EP21T2301H000032
Disponibile	Libera	CESENA - Viale Mazzoni, snc	501m	Enel Distribuzione	13EP21T2301H000028

**Figura 3: Esempio di ricerca nel sito EnelDrive [ENE16]**

- **Colonnine Elettriche:** a differenza di EnelDrive, permette anche l'inserimento di nuove colonnine per il miglioramento della mappa. In Figura 4 è possibile vedere l'Home page del sito web [COL16].

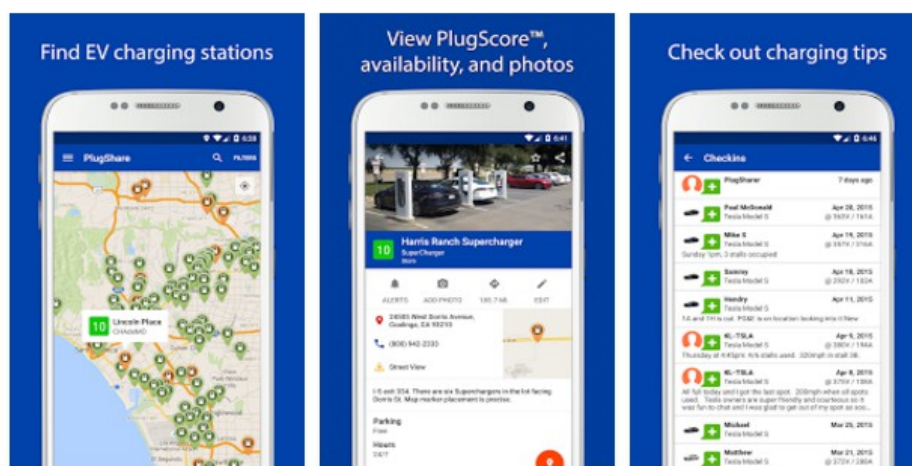


**Figura 4: Home page del sito web ColonnineElettriche [COL16]**

Presenteremo ora una lista di applicazioni già esistenti riguardanti le tematiche sopraelencate, definendone le caratteristiche principali e i servizi offerti.

- **PlugShare:** PlugShare offre una mappa mondiale contenente più di 50,000 stazioni di ricarica pubbliche e che raccoglie informazioni dalle maggiori reti Europee (RWE, Clever, Endesa, Enel) e del Nord America (Tesla SuperCharger, ChargePoint, Blink, SemaCharge, GE WattStation, Aerovironment, eVgo) ed è la comunità più grande di utenti di veicoli elettrici nel mondo. Gli utenti hanno contribuito fornendo oltre 100,000 recensioni sulle stazioni di servizio e 30,000 foto per rendere l'utilizzo di veicoli elettrici più semplice. PlugShare è il localizzatore ufficiale per le colonnine elettriche per il No Charge to Charge Program(sm) per Nissan LEAF e l'applicazione per smartphone MyFord Mobile. Tramite questa applicazione è possibile [PLS16]:
  - Localizzare le stazioni di ricarica pubbliche per il proprio veicolo elettrico

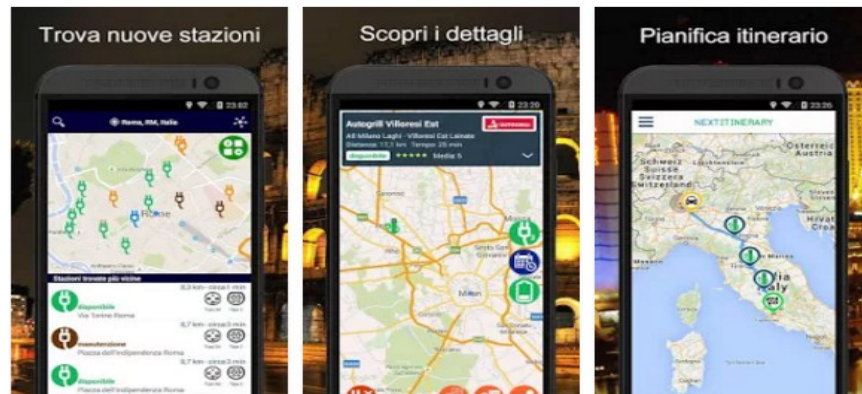
- Visualizzare le valutazioni su PlugScore™, tecnologia integrata nell'applicazione per visualizzare recensioni, la disponibilità in tempo reale, foto e descrizioni delle stazioni di ricarica.
  - Consultare recensioni e consigli dagli altri utenti
  - Filtrare i caricabatterie con quelli compatibili col proprio veicolo elettrico
  - Inserire nuove stazioni di ricarica
  - Coordinarsi con gli altri quando c'è la necessità di ricaricare
- Figura 5 mostra l'interfaccia grafica dell'applicazione con esempi delle sue funzionalità.



*Figura 5: Interfaccia utente di PlugShare [PLS16]*

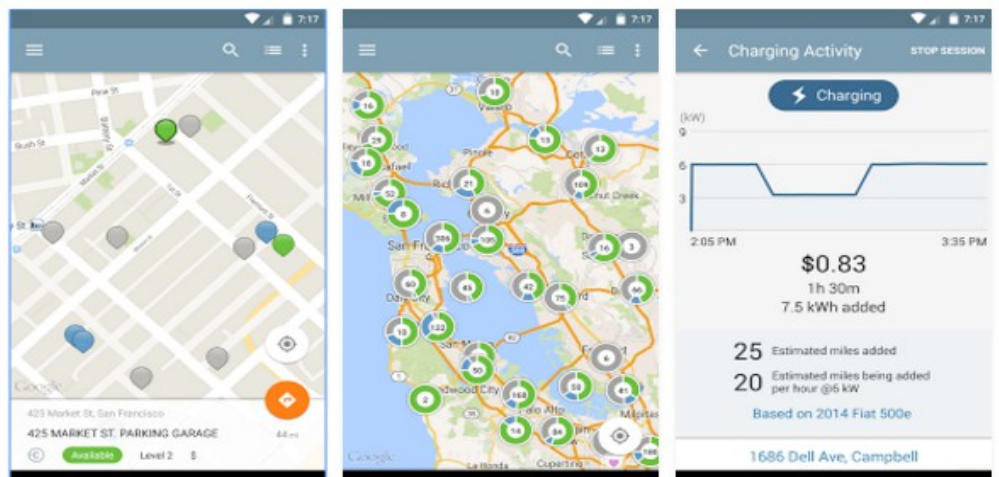
- **Next Charge:** Tramite NextCharge si possono visualizzare le stazioni di ricarica su mappa, la distanza, lo stato, il tempo di viaggio che separa la stazione dalla propria posizione corrente e avviare il navigatore per raggiungere la meta. Inoltre è possibile prenotare la ricarica in modo sicuro in zone con parcheggi altamente frequentati e pagando al momento della ricarica. Le principali funzionalità sono [NEC16]:
  - Ricerca colonnine elettriche
  - Stato disponibilità in tempo reale
  - Distanza e tempo di percorrenza per raggiungere la colonnina
  - Prenotazione e ricarica da App

Figura 6 mostra l'interfaccia grafica dell'applicazione con esempi di ricerca.



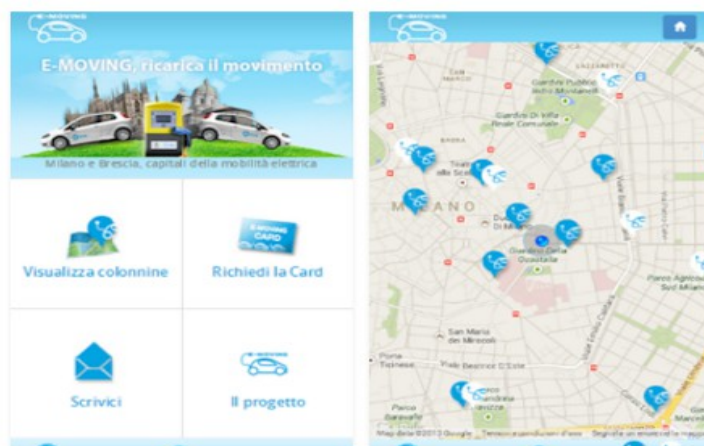
**Figura 6: Interfaccia grafica NextCharge [NEC16]**

- ChargePoint:** ChargePoint offre 33.000 posti di carica da tutte le reti principali e dati in tempo reale. Come le altre applicazioni visualizza le stazioni disponibili nei pressi della destinazione e ne fornisce le informazioni principali. Vista da satellite e navigatore integrati. Fornisce notifiche in tempo reale sullo stato di carica. In Figura 7 si può vedere l'interfaccia utente dell'applicazione con esempi sull'attività di ricarica e di ricerca [CP00].



**Figura 7: Interfaccia utente ChargePoint [CP00]**

- **E-moving:** Applicazione di visualizza delle colonnine di ricarica di Milano e Brescia. Tramite sottoscrizione ad un abbonamento tramite una apposita scheda elettronica, consente di effettuare un numero illimitato di ricariche per 3 mesi presso le colonnine di A2A. In Figura 8 si possono notare le funzionalità dell'applicazione attraverso l'interfaccia grafica [A2A16].

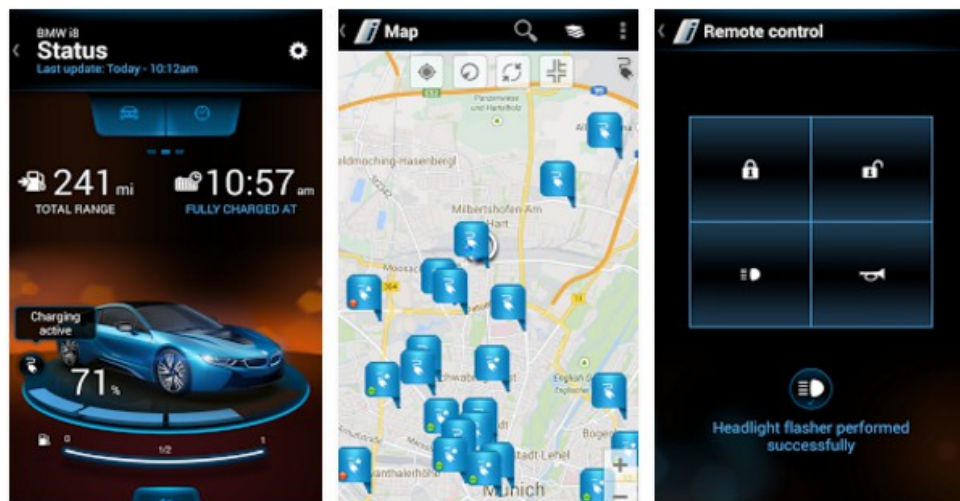


**Figura 8: Interfaccia utente E-moving [ASS13]**

- 
- **Bmw I Remote:** BMW iRemote rappresenta l'estensione fisica dei veicoli elettrici BMW i3/ i8 sullo smartphone. Grazie alle sue funzioni avanzate conseguente al guidatore di sfruttare in modo efficiente la propria auto elettrica BMW anche all'esterno del veicolo. Grazie ai BMW TeleServices il punto vendita BMW è sempre informato sulla eventuale necessità di assistenza. Le funzionalità supportate da questa applicazione sono: [BMW16]
  - Informazioni sullo stato della batteria e sull'autonomia residua del veicolo
  - Indicazione tramite mappa dinamica del fatto che la meta si trovi o meno nel raggio di autonomia
  - Indicazione della posizione del guidatore e del veicolo direttamente sullo smartphone
  - Sfruttamento del routing intermodale (coincidenze con treni e trasporti pubblici locali) e comoda navigazione fino alla meta
  - Impostazione degli orari di partenza per ottimizzare l'uso della batteria



- Visualizzazione di diverse colonnine di ricarica
  - Uso dei Remote Services (blocco e sblocco delle portiere, climatizzazione anticipata dell'interno dell'abitacolo e altre funzioni a distanza)
  - Indicazione di informazioni sul veicolo (chilometraggio complessivo, stato del veicolo, necessità di assistenza)
  - Semplice navigazione dalla posizione attuale fino al veicolo (Vehicle Finder)
  - Calcolo dell'efficienza del proprio stile di guida sulla base dei risultati degli ultimi tragitti, del contatore di CO2 e di altre statistiche di guida
  - Confronto di questi valori all'interno della community
  - Ottimizzazione dello stile di guida tramite tutorial integrati
- In Figura 8 si può vedere un esempio di alcune funzionalità dell'applicazione attraverso l'interfaccia grafica.



**Figura 9: Applicazione BMW i Remote [BMW16]**

- **MyFord Mobile:** Nata per il modello Focus Electric e sviluppata in collaborazione con Microsoft, questa applicazione permette di sfruttare al meglio la propria auto elettrica con funzioni come MapQuest che contiene un vasto database di punti di ricarica e permette di pianificare percorsi intelligenti. MyFord Mobile è stata presentata all'International Consumer Electronics Show di Las Vegas. Tramite l'applicazione gli utenti potranno

controllare l'auto anche a distanza e usufruire di alcuni servizi come monitorare il livello di carica, pianificare viaggi singoli o multibli, localizzare stazioni di ricarica e preriscaldare la vettura. Sarà possibile fissare degli obiettivi riguardo sul risparmio del pieno di elettricità e sull'emissione di CO2 non emessa [ELE16b]. In Figura 9 un esempio delle funzionalità dell'applicazione.

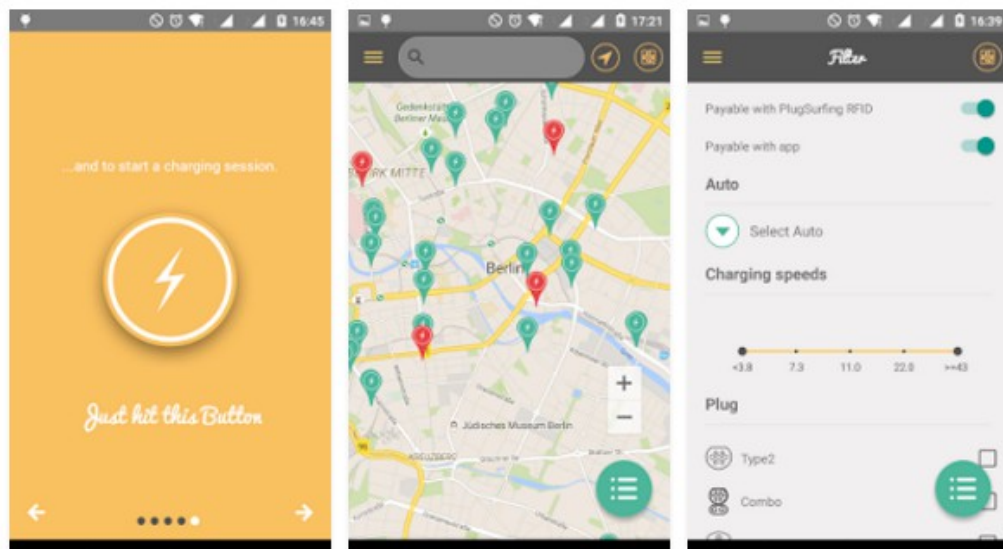


*Figura 10: myFord Mobile [FOR16]*

- **PlugSurfing:** PlugSurfing è stata classificata come il database di punti di ricarica più accurato dell'Europa e fornisce lo stato di attività delle stazioni di ricarica in tempo reale. È possibile vedere quale punto di ricarica è in funzione, informazioni a riguardo e aggiungere commenti. Più importante è la possibilità di trovare i prezzi della ricarica presso i vari punti di ricarica. Caricare la propria auto diventa più semplice grazie al pagamento attraverso l'applicazione che ha preso il posto delle carte di ricarica prepagate RFID. Il pagamento con PlugSurfing è attualmente disponibile in Germania, Paesi Bassi, Austria, Belgio e Lussemburgo e presto il servizio sarà esteso a tutta l'Europa. L'applicazione dispone delle seguenti funzionalità [PLS16]:
  - Trovare le stazioni disponibili attorno alla propria posizione o in un'area specifica

- Filtrare le stazioni secondo la velocità di carica o le specifiche della propria auto
- Ottenere i prezzi di ricarica per ogni stazione
- Aggiungere stazioni di ricarica ai preferiti
- Dare il via alla sessione di carica tramite l'applicazione
- Pagare attraverso l'applicazione tramite carta di credito o prepagata
- Tracciare l'attività dell'utente
- Fatturato mensile dei costi di carica
- L'applicazione necessita delle seguenti autorizzazioni:
- Località precisa (GPS) e approssimata (basata sulla rete): localizzazione tramite mappa
- Telefonata diretta al servizio di supporto tecnico
- Modificare o cancellare i contenuti della memoria USB: inserimento di foto delle stazioni di ricarica
- Scattare foto e video delle stazioni di ricarica
- Ricevere dati da Internet, visualizzare le connessioni di rete, cambiare la modalità di connessione alla rete

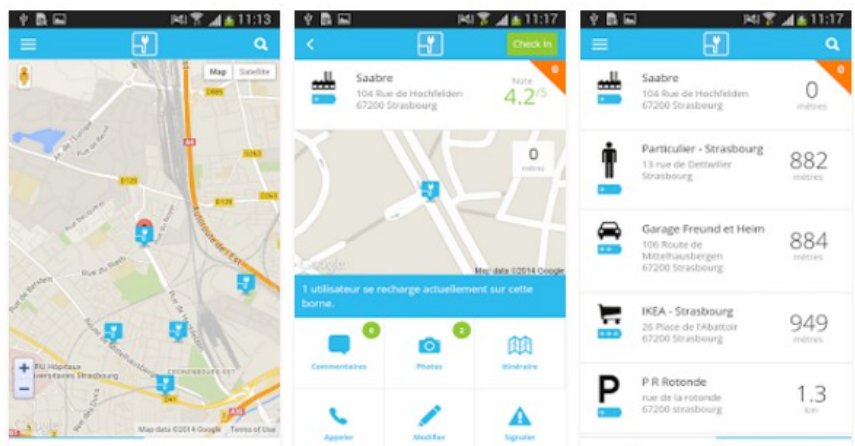
In Figura 10 possiamo vedere alcune funzionalità dell'applicazione.



**Figura 11: PlugSurfing [PLS16]**

- **ChargeMap:** Permette l'accesso a più di 55.000 prese per ricaricare la propria auto elettrica in tutta Europa fornendo informazioni sui tipi di prese, gli orari di accesso e i mezzi di accesso. Le principali funzionalità di ChargeMap sono [SAA16]:
  - Trovare i punti di ricarica che si trovano vicino a un determinato luogo o lungo il proprio tragitto
  - Controllare la disponibilità di un punto di ricarica in tempo reale
  - Lasciare commenti, pubblicare foto e effettuare il “check-in”
  - Condividere nuovi punti di ricarica con la comunità
  - Salire nella classifica degli utenti più attivi

In Figura 11 vediamo l'interfaccia utente dell'applicazione insieme ad alcune funzionalità.



*Figura 12: ChargeMap [SAA16]*

## 2 Tecnologie di progetto

---

In questo capitolo verranno introdotte le specifiche di progetto di tesi ed illustrate le tecnologie utilizzate per produrre una web app per il controllo della propria automobile elettrica.

### 2.1 Smartphone e Tablet

La diffusione di smartphone e tablet negli ultimi anni ha influenzato il mondo della tecnologia, rendendo applicazioni che prima erano disponibili solo sul browser o come applicativi del proprio computer sempre a portata di mano. Smartphone e tablet vantano di un processore molto evoluto, il quale si avvicina alle potenzialità di un PC, e dispongono di un sistema operativo ad hoc, come esempi: iOS, Android, Windows Phone. Le loro funzionalità, oltre a quelle dei telefoni normali, sono il supporto di file multimediali ad alta definizione (foto, video, audio), la lettura di documenti di diversi formati, connessione al web attraverso la rete dati o WiFi, uno schermo touch screen con un'interfaccia utente adeguata e la possibilità di poter personalizzare o aggiungere funzionalità tramite applicazioni scaricate dai rispettivi market di vendita.

Il punto di forza degli smartphone e dei tablet è la portabilità, cioè la possibilità di portarli ovunque si vada grazie alla loro leggerezza e al poco ingombro. Le funzionalità dell'applicazione sono state pensate per poter controllare la propria automobile soprattutto quando si è fuori casa, per questo è stata realizzata esclusivamente per dispositivi mobili.

Oggigiorno esistono diverse metodologie per creare un'applicazione per dispositivi mobili:

- *Responsive web design*: utilizzabile per rendere un'applicazione web o un normale sito web ridimensionabili per piccoli schermi e adattati per l'utilizzo del touch-screen.
- *Applicazioni native o applicazioni mobile*: vengono eseguite direttamente sul dispositivo come una applicazione software convenzionale.
- *Applicazioni ibride*: racchiudono un sito web all'interno di un'applicazione nativa, grazie all'utilizzo di un framework come Apache Cordova. Questa tecnologia permette lo sviluppo tramite l'utilizzo di tecnologie web pur mantenendo i vantaggi delle applicazioni native, come ad esempio l'operatività offline e l'accesso hardware.

Per lo sviluppo della nostra applicazione abbiamo scelto un approccio di sviluppo di una Web App come applicazione ibrida. Nei prossimi paragrafi offriremo una visuale dei concetti di Web App e applicazione ibrida come una sua evoluzione [WIK16d].

## 2.2 Web App

In informatica si definisce applicazione web (Web Application) un software applicativo client-server in cui il client è eseguito in un browser web. La distinzione tra una Web app e un sito web interattivo non è ben chiara. I siti web possono solitamente essere intesi come Web app quando hanno funzionalità simili a quelle di un'applicazione desktop o mobile. HTML5 ha introdotto un linguaggio esplicito per creare applicazioni che possono essere caricate come pagine web ma possono memorizzare dati localmente e funzionare anche offline.

Le applicazioni web sono in genere composte da una sola pagina, in quanto rifiutano il tipico paradigma della programmazione web di muoversi tra pagine distinte attraverso URLs differenti. Una web app può presentarsi con diverse strutture ed organizzazioni logiche riconoscibili su più livelli (architetture multi-tier) che va a mappare l'architettura a livello fisico-infrastrutturale sul quale l'applicazione web viene eseguita. Alcuni esempi di organizzazione logica sono, considerando la pila protocollare ISO/OSI:

- *Logica di business*: il secondo livello è costituito dal motore applicativo (back-end, cioè che elabora i dati ricevuti), presente tipicamente su application server e costituito da codice sorgente in linguaggio di sviluppo dinamico lato-server (PHP, ASP, ecc.). Il core applicativo back-end riceve, elabora e soddisfa le richieste del client. Il quarto livello, cioè trasporto, differenzia un'applicazione web da un normale sito web, in quanto implementa il servizio web sotto forma di codice sorgente fornito al client.
- *Logica di presentazione*: il livello fisico è costituito per lo più dall'interfaccia utente (HTML e CSS, JSP, JSF, Javascript, ecc.) la quale sostituisce le GUI (Graphic User Interface), applicazioni software non basate sul web.
- *Strato dati*: terzo livello riconducibile al motore database associato (MySQL, MSSql, ecc.). Riceve e soddisfa richieste di lettura/scrittura sul database da parte della logica applicativa. Può risiedere direttamente sul DB o insieme alla logica applicativa sull'application server.

L'utilizzo di un web browser come client è conveniente per via della compatibilità cross-platform (multiplatforma): permette di mantenere e aggiornare applicazioni web senza dover installare software in migliaia di client computer. In sostanza, il web browser del client invia richieste al motore applicativo dinamico attraverso il protocollo HTTP/HTTPS, il motore applicativo interpreta le interrogazioni al motore DBMS e genera un risultato diretto allo stesso browser, che lo interpreta e lo restituisce all'utente sotto forma di Web. [WIK16d]

Le applicazioni web hanno lo svantaggio di non poter interagire direttamente con l'hardware e il software del dispositivo e la dipendenza dalla rete, nonostante i costi computazionali siano decisamente minori grazie al fatto di essere scritte in HTML.

### 2.2.1 Applicazioni ibride come evoluzione delle Web App

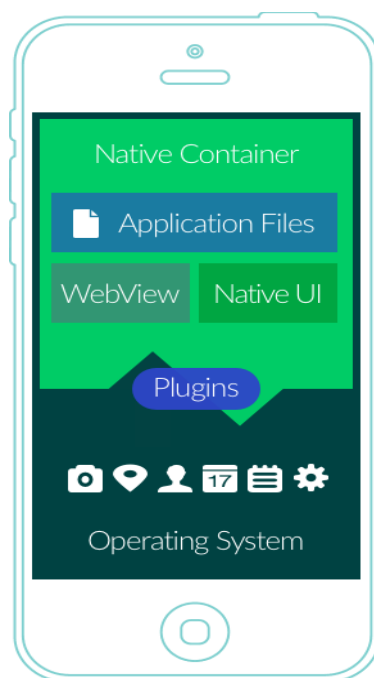
Le applicazioni ibride congiungono la tecnologia delle Web app con quella delle app native, in modo da renderle scaricabili tramite uno store e installabili direttamente sul proprio smartphone o tablet anziché visualizzarle sul browser e perlopiù utilizzabili anche in mancanza di connessione alla rete. Come le applicazioni web, le applicazioni ibride sono costituite da una combinazione di HTML, CSS e Javascript, l'unica

La differenza è quella di essere ospitate all'interno di applicazioni native, rendendo possibile l'accesso all'hardware e software e di poter usufruire quindi di accelerometro, fotocamera, contatti e altro ancora.

L'obiettivo delle applicazioni ibride non è essere eseguite su un browser per dispositivi mobili, bensì utilizzare una WebView contenuta nell'applicazione nativa.

Applicazioni ibride e native sono difficili da distinguere quando la prima è ben fatta poiché il comportamento e l'aspetto è identico.

La maggior parte delle applicazioni ibride per dispositivi mobili oggi sono progettate con l'ausilio di Apache Cordova, un framework di sviluppo che fornisce una serie di JavaScript API per accedere alle funzionalità del dispositivo attraverso plugin. Quest'ultimi sono scritti in codice nativo e includono le API per accedere all'accelerometro, contatti, fotocamera e altro. In Figura 12 si può notare come il contenuto nativo venga interfacciato con il sistema operativo tramite plugin.



**Figura 13: Rappresentazione applicazione ibrida [TEL15]**

La motivazione principale per la scelta dell'implementazione di un'applicazione ibrida, oltre alla semplicità di programmazione, è la possibilità di mirare a renderla disponibile su più piattaforme. Tra i problemi nella realizzazione di applicazioni ibride ci possono essere l'inconsistenza tra le WebViews dei vari dispositivi e la possibilità che i plugin non siano disponibili o non siano aggiornati per l'implementazione di una certa caratteristica. [TEL15]



## 2.3 Apache Cordova come framework di sviluppo

Apache Cordova è un framework open-source di sviluppo per dispositivi mobili. Permette di utilizzare le tecnologie web standard come HTML5, CSS3 e Javascript per lo sviluppo su più piattaforme, evitando qualsiasi contatto con il linguaggio delle applicazioni native. Le applicazioni vengono poi eseguite grazie ad API conformi agli standard per accedere ai sensori, dati e allo stato della rete e le altre funzionalità native del dispositivo [APC16]. Per superare le limitazioni dei browser mobile Cordova incorpora codice HTML5 all'interno di una WebView nativa sul dispositivo, utilizzando un'interfaccia esterna per accedere alle risorse native del dispositivo [WIK16e]. In pratica l'architettura di Apache Cordova si presenta come un contenitore di applicazione Web eseguita localmente; la sua interfaccia grafica è una WebView che occupa l'intero schermo del dispositivo e attraverso la quale si possono visualizzare l'HTML e CSS e viene eseguito il codice Javascript [HTM13].

Per comunicare tra loro, Cordova e i componenti nativi si interfacciano attraverso i plugin. In questo modo viene richiamato del codice nativo tramite le JavaScript API. Queste ultime, insieme alle WebView, dipendono dalla piattaforma mobile sulla quale verranno utilizzate. Esistono plugin creati da terze parti da poter utilizzare nella propria applicazione per potere sfruttare altre caratteristiche dell'applicazione. I plugin sono necessari per la comunicazione tra Cordova e i componenti nativi. In figura 13 si può osservare la struttura di una applicazione creata con Apache Cordova.



*Figura 14: Struttura applicazione con Apache Cordova [HTM13]*

Cordova non fornisce widget per la creazione e modifica dell'interfaccia utente (UI) o MV\* (Model-View) che devono essere provenienti da terze parti e inclusi nell'applicazione.

Componenti base di un'applicazione creata con Apache Cordova:

- `config.xml`: fornisce informazioni sull'applicazione e specifica i parametri sul funzionamento. Aderisce alle specifiche widget del W3C (World Wide Web Consortium).
- `index.html`: pagina web che conterrà l'HTML ed eventualmente il CSS, il codice Javascript e file multimediali. Viene eseguita come Web View.

Si possono seguire due diversi approcci per lo sviluppo di applicazioni ibride con Apache Cordova:

- *Sviluppo Cross-platform CLI (Command Line Interface)*: da seguire se si desidera che la propria applicazione giri sul maggior numero di sistemi operativi possibile. Questo metodo implica uno sviluppo generale e meno specifico per tipo di piattaforma. La CLI è uno strumento ad alto livello che permette di costruire un progetto compatibile per più piattaforme, astruendo molte delle funzionalità degli script di livello inferiore della shell. Le CLI forniscono anche un'interfaccia comune per applicare plugin nella propria applicazione.
- *Sviluppo dedicato a una piattaforma*: questo approccio è consigliabile nei casi in cui si desideri sviluppare un'applicazione compatibile con un'unica piattaforma. Per esempio, si può utilizzare questo approccio quando si vogliono unire componenti nativi con componenti web-based di Cordova e se è necessario modificare il progetto all'interno di un SDK (Software Development Kit). Questo metodo di lavoro è sconsigliabile per la progettazione di applicazioni compatibili con più piattaforme in quanto mancano strumenti ad alto livello, causando la separazione di cicli di costruzione e la necessità di modificare i plugin per ogni piattaforma.

Cambiare metodo di sviluppo da CLI cross-platform a quello dedicato alla piattaforma è possibile, però non è fattibile tornare indietro perchè verrebbe ignorato il codice comune alle piattaforme per basarsi invece su quello della specifica piattaforma [APC16].

Allo stato attuale Apache Cordova supporta le seguenti piattaforme mobile: Android, iOS, Blackberry, Bada, Tizen e Windows Phone [HTM13].

### 2.3.1 Adobe Phonegap e Apache Cordova: le differenze

Phonegap è un framework cross-platform mobile che permette di creare applicazioni ibride. Fu creato nel 2009 da una startup chiamata Nitobi come un metodo open source per accedere ad un ambiente nativo attraverso Web View incorporate in un applicazione nativa. L'obiettivo era quello di rendere possibile lo sviluppo di applicazioni mobile tramite l'utilizzo di tecnologie web come HTML, CSS e Javascript, rimanendo pur sempre in grado di utilizzare codice nativo quando necessario. Nel 2011 Adobe comprò la startup e con essa i diritti di Phonegap, mentre il nucleo open-source fu donato all' Apache Software Foundation con il nome di Cordova [ION15]. La rinominazione del software fu dovuta al fatto di dover assicurare che la proprietà intellettuale non fosse affetta da ambiguità del marchio di fabbrica dopo l'unione.

Il framework Phonegap è quindi diventato una distribuzione gratuita di Apache Cordova. Si può pensare ad Apache Cordova come al motore che fa funzionare Phonegap, come il WebKit è il motore che fa andare Chrome o Safari.

Inizialmente le differenze tra Cordova e Phonegap erano minimali, Adobe però aveva in mente dei piani per dotare l'ecosistema di Phonegap di un insieme di servizi che risulterebbero inappropriati per un progetto Apache [PHG12]. Il primo esempio di questi piani fu Phonegap Build, il quale si prende l'onore di compilare le applicazioni al posto del programmatore semplicemente inserendo HTML, CSS e Javascript nel cloud service e creando applicazioni pronte per essere inserite nello store senza dover mantenere gli SDK nativi [BPH16].

Quando si vuole creare un'applicazione ibrida si può scegliere se utilizzare Cordova o entrare nell'ecosistema Adobe usando la sua distribuzione Phonegap [ION15]. Per gli sviluppatori è tutto rimasto uguale a prima che Phonegap fosse comprato da Adobe: un framework gratuito a licenza aperta. I collaboratori invece, inteso come coloro che vogliono correggere bug, contribuire con il codice, aggiungere dei test o scrivere la documentazione, vengono indirizzati a lavorare con il progetto open source che fornisce energia a PhoneGap: Apache Cordova [PHG12].

## 2.4 HTML per la struttura dei contenuti

Abbiamo nominato nei paragrafi precedenti il linguaggio HTML (Hyper Text Markup Language) per la progettazione di Web App e per l'impaginazione di pagine web disponibili nel World Wide Web (WWW). Fu sviluppato alla fine del 1980 al CERN di Ginevra assieme al protocollo HTTP, destinato al trasferimento di documenti di tale formato. La sua sintassi è stabilita nel World Wide Web Consortium (W3C), inoltre è un linguaggio di pubblico dominio derivato dal linguaggio SGML (Standard Generalized Markup Language), il quale ha scopi più generici [WIK16f].

HTML non è un linguaggio di programmazione, bensì un linguaggio di markup. I linguaggi di markup sono insiemi di regole che descrivono i meccanismi per la rappresentazione di un testo, utilizzando una tecnica di formattazione per mezzo di marcatori di documenti [WP10]. A differenza di un linguaggio di programmazione non sono presenti meccanismi condizionali in base al verificarsi di una situazione o costrutti iterativi, strutture dati, definizione di variabili, nonostante supporti l'inserimento di oggetti esterni come immagini o filmati e script. L'unico scopo del linguaggio HTML o della sua variante XHTML (eXtended HTML) è quello di gestire i contenuti della pagina associandone un layout all'interno della pagina web realizzato grazie all'utilizzo di tag diversi, uno per ogni ruolo dei contenuti che esso contrassegna. I tag sono marcatori che descrivono caratteristiche come colore, funzione, dimensioni e posizione all'interno della pagina.

### 2.4.1 Gestione dei documenti HTML da parte del Browser

Un documento HTML contiene tutte le indicazioni sulla disposizione di elementi della pagina web che verrà aperta su un Browser. Il Browser web svolge due compiti: scaricare i file che si trovano sul web server e leggere i documenti scritti in HTML con i relativi file associati, disponendoli nella pagina. Il web server è un software installato sulle macchine contenenti i documenti HTML che si occupa di inviarli ai browser degli utenti richiedono di visualizzarli, facendo richiesta attraverso il protocollo HTTP, il quale si occupa del trasferimento dati. Esistono tipi di pagine web dinamiche con cui un utente può compiere operazioni interattive avanzate che vengono generate del tutto o parzialmente tramite un codice eseguibile risiedente sul server in modo da poter interagire con altre applicazioni presenti sul server, come un database. In

questo caso i documenti sono scritti in linguaggi come Java, PHP, ASP o Javascript. [WIK16f]

La visualizzazione di un file HTML da parte del browser è chiamata rendering della pagina, il motore di rendering quindi è la sezione del browser che si occupa di mostrarla agli utenti. La cache del browser si occupa di mantenere in memoria le pagine in modo di non doverle ricaricare ogni volta quando vengono richieste. [HTM06a]

## 2.4.2 Sintassi HTML

L'elemento, cioè la struttura base che ha la funzione di formattare i dati o indicare al browser delle informazioni è la componente principale del linguaggio HTML. Ogni elemento è racchiuso all'interno dei tag, marcature costituite da una sequenza racchiusa tra due parentesi angolari <>.

Durante la scrittura di un testo o codice i tag vanno aperti e chiusi. La chiusura viene indicata con una barra subito dopo l'apertura della parentesi triangolare. Esempio:

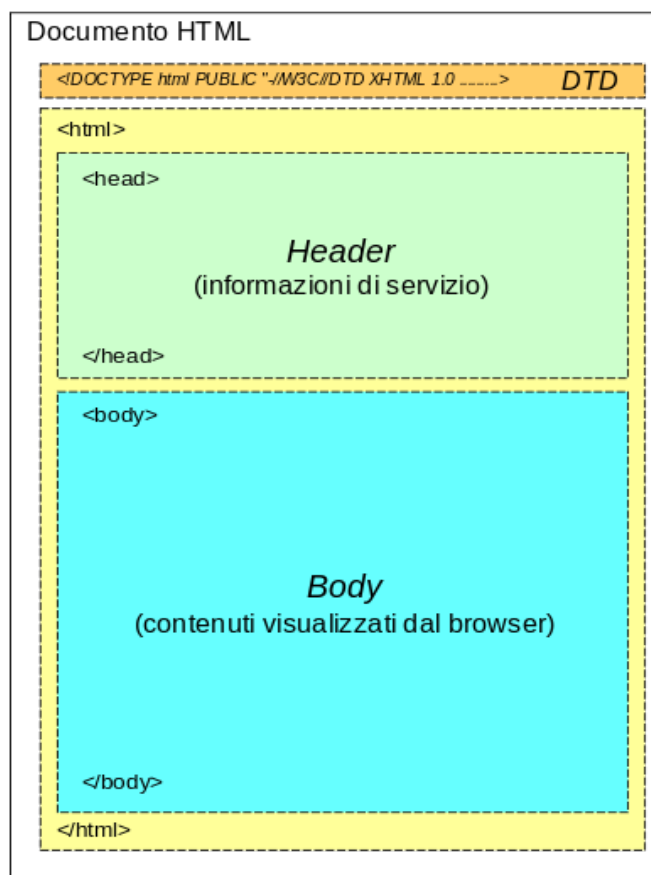
```
<h1>Hello World!</h1>
```

Alcuni tag non richiedono chiusura (tag di chiusura implicita), come ad esempio il tag <img> che serve per l'inserimento di un'immagine in un determinato punto della pagina, oppure il tag <link> che serve per includere, per esempio, un foglio di stile[WIK16f]. Nella Figura 14 possiamo vedere un chiaro esempio di struttura di codice HTML con i relativi tag.

```
<h1>This is a heading</h1>
<p>this is a paragraph</p>
<div>
  <p>These are links:</p>
  <a href="one.html">One</a>
  <a href="two.html">Two</a>
  <a href="three.html">Three</a>
</div>
<table>
  <thead>
    <tr>
      <th>
        <td>Heading 1</td><td>Heading 2</td><td>Heading 3</td>
      </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Cell 1</td><td>Cell 2</td><td>Cell 3</td>
    </tr>
  </tbody>
</table>
```

*Figura 15: esempio di codice HTML*

### 2.4.3 Struttura di un documento HTML



**Figura 16: struttura di un documento HTML [WIK16f]**

Come vediamo in Figura 15 un documento HTML inizia con una breve stringa ed è seguito da una sezione, contenente a sua volta due sezioni.

La stringa iniziale `<!DOCTYPE html PUBLIC "html version">` denota il tipo di documento, indicandone la sintassi e la versione. Questa dichiarazione è necessaria per il browser, il quale leggendola riesce ad interpretare in modo appropriato il documento ed a visualizzarlo.

I tag `<html>` e `</html>` delimitano l'intero documento. Al loro interno lo standard prevede due sezioni ben distinte e disposte nell'ordine seguente: [WIK16f]

- *Header*: sezione di intestazione che contiene informazioni di controllo non visualizzate dal browser, racchiuse all'interno dei tag `<head>` e `</head>`.

I tag contenuti al suo interno sono:

- Metadata per indirizzare le informazioni utili ad applicazioni esterne o al browser (ad esempio la codifica caretteri)
- Metadata di tipo http-equiv per controllare informazioni aggiuntive nel protocollo http
- Link a file di servizio esterni, script, icone visualizzabili nella barra degli indirizzi del browser (fav-icon)
- Inserimento di codice eseguibile (script)
- Informazioni di stile (CSS locali)
- Titolo associato alla pagina visualizzato nella finestra del browser
- *Body*: contiene la parte informativa, vale a dire il testo, le immagini e tutto ciò che andrà a costituire la parte visualizzata dal browser. I tag delimitatori sono `<body>` e `</body>`. I tag contenuti sono:
  - Intestazioni
  - Strutture di testo (indentazioni, paragrafi, ecc.)
  - Aspetto del testo (grassetto, corsivo, sottolineato, ecc.)
  - Elenchi e liste
  - Tabelle
  - Moduli elettronici (campi compilabili dall'utente)
  - Collegamenti ipertestuali ed ancore
  - Layout generico del documento
  - Inserimento immagini
  - Inserimento contenuti multimediali
  - Inserimento contenuti interattivi (script, applicazioni esterne)

#### 2.4.4 Novità introdotte da HTML5

HTML5 è la quinta revisione delle specifiche HTML, pubblicata dalla W3C Recommendation da ottobre 2014. Fu sviluppato dal gruppo di lavoro Web Hypertext Application Technology Working Group (WHATWG) con l'obiettivo di progettare

specifiche per lo sviluppo di applicazioni web piuttosto che documenti, migliorando HTML e le altre tecnologie. HTML5 racchiude in sé l'eredità semantica dell'XHTML2 e la volontà di aiutare lo sviluppo di applicazioni Web. Il modello di markup è stato evoluto per modificare le basi della sintassi, le regole per la disposizione dei contenuti sulla pagina e ampliato per accogliere nuovi elementi.



*Figura 17: logo HTML5 [WIK16g]*

L'evoluzione dell'HTML avviene in contemporanea a quella di CSS e altri importanti standard, di conseguenza sono state estese anche le API JavaScript (Figura 16 mostra quelle supportate da HTML5) per supportare nuove funzionalità utili nella realizzazione di applicazioni moderne :

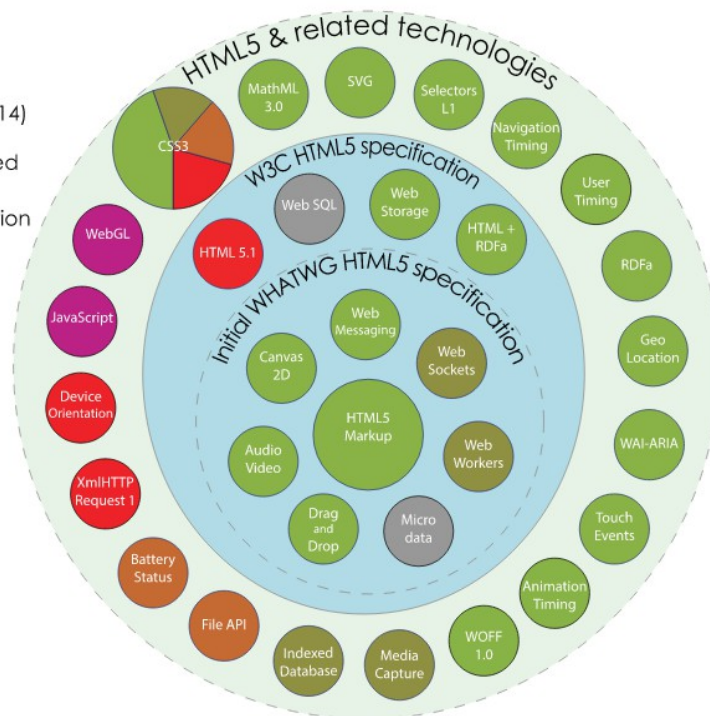
- salvare informazioni sul dispositivo dell'utente;
- accedere all'applicazione anche in assenza di connessione Web;
- comunicare in modo bidirezionale sia con il server sia con le altre applicazioni;
- eseguire operazioni in background;
- pilotare flussi multimediali;
- pilotare lo storico di navigazione;
- usare metafore di interazione tipiche di applicazioni desktop come il drag and drop;
- generare grafica 2D in tempo reale;
- generare grafica 3D in tempo reale;
- accedere e manipolare informazioni generate in tempo reale dall'utente attraverso sensori multimediali come microfono e webcam.



# HTML5

Taxonomy & Status (October 2014)

- Recommendation/Proposed
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated or inactive



**Figura 18: API collegate a HTML5 e specifiche. [WIM16]**

A livello semantico sono nate nuove specifiche per la generazione di microformati e l'integrazione tra HTML5 e delle sue possibili estensioni a livello di attributi. Sono state studiate altre iniziative per:

- accedere alle informazioni geografiche del dispositivo;
- mantenere un database sul dispositivo dell'utente;
- generare grafica 3D in tempo reale;

Il linguaggio HTML5 offre agli sviluppatori web un linguaggio modificabile secondo le più recenti necessità sia dal lato di strutturazione del contenuto che da quello dello sviluppo di applicazioni web [HTM11].

A livello di struttura del documento sono state apportate delle novità per supportare la memorizzazione locale di grosse quantità di dati scaricate dal web browser per consentire l'utilizzo di applicazioni basate sul web anche in assenza di collegamento alla rete internet, in particolare [WIK16g]:

- regole più stringenti per la strutturazione del testo in capitoli, paragrafi e sezioni;
- introduzione di elementi di controllo per i menù di navigazione;

- miglioramento ed estensione degli elementi di controllo per i moduli elettronici;
- introduzione di elementi specifici per il controllo di contenuti multimediali come il tag `<video>` e il tag `<audio>`;
- eliminazione di alcuni elementi che hanno dimostrato scarso utilizzo;
- estensione di una serie di attributi finalizzati all'accessibilità a tutti i tag;
- supporto di Canvas che permette di utilizzare JavaScript per la creazione di immagini e grafica bitmap;
- introdotta la geolocalizzazione;
- introdotto il Web Storage al posto dei normali cookie;
- standardizzazione dei programmi JavaScript;
- sostituzione della complessa stringa doctype con una più semplice `<!DOCTYPE html>`.

## 2.5 CSS per la presentazione dei contenuti HTML

Cascading Style Sheet (CSS) è un linguaggio per fogli di stile utilizzato per descrivere la presentazione di documenti scritti in linguaggi di markup. Solitamente è accoppiato a HTML e XHTML per definire lo stile di pagine web ma può essere applicato a qualsiasi documento XML, SVG, XUL e per il render di elementi multimediali. Insieme a HTML e JavaScript è una tecnologia utilizzata per la creazione di pagine web e interfacce utente per applicazioni mobile. La sua invenzione è dovuta al fatto che i tag di formattazione erano l'unico modo per definire la formattazione, creando un documento eccessivamente lungo. I tag di formattazione inoltre presentavano diversi problemi tra cui: la lunghezza dei tag, mancanza di logica del codice HTML che portava lavoro aggiuntivo al browser e le diverse risoluzioni dei dispositivi.

CSS separa il contenuto del documento dalla presentazione dello stesso inserendo aspetti come layout, colori e font, aumentando l'accessibilità del contenuto e la flessibilità nelle caratteristiche di presentazione. Può essere creato un codice CSS dedicato a un documento HTML oppure compatibile con più documenti.

Per ogni elemento HTML, CSS prevede una serie di istruzioni di formattazione.

Per esempio se tutti gli elementi scritti all'interno di un paragrafo `<p>` devono essere scritti in grassetto, verrà settata la proprietà CSS `font-weight:bold;` rendendo formattazione con il tag `<bold>` non necessaria. In questo modo, dividendo la formattazione e il contenuto, è possibile presentare la stessa pagina di markup con diversi stili o diversi metodi di rendering, come su schermo, stampati o a voce. Può essere utilizzato per rendere il formato di pagine web riadattabile in base alle dimensioni dello schermo del dispositivo sul quale viene visualizzato (responsive layout). Il file CSS incluso nel documento di markup da parte dell'autore può essere sovrascritto da uno diverso in qualora si voglia, nel caso invece che il documento non contenga nessun link a un foglio di stile, viene adottato lo stile predefinito del browser utilizzato.

Le specifiche dei fogli di stile CSS definiscono le priorità sulle regole da applicare quando una o più regole vengono assegnate ad un elemento. Le proprietà sono calcolate e assegnate alle regole con un metodo “a cascata”. Inoltre, le specifiche sono mantenute dal World Wide Web Consortium (W3C). Figura 18 mostra un esempio di foglio di stile CSS [WIK16h].



```

h1 { color: white;
      background: orange;
      border: 1px solid bla
      padding: 0 0 0 0;
      font-weight: bold;
    }
/* begin: seaside-theme */

body {
  background-color:white;
  color:black;
  font-family:Arial,sans-serif;
  margin: 0 4px 0 0;
  border: 12px solid;
}

```

**CSS**

**Figura 19: Esempio di CSS [WIK16h]**

Il codice CSS può essere inserito in una pagina HTML in tre modi diversi:

- Inserendo nel tag `<head>` un collegamento a un foglio esterno tramite

il tag:

```
<link rel="stylesheet" type="text/css" href="css/style.css" />
```

- Importando il foglio di stile all'interno del tag style:

```
<style type="text/css">
  @import url(style.css)
</style>
```

- In linea all'interno degli elementi:

```
<tag style="dichiarazioni CSS">...</tag>
```

Le regole CSS che vengono applicate al browser in fase di rendering degli elementi HTML interessati sono strutturate nel seguente modo [WIK16h] :

```
selettore{
  proprietà1:valore1;
  proprietà2:valore2, valore3;
}
```

I selettori possono essere [WIK16h]:

- Selettori di tipo: applicano la regola a tutti gli elementi della pagina del tipo determinato;
- Classi: applicano la regola a tutti gli elementi della pagina che presentano la proprietà `class="nome_classe"`;
- Identificatori: applicano la regola all'elemento della pagina che presenta la proprietà `id="nome_id"`;

## 2.6 JavaScript per l'interattività utente

JavaScript è un linguaggio di programmazione dinamico ed ad alto livello, orientato agli oggetti e agli eventi. Assieme a HTML e CSS rappresenta una delle tre essenziali tecnologie del WWW. Fa parte dei linguaggi di scripting le cui specifiche sono state standardizzate per la prima volta con il nome di ECMAScript tra il 1997 e il 1999 dall'ECMA (organizzazione no-profit degli standard per l'informazione e i sistemi di comunicazione). Contiene un'API per lavorare con documenti di testo, array, date e regular expressions, mentre per lavorare con gli I/O si affida al sistema sottostante in cui è incapsulato. JavaScript non solo viene utilizzato nella maggior parte di siti web ed è supportato da tutti i browser moderni senza la necessità di utilizzare plugin ma trova applicazione anche in ambienti non basati sul Web, come documenti PDF e widget per il desktop, programmazione client-side e grazie ad ambienti di programmazione

runtime come Node.js è utilizzabile anche per la programmazione server-side [WIK16i].

L'enorme diffusione di JavaScript è dovuta soprattutto alla nascita di nuove librerie con lo scopo di semplificare la programmazione sul browser, ma anche alla nascita di framework lato server e dall'essere il linguaggio principale dei dispositivi mobili. Gli utenti di JavaScript solitamente sono programmatori con esperienze di sviluppo in altri linguaggi di programmazione come Java, C++ o C# oppure sviluppatori che hanno prima imparato ad utilizzare le sue librerie per poi avvicinarsi al linguaggio di base. Non tutti gli utenti di JavaScript sono dei veri e propri sviluppatori, come coloro che si sono trovati a dover integrare le proprie pagine Web [HTM14].

A differenza di altri linguaggi di programmazione che permettono la scrittura di programmi completamente stand-alone, JavaScript viene utilizzato per il linguaggio di scripting, quindi viene implementato all'interno di un altro programma. Il programma ospite deve fornire un'API ben definita che consenta specifiche operazioni al quale lo script possa fare riferimenti. Questo processo è analogo all'inclusione di librerie in C o Java per effettuare operazioni non previste dal linguaggio in sé.

In ambito Web, JavaScript viene utilizzato per la scrittura di funzioni integrate nelle pagine HTML, il quale si rapporta con il browser tramite interfacce chiamate DOM (Document Object Model). Gli standard DOM imposti dal W3C vengono rispettati in base alle impostazioni dei vari browser, alcuni possono esporre diversi oggetti allo script ed è spesso necessario implementare controlli aggiuntivi per garantire la compatibilità con ciascun browser.

Gli interpreti JavaScript sono integrati anche in applicazioni al di fuori dal web, come Adobe Reader e Adobe Acrobat per file PDF o Windows Script Host di Microsoft per quanto riguarda lo scripting per di sistemi operativi.

### **2.6.1 Aspetti strutturali**

JavaScript è un linguaggio interpretato, vale a dire che il codice non viene compilato. Per esempio il browser nel caso di JavaScript lato client esegue direttamente il codice sul client e non sul server con il vantaggio di non sovraccaricare il web server a causa delle richieste dei client. Al contrario, in caso di script con codice sorgente particolarmente grande, il tempo di download aumenta notevolmente. JavaScript inoltre definisce le funzionalità tipiche dei linguaggi di programmazione di alto livello e

consente l'utilizzo del paradigma ad oggetti. La sua sintassi è simile a quella del C++ e del Java, si dice che sia debolmente orientato agli oggetti perchè alcuni meccanismi sono più simili ad altri linguaggi di programmazione rispetto a Java. Altre funzionalità degne di nota sono [WIK16i]:

- L'uso di Unicode, sistema di codifica che assegna un numero univoco ad ogni carattere per la scrittura di testi.
- La valutazione di espressioni regolari.
- Le espressioni JavaScript contenute in una stringa possono essere valutate usando la funzione `eval`.

Nei documenti HTML viene inserito tramite il tag `<script> ... </script>` che permette sia di importare il codice da un file JavaScript, sia di scriverlo al suo interno. Nel primo modo è possibile creare codice JavaScript riutilizzabile.

## 2.6.2 Sicurezza

JavaScript e DOM forniscono la possibilità per i malintenzionati di consegnare tramite il Web script eseguibili sul computer client. Esistono due tipi di restrizioni applicabili dagli autori del browser:

1. Eseguire gli script in sandbox nelle quali possono venire eseguite solo le azioni relative al Web e nessuna attività di programmazione generale come la creazione di file.
2. Gli script sono vincolati dalla politica di origine: gli script di un sito web non hanno accesso a informazioni personali dell'utente.

La maggior parte dei bug legati a JavaScript sono legati alla violazione di entrambi i punti, l'unico modo per assicurare che venga eseguito solo codice sicuro su una pagina Web è il Content Security Policy [WIK16i].

## 2.6.3 Eventi

Gli elementi di testo possono essere fonte di vari eventi che possono avviare un'azione, in HTML avviene attraverso funzioni richiamate nello script JavaScript tramite la seguente sintassi:

```
Oggetto.evento = handler;
```

Le categorie di eventi che JavaScript mette a disposizione dello sviluppatore sono le seguenti [WIK16i]:

- Eventi attivabili dai tasti del mouse;
- Eventi attivabili dai movimenti del mouse;
- Eventi attivabili dal trascinamento del mouse (drag and drop);
- Eventi attivabili dall'utente con la tastiera;
- Eventi attivabili dalle modifiche dell'utente;
- Eventi legati alla focalizzazione;
- Eventi attivabili dal caricamento degli oggetti;
- Eventi attivabili dai movimenti delle finestre;
- Eventi legati a particolari bottoni;

#### 2.6.4 Libreria jQuery e richieste Ajax

jQuery è un framework sviluppato da John Resig a partire dal 2006, il cui scopo è quello di rendere il codice più sintetico e limitare al minimo l'estensione degli oggetti globali per ottenere la massima compatibilità con le altre librerie [HTM09]. Tra gli obiettivi troviamo anche la semplificazione della selezione, la manipolazione, la gestione degli eventi e l'animazione di elementi DOM in pagine HTML implementando le funzionalità Ajax. Distribuito con licenza MIT (Massachusetts Institute of Technology), jQuery è un software libero.

Figura 19 mostra un chiaro esempio di codice jQuery e la sua sintassi semplificata, confrontandola con le stesse azioni in JavaScript. Il simbolo \$ sta ad indicare la parola jQuery, mentre il simbolo # si riferisce all'identificatore di un tag html.

```
// trovare il valore dell'attributo href del tag a con id miolink
// <a id="miolink" href="http://www.html.it">link</a>

document.getElementById("miolink").href; // JavaScript nativo
$('miolink').readAttribute('href'); // Prototype
document.getElementById("miolink").readAttribute('href') // Prototype, come detto, estende
il DOM
$("#miolink").attr("href"); // jQuery
```

**Figura 20: esempio sintassi jQuery [HTM09]**

Tramite jQuery si possono mandare delle richieste Ajax (Asynchronous JavaScript and XML). Ajax è una tecnica di sviluppo software per lo sviluppo di applicazioni web interattive, permettendo lo scambio di dati tra server e web browser, consentendo l'aggiornamento dinamico di una pagina web. Siccome i dati sono richiesti al server e caricati in background si dice che Ajax è sincrónico. Ajax è inoltre una tecnica multiplatforma che utilizza una combinazione di:

- HTML, CSS per il markup e lo stile;
- DOM per mostrare le informazioni e interagirvi;
- XMLHttpRequest per lo scambio asincrono di dati tra il web server e il browser (Esempio di richieste XMLHttpRequest in Figura 20);
- XML, JSON e altri come formato di scambio dati.

```
function loadDoc() {  
    var xmlhttp = new XMLHttpRequest();  
    xmlhttp.onreadystatechange = function() {  
        if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {  
            document.getElementById("demo").innerHTML =  
xmlhttp.responseText;  
        }  
    };  
    xmlhttp.open("GET", "ajax_info.txt", true);  
    xmlhttp.send();  
}
```

***Figura 21: Esempio di richiesta XMLHttpRequest [W3C16a]***

JavaScript è un linguaggio client-side per cui Ajax viene utilizzato al suo interno per effettuare richieste al web server, semplificando il procedimento. Grazie alle richieste Ajax si possono ottenere solo i dati necessari dal database, garantendo una maggiore velocità delle applicazioni e minor tempo di elaborazione da parte del server, mentre nelle applicazioni web tradizionali ogni richiesta al web server deve essere trasmessa su ogni interazione con l'applicazione, rendendo l'interfaccia utente più lenta. Le richieste Ajax possono essere effettuate anche tramite un'API jQuery che ne semplifica la sintassi (Esempio in Figura 21).



```
$("#button").click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
        $("#div1").html(result);
    }});
});
```

*Figura 22: Esempio di richiesta Ajax tramite API jQuery [W3C16b]*

## 2.7 Php per l'accesso al database

PHP (Hypertext Preprocessor) è un linguaggio di scripting interpretato concepito per la creazione di pagine web dinamiche. Fu creato nel 1994, il suo interprete è un software libero distribuito sotto la PHP License.

Trova applicazione maggiore nello sviluppo di applicazioni web server-side, nonostante possa essere utilizzato anche per script a riga di comando o applicazioni stand-alone con interfaccia grafica [WIK16j].

Nelle applicazioni web, un client effettua una richiesta ad un server remoto che attraverso il linguaggio di scripting server-side, in questo caso PHP risiedente nel server, interpreta la richiesta grazie al Web server e invia una risposta al client. Normalmente le azioni compiute dal browser e dal Web server durante una richiesta sono le seguenti:

- Viene effettuata una richiesta al server remoto tramite il protocollo di trasporto HTTP;
- Il Web server riceve la richiesta e la interpreta;
- Il Web server restituisce al client una risposta;

In questo processo, PHP si occupa di elaborare i dati ricevuti tramite la richiesta HTTP e di restituire una risposta al client [HTM09].

PHP trova applicazione anche nel caso di script da linea di comando oppure creazione di interfacce grafiche (GUI) lato client.

PHP agisce principalmente come un filtro, prendendo un input da un file contenente testo oppure istruzioni PHP e fornendo un altro flusso di dati in output. Spesso l'output può essere HTML, JSON, XML o dati binari come immagini o formati audio [WIK16j].

## 2.7.1 Caratteristiche di PHP

PHP è un linguaggio di scripting server-side con molte funzionalità. Tra i vantaggi si ha l'operatività multiplatforma, la compatibilità con la maggior parte dei server utilizzati oggi, cui ne deriva anche il supporto di una vasta gamma di database. Le funzionalità principali del linguaggio PHP sono [W3C16c]:

- *Risposta alle richieste HTTP:* le richieste effettuate al server remoto tramite HTTP possono essere di due tipi: GET o POST. GET è il metodo con cui vengono richieste le informazioni mentre POST consente di inviare dati, entrambe riferite ad un Web server. Il linguaggio PHP si occupa di elaborare i dati ricevuti tramite GET o POST e di inviare la risposta al client.
- *Pagine dinamiche:* può generare contenuto di pagine dinamiche, cioè pagine in cui il contenuto è generato in sul momento dal server. Una pagina dinamica può essere composta da codice PHP insieme a linguaggio di markup.
- *Operazione sul server:* PHP può creare, aprire, leggere, scrivere, cancellare e chiudere file sul server.
- *Raccolta dati dalle form:* tramite PHP si possono raccogliere dati dalle form e inserire o confrontare in un database.
- *Sessioni:* accesso in lettura/scrittura ai cookie del browser e supporto alle sessioni sul server. I cookie sono utilizzati per tenere traccia delle informazioni di un sito Web sul client dell'utente. Le sessioni invece vengono create per memorizzare informazioni utente sul server anziché sul client. Un tipico esempio di sessione è il mantenimento delle informazioni dell'utente sull'applicazione web dopo il login per accedere a contenuti riservati per gli utenti registrati, in modo da poter compiere operazioni fino al logout, cioè la chiusura della sessione [HTM09].
- *Operazioni sul database:* PHP è in grado di interfacciarsi con MySQL, PostgreSQL, MariaDB, Oracle, Microsoft SQL Server e altri tanti tramite librerie, potendo compiere azioni su di loro (manipolare le immagini, effettuare connessioni remote [HTM09]).

## 2.7.2 Sintassi

Lo script può iniziare in qualsiasi punto del documento semplicemente tramite la notazione: [W3C16c]

```
<?php
// PHP code goes here
?>
```

PHP appartiene alla famiglia di linguaggi con sintassi simile al C. Nella seguente tabella (Figura 22) si possono notare le differenze tra i due linguaggi.

[HTM09]

Linguaggio C	Linguaggio PHP
Tipizzazioni molto forte	Tipizzazione debole
Paradigmi supportati: imperativo, procedurale, strutturato	Paradigmi supportati: imperativo, funzionale, object-oriented, procedurale, riflessivo
Supporta i puntatori	Non supporta i puntatori
Utilizzato principalmente per software Desktop	Utilizzato principalmente per applicativi Web
Operazioni con le stringhe particolarmente complicate	Operazioni con le stringhe molto semplici da effettuare
Non supporta I/O filesafe	Supporta I/O filesafe
Non supporta system calls	Supporta system calls
Tipi di dati numerici supportati: int8_t, uint8_t, int16_t, uint16_t, int32_t, uint32_t, int64_t, uint64_t, int, unsigned int, signed char, unsigned char, short, unsigned short, long, unsigned long, long long, unsigned long long	Tipi di dati numerici supportati: 32-bit signed int, 64-bit signed int (long integer)
Tipi di dati float supportati: single precision float, doublelong double	Tipi di dati float supportati: double precision float
Supporta array di dimensioni fisse	Non supporta array di dimensioni fisse
Non supporta array associativi	Supporta array associativi

**Figura 23: differenze di linguaggio tra C e PHP [HTM09]**

# 3 Implementazione

---

In questo capitolo verranno illustrati processi decisionali adottati durante lo sviluppo dell'applicazione myEcoCar, comprese le tecnologie software utilizzate nell'implementazione, di cui ne verranno descritte le funzionalità e il motivo della loro adozione all'interno del progetto.

## 3.1 Introduzione

L'obiettivo del progetto è quello di sviluppare una applicazione web a supporto della mobilità elettrica, cioè che sia in grado di visualizzare i dati ricevuti dall'auto elettrica e di azionare alcuni comandi. Siccome per le funzionalità del progetto non era necessario lo sviluppo di un'applicazione nativa, abbiamo optato per lo sviluppo di un'applicazione ibrida. In particolare è stata sviluppata utilizzando il framework Adobe Phonegap. Tramite questo framework è stato possibile sviluppare l'applicazione utilizzando tecnologie web, tra cui il linguaggio di markup HTML, i fogli di stile CSS e il linguaggio di programmazione JavaScript per le interazioni con l'utente. Per interfacciarsi con il software e l'hardware dei dispositivi sono stati utilizzati plugin messi a disposizione da Apache Cordova e Adobe Phonegap e le varie JavaScript API, la combinazione di quest'ultimi permette di sfruttare al meglio le potenzialità del dispositivo. Le API utilizzate nello sviluppo del progetto sono: Google Maps API, Google Charts, jQuery, Ajax e JSON. Grazie a queste librerie è stato possibile sviluppare un'applicazione in grado di adoperare al meglio i servizi offerti da google, utilizzando un codice semplice, pulito e funzionale grazie a jQuery e Ajax. Durante lo svolgimento del progetto non è stata possibile l'interazione diretta con l'auto elettrica per cui ne è stato simulato il comportamento tramite l'utilizzo di un server locale. Per

questa funzionalità è stato utilizzato WAMP server, il quale permette di creare applicazioni web con Apache2, PHP e database MySQL.

## **3.2 Descrizione e motivazione delle scelte implementative**

L'applicazione è nata come web app principalmente per essere resa disponibile su più piattaforme con bassa complessità di programmazione. Si è poi optato per sviluppare un'applicazione ibrida che racchiudesse in sé la web app sottoforma di WebView incapsulata in un'applicazione nativa per i seguenti motivi: poter comunicare con hardware e software del dispositivo mobile, rendere l'applicazione scaricabile dagli store anziché visualizzabile attraverso il browser e poter essere operativa anche offline in alcuni casi. L'operabilità offline non è disponibile per l'applicativo sviluppato ma è stato preso in considerazione un possibile ampliamento delle caratteristiche anche senza rete. Il framework di sviluppo utilizzato è Adobe Phonegap, il quale essendo una distribuzione di Apache Cordova è gratuito e vanta di tutti i plugin e le guide disponibili sia per Cordova che Phonegap. L'applicazione è stata sviluppata per Android per la maggiore quantità di dispositivi sul quale testare l'applicazione e per la conoscenza dell'IDE Android Studio.

La presentazione dei contenuti è effettuata semplicemente tramite fogli di stile CSS per rendere la grafica personalizzabile a proprio piacimento senza doversi appoggiare a stili predefiniti da API JavaScript quali jQueryMobile o jQueryUi.

Come mappa per visualizzare la localizzazione della propria auto parcheggiata è stata utilizzata l'API di Google Maps, essendo una libreria molto solida e strutturata per mettere un ottimo servizio di geolocalizzazione e offre molteplici funzionalità difficilmente trovabili in mappe di altri autori.

Per quanto riguarda la costruzione dei grafici relativi all'econometro, i ciclibatteria per chilometro e alla percentuale di batteria consumata ad ogni frenata è stata scelta l'API di Google Charts. Quest'ultima mette a disposizione diverse tipologie di grafici da scegliere in base alle proprie necessità e facilmente modificabili a proprio piacimento.

La simulazione del server è stata effettuata grazie alla piattaforma software WampServer poiché l'applicazione è stata sviluppata su sistema operativo Windows e offre il vantaggio di utilizzare MySQL database.

### 3.3 Android Studio come IDE di sviluppo

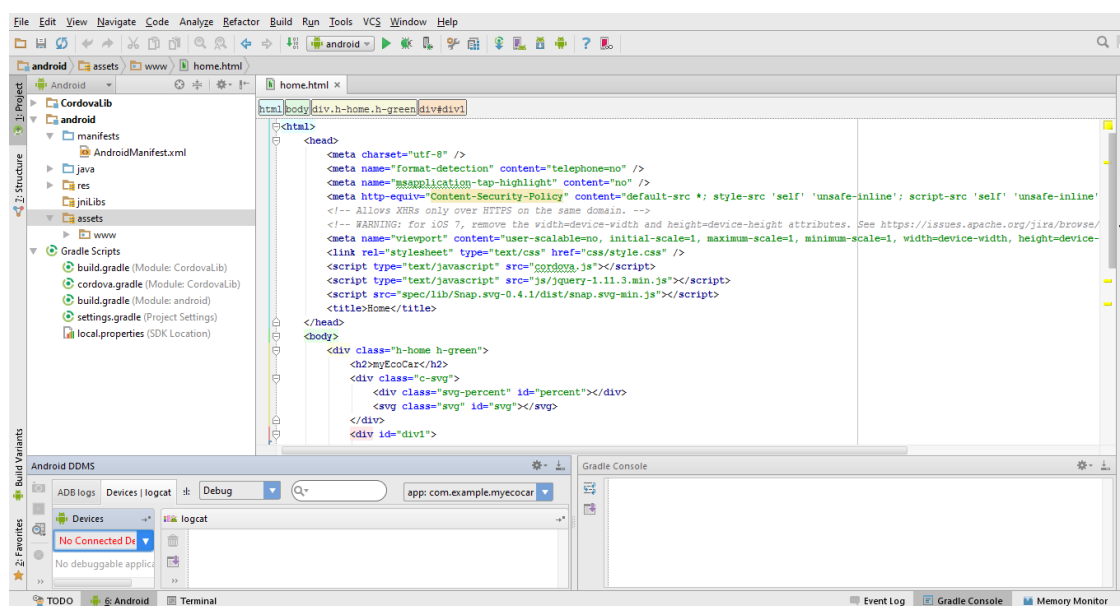
Come precedentemente accennato l'applicazione è stata sviluppata per il sistema operativo Android. Per ottenere una programmazione veloce e ottimale ci siamo affidati a un IDE (Integrated Development Environment), cioè software che con le loro funzionalità aiutano lo sviluppatore nella creazione di codice sorgente, segnalando errori, dando la possibilità di effettuare debug ed eseguire l'applicazione escludendo la necessità di inviare linee di comando dal terminale. Tra gli IDE a disposizione, tra cui NetBeans, Eclipse, ecc. abbiamo scelto di utilizzare Android Studio. Quest'ultimo è stato creato da Google nel Maggio 2013 e pensato proprio per semplificare la programmazione Android, fornendo strumenti e funzionalità adatti a progetti sia di applicazioni native che di applicazioni ibride.

L'editor di Android Studio permette di importare applicazioni ibride, quindi utilizzare tecnologie web come HTML, CSS, JavaScript oltre al linguaggio nativo di Android. Essendo un editor nuovo questi linguaggi sono gestiti in maniera basilare, soprattutto per i fogli di stile CSS e il JavaScript, per cui è stato fondamentale scaricare estensioni che rendessero il codice leggibile sotto diverse colorazioni e che segnalassero errori o autocompletamento di codice.

La scelta di utilizzare questo IDE di sviluppo è basata sulla possibilità di poter emulare l'applicazione su una vasta gamma di dispositivi Android, quindi osservare il comportamento della grafica su display di svariate dimensioni e risoluzioni e dell'interattività utente con diverse versioni del sistema operativo. L'emulazione avviene attraverso la creazione del dispositivo con le caratteristiche opportune attraverso l'Android Virtual Device (AVD) Manager. Durante l'emulazione si può controllare il comportamento dell'applicativo e gli errori attraverso l'Android Device Monitor. Inoltre, è stato possibile testare l'applicazione direttamente sul proprio dispositivo, rendendo possibile la prova dell'applicazione in modo reale. Tuttavia, durante lo sviluppo della grafica, si è optato per l'esecuzione dell'applicazione sulla piattaforma browser anziché l'emulatore. Il browser di default per l'esecuzione del progetto Phonegap è Google Chrome; grazie al suo strumento di ispezione è stato possibile provare aggiustamenti e combinazioni di HTML e CSS per il miglioramento della grafica.

Android Studio permette anche l'installazione di plugin per l'emulazione attraverso altre macchine virtuali più veloci e personalizzabili, come ad esempio GenyMotion.

Oltre a ciò, l'editor facilita anche la condivisione del progetto su GitHub, un servizio di hosting sul web per lo sviluppo di progetti software. Questo servizio permette di creare un repository sul server in cui importare il progetto e caricare ogni modifica tramite un comando di “push”. L'hosting sul web garantisce una maggiore sicurezza contro la perdita di dati e facilita la condivisione del proprio repository con più utenti.



*Figura 24: Panoramica di Android studio*

### 3.4 Adobe Phonegap

Adobe Phonegap è stato scelto come framework di sviluppo per la nostra applicazione. Come spiegato nel capitolo 2, Adobe Phonegap è un framework libero e sotto licenza, nonchè una distribuzione di Apache Cordova. La scelta del framework è dovuta alla necessità di realizzare un'applicazione ibrida, per cui poter usufruire delle caratteristiche di un'applicazione nativa pur sempre sviluppando con linguaggi web.

Phonegap è stato preferito ad Apache Cordova in quanto è stata la prima versione del progetto e mette a disposizione degli utenti un software per facilitare l'esecuzione del progetto sul proprio dispositivo grazie al download dell'applicazione



Phonegap dallo store. Il software in questione è Phonegap Desktop, il quale permette di creare l'applicazione senza passare dalle linee di comando del terminale. Da quest'ultimo si passa all'esecuzione sul dispositivo mobile grazie all'applicazione, la quale si connette con l'indirizzo IP del server, effettua il download dell'applicazione e la esegue. Tuttavia, usando Android Studio come IDE di sviluppo e avendo a disposizione l'emulazione su più dispositivi, questa caratteristica di Phonegap non è stata essenziale nel test della nostra applicazione.



*Figura 25: Aspetto iniziale di un'applicazione Phonegap*

### 3.4.1 Struttura del progetto con Adobe Phonegap

Adobe Phonegap crea un'applicazione nativa vuota nella quale verranno visualizzate le viste tramite una WebView incapsulate al suo interno.

La struttura di un'applicazione Phonegap suddivisa in cartelle è la seguente:

- *Manifest*: questa cartella contiene il file `AndroidManifest.xml`, il quale permette di gestire i comportamenti che l'applicazione può avere nei confronti dell'hardware e software del dispositivo fornisce queste informazioni al sistema Android. In modo più dettagliato, nell'Android Manifest troveremo una descrizione dei componenti dell'applicazione (attività, servizi, ricevitori, ecc.) e la gestione dei permessi che l'applicazione deve avere per interagire coi componenti dell'applicazione.

- *Java*: questa cartella contiene tutte le classi java che occorrono all'applicazione per funzionare. Android utilizza un linguaggio nativo java per la progettazione di applicazioni native. Man mano che si aggiunge del codice Javascript al codice sorgente vengono create nuove classe per implementarne le funzioni utilizzando i plugin di Cordova. Si può definire come il lavoro di incapsulamento della web app all'interno di una applicazione nativa.
- *Res*: questa cartella contiene tutti i riferimenti alle risorse necessarie all'applicazione. Le risorse possono essere immagini da inserire nell'applicazione o fav-icon, file binari o file xml. Qui troviamo il file `config.xml`, cioè il file di configurazione globale della nostra applicazione. Attraverso questo file possiamo gestire i plugin installati e le impostazioni della piattaforma utilizzata.
- *Assets*: in questa cartella si trovano tutti i file di codice scritti da noi durante l'implementazione della nostra applicazione. Al suo interno troviamo la cartella `www` nella quale risiede il nostro file `index.html`. La cartella `www` contiene a sua volta le seguenti cartelle:
  - *cordova-js-src*: contiene i file javascript relativi alla piattaforma, all'esecuzione dell'applicazione e alla comunicazione con la base di applicazione nativa sottostante.
  - *css*: contiene tutti i fogli di stile da importare all'interno del file html per definire lo stile delle pagine. La nostra applicazione dispone di un solo foglio di stile entro il quale sono definiti tutti gli stili ordinati per classe o identificativo, dividendoli in container, headers, paragrafi, ancore, immagini e molto altro.
  - *img*: contiene tutte le immagini inserite nelle schermate della nostra applicazione.
  - *js*: contiene tutti i file javascript per l'interattività della nostra applicazione.
  - *Plugins*: in questa cartella troviamo i file JavaScript dei plugin utilizzati nella nostra applicazione.
  - *Res*: contiene altre risorse necessarie al funzionamento dell'applicazione.
  - *Spec*: contiene librerie speciali aggiuntive. Nel nostro caso contiene la libreria Svg per il funzionamento della grafica della home page.

### 3.4.2 Plugin di Adobe Phonegap

- *Whitelist*: viene installato automaticamente durante la creazione di un progetto Phonegap. Tramite questo plugin si regolano le richieste alla rete dell'applicazione, cioè si può permettere o negare l'accesso al web.
- *Cordova Local-Notifications Plugin*: consente all'applicazione di lanciare notifiche push. Le notifiche della nostra applicazione riguardano dati ricevuti dal server come il livello di carica della batteria o la temperatura interna della macchina. Siccome non abbiamo ancora un server reale a nostra disposizione abbiamo simulato le notifiche in modo locale, semplicemente lanciandole dalla funzione di successo delle richieste Ajax.

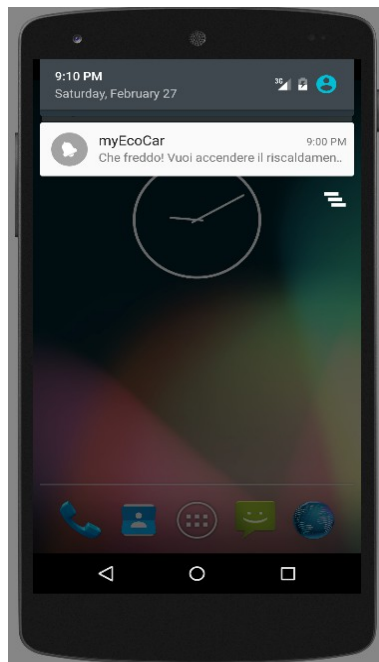
```

cordova.plugins.notification.local.schedule({
  id: 1,
  text: 'Temperatura interna bassa! Vuoi accendere il riscaldamento?',
  sound: null,
});

cordova.plugins.notification.local.on('click', function (notification) {
  $('#home_page').hide();
  $('#commands_page').show();
  loadComandi();
}, this);

```

**Figura 26: esempio codice notifica push**

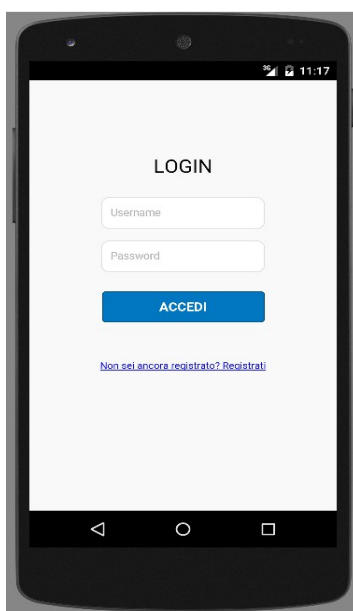


**Figura 27: schermata con notifica**

### 3.5 Le interfacce grafiche di myEcoCar

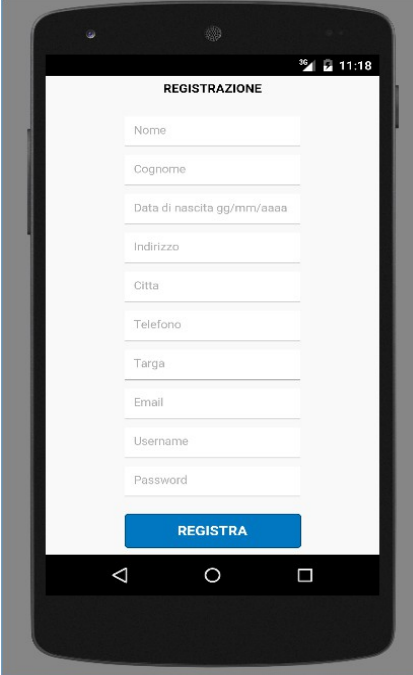
La visualizzazione dei contenuti di Phonegap è basata su un linguaggio di markup: il linguaggio HTML. Il file `index.html` della nostra applicazione, come in ogni app ibrida che si rispetti, contiene tutte le pagine presenti anziché dividerle per funzione come nel web per garantire una maggiore fluidità dell'applicazione. Phonegap e Cordova lavorano meglio con applicazioni single-page. Ogni pagina dell'app è contenuta nei tag `<div></div>` con l'apposito identificatore di pagina. Al suo interno la pagina è suddivisa tra due div “header” e “container”. La grandezza dell'header varia in base al contenuto da visualizzare all'interno di essa; un header grande viene utilizzato quando bisogna visualizzare informazioni importanti che colpiscano l'attenzione dell'utente tramite la parte grafica, mentre al container viene lasciato il compito di visualizzare informazioni più complesse o compiti riguardanti l'invio di informazioni ad database.

- *Login*: pagina di login utente. È composta da una form che accetta due input, uno di tipo testo e l'altro di tipo password, e da un input di submit della form. Tramite il click su “Accedi” i dati raccolti dalla form vengono confrontati con quelli presenti sul database. Se l'operazione va a buon fine viene eseguito il login, impostando in una sessione locale i dati e viene effettuato il redirect della pagina alla home.



**Figura 28: Login di myEcoCar**

- *Registrazione*: pagina di registrazione nuovo utente. È composta da una form che accetta in input di tipo testo i dati dell'utente (nome, cognome, data di nascita, indirizzo, città, targa, email, username, password) e da un input di submit della form. Tramite il click su “Registrazione” i dati raccolti dalla form vengono confrontati con quelli presenti sul database. Se l'operazione va a buon fine viene inserito un nuovo record nel database.



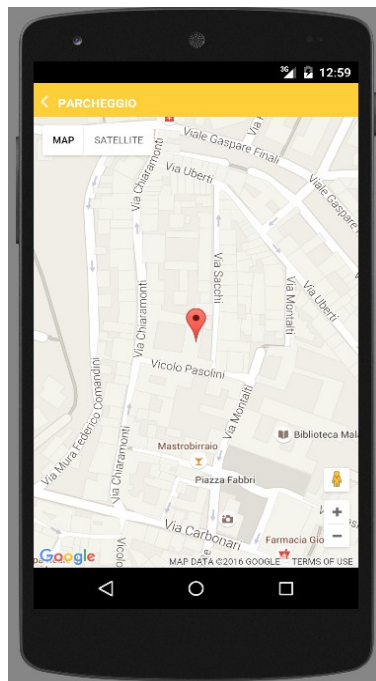
**Figura 29: interfaccia grafica Registrazione**

- *Home*: pagina principale della nostra applicazione. Attraverso l'interfaccia grafica possiamo vedere il livello di carica della batteria, le temperature interne ed esterne all'auto passate dal server e possiamo navigare nelle pagine attraverso il menù.



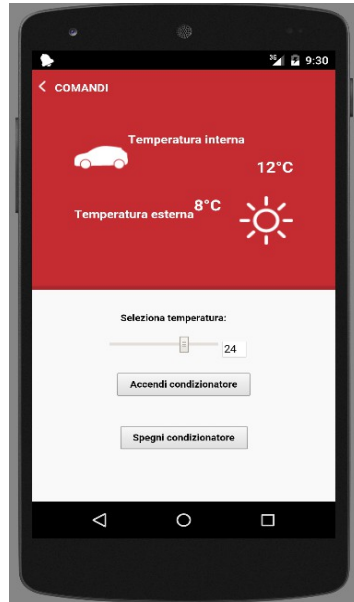
**Figura 30: home page di myEcoCar**

- *Parcheggio*: questa pagina contiene una mappa in cui viene localizzato dove è stata parcheggiata la propria auto elettrica. Le coordinate vengono passate dal server e mostrate grazie alle API di Google Maps.



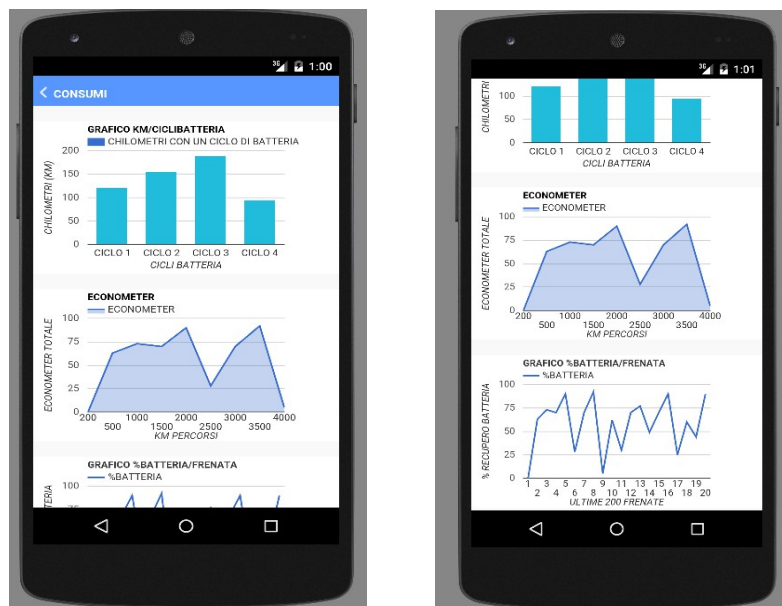
**Figura 31: interfaccia grafica Parcheggio**

- *Comandi*: in questa pagina è possibile visualizzare le temperature interne ed esterne dell'auto e selezionare una temperatura per l'accensione del condizionatore.



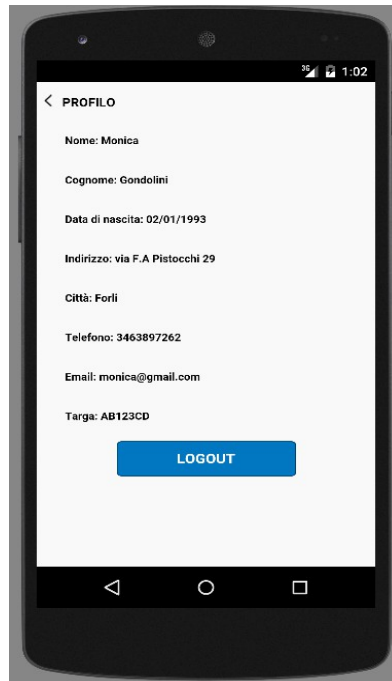
**Figura 32: pagina Comandi di myEcoCar**

- *Consumi*: questa pagina mostra tutti i grafici relativi alle prestazioni e ai consumi dell'auto. I dati vengono passati dal server per creare i grafici grazie alle API di Google Charts. I grafici presenti riguardano la percentuale di batteria utilizzata per ogni frenata, l'Econometer e ciclibatteria per chilometro.



**Figura 33: interfaccia grafica Consumi**

- *Profilo*: in questa pagina vengono visualizzati tutti i dati utenti ed è possibile effettuare il logout. I dati vengono richiesti al server mentre il logout viene effettuato disconnettendo la sessione locale creata durante il login.



*Figura 34: interfaccia grafica pagina Profilo*

## 3.6 JavaScript di myEcoCar

### 3.6.1 ProgressBar.js API

Per visualizzare il livello di carica dell'auto ricevuto dal server abbiamo utilizzato la libreria ProgressBar.js, la quale offre barre di avanzamento animate graficamente semplici da implementare. Nei casi più complessi questa libreria fa uso delle funzionalità della libreria SVG. Per simulare la carica abbiamo scelto di aumentare il livello di carica ricevuto dal server ogni due secondi, in modo da vedere il progressivo avanzamento della circonferenza da noi scelta.

Per usufruire delle sue funzionalità, la libreria deve essere importata nel file di linguaggio di markup tramite il tag `<script>`.



```

var circle = new ProgressBar.Circle('#c-percent', {
  color: '#FFFFFF',
  trailColor: '#64b87b',
  strokeWidth: 3,
  trailWidth: 2,
  text: {
    value: '0',
    className: 'progressbar_label'
  },
  step: function(state, bar) {
    bar.setText((bar.value() * 100).toFixed(0)+'%');
  }
});

var interval = setInterval(function(){
  circle.animate(battery/100, function() {
    circle.animate(battery/100, function(){
      battery++;
    });
  });
}, 2000);

```

*Figura 35: Il nostro codice per la creazione della progressbar della nostra applicazione*

### 3.6.2 Google Maps JavaScript API

Nello sviluppo della nostra applicazione, come già accennato, sono state utilizzate le API di Google Maps per la visualizzazione su mappe del luogo in cui è stata parcheggiata la propria auto. Questa libreria offerta da google offre molteplici funzionalità, poiché copre qualsiasi aspetto importante nell'utilizzo delle mappe.

L'API fornisce agli sviluppatori un'ampia gamma di funzionalità, come la scelta del tipo di mappa (stradale, satellite), le rappresentazioni grafiche su mappa (segnaposto, finestre di informazioni, simboli, ecc.), servizi come la geolocalizzazione, ricerca di direzioni, elevazione e molto altro.

Per comprendere a fondo la libreria non solo è stata necessaria la lettura delle guide ufficiali ma anche la ricerca di tutorial da parte di altri utenti e la lettura di quesiti e risposte su forum per programmatori.

Per poter utilizzare la libreria è stato necessario prima di tutto creare una chiave (API key) personale indicando il tipo di progetto, nel nostro caso Android. La chiave ottenuta è stata inserita nel collegamento all'API di Google Maps per renderlo privato e non utilizzabile da terzi. Una volta creato il link contenente l'API key generata e importato nel progetto, si potranno sfruttare tutte le potenzialità delle API di Google

Maps. L'importazione nel progetto avviene tramite il tag `<script></script>`, nel quale il l'attributo "scr" conterrà il link alla libreria con la chiave personale.

```
<script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyB5D7Y5jU54y-zA8a6qASewTgcVHcJVyxg&callback=initMap"></script>
```

**Figura 36: Importazione della libreria Google Maps con la nostra API key**

Nel nostro progetto abbiamo utilizzato la classe JavaScript `LatLng` di Google Maps, la quale ci ha consentito di creare un contenitore contenente le coordina geografiche passate dall'auto al server. Viene poi creato un marcatore per visualizzare graficamente la posizione sulla mappa.

```
latlng = new google.maps.LatLng(latitude, longitude);
```

**Figura 37: esempio di utilizzo della classe `LatLng`**

```
function drawMap(latlng) {  
    var myOptions = {  
        zoom: 17,  
        center: latlng,  
    };  
  
    var map = new google.maps.Map(document.getElementById("mappa"), myOptions);  
    // Add an overlay to the map of current lat/lng  
    var marker = new google.maps.Marker({  
        position: latlng,  
        map: map,  
        title: "Parcheggio"  
    });  
}
```

**Figura 38: Funzione della nostra applicazione attraverso la quale viene visualizzata la mappa e inserito il marcatore**

### 3.6.3 Google Charts API

Questa libreria è stata utilizzata per la creazione dei grafici dei consumi all'interno della nostra applicazione. Google Charts è uno strumento che permette di raccogliere dati dal server e offrirne una visualizzazione sottoforma di grafico. Google offre una grande varietà di grafici, ordinati per tipologie ed è stata scelta per la sua completezza e semplicità di implementazione.

Tra i grafici presenti sono stati utilizzati per rappresentare al meglio i valori ricevuti dal nostro server i seguenti: "ColumnChart" per il grafico rappresentanti i

chilometri effettuati con un ciclo di batteria, “LineChart” per il grafico sulla percentuale di recupero della frenata e “Area Chart” per il grafico dell'econometer.

L'importazione della libreria avviene, come per le altre API elencate, tramite il tag `<script>` del file HTML e tramite l'inserimento di due linee di codice nel file JavaScript, le quali servono per definire la tipologia di grafico e caricare la funzione per disegnarlo.

```
<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>
```

**Figura 39: Importazione della libreria**

```
google.charts.load('current', {'packages':['corechart']});
google.charts.setOnLoadCallback(drawChart);
```

**Figura 40: caricamento del tipo di grafico**

### 3.6.4 Collegamenti tra pagine

Abbiamo già accennato che le applicazioni ibride sono dette “single-page” poiché consistono in un'unica pagina HTML. La divisione interna in “pagine” del nostro file HTML è già stata spiegata precedentemente.

Normalmente in un'applicazione web composta da più pagine, il reindirizzamento tra pagine diverse avviene tramite i link alle pagine. Nel nostro caso abbiamo dovuto applicare un altro sistema: prima di tutto vengono assegnati degli identificatori ai div contenenti le pagine della nostra applicazione, tramite il foglio di stile viene impostata la visualizzazione di quest'ultimi a “display:none”, vale a dire invisibili. Il tag `<a>` per i collegamenti tra pagine cambia la sua sintassi da `<a href="pagina.html">`, viene inserito un identificatore e cambiato il contenuto dell'href con il link alla pagina stessa per mantenere il collegamento cliccabile, poiché il vero indirizzamento viene effettuato tramite javascript: `<a id="idpagina" href="#">`. Il comportamento che ogni pagina deve assumere quando viene richiamata è racchiuso in delle funzioni JavaScript che ne caricano il contenuto.

Il file JavaScript si occupa dell'indirizzamento delle pagine nel seguente modo:

1. mostrare il div di pagina attuale tramite la funzione `.show()`
2. richiamare una funzione sul click del link, la quale nasconde il div attuale tramite `.hide()` e mostra il div di pagina richiamato
3. chiamare le funzioni di caricamento della pagina richiamata

```
onDeviceReady: function() {
    app.receivedEvent('deviceready');

    $("#login_page").show();
    loadLogin();

    $("#signup_link").click(function(){
        $("#login_page").hide();
        $("#signup_page").show();
        loadSignUp();
    });
}
```

*Figura 41: esempio di reindirizzamento di pagine tramite jQuery*

### 3.6.5 Connessione al database tramite richieste Ajax

Nel secondo capitolo abbiamo introdotto Ajax, una tecnica di sviluppo software che permette lo scambio di dati tra server e web browser rendendo gli elementi della pagina dinamici. Il compito del codice Ajax all'interno della nostra applicazione è la comunicazione con i file php risidenti sul server per richiedere dati o per inserirne di nuovi nel database.

All'interno delle richieste Ajax relative alla nostra applicazione vengono definiti i seguenti parametri:

- *type*: specifica il tipo della richiesta “GET” o “POST”
- *url*: contiene il collegamento url al server. Siccome nel nostro caso ci colleghiamo ad un server simulato il collegamento è `http://localhost:80/myproject/file.php` nel caso di esecuzione sul browser, nel caso di emulazione su android abbiamo utilizzato l'indirizzo IP `10.0.2.2:80` al posto di localhost.
- *data*: contiene eventuali dati da passare al server
- *dataType*: tipo di dato per la codifica della richiesta
- *success*: contiene una funzione con gli aggiornamenti da eseguire in caso di successo della richiesta.

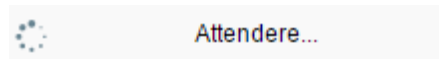
```
$.ajax({
  type: 'POST',
  url: 'http://localhost:80/myecocar/comandi.php',
  dataType: 'json',
  success: function(data){
    console.log(data);
    tempIn = data.tempIn.tmp1[0];
    tempOut = data.tempOut.tmp2[0];
    $("#tmpIn").text(tempIn+'°C');
    $("#tmpOut").text(tempOut+'°C');
  }
});
```

**Figura 42: Esempio di richiesta Ajax nella nostra applicazione**

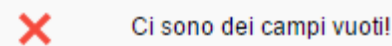
### 3.6.6 Funzione responseMessage()

Nelle realizzazione delle pagine di Login, Registrazione e Comandi della nostra applicazione è stato necessario implementare una funzione che mostri nell'interfaccia grafica il risultato dell'operazione compiuta sul server. Trattandosi per lo più di inserimenti o di confronto di dati ricevuti da una form con quelli già presenti nel database, non sempre le richieste potrebbero andare a buon fine ed è quindi stato opportuno effettuare un controllo sugli input per l'inserimento di informazioni corrette.

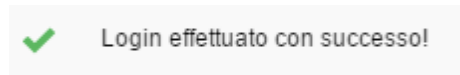
La funzione responseMessage() viene richiamata all'interno della funzione di successo della richiesta Ajax al server. Nella funzione di successo inoltre si imposta la classe del messaggio che verrà visualizzato all'interno del tag <p> nel file HTML a 'response-waiting' (Figura 40) e vengono controllati i dati ricevuti dal server. Se il messaggio ricevuto è un messaggio di successo, allora verrà chiamata la funzione responseMessage() che cambierà la classe attuale con la classe 'response-success' (Figura 41) e mostrerà il messaggio si operazione effettuata, in caso contrario verrà cambiata con la classe 'response-error' (Figura 42) e mostrerà un messaggio di errore.



**Figura 43: Messaggio di pausa**



**Figura 44: Messaggi di errore pagina di Login**



**Figura 45: Messaggio di successo, pagina di Login**

```
function responseMessage(responseMsg, responseData, changeClass){
    var responseMsg = responseMsg;
    changeClass = changeClass;
    responseData = responseData;

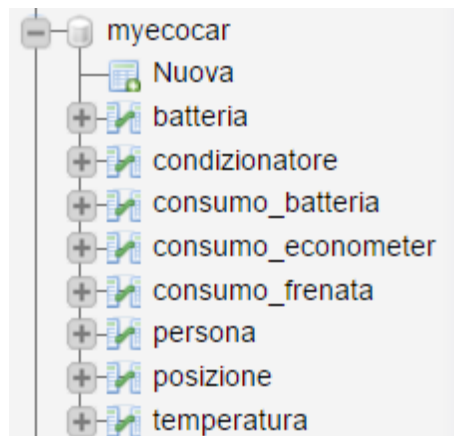
    responseMsg.fadeOut(200, function(){
        $(this).removeClass('response-waiting')
        .addClass(changeClass)
        .text(responseData.message)
        .fadeIn(200, function(){
            setTimeout(function(){
                responseMsg.fadeOut(200, function(){
                    $(this).removeClass(changeClass);
                });
            }, 3000);
        });
    });
}
```

**Figura 46: Funzione responseMessage()**

## 3.7 WampServer per la simulazione database

### 3.7.1 PhpMyAdmin

PhpMyAdmin è uno strumento software trovabile all'interno di WampServer il quale permette la visualizzazione del database e la semplificazione delle operazioni sulle tabelle, o l'importazione ed esportazione dei dati grazie all'interfaccia grafica. Nella seguente Figura viene visualizzata la struttura del database relativo all'applicazione. Il nome del database è “myecocar”, contenente le tabelle elencate come si può vedere in Figura 38.



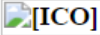
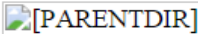










*Figura 47: Struttura database “myecocar”*

Ogni tabella è formata da un “id” come chiave primaria, contenente un numero incrementato automaticamente ad ogni nuovo inserimento e celle contenenti i dati di cui avrà bisogno l'applicazione.

### 3.7.2 Organizzazione delle pagine server-side

Le pagine server-side sono localizzate all'interno della cartella MyProjects su WampServer, contenute a loro volta in una cartella avente il nome del progetto. In tale modo vengono rese disponibili Online. Attraverso il codice server-side è possibile richiedere i dati alle tabelle tramite query, per poi essere passate al JavaScript client-side effettuante le richieste attraverso Ajax. I dati vengono passati sottoforma di JSON array.

## Index of /myEcoCar

	Name	Last modified	Size	Description
	<a href="#">Parent Directory</a>		-	
	<a href="#">batteria.php</a>	2016-01-30 15:44	754	
	<a href="#">comandi.php</a>	2016-02-01 15:34	1.1K	
	<a href="#">condizionatore.php</a>	2016-02-02 15:04	1.0K	
	<a href="#">connect.php</a>	2015-12-07 18:01	514	
	<a href="#">location.php</a>	2015-12-29 11:21	1.1K	
	<a href="#">login.php</a>	2016-02-02 16:39	1.2K	
	<a href="#">logout.php</a>	2016-02-01 13:45	172	
	<a href="#">longitude.php</a>	2015-12-21 17:36	1.0K	
	<a href="#">profilo.php</a>	2016-02-01 16:56	1.1K	
	<a href="#">registrazione.php</a>	2016-02-11 17:12	2.3K	

Apache/2.4.9 (Win64) PHP/5.5.12 Server at localhost Port 80

*Figura 48: rappresentazione file di progetto su phpMyAdmin*

### 3.7.3 Query per inserimenti e richieste al database

- *Login*: Seleziona i dati dalla tabella persona dove l'username e la password corrispondono a quelli inseriti del database

```
$query = "SELECT *
FROM persona
WHERE (username='".$username.'" AND password='".$password.'" )
LIMIT 1";
```

- *Registrazione*: inserisce tutti i dati ricavati dalla form nella tabella “persona”

```
$query = "INSERT INTO persona (nome, cognome, data_nascita, indirizzo, citta,
telefono, targa, email, username, password)
VALUES ('$nome', '$cognome', '$data_nascita', '$indirizzo', '$citta',
'$telefono', '$targa', '$email', '$username', '$password')";
```

- *Estrazione livello batteria*: seleziona dalla tabella “batteria” l'ultimo livello di batteria inserito nel database

```
$query = "SELECT BatteryLevel FROM batteria ORDER BY id DESC LIMIT 1";
```



- *Estrazione temperature:* vengono estratte dalla tabella “temperatura” l'ultima temperatura interna e l'ultima temperatura esterna inserite nel database.

```
$query1 = "SELECT t_interna
           FROM temperatura
           ORDER BY id
           DESC LIMIT 1";
$query2 = "SELECT t_esterna
           FROM temperatura
           ORDER BY id
           DESC LIMIT 1";
```

- *Localizzazione:*

```
$query1 = "SELECT latitudine FROM posizione ORDER BY id DESC LIMIT 1";
$query2 = "SELECT longitudine FROM posizione ORDER BY id DESC LIMIT 1";
```

- *Azionamento condizionatore:*

```
$query = "INSERT INTO condizionatore (tmpCond)
          VALUES ('$tmpCond')";
```

- *Dati consumi:*

- *Econometer:*

```
$query = "SELECT valore FROM consumo_econometer";
```

- *Percentuale di batteria per ogni frenata:*

```
$query = "SELECT valore FROM consumo_frenata";
```

- *Chilometri e cicli di batteria:*

```
$query = "SELECT valore FROM consumo_batteria";
```

- *Visualizzazione dati utente:* seleziona tutti i dati dalla tabella “persona” dove il nome utente corrisponde al nome utente contenuto nella sessione locale

```
$query = "SELECT *
          FROM persona
          WHERE (username = '".$_POST['username'].'')";
```

### 3.7.4 Codifica JSON

La codifica JSON serve per lo scambio di dati tra client e server. Le richieste Ajax della nostra applicazione richiedono infatti questo tipo di dato in risposta.

Nel lato server di myEcocCar i risultati delle query di interrogazione al database vengono controllati e codificati opportunamente per essere passati al lato client dell'applicazione. La codifica avviene prima di tutto controllando se la query è stata effettuata, cioè se un certo numero di righe sono state caricate nella tabella. Vengono poi caricate le righe di dati del risultato all'interno di un array, il quale verrà codificato come JSON.

```
$result = mysqli_query($conn, $query);
$return = array();
if(mysqli_num_rows($result)){
    while ($row=mysqli_fetch_row($result)){
        $return['level'][] = $row;
    }
}else{
    $return['level'][] = "Errore query";
}
echo json_encode($return);
```

*Figura 49: Esempio di codifica JSON in PHP della nostra applicazione*

L'array codificato ottenuto, come precedentemente spiegato, viene dato in risposta alla richiesta Ajax lato client, la quale ne ricaverà i dati per renderli disponibili all'utilizzo. Un esempio di struttura di un JSON array è il seguente:

```
{username: "mario", password: "mario", success: 1, message: "Login effettuato con successo!"}
message: "Login effettuato con successo!"
password: "mario"
success: 1
username: "mario"
```

*Figura 50: Esempio di struttura di JSON array riguardante il Login di un utente nella nostra applicazione.*

# Conclusioni

---

In questo progetto di tesi è stato introdotto il tema della mobilità elettrica. In particolare sono state descritte le caratteristiche principali delle auto elettriche come tipologia di EV. Sono stati confrontati i motori a propulsione elettrica con quelli a combustibile, traendone conclusioni sul rendimento e sull'impatto ambientale di queste due tecnologie.

Sono state affrontate le tematiche relative alle caratteristiche delle batterie e ai metodi di ricarica delle auto elettriche. Inoltre, sono stati elencati vantaggi e limitazioni della mobilità elettrica con relativi consigli per un miglioramento delle prestazioni della propria auto consigliando uno stile di guida ecologico, fornendo una panoramica del concetto di “Ecodriving” e delle sue regole principali.

Inoltre, sono state nominati metodi già esistenti per dispositivi mobili come aiuto alla risoluzione della “range anxiety”, evidenziando il ruolo importante del monitoraggio per via elettronica tramite smartphone e tablet.

È stata data una panoramica del punto di vista teorico di tutte le tecnologie utilizzare durante lo sviluppo del progetto per poter implementare al meglio l'applicazione di supporto alla mobilità elettrica “myEcoCar” e simularne i comportamenti. Oltre alle tecnologie è stato dimostrato come stia diventando sempre più importante il ruolo di smartphone e tablet in ambito automobilistico, come aiuto alla gestione della propria auto a distanza.

Successivamente è stata affrontata la tematica di applicazione ibrida, quale myEcoCar, illustrandone le particolarità e le differenze, comprendendo i vantaggi e gli svantaggi rispetto allo sviluppo di applicazioni con le altre due tipologie esistenti, ovvero le applicazioni web e le applicazioni native.

Inoltre è stato spiegato il concetto di simulazione del server per il funzionamento dell'applicazione, necessariamente introdotto poiché non sono ancora a disposizione dati reali su cui lavorare.

La nostra applicazione myEcoCar attualmente non si può definire come una versione definitiva, bensì come una prima versione funzionante e pronta a modifiche e a integrazioni con nuove funzionalità che la rendano affidabile e completa sotto tutti gli aspetti.

In futuro potrebbero essere introdotte le seguenti funzionalità o miglioramenti:

- Funzionamento offline: allo stato attuale la nostra applicazione funziona solo con collegamento alla rete internet. In futuro potrebbero essere introdotte delle funzionalità offline, ad esempio:
  - un sistema di gestione delle informazioni riguardanti l'assicurazione e il bollo della propria auto;
  - annotazioni sulle ultime revisioni o su eventuali problemi meccanici da risolvere e reminder degli appuntamenti;
  - salvataggio di file multimediali come foto e video della propria auto;
- Al passo con la gamification: possono essere introdotte nuove caratteristiche corrispondenti a quelle implementate nell'applicazione di gamification presente nel cruscotto dell'auto elettrica per tenere sempre d'occhio gli obiettivi raggiunti;
- Miglioramento dell'interfaccia utente: rendendola facilmente accessibile a qualsiasi persona tramite modifiche all'interfaccia grafica e all'interattività utente, in modo da rendere i gesti più fluidi;
- Utilizzo al meglio delle API di Google Maps: come altre applicazioni già esistenti, potrebbero essere introdotte mappe per la localizzazione delle colonnine di ricarica sul proprio territorio e l'inserimento di nuove. Un'altra funzionalità potrebbe essere l'inserimento di un navigatore per il raggiungimento del parcheggio dalla posizione attuale;
- Annotazione dei tempi di ricarica con i relativi costi dell'energia erogata al motore;
- Progettazione dell'applicazione per altri sistemi operativi: myEcoCar attualmente è stata progettata per smartphone o tablet con sistema operativo

Android per la possibilità di testare l'applicazione su più dispositivi ma in futuro si può pensare allo sviluppo di una versione per iOS e Windows Phone.



# Bibliografia

- [WIK16a] Wikipedia, “Electric car”, [https://en.wikipedia.org/wiki/Electric\\_car](https://en.wikipedia.org/wiki/Electric_car)
- [WIK16b] Wikipedia, “Electric vehicle”  
[https://en.wikipedia.org/wiki/Electric\\_vehicle](https://en.wikipedia.org/wiki/Electric_vehicle)
- [WIK16c] Wikipedia, Normative sui connettori per il MODO 3 di ricarica (IEC 62196-1 e 2),  
[https://it.wikipedia.org/wiki/Colonnina\\_di\\_ricarica\\_per\\_veicoli\\_elettrici](https://it.wikipedia.org/wiki/Colonnina_di_ricarica_per_veicoli_elettrici)
- [WIK16d] Wikipedia, “Web Application”  
[https://en.wikipedia.org/wiki/Web\\_application](https://en.wikipedia.org/wiki/Web_application)
- [WIK16e] Wikipedia, “Apache Cordova, Design and Rationale”  
[https://en.wikipedia.org/wiki/Apache\\_Cordova](https://en.wikipedia.org/wiki/Apache_Cordova)
- [WIK16f] Wikipedia, “HTML” <https://it.wikipedia.org/wiki/HTML>
- [WIK16g] Wikipedia “HTML5” <https://it.wikipedia.org/wiki/HTML5>
- [WIK16h] “Cascading Style Sheet”  
[https://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [WIK16i] “JavaScript” <https://en.wikipedia.org/wiki/JavaScript>
- [WIK16j] “PHP” <https://it.wikipedia.org/wiki/PHP>
- [WIM16] WikimediaFigura 17, “HTML5 related API” By Mercury999 - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=36352535>
- [GMH16] GM, Detailed GM History, “History of the Automobile”  
<http://www.gm.ca/gm/english/corporate/about/ourhistory/detail>
- [DTE11] Ralph DTE, “Che cos'è un freno rigenerativo?” <http://www.ralph-dte.eu/2011/05/10/che-cose-un-freno-rigenerativo/>
- [SCA16d] SCAME, “Norme sui metodi di ricarica”,  
[http://www.scame.com/it/infopoint/newmobility/focus\\_normativo\\_internazionale.asp](http://www.scame.com/it/infopoint/newmobility/focus_normativo_internazionale.asp)
- [SCH09] Schott Blogs, “Range Anxiety”, 15 Gennaio 2009  
[http://schott.blogs.nytimes.com/2009/01/15/range-anxiety/?\\_r=0](http://schott.blogs.nytimes.com/2009/01/15/range-anxiety/?_r=0)
- [HTM06b] HTML.it “Che cos'è un linguaggio di markup”  
<http://www.html.it/faq/che-cose-un-linguaggio-di-markup/> 3 Aprile 2006
- [SCA16a] “I vantaggi della mobilità elettrica”  
[http://www.scame.com/it/infopoint/newmobility/vantaggi\\_mobilita\\_elettrica.asp](http://www.scame.com/it/infopoint/newmobility/vantaggi_mobilita_elettrica.asp)

- [SCA16b] SCAME, “Benefici della mobilità elettrica”  
<http://www.scame.com/it/infopoint/newmobility/benefici.asp>
- [SCA16c] SCAME, “Infrastrutture di ricarica”  
[http://www.scame.com/it/infopoint/newmobility/infrastrutture\\_ricarica.asp](http://www.scame.com/it/infopoint/newmobility/infrastrutture_ricarica.asp)
- [ELE16a] Electricmobility.it, “L'automobile elettrica”  
<http://www.electricmobility.it/index.php/en/speciali/lautomobile-elettrica?showall=1>
- [ELE16b] Electricmobility.it, “Las Vegas. Con myFord Mobile la gestione dell'auto elettrica possibile via smartphone”  
<http://www.electricmobility.it/index.php/it/tecnologie/altre/235-las-vegas-con-myford-mobile-la-gestione-dellauto-elettrica-possibile-via-smartphone>
- [ALB13] Albanesi.it, “L'auto elettrica”  
[http://www.albanesi.it/inchieste/auto\\_elettrica.htm](http://www.albanesi.it/inchieste/auto_elettrica.htm)
- [ENE16] EnergoClub, “Progetto svolta elettrica”  
<http://www.energoclub.org/page/progetto-svolta-elettrica>
- [GAA16] Gruppo Acquisto Auto, “I vantaggi dell'auto elettrica”  
<http://www.gruppoacquistoauto.it/i-vantaggi-dell-auto-elettrica/>
- [IDG16] Ideegreen, “Batterie auto elettriche, impariamo a conoscerle”  
<http://www.ideegreen.it/batterie-auto-elettriche-37724.html>
- [ECD16a] Ecodrive, “Ecodriving” <http://www.ecodrive.org/en/home/>
- [ECD16b] Ecodrive, “Golden rules of Ecodriving”  
[http://www.ecodrive.org/en/what\\_is\\_ecodriving/the\\_golden\\_rules\\_of\\_ecodriving/](http://www.ecodrive.org/en/what_is_ecodriving/the_golden_rules_of_ecodriving/)
- [ECD16c] “Silver rules of Eco-driving”  
[http://www.ecodrive.org/en/what\\_is\\_ecodriving/the\\_silver\\_rules\\_of\\_ecodriving/](http://www.ecodrive.org/en/what_is_ecodriving/the_silver_rules_of_ecodriving/)
- [ASS13] Assicurazione auto, “Le migliori app per l'auto”  
<http://www.assicurazioneauto.it/le-migliori-app-per-lauto/>
- [COR13] Corriere della sera, sezione Tecnologia, “Le migliori app per l'automobile”  
[http://motori.corriere.it/tecnologia/13\\_dicembre\\_20/app-automobile-migliori-685d3aaa-698e-11e3-95c3-b5f040bb6318.shtml](http://motori.corriere.it/tecnologia/13_dicembre_20/app-automobile-migliori-685d3aaa-698e-11e3-95c3-b5f040bb6318.shtml), 23 Dicembre 2013
- [GRE16] iTunes, “GreenMeter”  
<https://itunes.apple.com/us/app/greenmeter/id289973918?mt=8>
- [FCR13] R. Frank, G. Castignani, R. Schmitz e T. Engel “A novel eco-driving application to reduce energy consumption of electric vehicles”*In Connected Vehicles and Expo (ICCVE), 2013 International Conference on. IEEE*, pp. 283-288, 2013.
- [ENE16] Enel Drive, “Enel Drive” <https://www.eneldrive.it/>
- [COL16] Colonnine Elettriche, “Colonnine elettriche”  
<http://www.colonnineelettriche.it/>
- [PLS16] Recargo Inc. “PlugShare” <https://play.google.com/store/apps/details?id=com.xatori.Plugshare&hl=it>
- [NEC16] Go Electric Stations “Next Charge colonnine EV”  
<https://play.google.com/store/apps/details?id=com.goelectricstations.nextcharge>



- [CP00] ChargePoint “ChargePoint: Find EV Charging”  
<https://play.google.com/store/apps/details?id=com.coulombtech>
- [PLS16] PlugSurfing “PlugSurfing - Chargin stations”  
<https://play.google.com/store/apps/details?id=com.xitaso.plugsurfing&hl=it>
- [A2A16] A2A “E-moving” <http://www.e-moving.it/home/cms/emv/progetto/app.html>
- [BMW16] BMW “BMW iRemote” <https://play.google.com/store/apps/details?id=com.bmwi.remote>
- [FOR16] Ford Motor Co. “myFord Mobile”  
<https://play.google.com/store/apps/details?id=com.ford.mfm&hl=it>
- [SAA16] SAABRE “ChargeMap” [https://play.google.com/store/apps/details?id=com.chargemap\\_beta.android](https://play.google.com/store/apps/details?id=com.chargemap_beta.android)
- [HTM13] HTML.IT, “Introduzione ad Apache Cordova”  
<http://www.html.it/pag/42121/introduzione-ad-apache-cordova/>, 26 Giugno 2013
- [HTM06a] HTML.IT, “HTML e i browser”  
<http://www.html.it/pag/16026/introduzione22/>, 17 Marzo 2006
- [HTM11] HTML.IT, “Che cos'è HTML5” <http://www.html.it/pag/19263/dadhtml-4-ad-html5/>, 28 febbraio 2011
- [HTM14] HTML.IT, “Introduzione a JavaScript”  
<http://www.html.it/pag/45343/introduzione-a-javascript/>, 28 Gennaio 2014
- [HTM09] HTML.IT, “Introduzione a jQuery”  
<http://www.html.it/pag/18382/introduzione63/>, 29 Giugno 2009
- [HTM15] HTML.IT, “Introduzione a PHP” <http://www.html.it/pag/16673/cosphp/>, 20 Marzo 2015
- [APC16] Apache Cordova, “Overview – Apache Cordova”  
<http://cordova.apache.org/docs/en/latest/guide/overview/>
- [TEL15] Telerik Developer Network, “What is a hybrid mobile app?”  
<http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>, 25 Marzo 2015
- [ION15] Ionic Blog, “The Last Word on Cordova and PhoneGap”,  
<http://blog.ionic.io/what-is-cordova-phonegap/>, 6 Marzo 2014
- [PHG12] Phonegap.com, “Phonegap Cordova and what's in a name”  
<http://phonegap.com/2012/03/19/phonegap-cordova-and-what's-in-a-name/>, 12 Marzo 2012
- [BPH16] Build Phonegap, “Adobe PhoneGap Build”  
<https://build.phonegap.com/>
- [W3C16a] W3C Schools, “Ajax Tutorial” <http://www.w3schools.com/ajax/>“
- [W3C16b] W3C Schools, “jQuery ajax() Method”  
[http://www.w3schools.com/jquery/ajax\\_ajax.asp](http://www.w3schools.com/jquery/ajax_ajax.asp)
- [W3C16c] W3C Schools, “PHP 5 Introduction”  
[http://www.w3schools.com/php/php\\_intro.asp](http://www.w3schools.com/php/php_intro.asp)