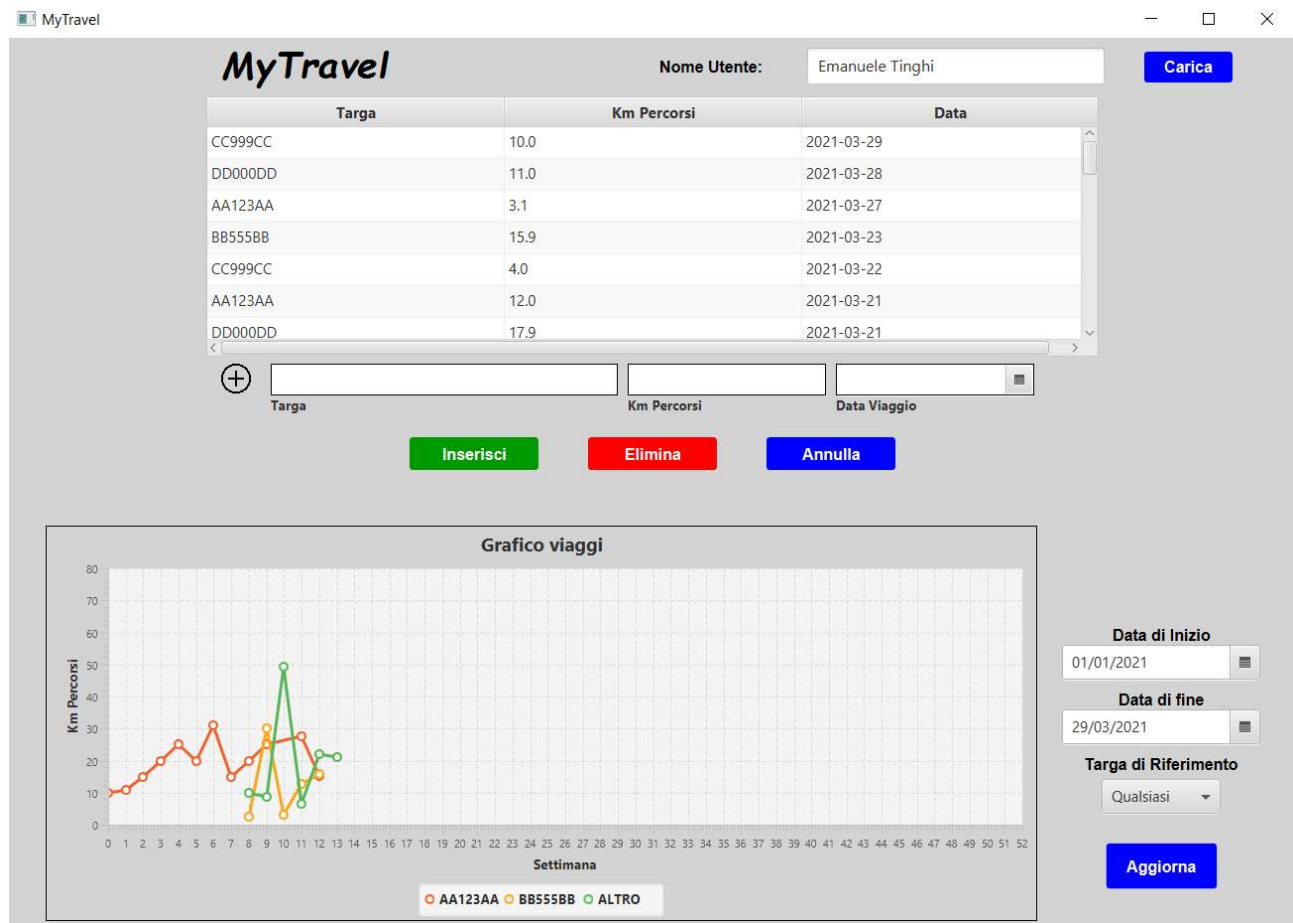


# PROGETTO DI PROGRAMMAZIONE AVANZATA

## Visione statica (Mockup)



## Visione dinamica (Scenario)

### *Caricare i dati*

1. L'utente inserisce il proprio nome utente nel campo Nome Utente in alto.
2. L'utente preme il pulsante Carica.
3. **IF** l'utente non è presente nel database
  - 3.1. L'utente viene registrato.
4. Il sistema carica nella tabella e nel grafico le informazioni relative all'utente.

### ***Inserimento di un viaggio***

1. L'utente inserisce la Targa, i Km e la Data. La data non deve essere successiva a quella attuale.
2. **IF** l'utente preme Nuovo
  - 2.1. Il sistema salva i nuovi dati e svuota il form.
  - 2.2. **IF** la data inserita è compresa nel periodo temporale preso in considerazione
    - 2.2.1 Il sistema aggiunge una nuova riga nella tabella con i nuovi dati.
    - 2.2.2 Il sistema aggiorna il grafico.
3. **ELSE IF** l'utente preme Annulla.
  - 3.1. Il sistema scarta i dati inseriti.

### ***Eliminazione di un viaggio***

1. L'utente clicca sulla riga della tabella in cui si trovano i dati da eliminare.
2. L'utente preme Elimina.
3. Il sistema elimina la riga selezionata dalla tabella.
4. Il sistema aggiorna il grafico.

### ***Modifica del periodo temporale in analisi***

1. L'utente inserisce le nuove Data di inizio e Data di fine.
2. L'utente preme il pulsante Aggiorna.
3. Il sistema aggiorna la tabella.
4. Il sistema aggiorna il grafico.

### ***Modifica della targa di riferimento***

1. L'utente inserisce la nuova Targa di riferimento.
2. L'utente preme il pulsante Aggiorna.
3. Il sistema aggiorna la tabella.
4. Il sistema aggiorna il grafico.

## **File di configurazione locale XML**

All'avvio il sistema legge dal file di configurazione i seguenti dati (se presenti):

- Il numero di righe da mostrare.
- Il numero di giorni **g** precedenti alla data attuale di cui mostrare i dati.
- I colori dei pulsanti NUOVO, ELIMINA, ANNULLA, AGGIORNA e CARICA.
- L'indirizzo IP e la porta del server di backlog.
- L'indirizzo IP e la porta del server database.
- Il nome del database cui accedere
- Username e password per accedere al database.

## **Cache locale degli input**

Alla chiusura il sistema salva su file binario: il Nome Utente, Targa, Km e Data già inseriti dall'utente ma non inviati. Verranno inoltre salvati la Data di inizio, la Data di fine e la Targa di riferimento. All'apertura il sistema carica i suddetti dati.

## **Archivio**

Il sistema archivia i seguenti dati:

- Il Nome Utente di chi ha inserito il viaggio.
- La targa della macchina che ha effettuato il viaggio.
- La distanza percorsa.
- La data del viaggio.

## **File di log remoto in XML**

Il sistema invia una riga di log al verificarsi dei seguenti eventi:

- Avvio dell'applicazione. ("AVVIO")
- Selezione di una riga della tabella. ("SELEZIONE")
- Pressione dei pulsanti ("AGGIORNA", "CARICA", "NUOVO", "ELIMINA", "ANNULLA").
- Chiusura dell'applicazione. ("CHIUSURA")

La riga di log conterrà inoltre: nome dell'applicazione, indirizzo IP del client, timestamp dell'evento e l'etichetta relativa all'azione effettuata

## Documento delle Classi

### Front-end

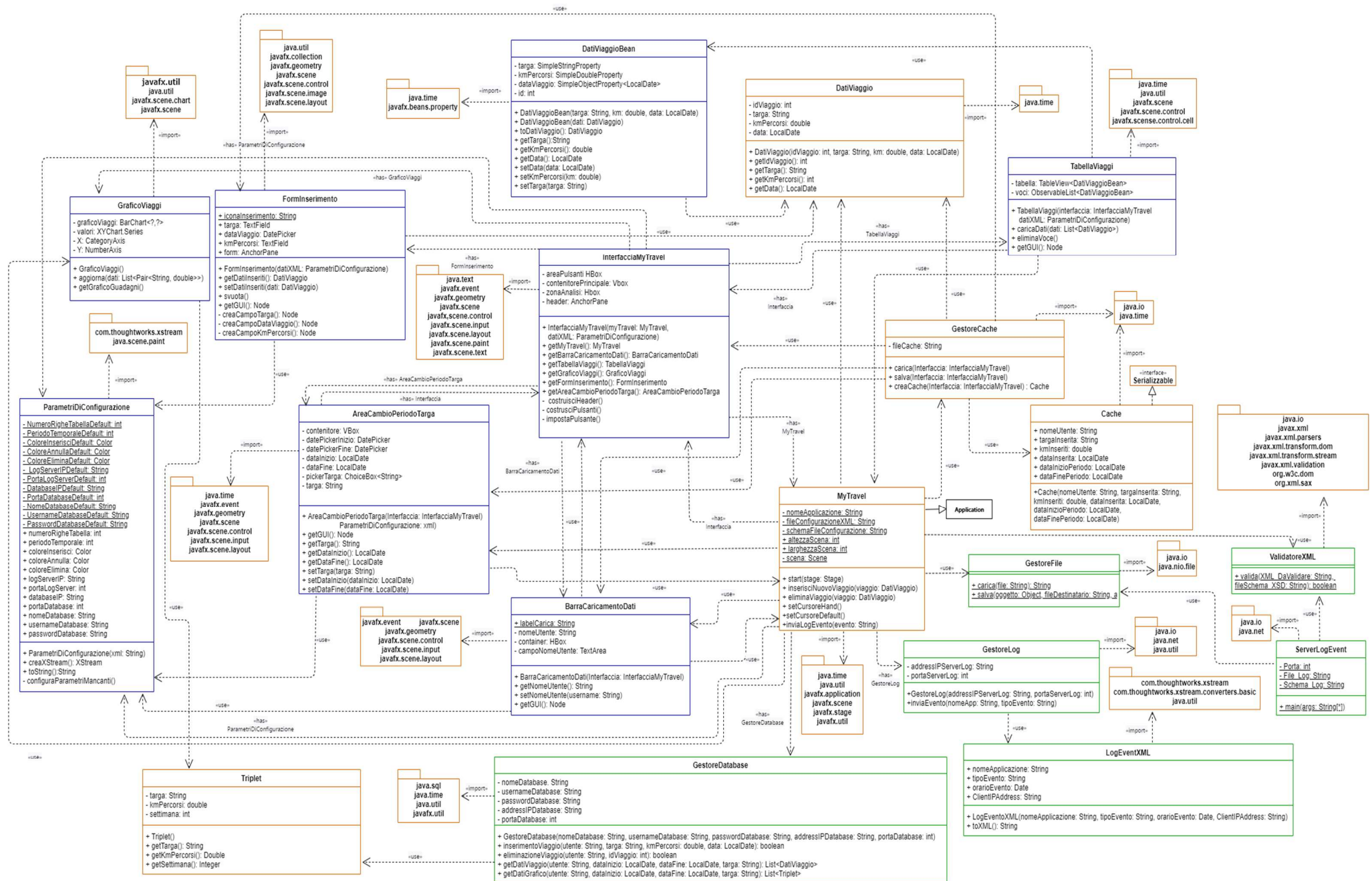
- **InterfacciaMyTravel:** interfaccia grafica dell'applicazione che avrà come obiettivo quello di istanziare gli elementi grafici della GUI e di gestirne l'aspetto. Offre metodi utilizzati dal middleware per inserire/prelevare dati dall'interfaccia. Gestisce inoltre i pulsanti, rispondendo in modo adeguato (mediante apposite procedure) alle richieste dell'utente per quello che riguarda la modifica del database
- **TabellaViaggi:** mostra un certo numero di voci (il numero è definito nel file di configurazione) relative ai viaggi già effettuati. Consente la selezione di una riga al fine di renderla eliminabile.
- **DatiViaggioBean:** classe Bean contenente i dati mostrati in TabellaViaggi
- **BarraCaricamentoDati:** classe che ha la funzione di gestire l'area nella quale l'utente inserisce il "NomeUtente", fornendo inoltre questo dato alle classi middleware che lo richiedono.
- **AreaCambioPeriodoTarga:** area grafica nella quale l'utente può modificare la targa e il periodo temporale presi di riferimento, preoccupandosi anche di lanciare la giusta procedura quando l'utente clicca il pulsante "AGGIORNA"
- **ParametriDiConfigurazione:** deserializza una stringa XML (deve essere già stata validata) passata al costruttore in un oggetto ParametriDiConfigurazione, che viene utilizzata per inizializzare gli attributi della classe, mettendoli quindi a disposizione delle altre classi. In caso di valori mancanti dal file XML, i corrispondenti attributi della classe verranno inizializzati al loro valore di default.
- **FormInserimento:** classe che si occupa di gestire i campi del form adibiti all'inserimento di nuovi dati di viaggio. Si occuperà quindi di passare al gestore del database i dati necessari alla memorizzazione di tale viaggio e di effettuare il salvataggio in cache di queste informazioni nel caso in cui l'utente chiuda l'applicazione prima di aver premuto il tasto "NUOVO".
- **GraficoViaggi:** classe che si occupa di gestire il grafico presente nell'interfaccia grafica mostrando il numero di chilometri percorsi nel periodo di tempo e con la targa specificata dall'utente, dividendo il periodo specificato in settimane. Nel grafico verranno riportati i chilometri percorsi dalle due vetture che nel periodo selezionato hanno viaggiato di più, insieme ai dati aggregati di tutte le altre vetture con le quali l'utente ha viaggiato in tale periodo.

## Middleware

- **MyTravel:** estende Application e rappresenta il controller applicativo. Legge e valida il file di configurazione, istanzia l'interfaccia grafica e gestisce il caso di ripristino della cache. Ha anche il compito, mediante l'utilizzo di specifici metodi, di consentire lo scambio di informazioni tra front-end e back-end.
- **GestoreCache:** offre metodi per caricare/salvare su file binario i dati inseriti, prendendoli dal form.
- **Cache:** implementa Serializable. Rappresenta la struttura che deve avere il file di cache.
- **DatiViaggio:** contiene le informazioni relative a un certo viaggio memorizzato su database
- **Triplet:** contiene le tre informazioni (targa, kmPercorsi, settimana) necessarie al grafico.

## Back-end

- **GestoreFile:** classe finalizzata a fornire metodi necessari alla lettura/scrittura da e su file di testo.
- **ValidatoreXML:** classe dedicata alla validazione dei file di log inviati dalla classe GestoreLog alla classe ServerLogEvent.
- **ServerLogEvent:** classe server dedicata alla raccolta e al salvataggio degli eventi che comportano la creazione di un record di log. Una volta validati, tali file di log vengono salvati e stampati a video.
- **GestoreLog:** classe che ha lo scopo di creare i log e di inviarli al ServerLogEvent.
- **LogEventXML:** contiene le informazioni di un evento di log. Tale classe è serializzabile e viene inviata dalla classe GestoreLog al serverLogEvent.
- **GestoreDatabase:** classe che si occupa di gestire l'accesso al database fornendo metodi per l'inserimento/eliminazione di viaggi e per la lettura di dati di viaggi effettuati in un certo periodo. Si occupa anche di prelevare i dati necessari al grafico.



## Guida all'utilizzo

### Prima di avviare l'applicazione

Prima di avviare l'applicazione sarà necessario avviare il server di log mediante un apposito file batch fornito insieme al codice. Dopo averlo avviato mediante doppio-click sarà possibile avviare l'applicazione.

Il server appena avviato rimarrà in attesa di file di log che, una volta convalidati verranno salvati su file di testo presente nella cartella *files* in modalità append.

Nel caso in cui si ritenga necessario cambiare i parametri di configurazione di default questa azione sarà possibile modificando il file *config.xml* presente nella medesima cartella.

### Avvio dell'applicazione

All'avvio dell'applicazione ci si troverà davanti alla seguente schermata (opportunamente inizializzata con i valori presenti in cache nel caso in cui non si tratti del primo avvio).

MyTravel

Nome Utente:  Carica

Targa	Km Percorsi	Data
Nessun contenuto nella tabella		

Targa Km Percorsi Data Viaggio

Inserisci Elimina Annulla

**Grafico viaggi**

Km Percorsi

Settimana

Data di Inizio: 27/02/2021

Data di fine: 29/03/2021

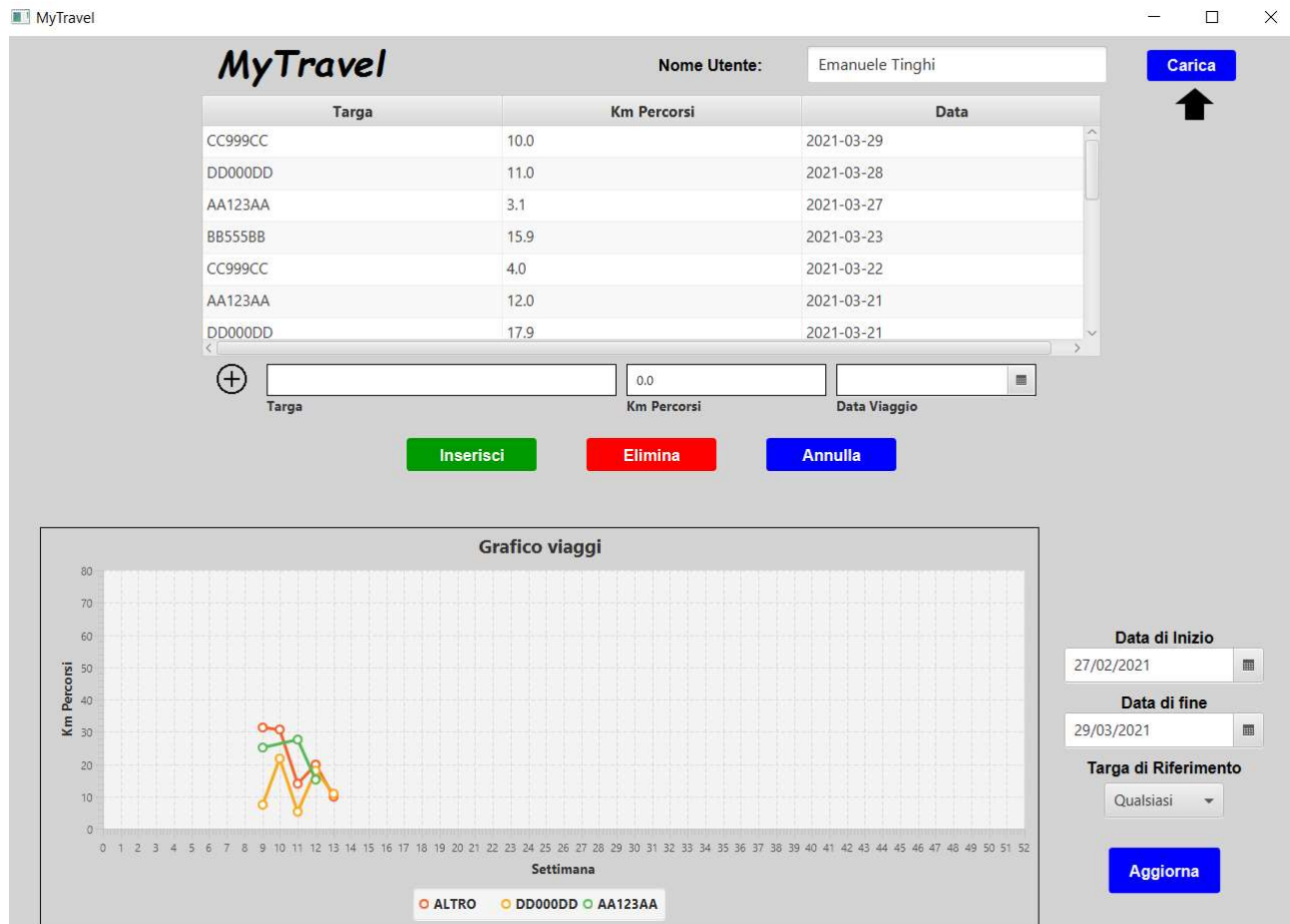
Targa di Riferimento:

Aggiorna

Il server di log riceverà un log che testimonierà l'effettivo avvio dell'applicazione.

## 1° Caso d'uso: "Caricamento dei dati"

L'utente che ha avviato l'applicazione potrà quindi richiedere che i propri dati vengano caricati. Per effettuare questa azione sarà necessario che l'utente inserisca il proprio nome utente nel campo NomeUtente e che prema il tasto Carica.



Anche in questo caso il server di log riceverà un log relativo all'avvenuto caricamento dei dati.



## 2° Caso d'uso: "Inserimento di un viaggio"

Una volta caricati i propri dati sarà possibile per l'utente inserire nuovi viaggi. Per fare ciò l'utente dovrà riempire l'apposito form posizionato sotto la tabella e premere in seguito il pulsante "Inserisci". Nel caso in cui l'utente esca dall'applicazione dopo aver inserito i dati ma prima di averli inviati, il sistema si preoccuperà di salvarli in cache in modo da ripresentarli immediatamente al successivo avvio.

MyTravel

Nome Utente: Emanuele Tinghi

Carica

Targa	Km Percorsi	Data
CC999CC	10.0	2021-03-29
DD000DD	11.0	2021-03-28
AA123AA	3.1	2021-03-27
BB555BB	15.9	2021-03-23
CC999CC	4.0	2021-03-22
AA123AA	12.0	2021-03-21
DD000DD	17.9	2021-03-21

AA123AA

25.0

29/03/2021

TargaKm PercorsiData Viaggio

InserisciEliminaAnnulla

Grafico viaggi

Km Percorsi

Settimana

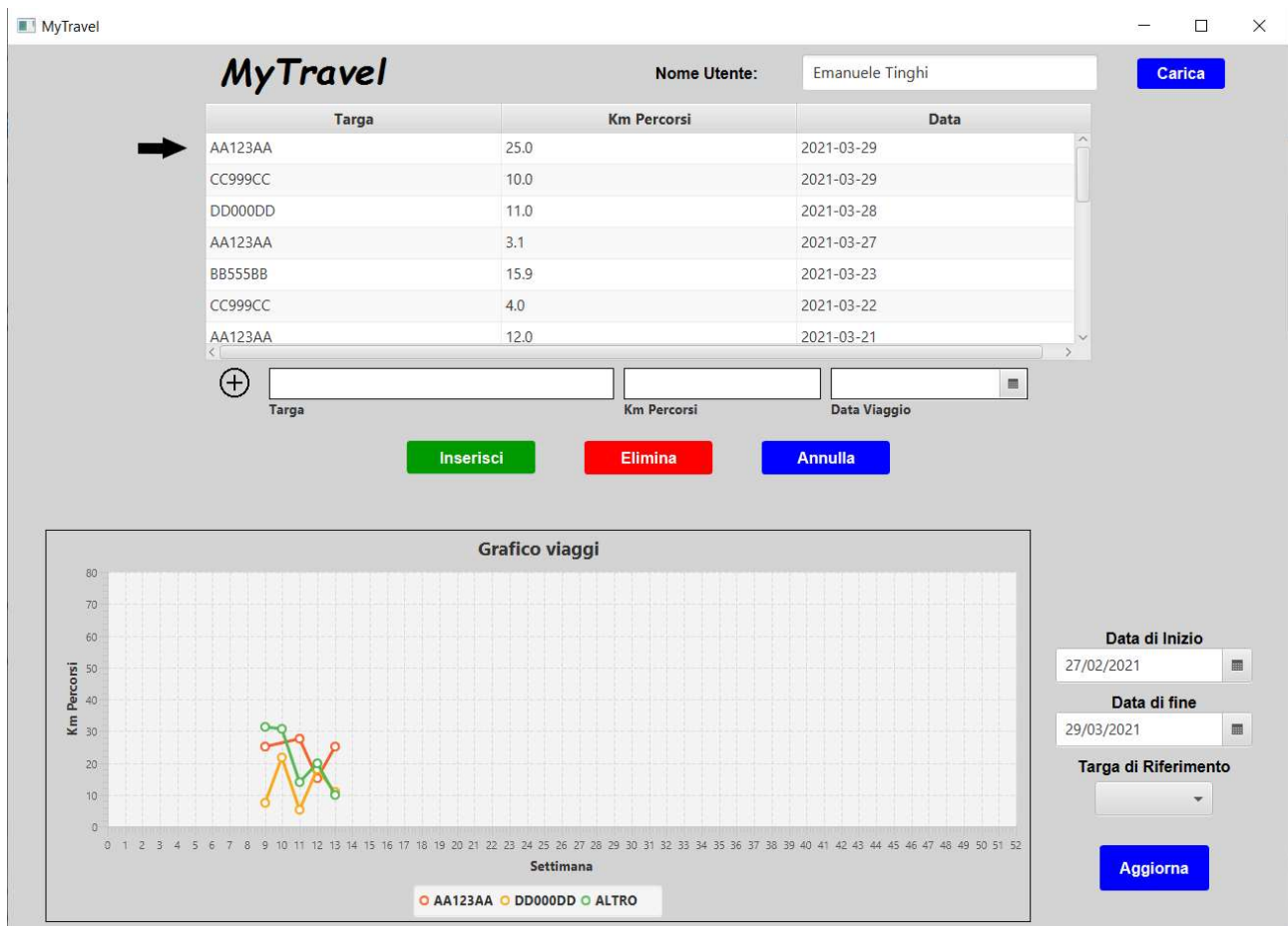
ALTRO DD000DD AA123AA

Data di Inizio27/02/2021

Data di fine29/03/2021

Targa di RiferimentoQualsiasi

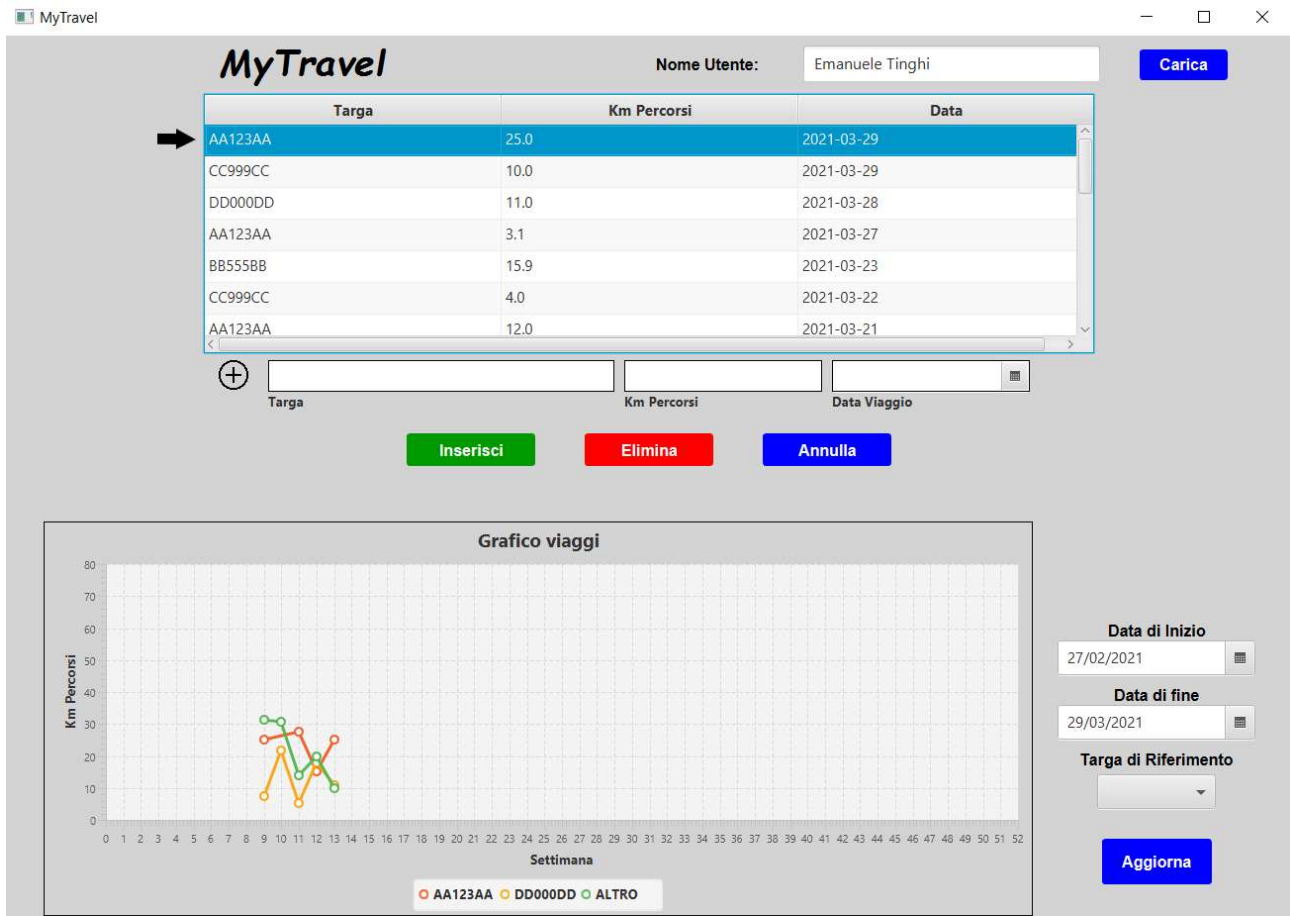
Aggiorna

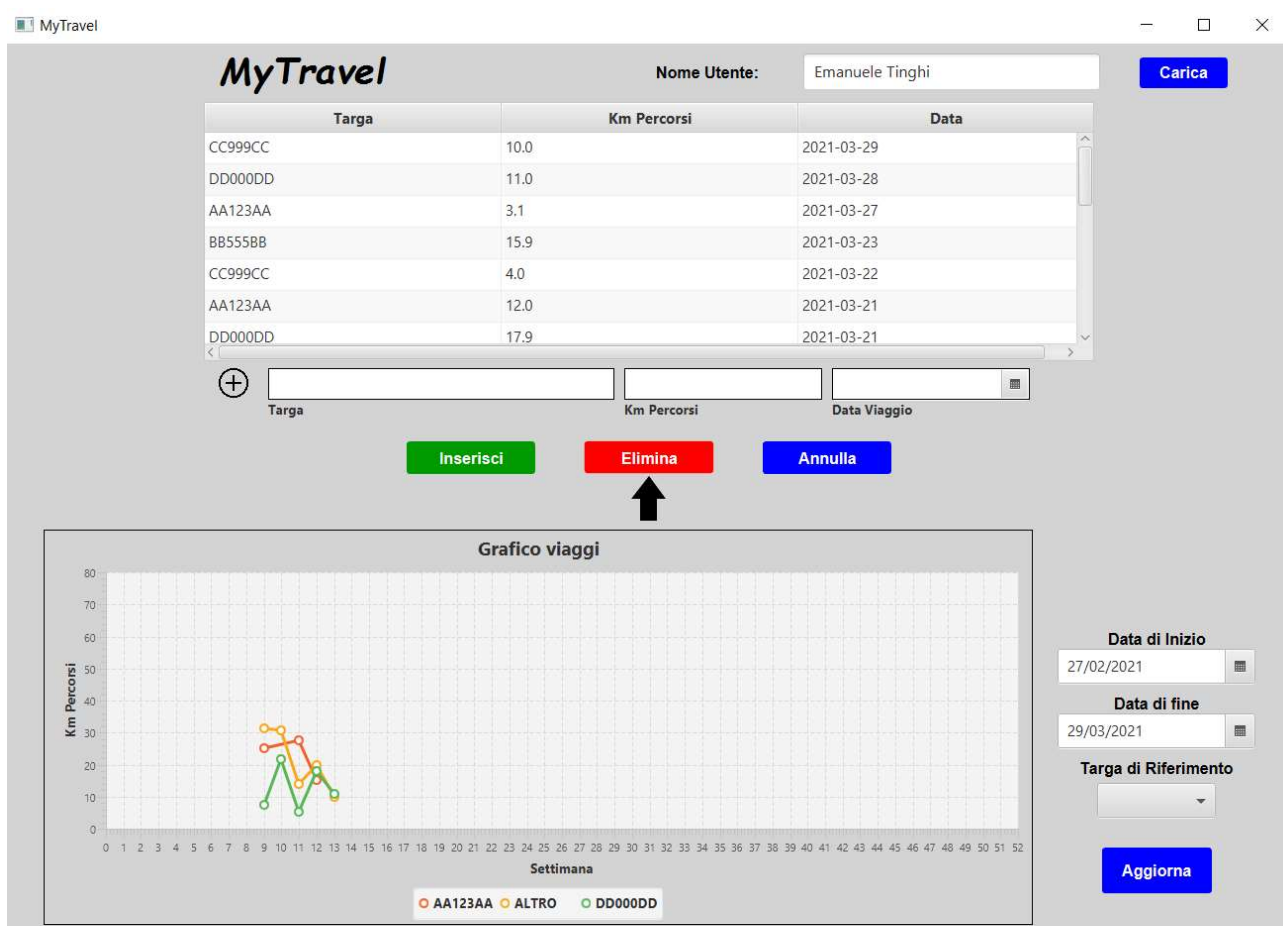


Il server di log riceverà un log relativo all'avvenuto inserimento dei dati.

### 3° Caso d'uso: "Eliminazione di un viaggio"

Una volta caricati i propri dati sarà possibile per l'utente eliminare vecchi viaggi. Per effettuare tale azione sarà necessario selezionare dalla tabella la riga da eliminare e in seguito premere il pulsante Elimina.

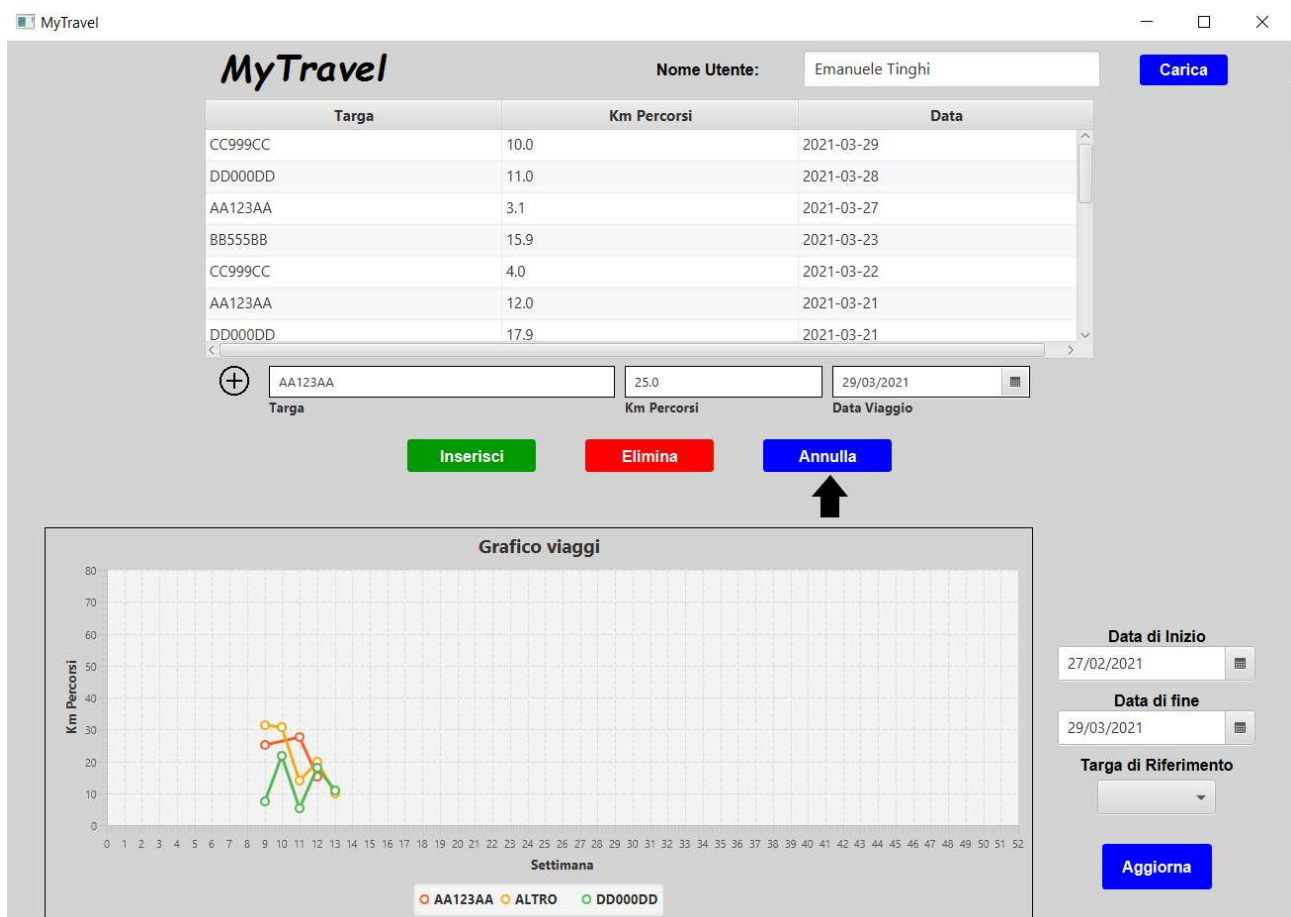


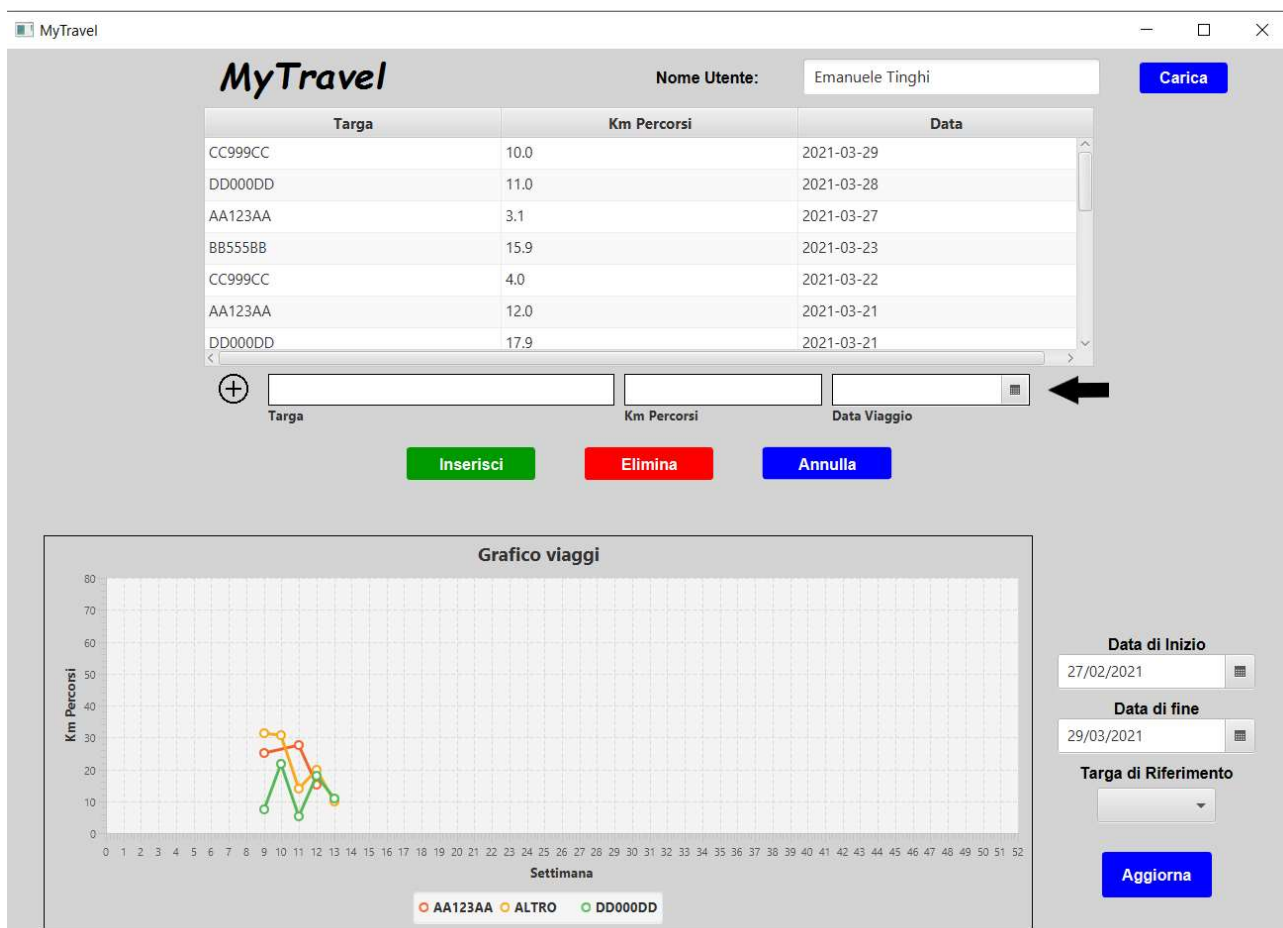


Il server di log riceverà un log relativo all'avvenuta eliminazione dei dati.

#### 4° Caso d'uso: "Svuotamento del form"

Una volta inseriti i dati di un viaggio nel form sarà possibile per l'utente eliminarli da quest'ultimo premendo sul pulsante Annulla.

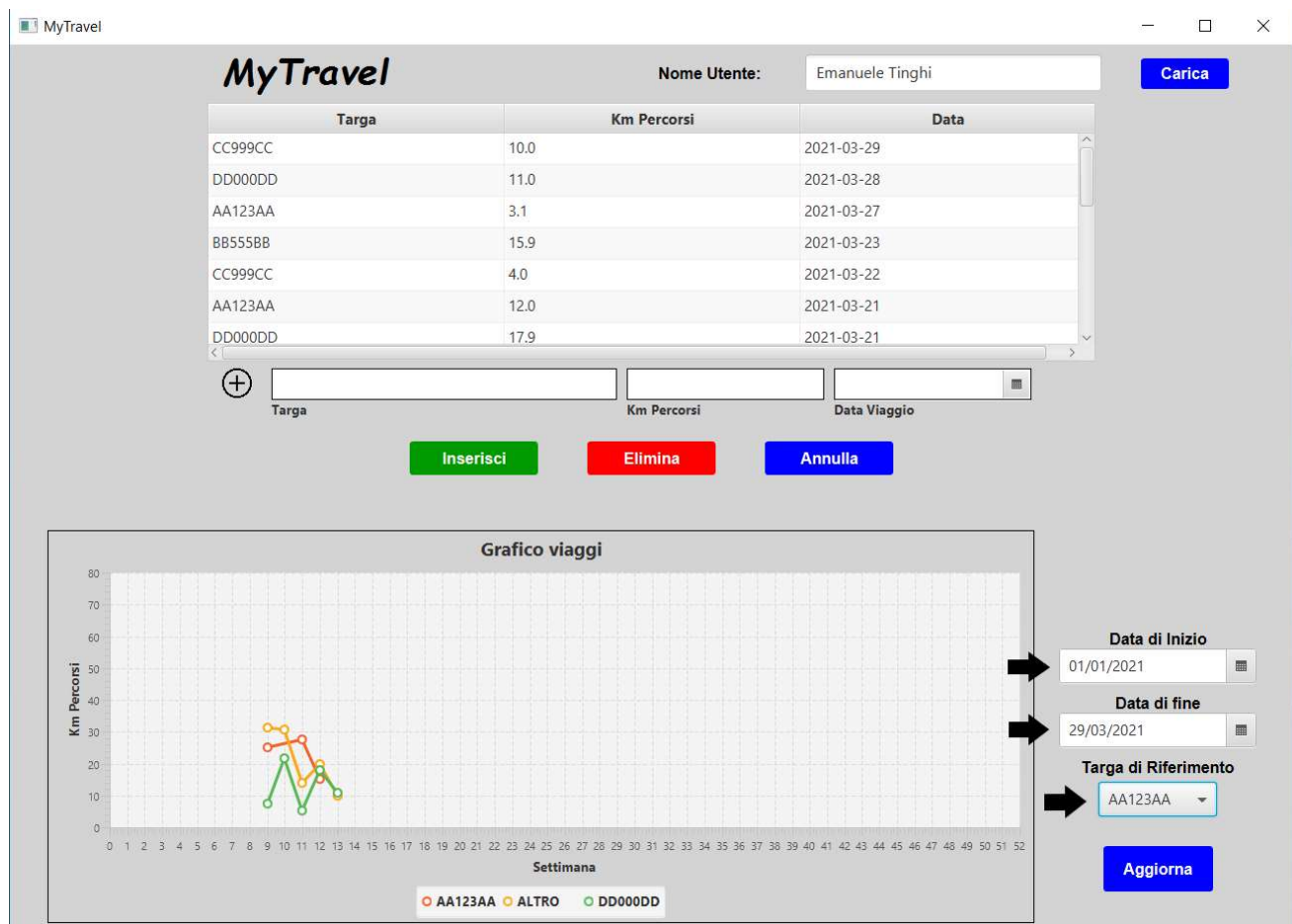


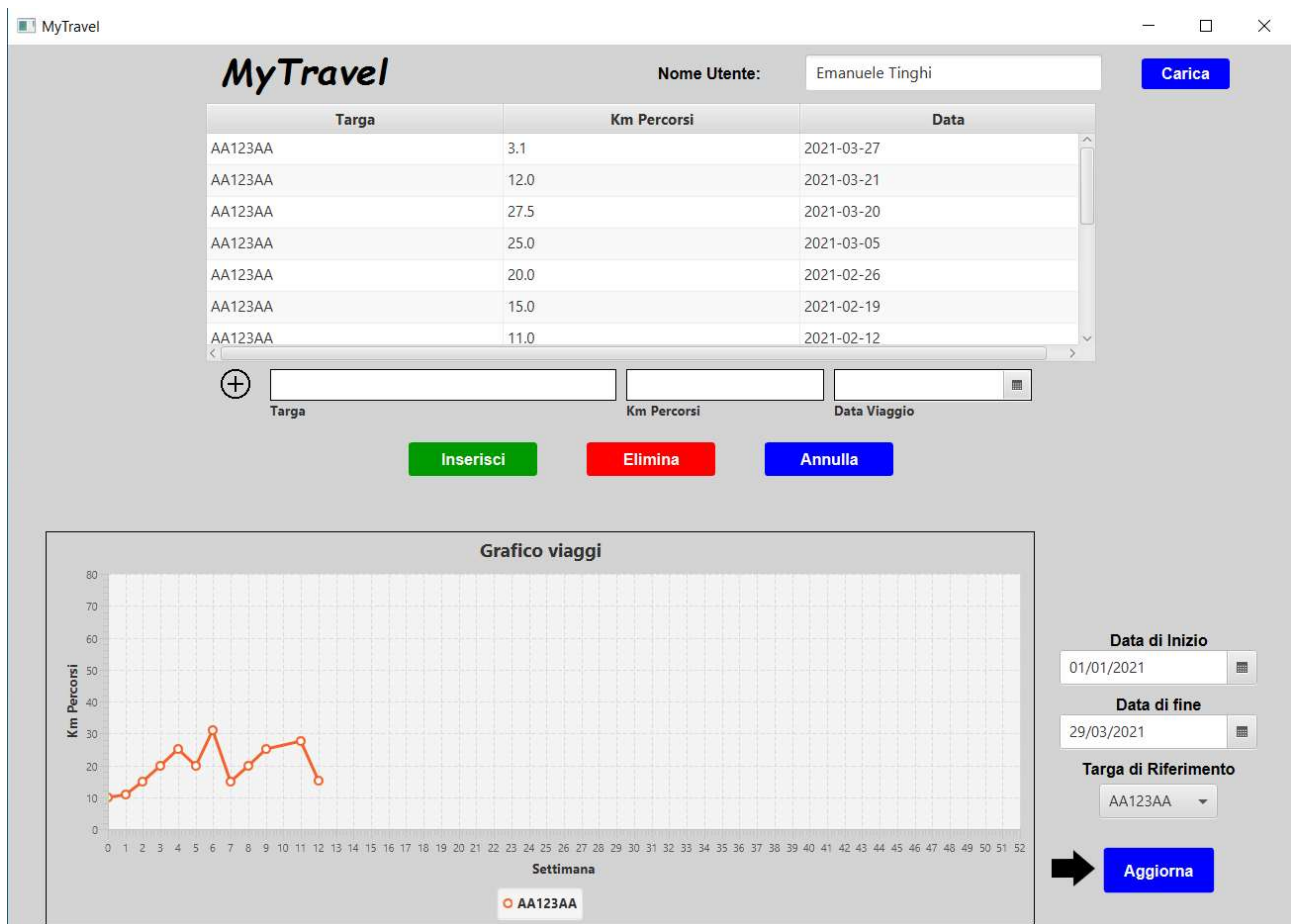


Il server di log riceverà un log relativo all'avvenuta svuotamento del form.

## 5° Caso d'uso: "Modifica del periodo temporale/targa di riferimento"

Una volta caricati i propri dati sarà possibile per l'utente modificare il periodo temporale e la targa di riferimento in modo da aggiornare il grafico e la tabella per vedere le informazioni volute. Per effettuare questa azione l'utente dovrà selezionare le date nei due datePicker Data di Inizio e Data di Fine (nel caso voglia cambiare il periodo) e la targa nel ChoiceBox Targa di Riferimento (nel caso voglia cambiare la targa). Una volta selezionate le informazioni volute l'utente dovrà cliccare sul pulsante Aggiorna per provocare l'aggiornamento della tabella e del grafico.





Il server di log riceverà un log relativo all'avvenuta modifica dei parametri di riferimento.

## Chiusura dell'applicazione

Alla chiusura dell'applicazione le informazioni contenute nel Form di inserimento, NomeUtente, i valori dei DatePicker e della targa selezionati, il contenuto della tabella e la riga eventualmente selezionata vengono memorizzate nel file di cache e viene inviato al server di Log un evento di chiusura.