

MANUALE UTENTE

Da quanto detto in precedenza, si è evinto che il sistema software comprende tre distinti componenti: **un server** responsabile delle attività di clustering, del salvataggio e del caricamento di file, e **due client** che consentono l'interazione con il server. Entrambi i client, che siano basati su console o presentino un'interfaccia grafica, sono in grado di utilizzare lo stesso server. Per agevolare l'usabilità, sia il client che il server operano con un indirizzo IP e una porta predefiniti, i quali possono essere modificati intervenendo sui file.bat appropriati.

REQUISITI PER IL CORRETTO FUNZIONAMENTO DEL SOFTWARE

Al fine di avviare l'applicazione software in questione, per garantire un funzionamento corretto, è essenziale soddisfare i seguenti requisiti d'uso del programma:

1. La presenza e l'attivazione del sistema di gestione dei database MySQL.
2. L'utilizzo della versione adeguata del kit di sviluppo Java (JDK).
3. L'avvio preliminare del server prima di avviare il client.
4. L'esecuzione dello script denominato "createDB.sql" al fine di inizializzare il database.

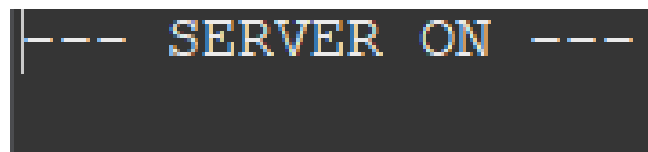
GUIDA DI INSTALLAZIONE PER CODICE SORGENTE

Prima di utilizzare il software, è essenziale garantire il rispetto dei requisiti menzionati in precedenza.

Per automatizzare l'avvio del server e del client, è possibile utilizzare i file batch denominati *K-Means_Server.bat* e *K-Means_Client.bat* tramite una semplice doppia pressione. Per gli utenti più esperti, c'è anche la possibilità di avviare il server e il client tramite il terminale, utilizzando il comando ``java -jar nomeFile.jar`` all'interno della cartella dei file *JAR*.

AVVIO SERVER:

In seguito all'esecuzione del file *K-Means_Server.bat*, il server è stato avviato con successo. L'interfaccia visualizzata conferma che il server è pronto per accogliere e gestire le richieste in arrivo dai client.



AVVIO CLIENT:

All'avvio di *K-Means_Client.bat*, viene inizializzato il client e vengono mostrate le informazioni di connessione insieme al menu delle opzioni disponibili per la scelta. L'utente ha la possibilità di scegliere tra due alternative: caricare i dati da un file o individuare nuovi cluster da una tabella di un database.

```
addr = localhost/127.0.0.1
Socket[addr=localhost/127.0.0.1,port=8080,localport=50502]
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta:
```

Nel caso si selezioni l'opzione 1 dal menu precedente, sarà consentito il recupero dei dati da un file. Sarà richiesto di specificare il nome del database, il nome della tabella e il numero dei cluster precedentemente calcolati. Questo perché il file è salvato con il formato nomeDatabaseNomeTabellaNumeroCluster.dat:

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 1
Inserisci nome database: MapDB
Inserisci nome tabella: playtennis
Inserisci numero di cluster: 3
0: Centroid=(overcast 16.862000000000002 normal strong yes )
1: Centroid=(rain 5.0200000000000005 normal weak yes )
2: Centroid=(sunny 21.525 high strong no )

Vuoi ripetere l'esecuzione? (y/n) n
Vuoi scegliere una nuova operazione da menu? (y/n)
```

Dopo aver visualizzato l'output, il programma offre la possibilità di scegliere se si desidera ripetere l'esecuzione. Se si dovesse optare di non ripetere l'operazione, si può decidere di ritornare al menu principale, in caso contrario il programma termina.

Se si dovesse selezionare l'operazione 2 dal menu, sarà consentito eseguire nuove operazioni di clusterizzazione, permettendo di salvare i risultati in un file. Per eseguire la clusterizzazione, sarà richiesto di specificare se si vogliono usare i valori predefiniti (localhost-MapDB-playtennis-UserMap-map) o di inserirli manualmente.

```
Scegli una opzione
(1) Carica Cluster da File
(2) Carica Dati
Risposta: 2
Vuoi usare i valori di default? (y/n)
```

si dovesse decidere di inserirli:

```
Vuoi usare i valori di default? (y/n) n
Inserisci il nome del server (ad esempio localhost): input1
Inserisci il nome del database (ad esempio MapDB): input2
Inserisci il nome della tabella (ad esempio playtennis): input3
Inserisci il nome dell'utente (ad esempio MapUser): input4
Inserisci la password dell'utente (ad esempio map): input5
```

In seguito si deve controllare l'esistenza del database con i dati inseriti. Se non dovesse esistere:

```
Vuoi usare i valori di default? (y/n) n
Inserisci il nome del server (ad esempio localhost): input1
Inserisci il nome del database (ad esempio MapDB): input2
Inserisci il nome della tabella (ad esempio playtennis): input3
Inserisci il nome dell'utente (ad esempio MapUser): input4
Inserisci la password dell'utente (ad esempio map): input5
ClientServerException: [!] SQLException: Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not received any packets from the server.
[!] SQLState: 08S01
[!] VendorError: 0

Reinserisci i dati

Vuoi usare i valori di default? (y/n)
```

Altrimenti se il DB dovesse esistere, in entrambi i casi di scelta (valori di default o inserimento manuale) viene richiesto il numero dei cluster. Una volta inserito, ci sarà la visualizzazione dei risultati con conseguente richiesta di ripetere l'operazione o meno

```
Numero di cluster: 4
Numero di Iterazioni: 3
0: Centroid=(overcast 13.5775 normal strong yes)
Examples:
[overcast 0.1 normal strong yes] dist=0.4448019801980198
[sunny 12.5 normal strong yes] dist=1.0355610561056106
[overcast 12.5 high strong yes] dist=1.0355610561056106
[overcast 29.21 normal weak yes] dist=1.515924092409241
AvgDistance=1.0079620462046206

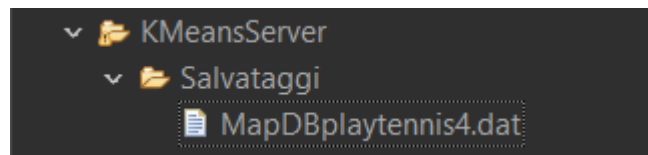
1: Centroid=(rain 6.275 normal weak yes)
Examples:
[rain 13.0 high weak yes] dist=1.221947194719472
[rain 0.0 normal weak yes] dist=0.2070957095709571
[sunny 0.1 normal weak yes] dist=1.203795379537954
[rain 12.0 normal weak yes] dist=0.1889438943894389
AvgDistance=0.7054455445544555

2: Centroid=(rain 6.25 normal strong no)
Examples:
[rain 0.0 normal strong no] dist=0.20627062706270627
[rain 12.5 high strong no] dist=1.2062706270627062
AvgDistance=0.7062706270627062

3: Centroid=(sunny 25.9 high weak no)
Examples:
[sunny 30.3 high weak no] dist=0.1452145214521453
[sunny 30.3 high strong no] dist=1.1452145214521452
[overcast 30.0 high weak yes] dist=2.1353135313531353
[sunny 13.0 high weak no] dist=0.4257425742574257
AvgDistance=0.9628712871287128

Vuoi ripetere l'esecuzione? (y/n) y
```

Successivamente verrà salvato il file MapDBplaytennis4.dat nella cartella Salvataggi:



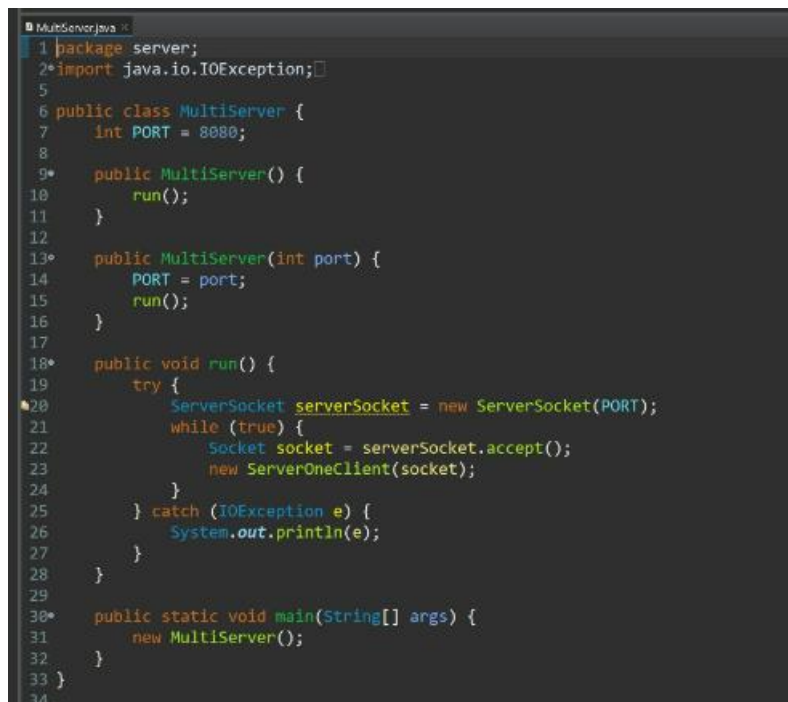
Va specificato che durante l'esecuzione del client, se l'utente dovesse inserire informazioni sbagliate (ex: tabelle di mapDB vuote, inesistenti oppure un numero di cluster non valido), verrebbero visualizzati messaggi di errore. Ad ogni errore, sarà necessario reinserire i dati corretti per poter procedere con i calcoli e le letture dei file. Qualora il server venga spento durante l'esecuzione del client, quest'ultimo verrà chiuso.

GUIDA DI INSTALLAZIONE PER CODICE CON INTERFACCIA

Prima Se si dovesse decidere di avviare l'applicazione scegliendo come IDE Eclipse, di seguito verrà riportato la configurazione lato server e client al fine di un corretto utilizzo del programma.

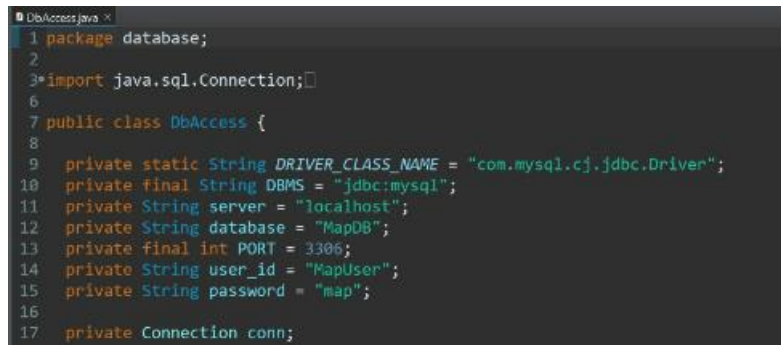
AVVIO SERVER

L'applicazione si avvierà di default sulla porta 8080. Se si volesse modificare in seguito la porta, basterà andare nella classe *MultiServer* posizionato nel package *Server* in *KMeansServer*. e procedere con la modifica.



```
1 package server;
2 import java.io.IOException;
3
4
5
6 public class MultiServer {
7     int PORT = 8080;
8
9     public MultiServer() {
10         run();
11     }
12
13     public MultiServer(int port) {
14         PORT = port;
15         run();
16     }
17
18     public void run() {
19         try {
20             ServerSocket serverSocket = new ServerSocket(PORT);
21             while (true) {
22                 Socket socket = serverSocket.accept();
23                 new ServerOneClient(socket);
24             }
25         } catch (IOException e) {
26             System.out.println(e);
27         }
28     }
29
30     public static void main(String[] args) {
31         new MultiServer();
32     }
33 }
34
```

Il server è stato allegato al progetto, e per il corretto funzionamento sarà necessario avere attivo il software MySQL e avere le corrette credenziali per poter accedere al server (di seguito riportate).



```

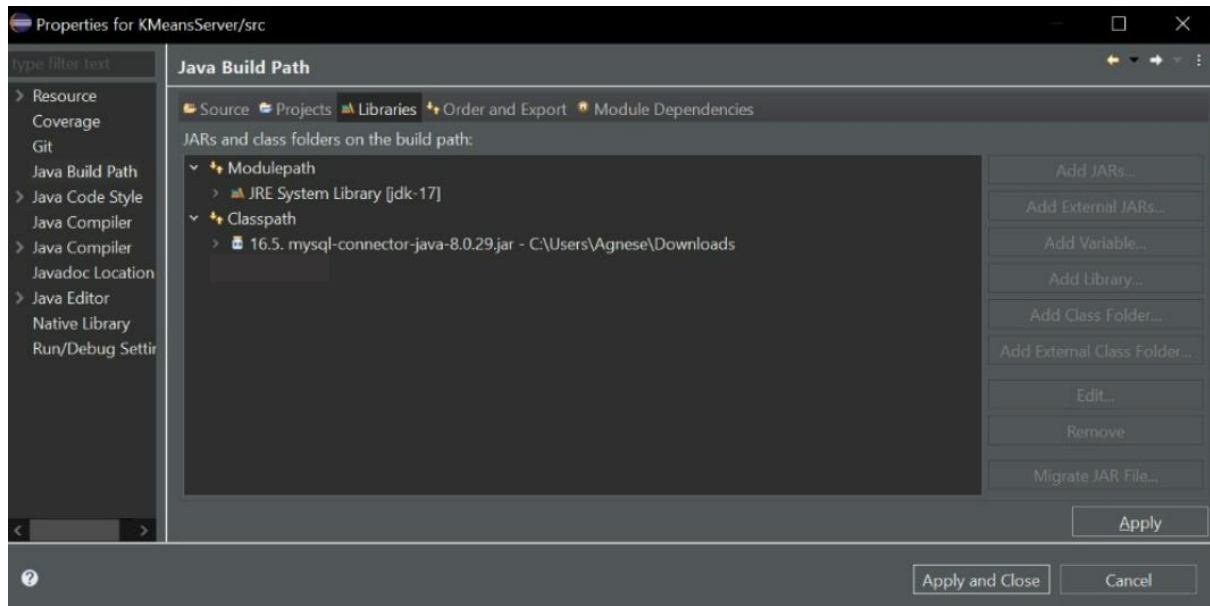
1 package database;
2
3 import java.sql.Connection;
4
5
6
7 public class DbAccess {
8
9     private static String DRIVER_CLASS_NAME = "com.mysql.cj.jdbc.Driver";
10    private final String DBMS = "jdbc:mysql";
11    private String server = "localhost";
12    private String database = "MapDB";
13    private final int PORT = 3306;
14    private String user_id = "MapUser";
15    private String password = "map";
16
17    private Connection conn;

```

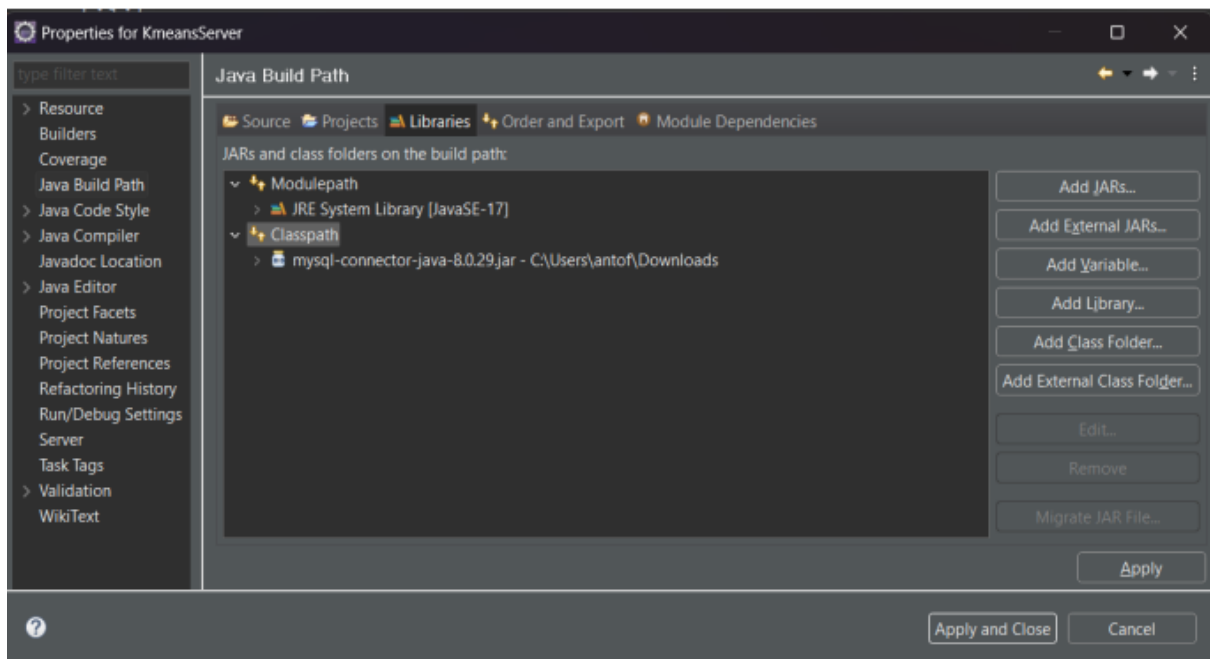
È possibile apportare modifiche alle informazioni presenti nella classe *DbAccess*, ma ciò richiede la creazione di un nuovo utente con credenziali identiche all'interno del database MySQL.

È essenziale tenere a mente l'aggiunta del connettore necessario per stabilire la connessione con il database. Per incorporarlo nel progetto, si devono seguire i seguenti passaggi:

1. Fare clic con il tasto destro del mouse su *KmeansServer*.
2. Selezionare Proprietà (*Properties*).
3. Successivamente, accedere a *Java Build Path*.
4. All'interno della sezione *Libraries*, selezionare *Classpath*.
5. Cliccare sull'opzione "*Add External Jars*" posizionata a destra.
6. Scegliere il connettore MySQL compatibile con la versione utilizzata.
7. Terminare con "*Apply and close*"



È fondamentale seguire attentamente queste istruzioni per garantire che il connettore appropriato venga integrato nel progetto, consentendo così l'efficace collegamento con il database MySQL senza compromettere il funzionamento della classe DbAccess.



AVVIO CLIENT

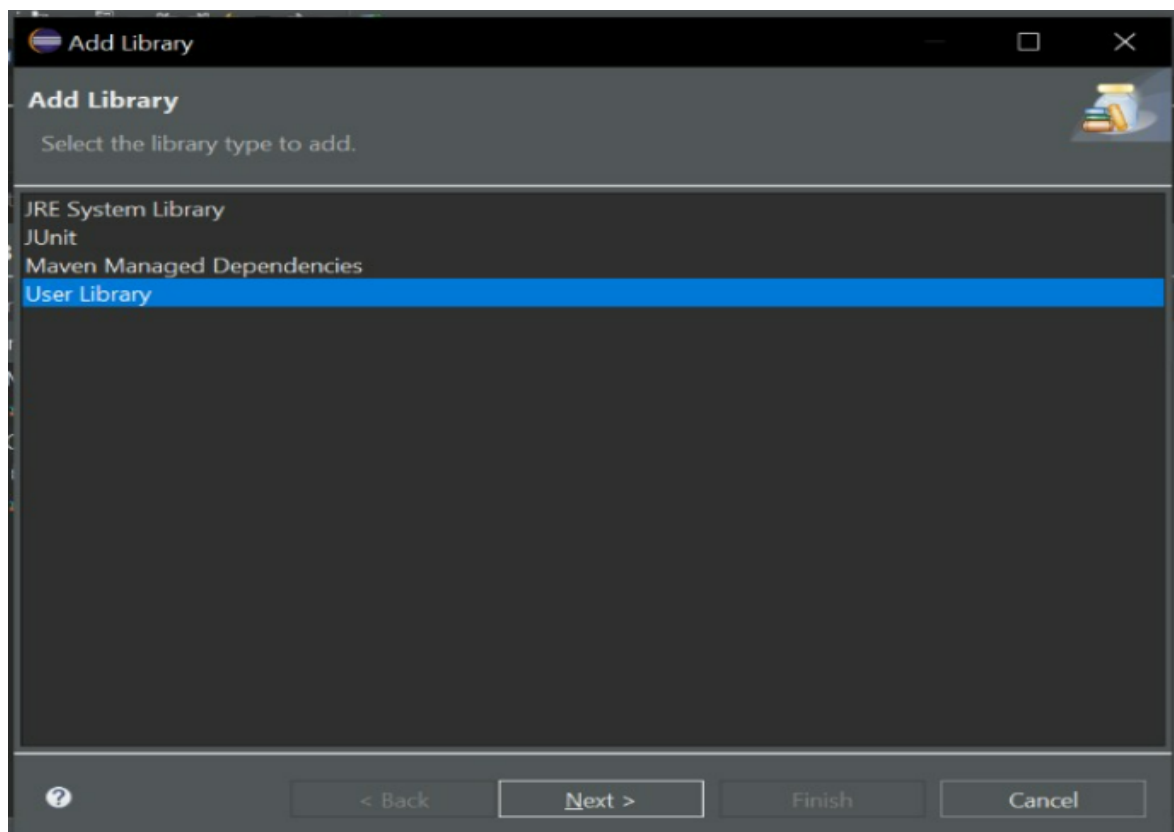
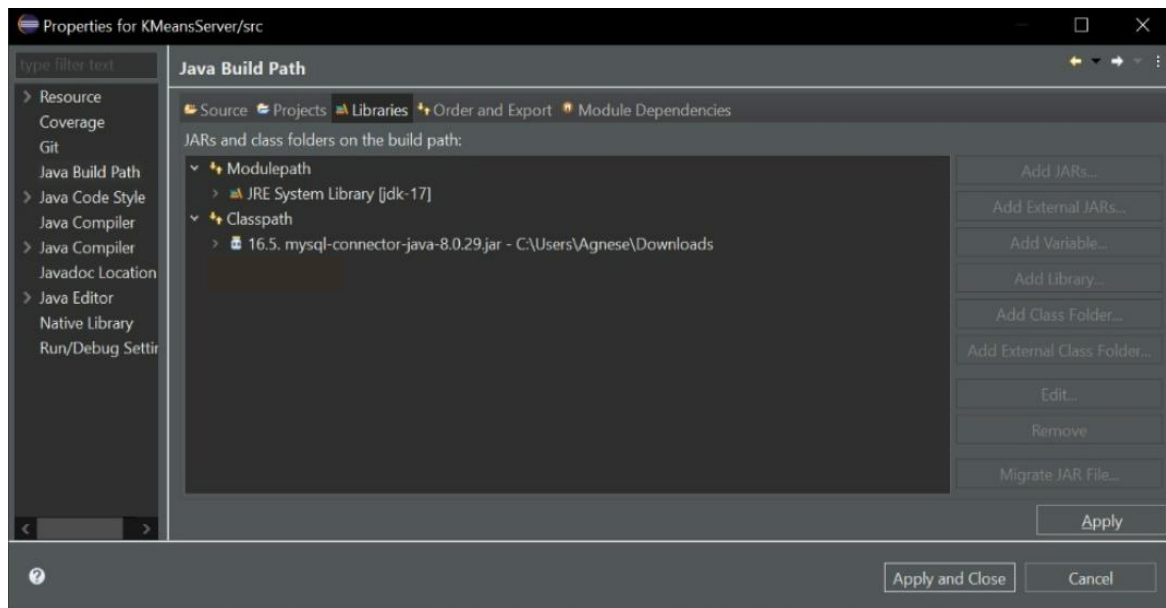
Nel lato del client, è essenziale configurare correttamente l'ambiente di esecuzione e integrare la libreria *JavaFX* per aprire l'interfaccia. Per ottenere una configurazione adeguata, segui le istruzioni di setup.



Per aggiungere la libreria al progetto, avvia con un clic destro sul progetto "*KMeansClient*" e seleziona "Proprietà". Successivamente, vai su "*Java Build Path*" e quindi seleziona "*Librerie*". Quando sei nella sezione "*Classpath*", procedi a fare clic su "*Aggiungi libreria*", dopodiché seleziona "*Libreria utente*" e poi "*Librerie utente*". Verrà visualizzata una nuova finestra, dove dovrai premere il pulsante "*Nuovo*" per creare una libreria con un nome a tua scelta. Successivamente, seleziona "*Aggiungi file esterni*" e incorpora tutti i file contenuti nella cartella "lib" che si trova all'interno del pacchetto "*javafx-sdk-20.0.1*".

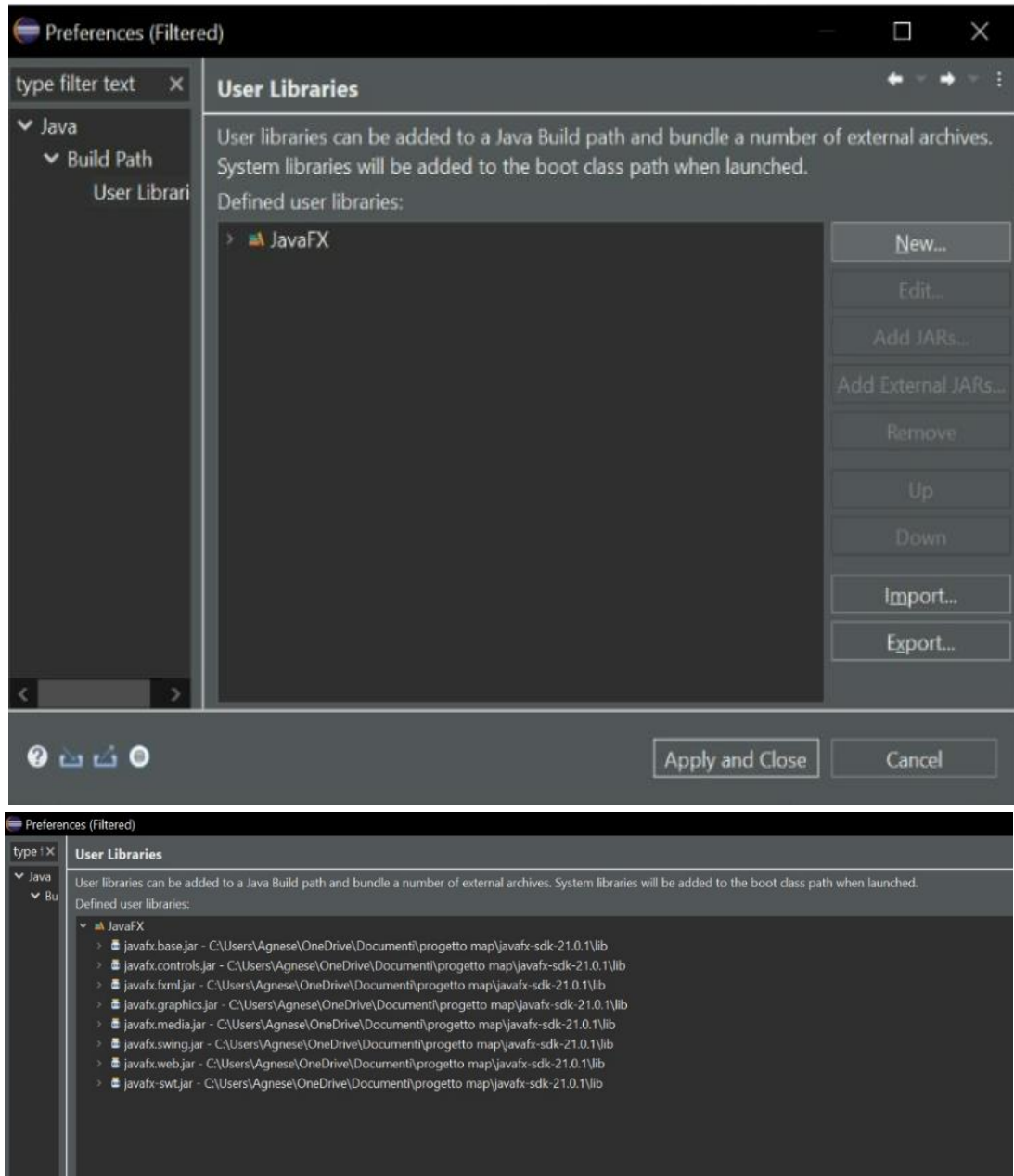
Per impostare correttamente la configurazione di esecuzione nel lato client e integrare la libreria *JavaFX* per aprire l'interfaccia, segui questi passaggi:

1. Apri il progetto "*KmeansClient*" e accedi alle "*Proprietà*" facendo clic destro sul progetto.
2. Seleziona "*Java Build Path*" e successivamente fai clic su "*Librerie*".
3. All'interno della sezione "*Classpath*", premi su "Add Library", quindi opta per "*User Library*" e successivamente "*User libraries*".



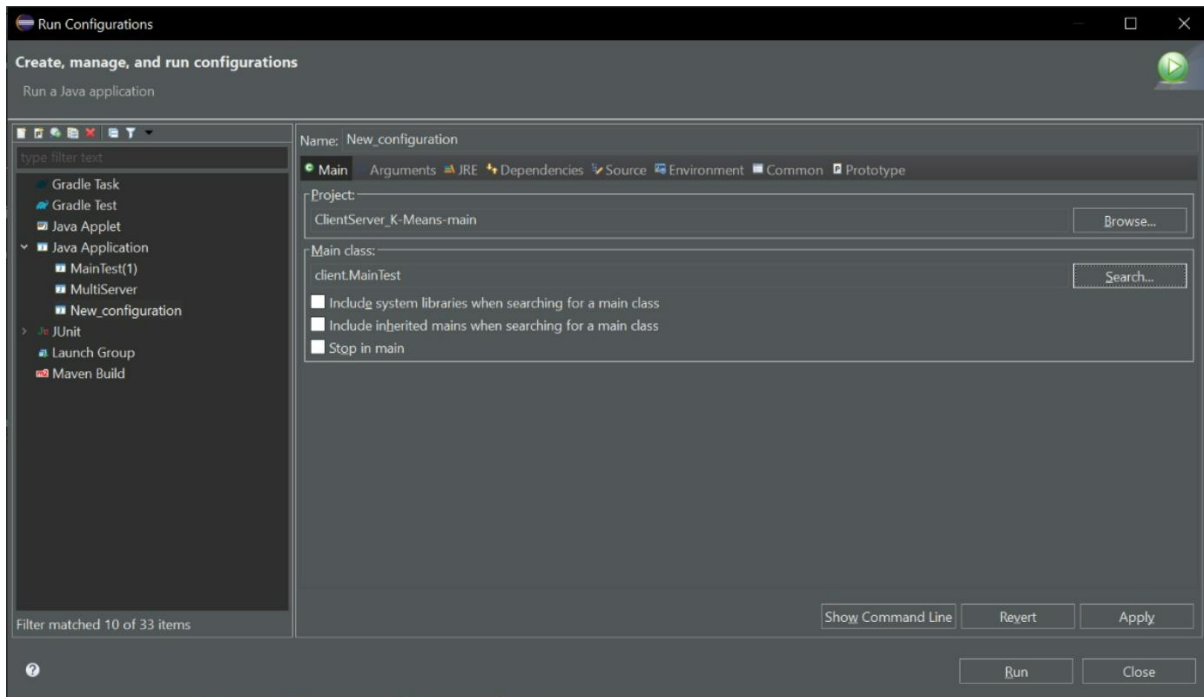
4. Si aprirà una finestra; premi su "New" per creare una libreria e assegna un nome a tua scelta.

5. Successivamente, seleziona "*Aggiungi file esterni*" e integra tutti i file contenuti nella cartella "*lib*" che si trova nel pacchetto "*javafx-sdk-20.0.1*" (questa cartella è inclusa nello zip allegato al progetto).



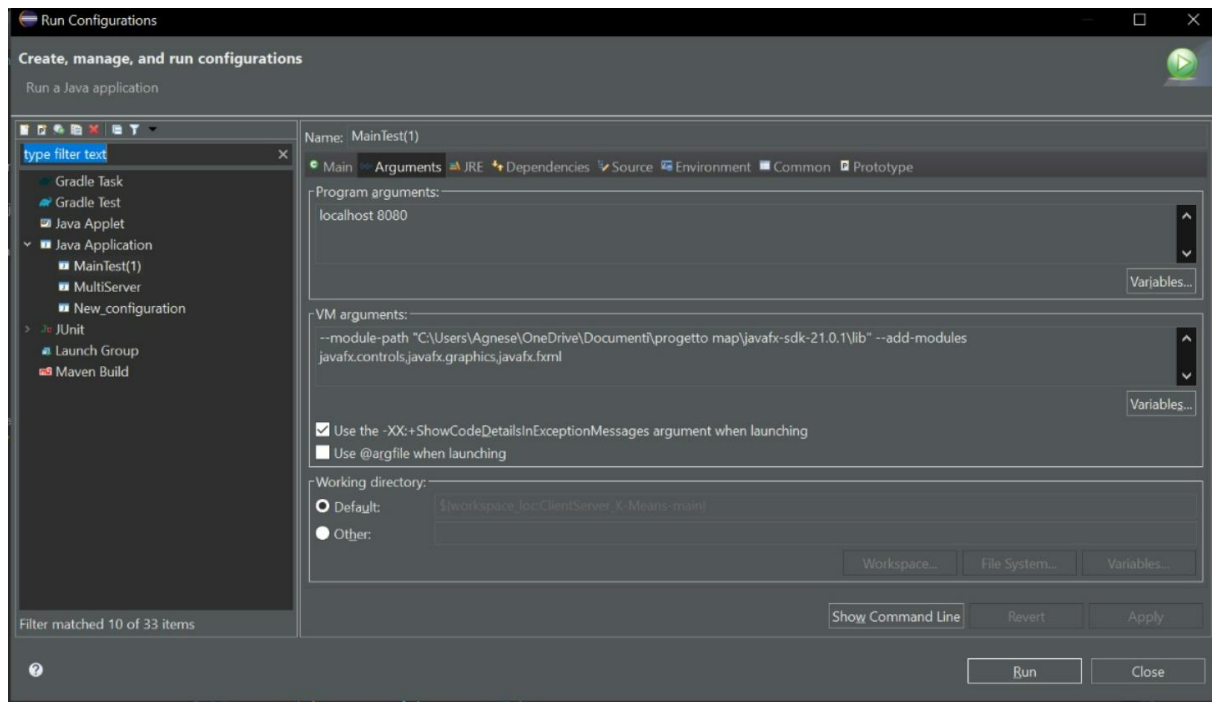
Questi passaggi assicureranno che la libreria *JavaFX* sia integrata correttamente nel progetto "*KmeansClient*" per consentire l'avvio dell'interfaccia grafica.

Creiamo in seguito la configurazione per l'avvio del programma: cliccare con il tasto destro su *KmeansClient* -> *Selezionare Esegui come* -> poi *Configurazioni* di Esecuzione. Cliccando sulla prima icona in alto a destra apparirà la seguente schermata:



Inserire *KMeansClient* nel campo Project dove è presente il nostro file *MainTest* per avviare il client, mentre nel campo Classe Principale inserire il file *MainTest.java* all'interno di *KmeansClient*.

Successivamente, configurare gli Argomenti, ovvero l'indirizzo e la porta a cui connettersi e un argomento per l'interfaccia *JavaFx*. Di seguito sono mostrati l'indirizzo e la porta predefiniti, ma è possibile modificarli. L'argomento da aggiungere per l'interfaccia è il seguente: `--module-path "percorso" --add-modules javafx.controls,javafx.graphics,javafx.fxml`. Per "percorso", si intende il percorso della cartella "lib" contenuta in "javafx-sdk-20.0.1".



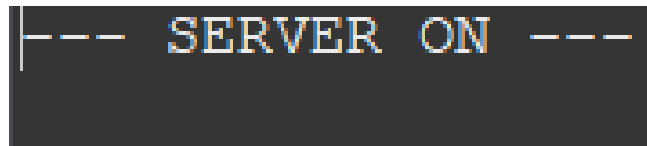
Infine, fare clic su "Applica" e su "Esegui".

GUIDA PER INTERFACCIA GRAFICA

Come nel caso della versione base, al fine di un corretto utilizzo del software, è essenziale garantire il rispetto dei requisiti menzionati in precedenza, mentre per quanto concerne l'automatizzare dell'avvio del server e del client, è anche qui possibile utilizzare il file batch denominato *K-Means_Server.bat* tramite una semplice doppio click.

AVVIO SERVER:

Anche qui, in seguito all'esecuzione del file *K-Means_Server.bat*, il server è stato avviato con successo. Il display mostra che il server è completamente preparato per ricevere e gestire qualsiasi richiesta proveniente dai clienti.

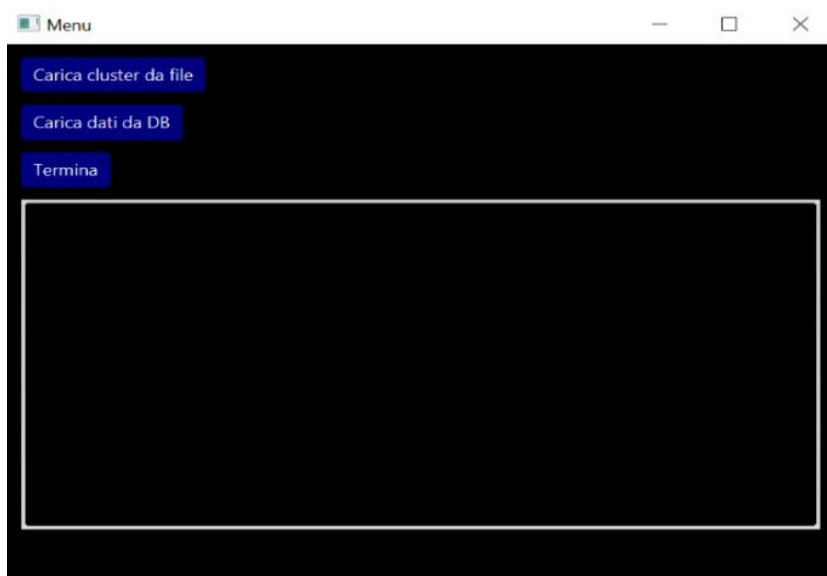


AVVIO CLIENT:

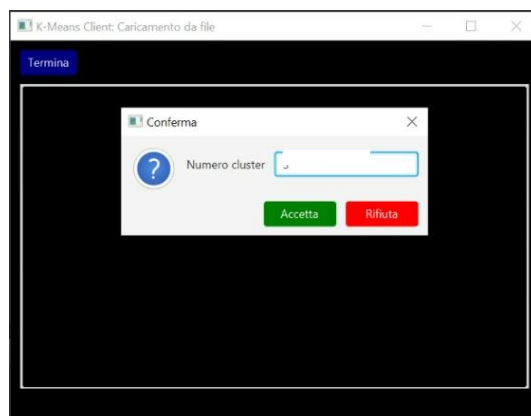
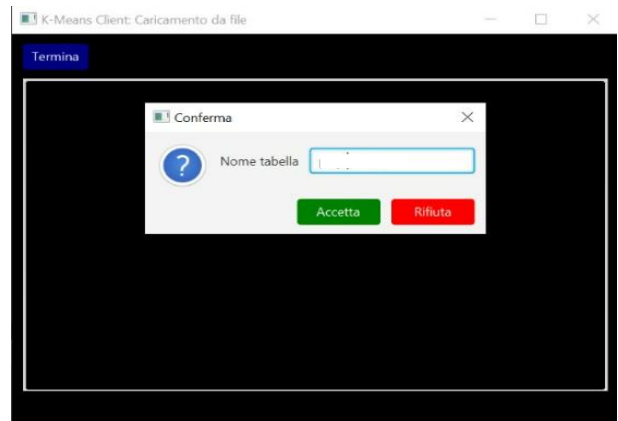
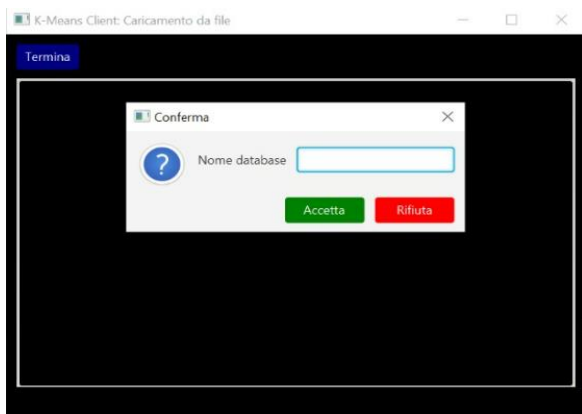
Quando il client verrà mandato in esecuzione verrà mostrata una finestra la quale conterrà il logo del progetto con la barra per il caricamento del software.



Quando il software avrà ultimato il caricamento apparirà il menu contenente sia le due opzioni per il caricamento dei Cluster, sia l'opzione di voler terminare il programma prematuramente.



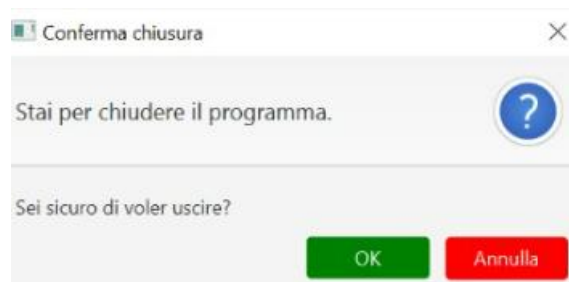
Se si dovesse optare per la scelta “*Carica cluster da file*”, si procederà alla compilazione del nome del DB, del nome della tabella e del numero di Cluster da inserire.



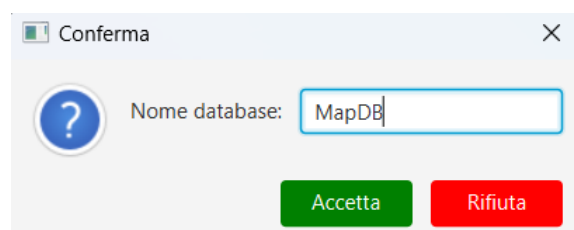
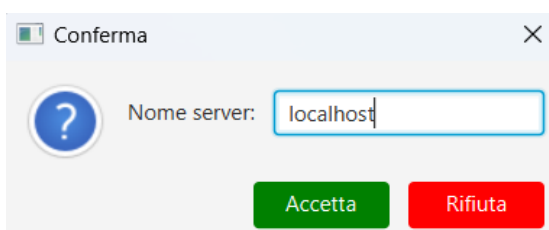
Se i dati inseriti saranno corretti, al termine dell'inserimento del database, della tabella e dei cluster verrà visualizzata la seguente schermata

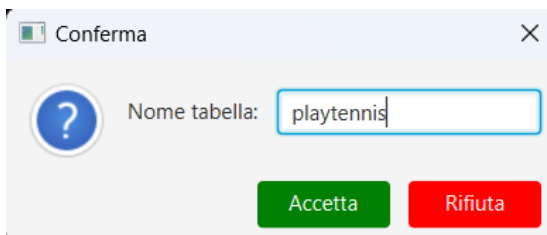


Si ricorda che in qualsiasi momento è possibile terminare il programma, e in tal caso apparirà il seguente messaggio:

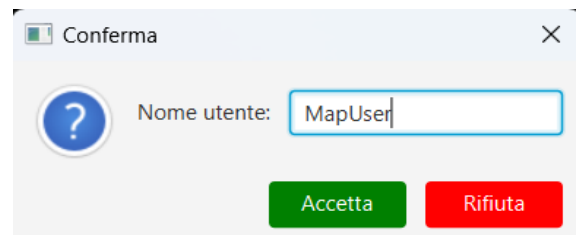


Se si dovesse optare per la scelta "Carica dati da DB", si potrà decidere di utilizzare l'inserimento dei dati di default o meno. Se l'utente dovesse optare per la seconda opzione, ovvero quella di inserire i dati manualmente, dovrà inserire il nome del Server, nome del database, nome della tabella, nome utente e password.

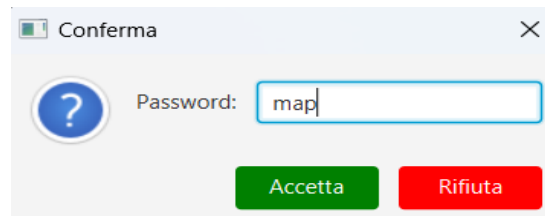




A dialog box titled "Conferma" with a close button (X) in the top right corner. It contains a blue circular icon with a question mark on the left. To its right is the label "Nome tabella:" followed by a text input field containing the text "playtennis". At the bottom, there are two buttons: a green "Accetta" button and a red "Rifiuta" button.

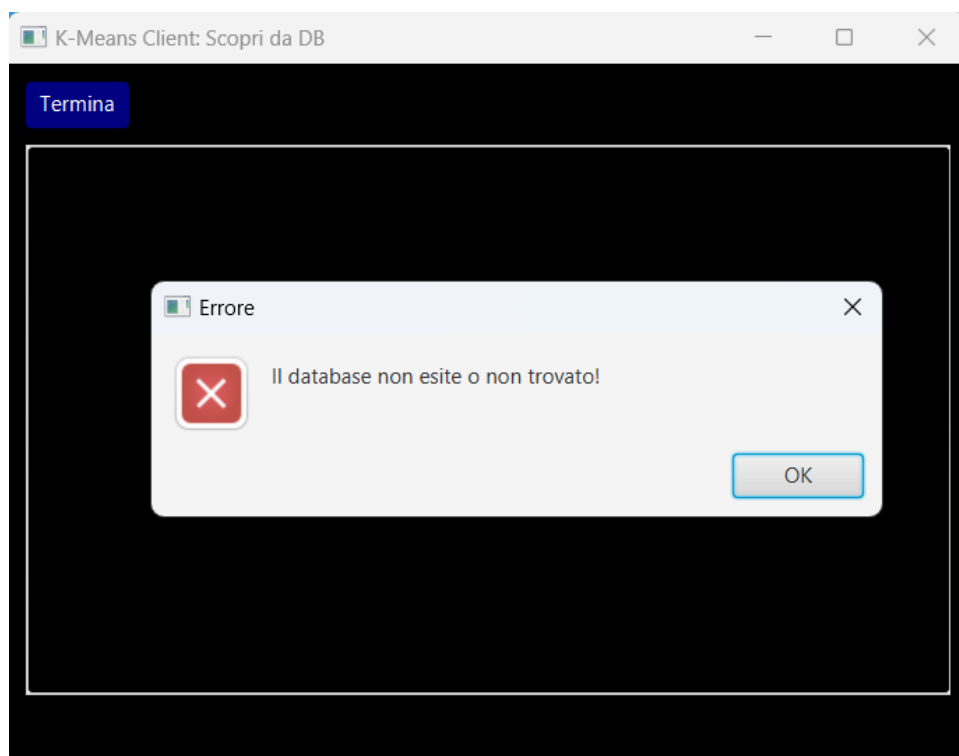


A dialog box titled "Conferma" with a close button (X) in the top right corner. It contains a blue circular icon with a question mark on the left. To its right is the label "Nome utente:" followed by a text input field containing the text "MapUser". At the bottom, there are two buttons: a green "Accetta" button and a red "Rifiuta" button.

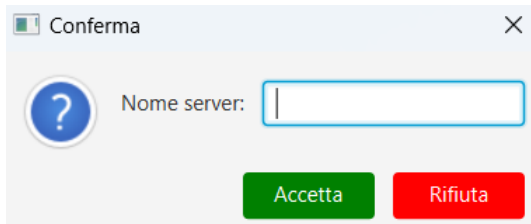


A dialog box titled "Conferma" with a close button (X) in the top right corner. It contains a blue circular icon with a question mark on the left. To its right is the label "Password:" followed by a text input field containing the text "map". At the bottom, there are two buttons: a green "Accetta" button and a red "Rifiuta" button.

I campi citati in precedenza dovranno essere esistenti, in caso contrario apparirà un messaggio di errore che segnalerà il non aver compilato i campi in maniera corretta.



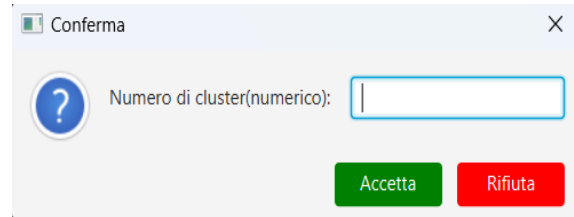
Se l'utente opterà per l'inserimento dei dati di default, verrà richiesto il nome del server e l'inserimento del numero di cluster(numerico).



Conferma

Nome server:

Accetta Rifiuta



Conferma

Numero di cluster(numerico):

Accetta Rifiuta

Nel corso dell'utilizzo dell'app, ogni volta che si fa una scelta, si noterà che nel menu saranno presenti solo i tasti relativi all' ultima decisione. Questo approccio mira a semplificare l'esperienza di navigazione, presentando solo le opzioni pertinenti in base alle tue azioni recenti.