ASSE I – RICERCA, INNOVAZIONE E SVILUPPO TECNOLOGICO del PO FESR 2014-2020- Azione 1B.1.2.1

PROGETTO MOBAS 4.0

Mobilità s**O**stenibile in **BAS**ilicata **4.0**

Work Package 5

"MOBILITA' SOSTENIBILE PER IL TRASPORTO PUBBLICO URBANO"

Deliverable 5.1

"Progettazione di un'area di prova per la sperimentazione delle tecnologie Automotive del WP"

Stato di avanzamento n. 2 dal 01/01/2023 al 31/12/2023

Data	***Redazione a cura di:***	***Persona di contatto per il progetto:***
30/11/2023	Consorzio TRAIN *Altri partner: UNIBAS, ENEA*	Mario Zagaria COM SCPA E-mail: mario.zagaria@com-scpa.it telefono: 0972 460130

Introduzione

Il presente documento è finalizzato alla presentazione delle soluzioni sviluppate nell'ambito della mobilità sostenibile per il trasporto urbano e, più precisamente, degli applicativi sviluppati nell'ambito di un'area di prova per la sperimentazione delle tecnologie Automotive del Work Package 5.

Scenari di simulazione di servizio di bus a prenotazione

Il servizio DRT si basa su un modello di pianificazione dei percorsi e dei tempi di passaggio in diverse fermate in base alle richieste pervenute al sistema in modo da ottimizzare il servizio per gli utenti e i costi di gestione. In generale, è considerato un servizio intermedio tra servizio pubblico e taxi e il problema di ottimizzazione ad esso collegato è riconducibile a situazioni ed algoritmi noti di ricerca operativa (VRP).

Le caratteristiche principali di un DRT sono:

- flotta di veicoli di piccola dimensione
- rotte flessibili nel tempo e nello spazio
- piattaforma di prenotazione delle corse

Il sistema DRT preso in considerazione per la simulazione nell'ambito del progetto MOBAS sarà di tipo statico: si assume cioè che tutte le richieste da servire siano conosciute prima della pianificazione (acquisite tramite un'apposita applicazione entro un orario prestabilito che precede quello in cui è richiesto il servizio) e avrà un modello del tipo "linee fisse a prenotazione", con percorsi definiti ma corse effettuate solo in presenza di prenotazioni, e di tipo "molti a molti", che offre la flessibilità più completa sia in origine che in destinazione.Le richieste dovranno quindi essere caratterizzate da un'origine e una destinazione e da un orario desiderato di partenza e/o di arrivo (a meno di una finestra di tolleranza da concordare).

Il progetto MOBAS 4.0 prevede la sperimentazione di uno scenario simulato di servizio di bus a prenotazione prendendo come riferimento il Centro Ricerche Trisaia di ENEA, sia per quel che riguarda la rete interna di strade e edifici collegati, sia per l'utenza del servizio di prenotazione.

Per lo scenario di simulazione ipotizzato sul sito del Centro Ricerche Trisaia di ENEA sono state assunte alcune ulteriori ipotesi:

- la flotta in servizio è costituita da un singolo veicolo
- il veicolo è elettrico (con eventuale vincolo sulla ricarica delle batterie)
- è specificata una località di partenza/arrivo/deposito per il servizio "navetta" (abilitata come base di ricarica per il bus)
- le stazioni raggiunte dal servizio sono in numero prestabilito di 6 (5 FERMATE + 1 CAPOLINEA, base di partenza e di ricarica)
- le varie rotte di collegamento tra le località sono prestabilite
- si valuta la possibilità di definire vincoli sulla capacità del veicolo

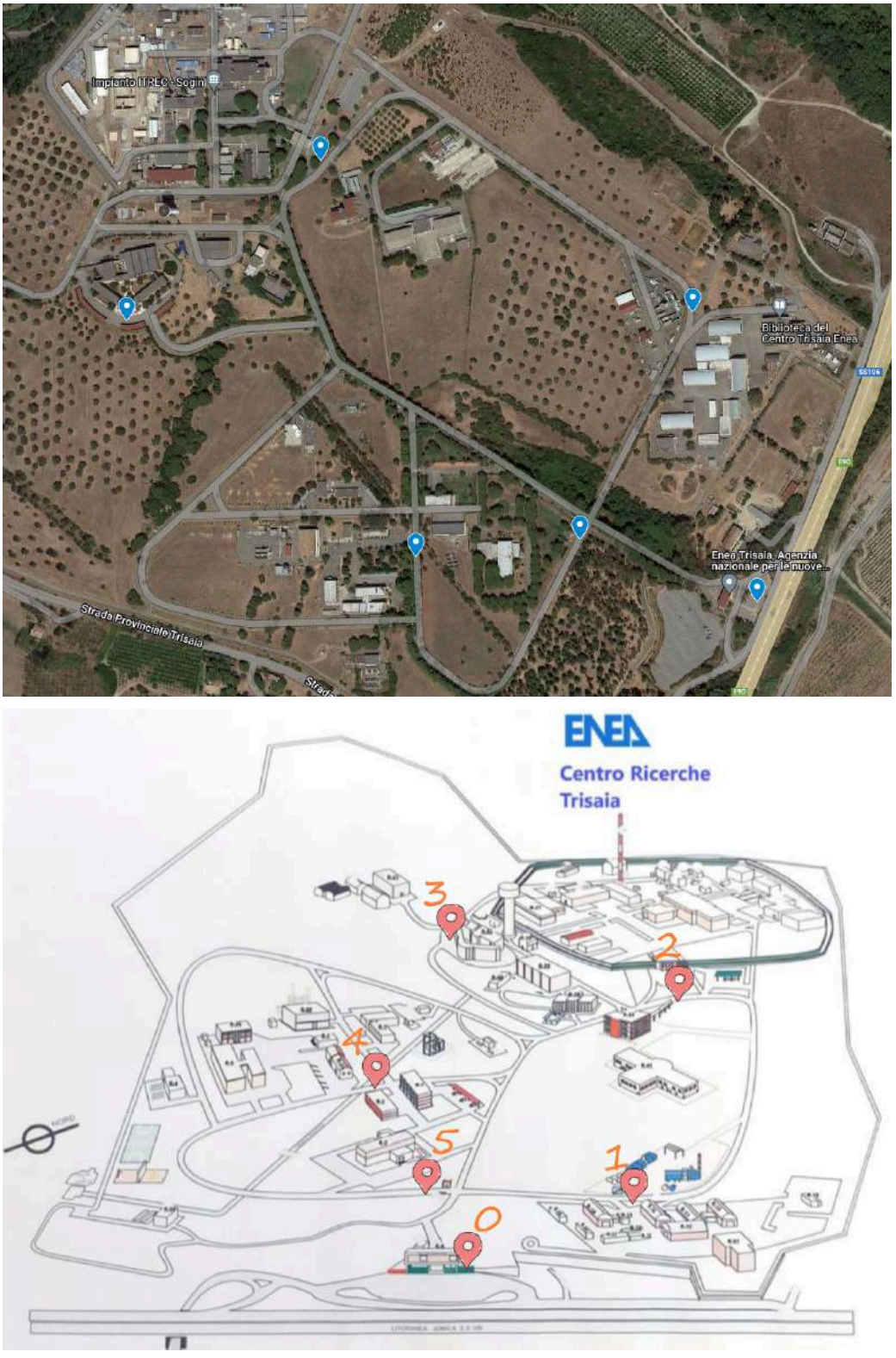

Fig.1 - Il sito ENEA di Trisaia e la rete del servizio

Caratteristiche generali del Vehicle Routing Problem

Il VRP è stato introdotto per la prima volta da Dantzig e Ramser nel 1959 [1], con lo scopo di trovare il percorso ottimale di veicoli omogenei che servono un gruppo di stazioni di servizio soddisfacendo la domanda di petrolio richiesta da ciascuna stazione. Nel 1964 Clarke e Wright [2] hanno generalizzato questo problema definendolo come un problema di ottimizzazione lineare che si presenta spesso nell'ambito della logistica e dei trasporti, ovvero come servire un insieme di clienti, geograficamente dispersi attorno ad un deposito centrale, utilizzando un insieme di veicoli di differente portata. Il classico VRP è definito da un deposito da cui partono e ritornano i veicoli e un insieme di clienti che devono essere serviti da uno solo dei veicoli. Ci sono molte specializzazioni possibili del VRP, ma in generale è sempre formato da sei elementi principali: rete stradale, clienti, depositi, veicoli, vincoli e obiettivo operazionale.

La rete stradale è descritta tramite un grafo pesato, dove i nodi rappresentano i depositi e i clienti, mentre gli archi rappresentano il percorso che li unisce. Gli archi possono essere diretti o indiretti in base al senso di marcia possibile nella strada rappresentata. Ogni arco ha assegnato un peso che può rappresentare la lunghezza, il costo, o un valore rappresentativo di entrambi.

I clienti sono rappresentati da un nodo del grafo. Essi sono caratterizzati dalla domanda, che varia dal tipo di problema rappresentato, essa può essere di carico o scarico, o entrambi. Il cliente può richiedere una finestra di tempo in cui devono avvenire le operazioni di carico o scarico.

Nel classico VRP c'è un unico deposito da cui partono i veicoli e a cui devono ritornare. I veicoli possono trasportare merce dal deposito ai clienti o possono caricare merce in un cliente e scaricarla in un altro cliente.

La flotta dei veicoli può essere omogenea o formata da diversi tipi di veicoli. I veicoli sono descritti dalla capacità di carico, dal costo di utilizzo e dalla durata di consegna.

I vincoli esterni possono essere di molti tipi, dipendono dal tipo di problema che si vuole rappresentare. Essi possono essere, ad esempio: tutti i clienti devono essere serviti, la domanda di ogni cliente può essere completata solo da un veicolo, la distanza o il tempo di percorrenza totale di ogni veicolo deve essere minore di una certa soglia, la somma delle domande dei clienti serviti da un veicolo non può essere maggiore della capacità di carico del veicolo.

L'obiettivo operazionale dipende dal problema rappresentato, in generale può essere mono obiettivo o multi-obiettivo. Gli obiettivi singoli classici sono: minimizzare la distanza totale o il tempo totale, minimizzare il numero di veicoli utilizzati, minimizzare il costo totale. Una combinazione di questi obiettivi può rappresentare più accuratamente una situazione reale che si vuole ottimizzare.

Specializzando queste parti fondamentali del VRP si ottengono estensioni del problema.

Per esempio, nel Capacitated Vehicle Routing Problem (CVRP) la caratteristica principale è il limite di capacità di carico dei veicoli. In questo problema tutti i clienti hanno domande di carico o scarico, le richieste sono note prima della partenza e non sono frazionabili. I veicoli sono tutti uguali, hanno sede in un unico deposito e vengono imposti dei limiti sulle loro capacità. L'obiettivo è quello di minimizzare il costo totale per servire tutti i clienti. Il CVRP è un problema NP-hard, infatti è una generalizzazione del noto Traveling Salesman Problem (TSP) che richiede la determinazione di un circuito semplice di costo minimo che visiti tutti i vertici del grafo, ossia un circuito hamiltoniano.

Varianti del VRP, molto utili per la rappresentazione di istanze reali, sono il VRP with Time Windows (VRPTW) e la VRP with Pickup and Delivery (VRPPD).

Il primo è l'estensione del CVRP in cui vengono imposti vincoli di capacità e ad ogni cliente è associato un intervallo di tempo, chiamato finestra temporale. Nel secondo caso, ogni cliente può richiedere un'operazione di scarico, di carico o entrambe.

Entrambi sono problemi NP-hard, poiché possono essere considerati delle generalizzazioni del CVRP.

Il Multi-Trip Vehicle Routing Problem considera invece un VRP classico in cui i veicoli possono eseguire più viaggi.

Nel nostro caso, Il problema considerato e definito corrisponde ad un VRP con Time Windows, Pickup and Delivery di tipo Multi-Trip e con stazioni di sosta opzionali. Inoltre il problema è anche Time Constrained e Capacity Constrained.

Evidentemente la complessità computazionale di questo problema è non inferiore a quelle di tutte le classi di problemi di cui esso è una generalizzazione, ossia il VRP-TW, VRP-PD, lo stesso capacitated VRP e in ultimo il problema del Traveling Salesman (TSP). Tutti questi problemi appartengono alla classe dei problemi NP-hard in senso forte. Per tali problemi quindi non è possibile definire un algoritmo di tipo pseudo-polinomiale anche limitando la massima dimensione dei dati di input per le sue istanze. In termini pratici, essere NP-hard significa non avere la possibilità di determinare la soluzione ottimale per le istanze del problema in tempi ragionevoli, a meno che queste non siano di dimensioni molto ridotte.

La conoscenza del tipo di complessità per un problema è fondamentale per determinare il tipo di approccio risolutivo. Nel caso dei problemi NP-hard non è praticamente pensabile utilizzare algoritmi "esatti", ovvero in grado di determinare la soluzione ottimale, perché questi sarebbero messi in crisi da istanze appena poco più grandi di quelle banali. Per tali tipi di problemi è necessario ricorrere ad approcci euristici in grado di determinare soluzioni di buona qualità, anche se in genere sub-ottime, entro tempi di calcolo ridotti e quindi accettabili.

Di seguito, un paio di rappresentazioni grafiche di semplici casi di problemi VRP:

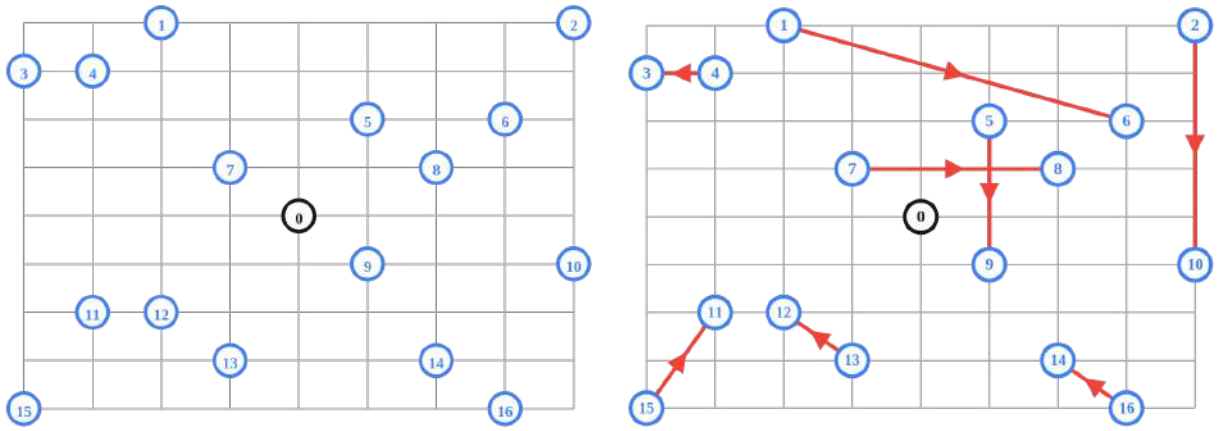

Fig.2 - Rappresentazione grafica di un problema VRP e di un VRP with Pick-up and delivery

Soluzioni del Vehicle Routing Problem

Per risolvere un problema di ottimizzazione combinatoria nel miglior modo possibile, si possono utilizzare dei metodi esatti, ossia che ci restituiscono una soluzione ottima della funzione obiettivo tra tutte le soluzioni ammissibili. Anche se recentemente sono state proposte molte formulazioni di programmazione matematica per la soluzione del VRP, questa strada non è percorribile, poiché il tempo computazionale e la complessità del problema sono spesso molto elevate. Quindi pur avendo una buona formulazione matematica, non si ha la disponibilità temporale adeguata ad ottenere una soluzione ottimale di un problema di cui si vuole ottenere una soluzione in tempi brevi.

Per questi motivi, problemi complessi come il VRP vengono risolti con algoritmi euristici, i quali forniscono soluzioni sub ottime in tempi molto brevi. In effetti delle soluzioni approssimate possono essere sufficienti nella risoluzione di istanze reali tipicamente di grandi dimensioni, poiché i parametri reali utilizzati sono essi stessi delle approssimazioni sottoposte ad errore. Inoltre, generalmente si eseguono ottimizzazioni in tempo reale, quindi è necessario ottenere una soluzione ammissibile in tempi brevi e molto spesso è sufficiente una soluzione ammissibile per poter attuare in modo rapido uno scenario di lavoro che rispetti tutte le richieste. Quindi è stato necessario sviluppare euristiche abbastanza flessibili da poter gestire i diversi obiettivi e i vincoli laterali. Sono state sviluppate molte euristiche che possiamo suddividere in euristiche costruttive ed in euristiche di miglioramento.

Gli algoritmi euristici sono progettati per i problemi di ottimizzazione combinatoria specifici che si vogliono risolvere, mentre negli ultimi anni hanno acquisito importanza le metaeuristiche, approcci euristici di tipo generale. La struttura e l'idea di fondo di ciascuna metaeuristica sono sostanzialmente fissate, ma la realizzazione delle varie componenti dell'algoritmo dipende dai singoli problemi

Il primo passo per risolvere il VRP con un'euristica è costruire una prima soluzione ammissibile. Solitamente le euristiche costruttive seguono l'idea che i clienti sono selezionati attraverso un

qualche criterio di minimizzazione del costo e le rotte sono costruite in modo da rispettare la capacità e i vincoli temporali. Esistono sia metodi sequenziali in cui viene costruita una soluzione alla volta, sia metodi paralleli in cui più soluzioni vengono costruite simultaneamente. Alcuni metodi costruttivi sono composti da due fasi e si possono suddividere in due parti: cluster-routes e routes-cluster. Nel primo caso i clienti vengono inizialmente suddivisi in sottoinsiemi ammissibili e poi viene costruito un percorso per ognuno di questi gruppi. Nel secondo caso viene prima costruita un percorso che comprende tutti i clienti e poi suddivisa in percorsi ammissibili.

Dopo una fase costruttiva molte euristiche hanno una fase migliorativa. Le euristiche di miglioramento per il VRP operano sia sui percorsi singoli che su un insieme di percorsi allo stesso tempo. L'idea base è quella di partire da una soluzione ammissibile e migliorarla attraverso piccoli cambiamenti. È possibile vedere questo miglioramento come un neighborhood search process, dove ogni percorso ha associato un vicinato di percorsi adiacenti. Le euristiche di miglioramento procedono con modifiche intra-route e inter-route.

Le metaeuristiche infine cercano di ovviare al problema di approcci globali che, in casi complessi in cui la funzione di costo ammetta un grande numero di minimi locali, rischiano di rimanere intrappolati in un minimo locale, troppo distante da una soluzione efficace del problema.

Di seguito, le soluzioni degli esempi precedenti in formato grafico:

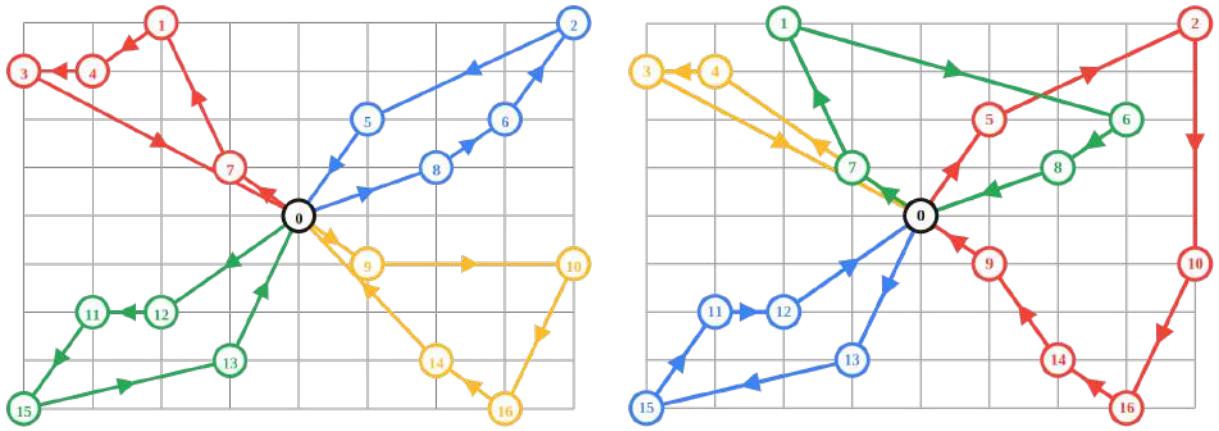

Fig.3 - Rappresentazione grafica delle soluzioni dei problemi precedenti

Il modulo di calcolo per il progetto MOBAS 4.0

Le seguenti sezioni descrivono quanto realizzato per il progetto.

Configurazione del modulo di calcolo

Il modulo di risoluzione del problema VRP associato allo scenario in oggetto deve fornire come output una tabella oraria del tipo località/orario con indicazioni georeferenziate del percorso scelto e un riferimento alle prenotazioni soddisfatte di diversi viaggi.

A questo scopo si è scelto di appoggiarsi a tool e librerie di risoluzione open source come la piattaforma SUMO oppure OSRM (Open Source Routing Machine) per lo studio del problema e infine di utilizzare OR tools (Google) per il calcolo della soluzione.

Questi strumenti, opportunamente customizzati e configurati per il nostro caso specifico, sono stati inseriti e utilizzati in alcuni script Python che costituiscono il core del servizio di routing.

La libreria di OR-Tools permette di configurare gli algoritmi e le opzioni di ricerca della soluzione (vedi https://developers.google.com/optimization/routing/routing_options), in particolare per la first solution strategy e le local serach options (metaeuristiche) e di approcciare al meglio il caso in esame.

Il modulo di calcolo dei percorsi e dei tempi di passaggio (routing & scheduling) necessita di alcuni file di configurazione, alcuni dei quali caratteristici della rete in esame, altri relativi ai dati di input sulle prenotazioni delle corse del bus.

Si tratta da una parte di dati statici standard che la libreria OR-tools utilizza per definire il problema e i suoi vincoli (matrice dei tempi o delle distanze, numero e capacità dei veicoli etc…) e che servono prettamente all'algoritmo per calcolare la soluzione, come ad esempio:

```
{
    "time_matrix": [
                [  0, 999, 999, 999, 999, 999,   0, 999],
                [999,   0,  80, 145, 225, 283, 999, 194],
                [999, 999,   0,  65, 146, 203, 999, 115],
                [999, 999, 999,   0,  81, 138, 999, 151],
                [999, 999, 999, 999,   0,  58, 999, 115],
                [999, 999, 999, 999,  30,   0, 999,  58],
                [999,  71, 151, 196, 157, 254,   0,   0],
                [  0, 999, 999, 999, 999, 999,   0,   0]
                ],

    "starts": [0],
    "ends":   [0],
    "depot": 0,
    "time_windows": [[0, 3600],
                     [1, 3600],
                     [1, 3600],
                     [1, 3600],
                     [1, 3600],
                     [1, 3600]],

    "num_vehicles": 1,
    "vehicle_capacity": [12],
    "vehicle_load_time": 10,
    "vehicle_unload_time": 10,
    "depot_capacity": 1
}
```

Fig.4 Esempio di file di configurazione data.json

Dall'altra di dati statici geografici per meglio definire l'output di routing e la sua eventuale rappresentazione su mappe, come nell'esempio:


```
[
    {
        "name"   : "route_1to2",
        "start" : "1",
        "end"    : "2",
        "coordinates": [[16.64373137677596, 40.16301862633316, 0.0],
                        [16.64350954526544, 40.16332345735884, 0.0],
                        [16.63994914970874, 40.16526169892004, 0.0],
                        [16.63956009728916, 40.16526790810651, 0.0],
                        [16.63870553159681, 40.16463002986256, 0.0]]
    },
    {
        "name": "route_2to3",
        "start" : "2",
        "end"    : "3",
        "coordinates": [[16.63870700097245, 40.16458321527378, 0.0],
                        [16.63814580620158, 40.16422302977829, 0.0],
                        [16.63801239165037, 40.16391770382961, 0.0],
                        [16.63579850326296, 40.16388017594545, 0.0],
                        [16.63539986942726, 40.16371457948282, 0.0],
                        [16.63522431746965, 40.16341533301366, 0.0],
                        [16.63544534875582, 40.16331463248869, 0.0],
                        [16.63554239304019, 40.16310103598007, 0.0],
                        [16.63606664747905, 40.16292514680515, 0.0]]
    },
    {
        "name": "route_3to4",
        "start" : "3",
        "end"    : "4",
        "coordinates": [[16.63609434795434, 40.16292505651972, 0.0],
                        [16.63631589192575, 40.16296694817747, 0.0],
                        [16.6366712697508, 40.16259771720501, 0.0],
                        [16.63732473419663, 40.16280792489939, 0.0],
                        [16.63867278608915, 40.16286641265582, 0.0],
                        [16.63895241351288, 40.16244969238129, 0.0],
                        [16.63967669844781, 40.16209720676127, 0.0],
                        [16.63989927322008, 40.16168196745819, 0.0],
                        [16.64000708981687, 40.16047629018312, 0.0]]
    },
    {
        "name": "route_4to5",
        "start" : "4",
```


Fig.5 - Esempio di file di configurazione routes.json

Comunicazione e formato dei dati scambiati

In base all'ipotesi di staticità assunta per lo scenario di simulazione di servizio bus a prenotazione, l'ottimizzazione verrà operata in modalità offline, una volta acquisiti i dati necessari.

In questo contesto risulta sufficiente, oltreché conveniente, che lo scambio di informazioni tra il modulo di calcolo e l'interfaccia utente avvenga in modalità asincrona tramite *file* di input/output salvati su cartella dedicata su un server (da concordare) e fasi successive di scrittura/lettura da parte degli stessi moduli ("a chiamata" oppure "a tempi fissati").

A tale scopo, i *file* di tipo ***.json*** (*Java Script Object Notation*) sono stati scelti come migliore opzione.

Di seguito vengono presentati degli estratti del file di prenotazione del servizio (***booking.json*** input per l'ottimizzatore) e del file di schedulazione delle corse (***routing.json***, output della soluzione di routing e scheduling) dove i campi significano per quel che riguarda l'input delle prenotazioni:

> "**booking_id**": stringa (o intero) univoca di identificazione della prenotazione,
>
> "**booking_time**" : stringa di identificazione dell'orario a cui è avvenuta la prenotazione in formato date/time (ad esempio YYYY-MM-ddThh:mm:ss),
>
> "**start_point**" : stringa (o intero) univoca di identificazione del punto di pick-up,
>
> "**end_point**" : stringa (o intero) univoca di identificazione del punto di delivery,
>
> "**depart_date_time**" : stringa di identificazione del tempo di prenotazione (inteso come tempo desiderato di prelievo allo start-point) in formato date/time (ad esempio YYYY-MM-ddThh:mm:ss)

e per quel che riguarda l'output della soluzione:

"trip": stringa (o intero) univoca di identificazione del viaggio che comprende schedule e route,

> **"schedule"**: lista che include nodo e tempo di raccolta e la lista delle prenotazioni previste
>
> > "**point_id**": stringa (o intero) univoca di identificazione del punto di pick-up,
> >
> > "**time**" : stringa di identificazione dell'orario di passaggio previsto nel punto indicato, in formato date/time (ad esempio YYYY-MM-ddThh:mm:ss),
> >
> > "**booking_list**": lista di booking_id, cioè di identificativi di prenotazione
>
> **"route":** comprende una sequenza di coordinate georeferenziate che rappresenta il percorso del veicolo durante il viaggio (trip) di riferimento.

Abbiamo lasciato indicato anche nell'output il numero di prenotazione, con l'idea di poter eventualmente comunicare modifiche e slittamenti di orario dovute a limiti di capacità, anche se questa ipotesi al momento non è nello stato dell'arte e negli obiettivi della simulazione.

```
[
    {
        "booking_id": "00012",
        "booking_time" : "2023-11-12T20:20:00Z",
        "start_point" : "3",
        "end_point" : "4",
        "depart_date_time" : "2023-11-13T12:30:00Z"
    },
    {
        "booking_id": "00013",
        "booking_time" : "2023-11-12T20:27:00Z",
        "start_point" : "4",
        "end_point" : "5",
        "depart_date_time" : "2023-11-13T13:00:00Z"
    },
    {
        "booking_id": "00014",
        "booking_time" : "2023-11-12T20:31:00Z",
        "start_point" : "3",
        "end_point" : "5",
        "depart_date_time" : "2023-11-13T12:40:00Z"
    },
    {
        "booking_id": "00015",
        "booking_time" : "2023-11-12T20:20:00Z",
        "start_point" : "1",
        "end_point" : "2",
        "depart_date_time" : "2023-11-13T12:30:00Z"
    },
    {
        "booking_id": "00016",
        "booking_time" : "2023-11-12T20:27:00Z",
        "start_point" : "4",
        "end_point" : "7",
        "depart_date_time" : "2023-11-13T14:00:00Z"
    },
    {
        "booking_id": "00017",
        "booking_time" : "2023-11-12T20:31:00Z",
        "start_point" : "3",
        "end_point" : "5",
        "depart_date_time" : "2023-11-13T12:42:00Z"
    },
    {
        "booking_id": "00018",
```


Fig.6 - Esempio di file di input booking.json

Di seguito invece ecco il corrispondente file di output che riporta la soluzione proposta dal modulo di routing/scheduling.


```
[
    {
        "trip": "1",
        "schedule": [
            {
                "point_id": "CAPOLINEA",
                "time": "2023-11-13T12:30:00Z",
                "booking_list": "[]"
            },
            {
                "point_id": "FERMATA_1",
                "time": "2023-11-13T12:31:11Z",
                "booking_list": "['00015']"
            },
            {
                "point_id": "FERMATA_2",
                "time": "2023-11-13T12:32:31Z",
                "booking_list": "[]"
            },
            {
                "point_id": "FERMATA_3",
                "time": "2023-11-13T12:33:36Z",
                "booking_list": "['00012', '00014', '00017']"
            },
            {
                "point_id": "FERMATA_4",
                "time": "2023-11-13T12:34:57Z",
                "booking_list": "[]"
            },
            {
                "point_id": "FERMATA_5",
                "time": "2023-11-13T12:35:55Z",
                "booking_list": "[]"
            },
            {
                "point_id": "CAPOLINEA",
                "time": "2023-11-13T12:36:53Z",
                "booking_list": "[]"
            }
        ],
        "route": [
            [
                16.64463206024319,
                40.16002552552391,
                0.0
            ],
            [
                16.64465587138461,
```


Fig.7 - Esempio di file di output routing.json

Configurazione della dashboard per prenotazione e visualizzazione dei percorsi

Facendo seguito a quanto sviluppato, si è proceduto con la realizzazione di una dashboard, accessibile online tramite dispositivi fissi e mobile per la prenotazione dei percorsi e per la conseguente visualizzazione dei percorsi ottimizzati.

La dashboard realizzata per il progetto MOBAS 4.0 è accessibile al link: https://app.s4mobas.it/

Di seguito alcuni esempi di visualizzazione:

Fig. 8 - Visualizzazione generale pagina prenotazione percorso

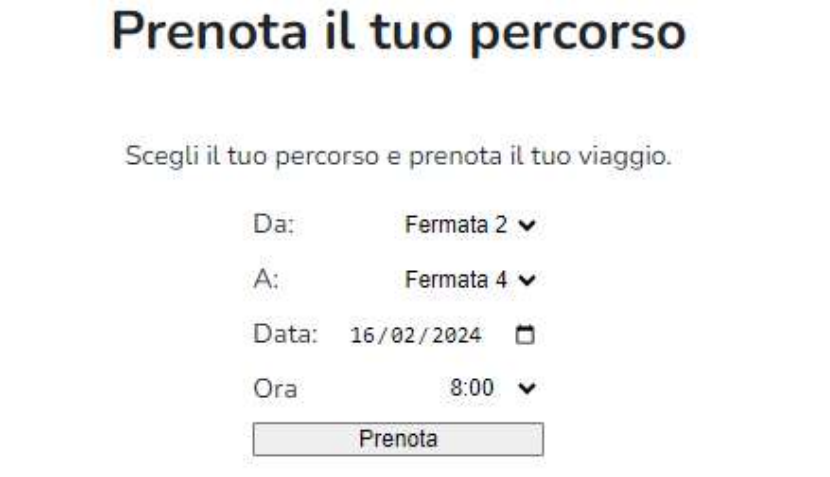

Fig. 10 - Dettaglio prenotazione percorso

A seguito della prenotazione, l'algoritmo precedentemente descritto procede a generare il percorso ottimale, rappresentato in dashboard come di seguito mostrato:

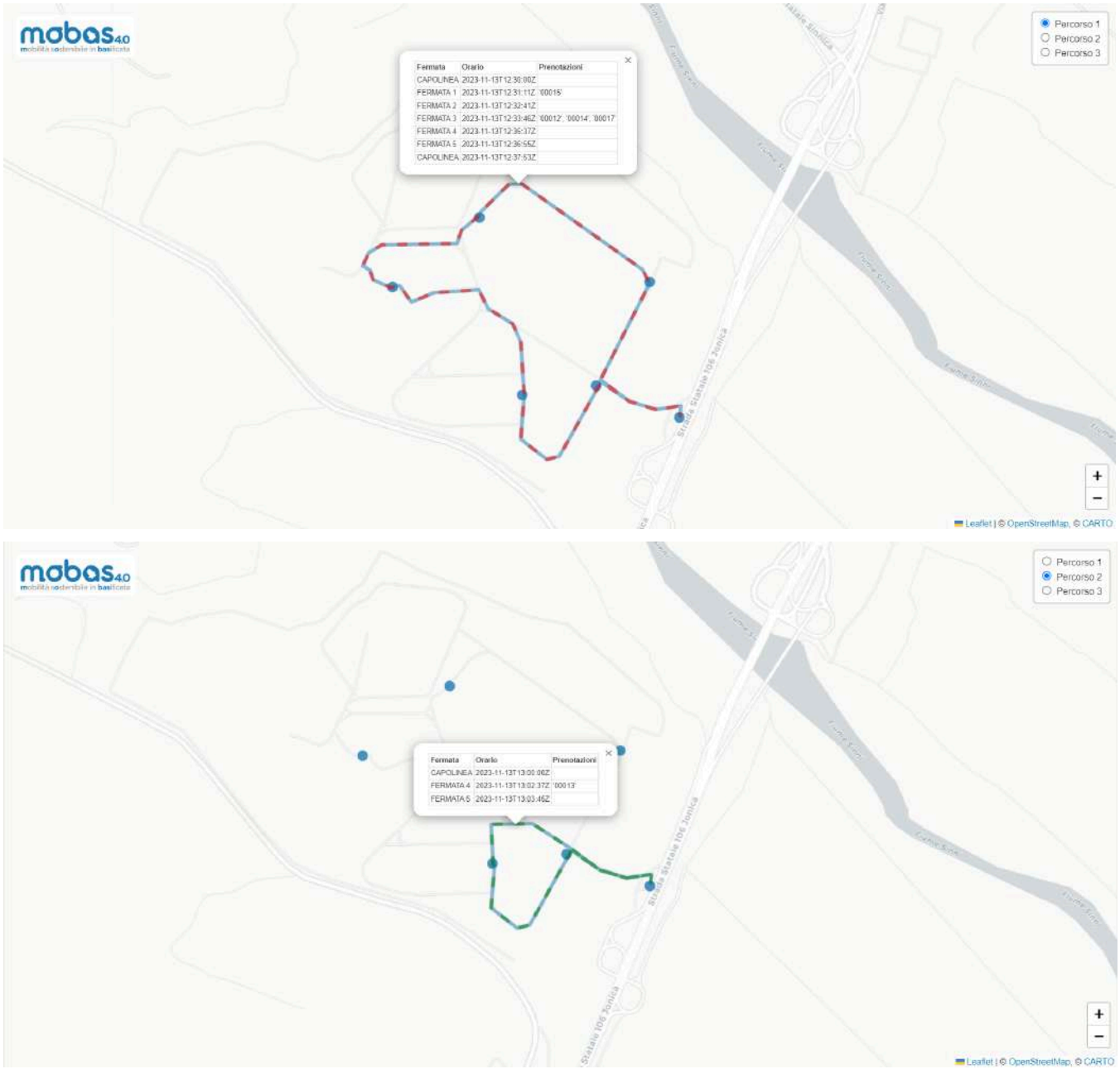

Fig. 11 e 12 - Esempio percorsi ottimali

Di seguito visualizzazione tramite smartphone:

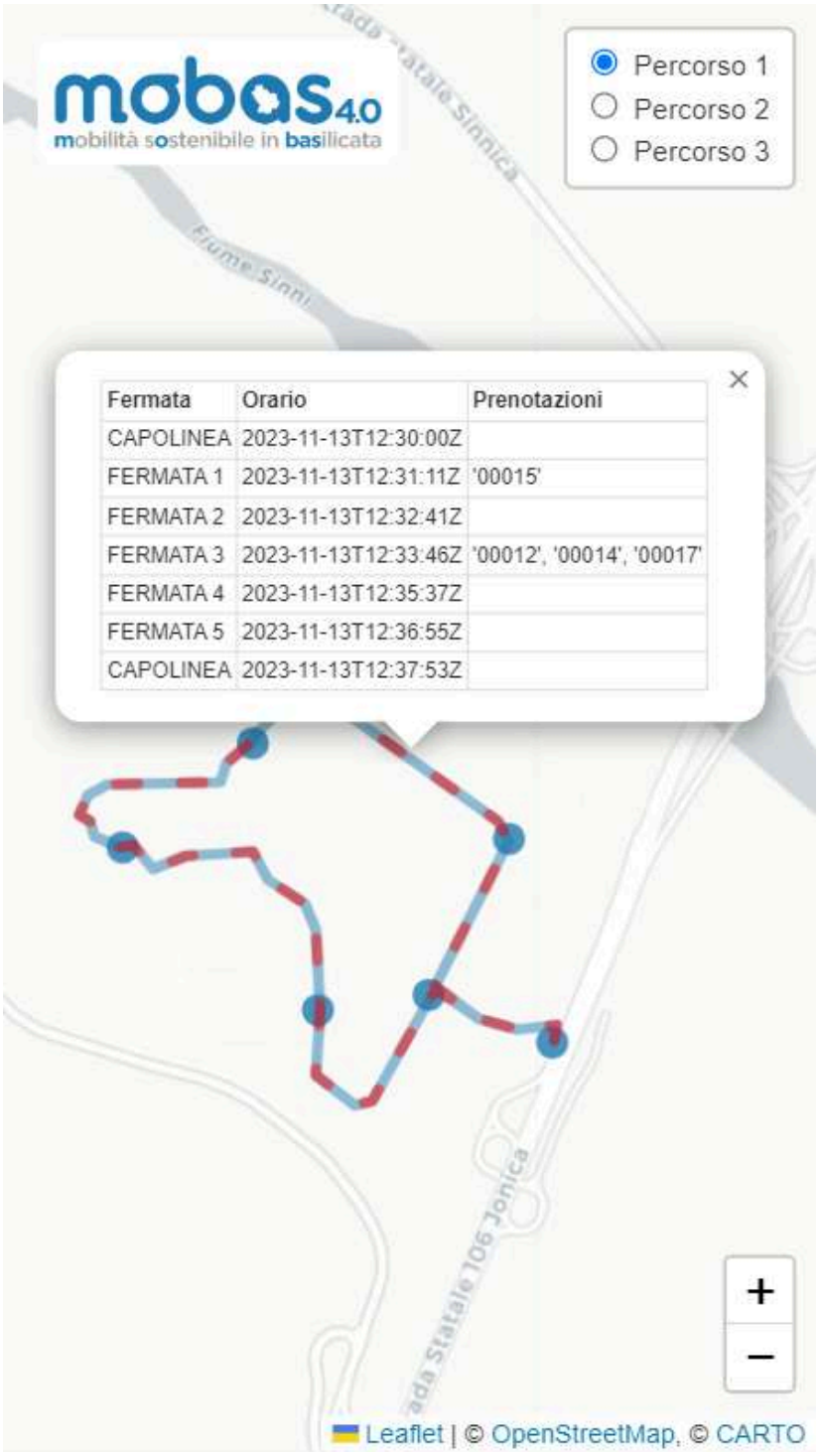

Fig. 13 - Esempio percorso ottimale (visualizzazione smartphone)

Ulteriori funzionalità della dashboard

In relazione alle ulteriori aree di sperimentazione di progetto, la dashboard presenta due ulteriori funzionalità, connesse alla prenotazione di slot di ricarica nell'area interessata dall'intervento di progetto e alla ricezione di dati da un altro dimostratore di progetto, la carrozzina elettrica sviluppata nell'ambito del Work Package 4.

Prenotazione slot di ricarica

In generale, l'incentivazione all'uso di auto elettriche rappresenta un pilastro fondamentale per la transizione verso una mobilità più sostenibile e a basse emissioni di carbonio. Gli incentivi possono assumere diverse forme, tra cui sgravi fiscali sull'acquisto di veicoli elettrici, agevolazioni per l'installazione di infrastrutture di ricarica, e vantaggi in termini di accesso a zone a traffico limitato e parcheggi dedicati. Tuttavia, è cruciale che gli incentivi siano supportati da politiche a lungo termine e da investimenti consistenti per assicurare una transizione stabile e sostenibile verso la mobilità elettrica.

Queste misure non solo favoriscono la diffusione di veicoli elettrici, ma anche la crescita di un'infrastruttura di ricarica capillare e accessibile. L'incentivazione all'uso di auto elettriche è infatti strettamente legata alla presenza capillare sul territorio di colonnine di ricarica. Infatti, una rete di infrastrutture di ricarica diffusa e accessibile è fondamentale per garantire la praticità e l'attrattività dell'adozione di veicoli elettrici.

Le colonnine di ricarica devono essere strategicamente posizionate in luoghi ad alto flusso di traffico, come aree urbane, centri commerciali, stazioni di servizio e parcheggi pubblici. In questo modo, gli utenti di auto elettriche possono contare su una rete affidabile di punti di ricarica che permette loro di effettuare viaggi senza preoccupazioni di autonomia. Inoltre, la presenza di colonnine di ricarica può incoraggiare l'adozione di veicoli elettrici anche tra coloro che vivono in condomini o abitazioni senza possibilità di installare una ricarica privata. Pertanto, investire nella costruzione e nell'ampliamento della rete di colonnine di ricarica è essenziale per favorire una transizione efficace verso la mobilità elettrica e per massimizzare gli effetti degli incentivi sul mercato automobilistico.

In relazione al progetto MOBAS 4.0, la dashboard sviluppata consente - a titolo puramente dimostrativo - di visualizzare le stazioni di ricarica presenti sul suolo pubblico nell'area oggetto dell'intervento di progetto. La lista e il posizionamento delle colonnine considerate nel dimostratore sono informazioni estrapolate dal sito web Enel X: https://www.mobility.enelx.com/it/mappa-stazioni-ricarica.

La funzionalità realizzata a scopo dimostrativo, prevede la possibilità di selezionare una colonnina e richiedere le indicazioni di navigazione verso la stessa. Tale funzionalità potrà essere, in prospettiva e in eventuali future evoluzioni del progetto, essere implementata tramite sensoristica che consenta di visionare lo stato della colonnina di ricarica selezionata (disponibile / occupata).

Di seguito la visualizzazione su mappa:

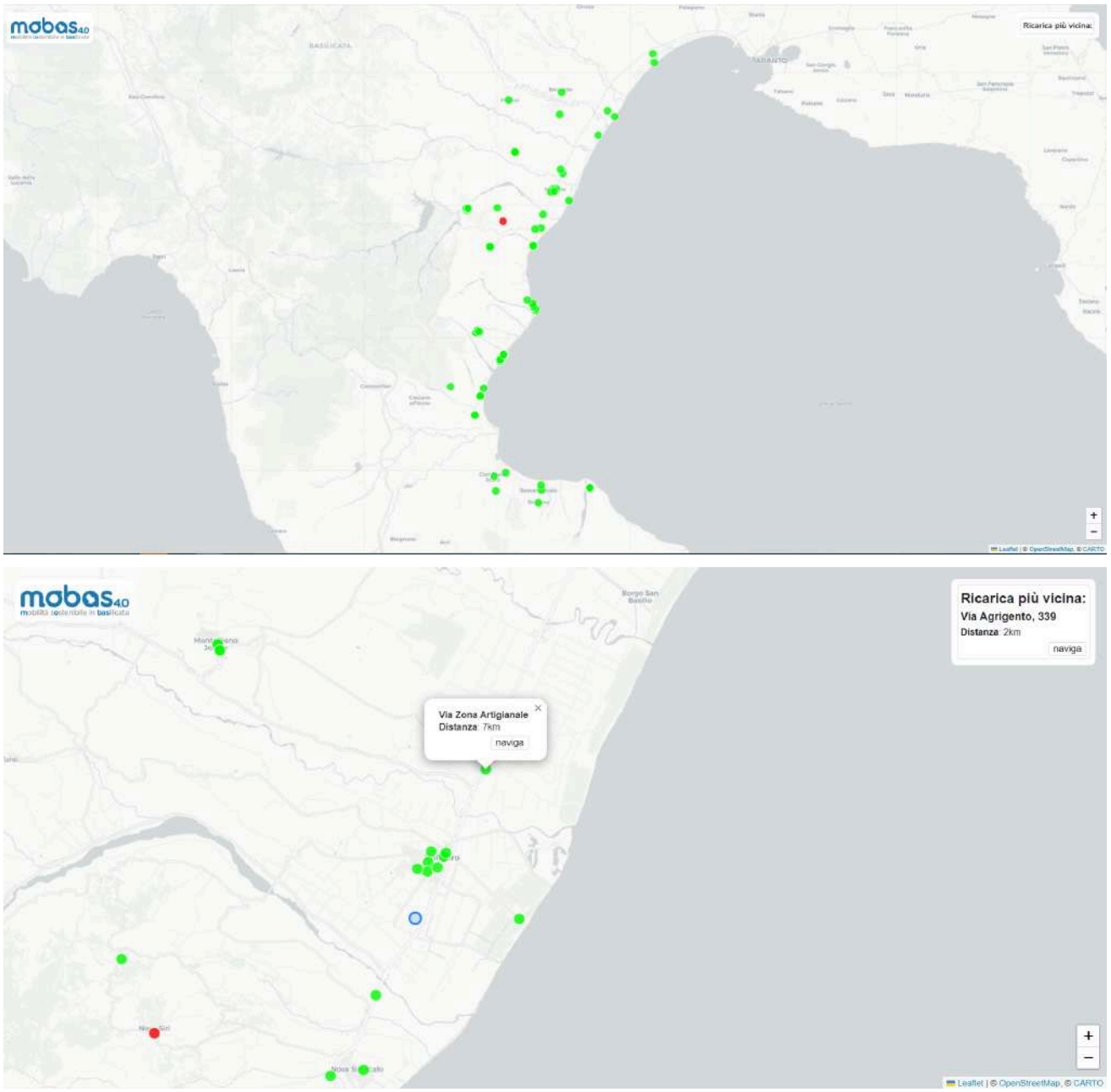

Fig. 14 e 15 - Posizione colonnine di ricarica sul territorio (visualizzazione PC)

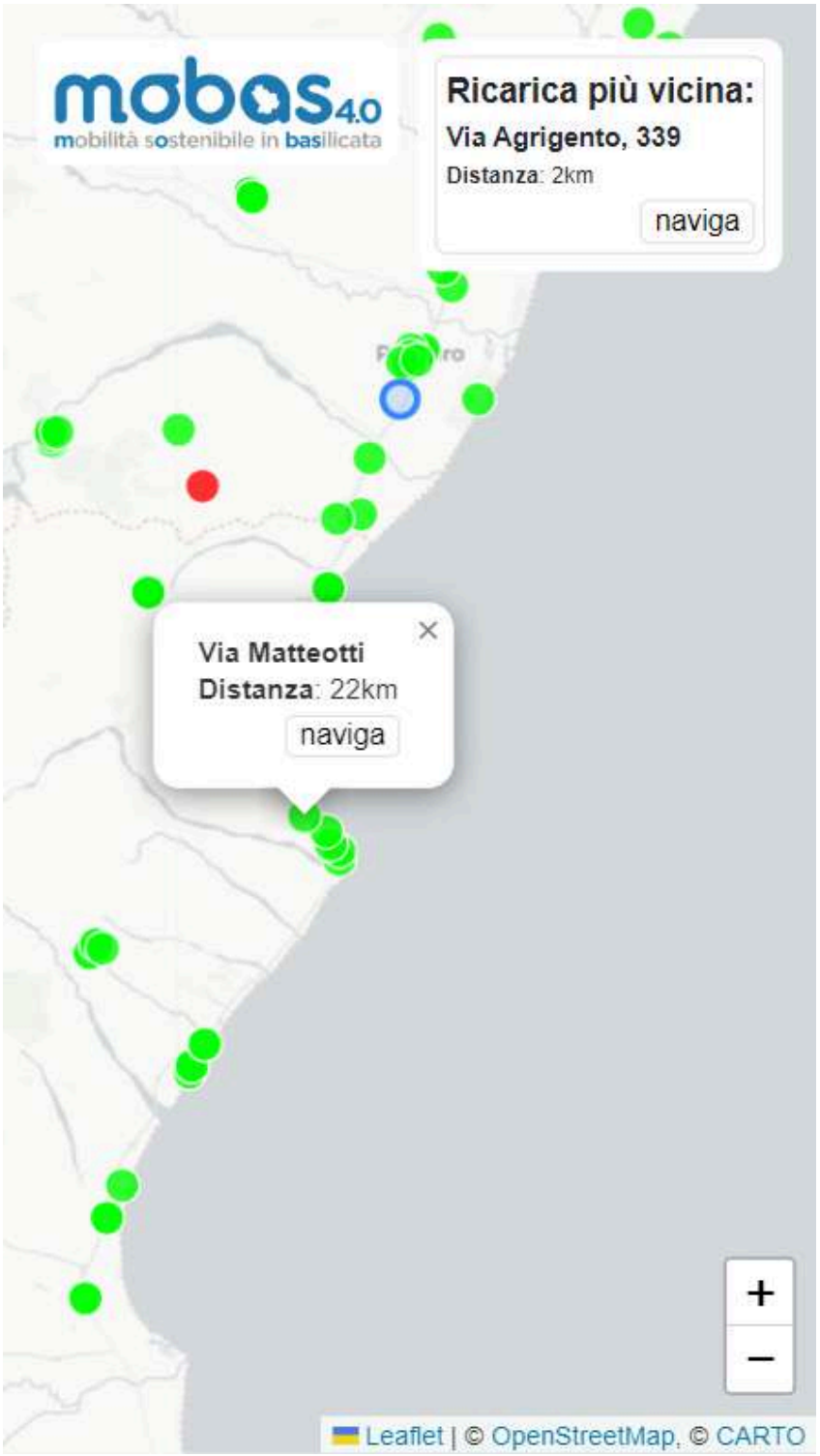

Fig. 16 - Posizione colonnine di ricarica sul territorio (visualizzazione smartphone)

Ricezione dati da carrozzina elettrica sviluppata nel WP4

Il progetto MOBAS 4.0 ha previsto la progettazione e la realizzazione di un prototipo avanzato di carrozzina elettrica, dotato di sensori per monitorare il movimento del mezzo, registrare le condizioni di salute dell'utente e trasmettere i dati a una centrale di controllo per l'analisi. Le tecnologie impiegate hanno risposto a specifiche esigenze, tra cui garantire il corretto funzionamento meccanico ed elettrico, migliorare la sicurezza degli utenti individuando ostacoli, consentire la localizzazione della carrozzina e raccogliere dati sullo stato di salute dell'utente. Le informazioni di dettaglio circa le attività realizzate nell'ambito del Work Package 4 sono descritte nel dettaglio dei Deliverables dedicati.

Rispetto alla ricezione delle informazioni raccolte nel WP4, il dashboard riceve i dati dalla sensoristica sviluppata e utilizzata tramite l'utilizzo di una API (Application Programming Interface). In generale, le API sono insiemi di regole e definizioni che consentono a diversi software di comunicare tra loro e scambiare dati e funzionalità in modo efficiente e standardizzato. Le API definiscono come diversi componenti software possono interagire tra loro, consentendo agli sviluppatori di integrare servizi e funzionalità offerti da altri programmi o piattaforme nei propri progetti senza la necessità di conoscere i dettagli interni di come funzionano. Le API semplificano lo sviluppo di software, migliorano la produttività degli sviluppatori e favoriscono l'integrazione e l'interoperabilità tra sistemi informatici diversi.

Di seguito la visualizzazione dei dati raccolti tramite la sensoristica sviluppata nel WP4:

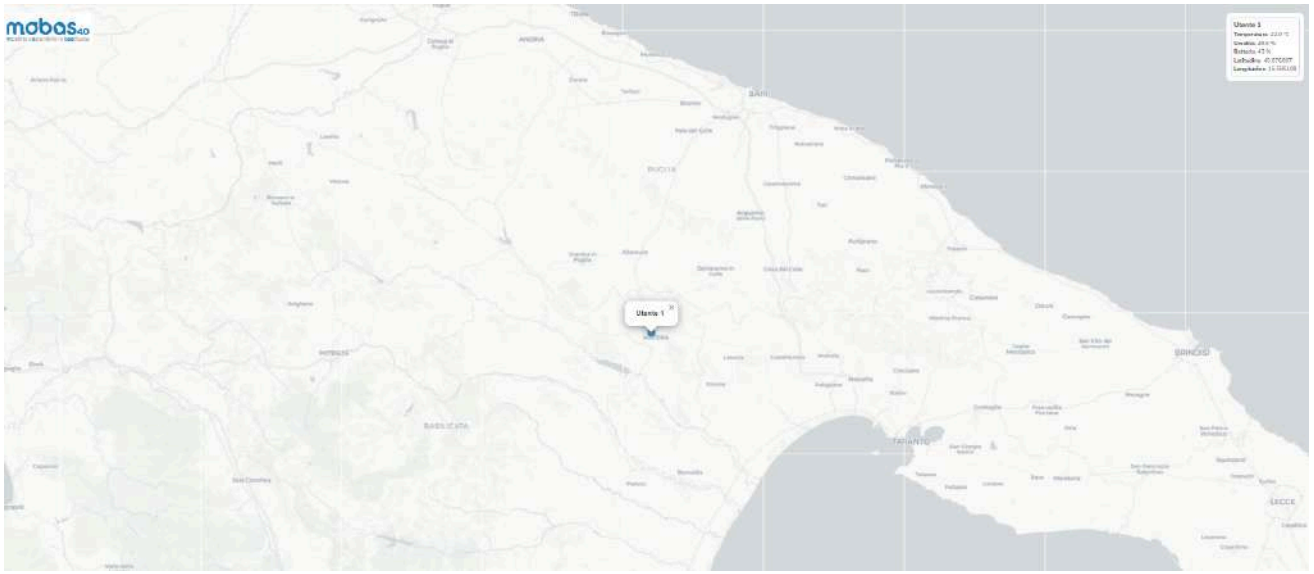

Fig. 17 - Visualizzazione dati carrozzina elettrica su mappa

Fig. 17 - Dettaglio visualizzazione dati carrozzina elettrica su mappa

Applicazione mobile per la fruizione di contenuti interesse culturale e turistico

Negli ultimi anni, l'uso delle applicazioni mobile per la veicolazione di informazioni di carattere culturale e turistico ha conosciuto una diffusione sempre più ampia e significativa.

Queste applicazioni rappresentano strumenti versatili e accessibili che consentono ai viaggiatori di esplorare luoghi di interesse culturale, storico e turistico in modo interattivo e personalizzato. Grazie alla loro portabilità e alla disponibilità di connessione internet, le app culturali offrono agli utenti la possibilità di accedere a una vasta gamma di contenuti multimediali, quali guide audio, video, immagini e testi informativi, direttamente dai loro dispositivi mobili. Attraverso funzionalità come mappe interattive, suggerimenti personalizzati e percorsi tematici, queste applicazioni sono in grado di arricchire l'esperienza di viaggio, consentendo agli utenti di scoprire e approfondire la storia, l'arte e la cultura dei luoghi visitati in modo coinvolgente e informativo. Inoltre, le applicazioni mobili favoriscono la diffusione e la promozione del patrimonio culturale e turistico, raggiungendo un pubblico sempre più ampio e diversificato, e contribuendo così a valorizzare e preservare il patrimonio culturale delle diverse comunità e destinazioni turistiche.

In questa ottica, un ulteriore dimostratore è stato realizzato per il progetto MOBAS 4.0. Si tratta di un'applicazione mobile dedicata alla presentazione e alla fruizione di contenuti a carattere culturale e turistico. Per il dimostratore sono stati implementati uno spazio *back end* per la gestione dei contenuti, definito anche CMS dall'inglese *Content Management System*, e il *front end*, ovvero l'applicativo mobile con cui si interfaccia direttamente l'utente finale.

I contenuti dimostrativi

Nello specifico, il dimostratore mobile presenta un Percorso Test, con punti geolocalizzati sulla mappa della Basilicata tramite coordinate e con schede di dettaglio su punti turistici selezionati per simulare quella che potrebbe essere l'effettiva creazione di un percorso di scoperta del territorio.

Tramite la geolocalizzazione su mappa, si ha accesso a delle schede esplicative che presentano un testo descrittivo e una gallery di fotografie dedicate al luogo o alla vicenda trattata. Queste schede esplicative sono predisposte per essere implementate con ulteriori contenuti multimediali come, ad esempio: audio, video, modelli 3D da visualizzare in Realtà Aumentata, ma anche fotografie a 360° etc.

Per il dimostratore sono stati selezionati 5 Punti di Interesse (POI, dall'acronimo inglese Points Of Interest), a scopo puramente dimostrativo e che sono rappresentativi, ma non limitativi, alla realtà territoriale di riferimento del progetto.

Nello specifico, si riportano di seguito i contenuti testuali individuati per il Percorso Test: :

- POI 1 - MATERA
  Matera, cuore della Basilicata, è una città dalle radici millenarie, dichiarata Patrimonio dell'Umanità dall'UNESCO nel 1993. Conosciuta per i suoi antichi sassi, Matera offre un viaggio nel tempo attraverso abitazioni scavate nella roccia calcarea, straordinari esempi dell'adattamento dell'uomo all'ambiente naturale che creano un paesaggio unico e suggestivo. La Cattedrale di Matera, con la sua facciata romanica, e la Chiesa di San Pietro Caveoso sono solo alcune delle gemme storiche di una città in cui la mano dell'uomo si fonde con quella della natura. Oltre che di storie, la cultura materana è ricca di tradizioni gastronomiche, come quella del famoso Pane di Matera, cui è associata anche la produzione artigianale dei timbri del pane che ancora oggi è forte attrattiva turistica. Vittima, durante il secondo dopoguerra, dell'arretratezza del meridione fino ad essere bollata "vergogna nazionale", dalla fine del Novecento in avanti la città si impegna in un riscatto talmente efficiente da essere nominata Capitale Europea della Cultura del 2019, divenendo così simbolo di rinascita e adattamento che crea un legame indelebile tra passato e presente.

- POI 2 - METAPONTO
  Situata sulla costa ionica della Basilicata, Metaponto è una località ricca di storia e bellezze naturali. Famosa per i suoi siti archeologici, rappresenta una testimonianza del passato magnificente della Magna Grecia. Il Parco Archeologico di Metaponto ospita antichi templi, come quello dedicato a Hera, e i resti di una maestosa città greca che, con le sue colonne e le sue fondamenta, narrano storie di civiltà passate. Il Museo Archeologico Nazionale di Metaponto offre un'ulteriore immersione nella storia, con reperti risalenti all'epoca greca e romana e il teatro greco, con la sua struttura ottimamente conservata, è un luogo imperdibile per gli amanti dell'archeologia. La città moderna di Metaponto, circondata da uliveti e vigneti, affacciata sulle acque cristalline del Mar Ionio, offre un rifugio accogliente in cui la tradizione contadina e marina si fondono nell'esperienza gastronomica locale. Con la sua dualità di antichità e bellezze naturali, Metaponto incanta i visitatori offrendo loro un viaggio unico nel tempo e nella cultura della Magna Grecia.

- POI 3 - CASTELMEZZANO E PIETRAPERTOSA
  Iscritte nell'elenco dei "Borghi più belli d'Italia", Castelmezzano e Pietrapertosa sono due incantevoli realtà situate nel cuore del Parco Naturale delle Dolomiti Lucane. Con le loro antiche pietre e l'atmosfera fiabesca, trasportano i visitatori in un mondo senza tempo. Il borgo di Castelmezzano, incastonato tra montagne imponenti, è caratterizzato da strette vie lastricate ed edifici in pietra. Il borgo di Pietrapertosa, anch'esso dominato da rocce e montagne, offre uno scenario altrettanto suggestivo. Il suo nucleo antico conserva l'atmosfera di un tempo passato, con vicoli tortuosi e architetture tradizionali. Entrambi i borghi sono noti per il famoso "Volo dell'angelo", un'emozionante esperienza di zip-line che permette ai visitatori di volare tra le valli sospesi a un cavo d'acciaio. La vista panoramica durante il volo è mozzafiato e offre una prospettiva unica sulle maestose Dolomiti Lucane, che accolgono i visitatori in un paesaggio ricco di storia oltre che di natura.

- POI 4 - VENOSA
  Gemma ricca di storia, cultura e bellezze artistiche, Venosa è stata fondata dagli antichi Romani e ha conservato intatto il suo fascino nel corso dei secoli, arricchendosi di cultura epoca dopo epoca. Ancora visibili e visitabili, infatti, sono i resti dell'antica città romana, tra le cui costruzioni spicca per importanza il nucleo abitativo originale del poeta Orazio. Di notevole rilievo anche il Castello Aragonese, un'imponente fortezza che non solo domina il panorama e offre uno sguardo privilegiato sulla città, ma ospita anche la ricca collezione di reperti del Museo Archeologico Nazionale, che racconta la storia della zona dalla preistoria all'epoca romana. La Cattedrale di Santa Maria Maggiore, con la sua maestosa facciata romanica, è un'altra perla architettonica di Venosa e racchiude affreschi e opere d'arte che offrono una testimonianza dell'importanza culturale e religiosa della città nel corso dei secoli. Immersa nella campagna verde e collinare, la città di Venosa dona ai visitatori una combinazione di storia, arte e paesaggio incantevole e indimenticabile.

- POI 5 - POTENZA
  Potenza, gioiello della Basilicata e suo capoluogo, svetta fieramente a 819 metri sul livello del mare, rendendola una delle città italiane più elevate. Conosciuta come la "Città verticale", il suo centro storico si erge sulla cima più alta, mentre i quartieri si snodano lungo pendii graduali. Le sue antiche scale e il sistema di scale mobili la distinguono come la "Città delle cento scale". Potenza ha subito le vicissitudini della storia, dalle invasioni greche e romane ai terremoti devastanti del 1273 e del 1857. Tuttavia, ha rinato con forza, mostrando una ricca eredità culturale e architettonica. Le sue chiese millenarie, come San Michele Arcangelo e il Duomo di San Gerardo, testimoniano la sua devozione e la sua magnificenza. Il Museo Archeologico Nazionale della Basilicata Dinu Adamesteanu e le porte antiche raccontano le sue storie millenarie. Potenza offre anche oasi di tranquillità, come la Villa Comunale di Santa Maria. In breve, Potenza incarna la grandezza e la bellezza della Basilicata.

Il back end

Il back end di un'app mobile, gestito attraverso un Content Management System (CMS), svolge un ruolo fondamentale nell'organizzazione, gestione e distribuzione dei contenuti che vengono visualizzati tramite l'app stessa. In generale, tra le principali funzionalità e componenti dei back end CMS di un'app mobile si possono annoverare:

- Gestione dei contenuti: Il CMS consente agli amministratori di caricare, modificare e organizzare i contenuti dell'app mobile. Questi contenuti possono includere testi, immagini, video, audio, e altri tipi di file multimediali;
- Struttura dei dati: Il back end del CMS definisce la struttura dei dati che vengono utilizzati dall'app mobile. Questo può includere la definizione di campi e attributi per diversi tipi di contenuti, nonché le relazioni tra di essi;

- Autenticazione e autorizzazione: Il CMS gestisce l'autenticazione degli utenti e l'assegnazione dei permessi di accesso. Questo è particolarmente importante per garantire che solo gli utenti autorizzati possano accedere a determinati contenuti o funzionalità dell'app;
- Gestione degli utenti: Il CMS consente agli amministratori di gestire gli utenti dell'app mobile, inclusi la registrazione degli utenti, la gestione dei profili utente e l'assegnazione dei ruoli e dei privilegi;
- Analytics e monitoraggio delle prestazioni: Il CMS può integrare strumenti di analisi che consentono agli amministratori di monitorare l'utilizzo dell'app, raccogliere dati sui comportamenti degli utenti e valutare le prestazioni dell'app;
- Gestione delle notifiche push: Il back end del CMS può includere funzionalità per la gestione delle notifiche push, consentendo agli amministratori di inviare messaggi agli utenti dell'app mobile per comunicazioni importanti, aggiornamenti o promozioni;
- Integrazione con servizi esterni: Il CMS può integrarsi con servizi esterni, come sistemi di pagamento, piattaforme di social media, servizi di geolocalizzazione e altro ancora, per arricchire le funzionalità dell'app e migliorare l'esperienza dell'utente.

In sintesi, il back end CMS di un'app mobile svolge un ruolo chiave nell'organizzazione e nella gestione dei contenuti, nell'autenticazione degli utenti, nell'analisi delle prestazioni e nella gestione delle interazioni con servizi esterni, contribuendo così a garantire un'esperienza utente ottimale e un funzionamento efficace dell'app. Nel dettaglio del progetto MOBAS 4.0, il CMS strutturato permette in modo intuitivo di gestire, modificare, aggiungere ed eliminare contenuti informativi.

Il front end

La creazione del front end di un'app mobile comporta la progettazione e lo sviluppo dell'interfaccia utente (UI - User Interface) e dell'esperienza utente (UX - User Experience) per rendere l'applicazione intuitiva, piacevole e funzionale per gli utenti.

Per l'UI, si tratta di creare un design visivamente accattivante, che includa la disposizione degli elementi visivi come pulsanti, icone e campi di input in modo chiaro e coerente. La scelta di colori, caratteri e stili grafici gioca un ruolo importante nell'aspetto complessivo dell'app e nell'attrarre gli utenti. Per quanto riguarda l'UX, l'obiettivo è di rendere l'utilizzo dell'app semplice e intuitivo per gli utenti. Questo significa progettare flussi di navigazione chiari e logici, garantendo che gli utenti possano trovare facilmente ciò di cui hanno bisogno all'interno dell'app e che le azioni siano comprensibili e facili da eseguire.

Inoltre, è fondamentale testare l'UI e l'UX durante il processo di sviluppo per identificare e risolvere eventuali problemi di usabilità o di design. Il feedback degli utenti è prezioso per migliorare continuamente l'esperienza complessiva dell'app e assicurarsi che sia adattata alle esigenze e alle preferenze degli utenti. In sintesi, la creazione del front end di un'app mobile si concentra sull'aspetto visuale e sull'esperienza di utilizzo dell'applicazione, con l'obiettivo di fornire agli utenti un'interfaccia intuitiva, piacevole e funzionale.

Per il progetto MOBAS 4.0, si riportano di seguito alcune schermate dell'applicativo mobile realizzato come dimostratore.

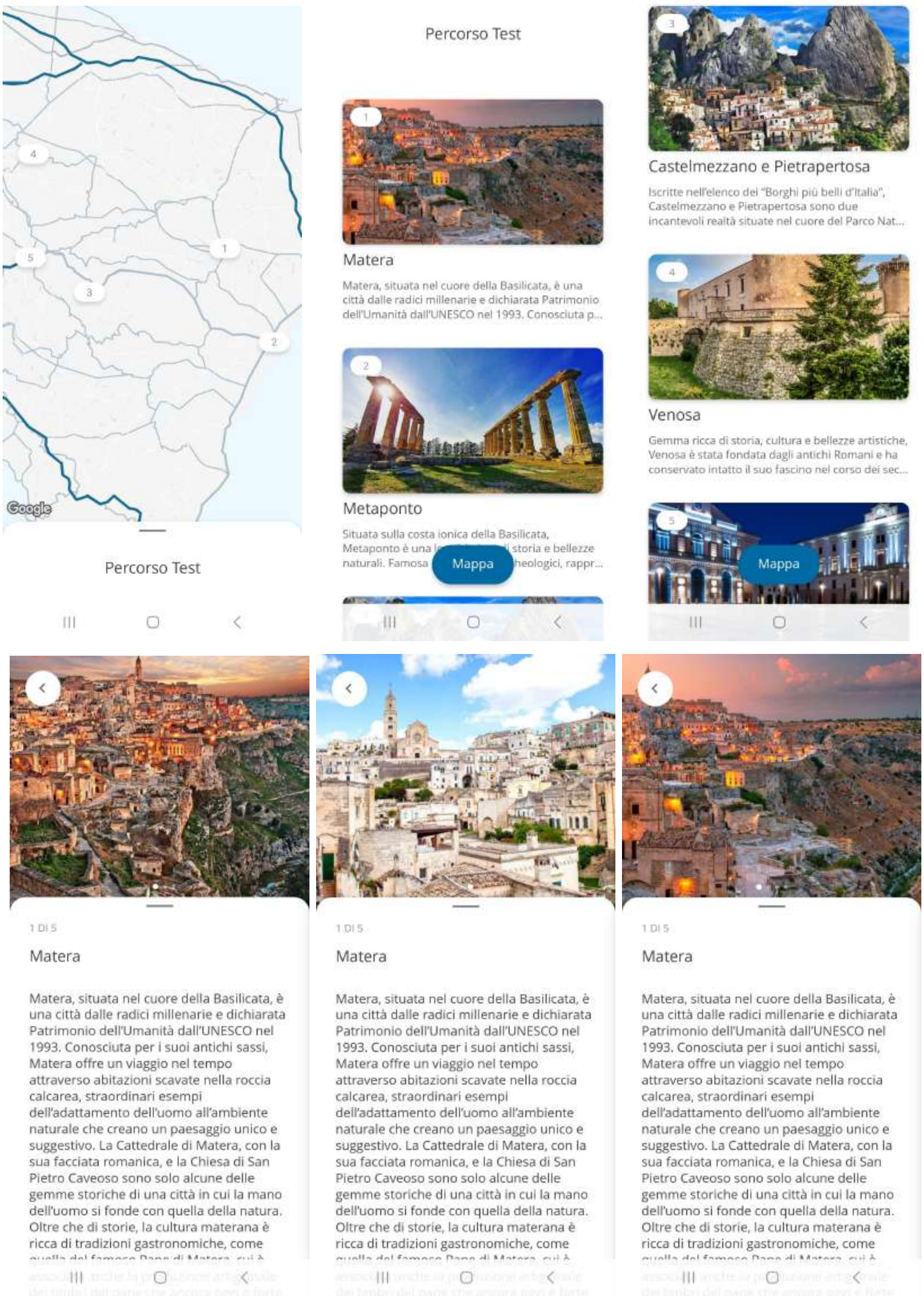

Fig. 18, 19, 20, 21, 22, 23 - Visualizzazione applicativo mobile

Conclusioni

I dimostratori tecnologici realizzati nell'ambito del progetto MOBAS 4.0 si sono rivelati strumenti utili nella verifica e validazione di quanto previsto e implementato nel progetto. Essi hanno consentito di testare le funzionalità in un contesto pratico e di simulare le condizioni reali di utilizzo, rappresentando un ottimo punto di partenza per lo sviluppo futuro di ulteriori funzionalità, offrendo una solida base su cui implementare l'interazione con future tecnologie del settore Automotive e in relazione alle mobilità sostenibile.