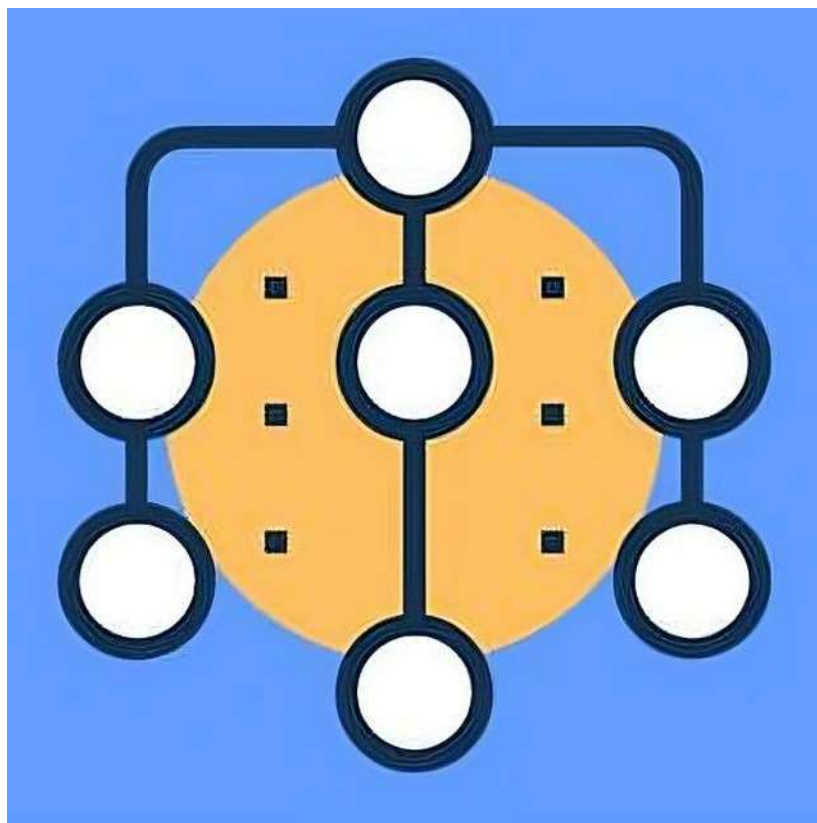


Metodi Avanzati di programmazione A.A. 2023-2024

Docente del corso: Professoressa A. Appice



Progetto di:

Elisa Vittoria Cosmai (758303) e.cosmai4@studenti.uniba.it

Michele Grieco (757059) m.grieco31@studenti.uniba.it

Giuseppe D'Urso(758311) g.durso15@studenti.uniba.it

Indice

1. Introduzione	3
1.1 Clusterizzazione agglomerativa dei dati.....	3
Vantaggi e limiti.....	3
2. Implementazioni.....	3
2.1 Progetto Base.....	3
2.2 Estensione	4
3. Installazione e Avvio	4
3.1 Requisiti Server.....	4
3.2 Requisiti Client	4
3.3 Avvio Server	5
3.4 Avvio Client	5
4. Diagrammi UML.....	5
4.1 Server	5
4.2 Client.....	9
Diagramma delle classi	9
5. Test.....	9
6. Javadoc.....	15

1. Introduzione

I **dati** sono diventati un elemento fondante della società moderna, rappresentando una delle principali risorse strategiche per aziende e organizzazioni. Ogni giorno, miliardi di dati vengono generati da dispositivi connessi, transazioni, piattaforme online, ma più in generale da quasi ogni tipologia di dispositivo. Questa enorme mole di informazioni, se opportunamente analizzata, consente di prendere decisioni più informate e di mantenere un vantaggio competitivo in campo aziendale e non solo. Tuttavia, l'abbondanza di dati non è sufficiente: è necessaria un'analisi efficace per interpretarli e trasformarli in informazioni utili, in modo tale da ridurre il più possibile l'incertezza.

Tra i metodi utilizzati per analizzare i dati, la clusterizzazione si distingue per la capacità di raggruppare elementi con caratteristiche simili. Si tratta di una tecnica fondamentale che mira a suddividere gli oggetti in insiemi distinti, detti **cluster**, in modo che gli elementi di ciascun cluster siano tra loro più simili rispetto a quelli appartenenti ad altri cluster. Nel esistono di diverse tipologie: clusterizzazione gerarchica o agglomerativa, K-Means, modelli basati sulla probabilità...

1.1 Clusterizzazione agglomerativa dei dati

La clusterizzazione agglomerativa è particolarmente utile per i dataset in cui non si conosce a priori il numero di cluster. Il processo è determinato da un certo numero di fasi:

- **Inizializzazione:** ogni punto del dataset si può considerare come un cluster singolo
- **Calcolo della somiglianza:** si calcola la distanza tra tutti i cluster utilizzando una misura come la distanza euclidea
- **Unione iterativa:** si uniscono i due cluster più simili tra loro
- **Creazione del dendrogramma:** il processo si ripete fino a quando tutti i punti sono inclusi in un unico grande cluster. Il grafico che si ottiene da questa operazione è definito dendrogramma e mostra le relazioni gerarchiche tra i dati

Andando a "tagliare" il dendrogramma a profondità diverse si può scegliere il numero di cluster da considerare.

Vantaggi e limiti

Tra i vantaggi principali c'è l'**identificazione automatica** di gruppi omogenei e la capacità di lavorare con dati **non etichettati**, oltre al fatto che molto spesso siano in grado di **rivelare relazioni** tra i dati che potrebbero non essere state prese in considerazione inizialmente.

Bisogna però prestare attenzione alla scelta della misura di distanza e del metodo di collegamento che si sceglie di sfruttare, in quanto potrebbero influenzare significativamente i risultati. Inoltre, su dataset di grandi dimensioni potrebbe richiedere molte risorse di calcolo.

2. Implementazioni

2.1 Progetto Base

Il progetto **H-CLUS** mira a sviluppare un sistema Client-Server avanzato per l'analisi dei dati tramite il clustering gerarchico di valori interi. Questo sistema è composto da due componenti principali:

- **Server:** È alla base del sistema, responsabile dell'elaborazione dei dati. Utilizza algoritmi di analisi quale il clustering agglomerativo, per scoprire strutture nascoste nei dati e rappresentarle attraverso dendrogrammi. Questi dendrogrammi permettono di visualizzare le relazioni tra i dati e di identificare raggruppamenti omogenei in modo gerarchico.
- **Client:** Applicazione in Java che ha il ruolo di interfaccia per gli utenti. Attraverso il client, gli utenti possono connettersi al server, accedere ai servizi di analisi e visualizzare i risultati dell'elaborazione, come i cluster scoperti.

2.2 Estensione

La versione base del progetto è stata estesa tramite un **bot telegram** con l'obiettivo di sostituire l'interfaccia client. Il bot consente all'utente di accedere ai servizi e visualizzare i risultati dell'elaborazione per più utenti in contemporanea.

Il bot permette di selezionare e caricare dati dal database o da file presenti sul server, rendendo il sistema facilmente configurabile. L'estensione offre una soluzione di visualizzazione user-friendly, rendendo le funzionalità del progetto accessibili da qualsiasi luogo e dispositivo dotato di Telegram, migliorando l'esperienza utente e supportando scenari d'uso più ampi.

3. Installazione e Avvio

Il software è strutturato da due sezioni: un **server**, il quale si occupa di applicare algoritmi di data mining per la scoperta di dendrogrammi di cluster di dati, sfruttando la clusterizzazione gerarchica; e un **client**, il quale consente di accedere alle funzionalità del sistema.

3.1 Requisiti Server

Per utilizzare il sistema, è necessario configurare un server compatibile con il progetto base e l'estensione Telegram

- Sistema operativo: Windows, macOS o Linux.
- Java: JDK 21 o superiore.
- Database: MySQL 8.0 o superiore.
 - o Inserire MySQL tra le variabili d'ambiente
 - Cercare "variabili d'ambiente" nel menu Start.
 - Nella sezione Variabili di sistema selezionare la variabile Path e cliccare su modifica.
 - Cliccare su Nuovo e aggiungere il percorso della cartella bin di MySQL. Il percorso predefinito è solitamente: "C:\Program Files\MySQL\MySQL Server [versione]\bin" (nella cartella di installazione di MySQL).
 - Cliccare su OK.
 - Controllare che sia stato digitando `mysql --version` nel prompt dei comandi e premendo Invio. In caso positivo, ci sarà la versione del programma.
- Librerie richieste:
 - o `mysql-connector-java`
- Configurazione del database:
 - o Creare un database utilizzando lo script `setup.sql` fornito nella directory `H_CLUS`.

Estensione

- Librerie richieste:
 - o `telegrambots` (versione 6.9.7.1 o compatibile)
 - o `slf4j-jdk14`

3.2 Requisiti Client

- Per utilizzare il bot Telegram come client:
 - o Telegram: Applicazione installata su smartphone o desktop.
 - o Connessione Internet: Necessaria per accedere al server remoto.

3.3 Avvio Server

Per poter utilizzare il software o il bot sarà necessario avviare prima il server. Per poter eseguire il server sarà necessario effettuare i seguenti passaggi:

- Eseguire lo script **setup.sql** per configurare la tabella.
- Nella directory principale del progetto individuare il file “**server.bat**” e avviarlo. Il progetto è impostato per sfruttare di default la porta 8080 e l’indirizzo ip 0.0.0.0. Per avviare il file è possibile fare doppio click oppure da terminale eseguendo il comando “server.bat” una volta arrivati alla directory in cui si trova il file.

Si potrà avviare un solo server alla volta, quindi quando si vuole terminare la connessione usare la combinazione di tasti CTRL+C, non chiudere direttamente il terminale. (Nel caso in cui dovesse capitare, bisognerà terminare il processo dal task manager)

Il file batch avvierà il file jar corrispondente al server rendendo possibile l’utilizzo del bot telegram.

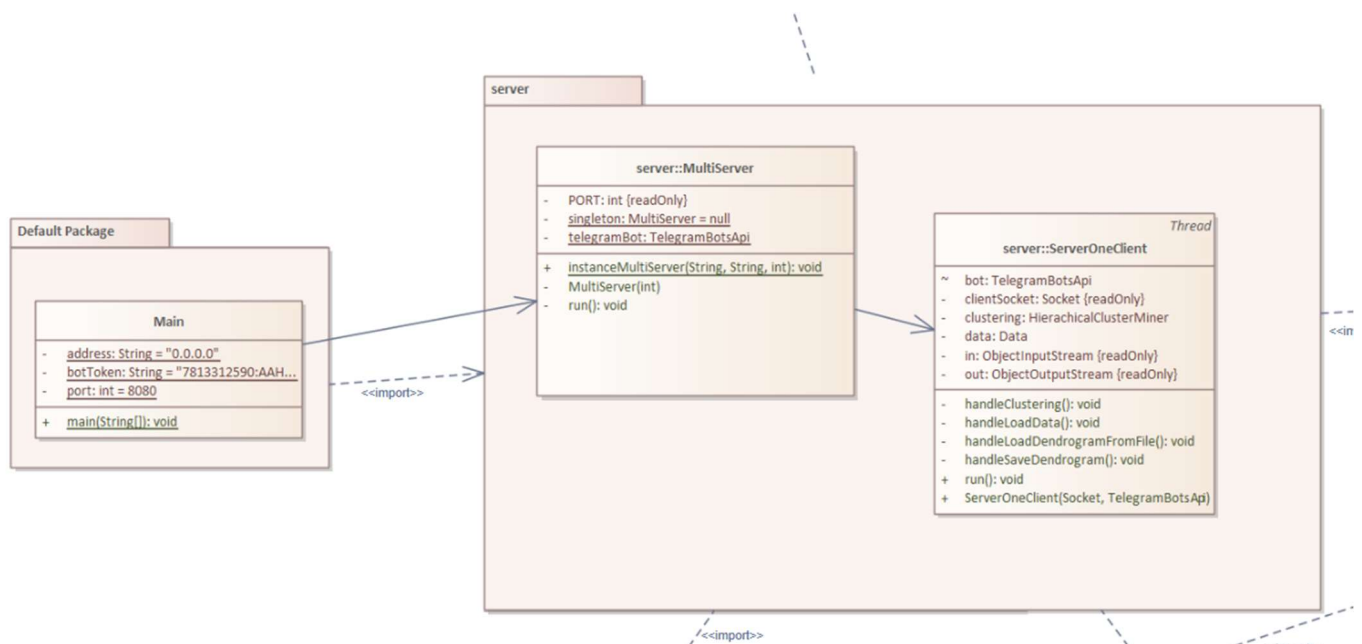
3.4 Avvio Client

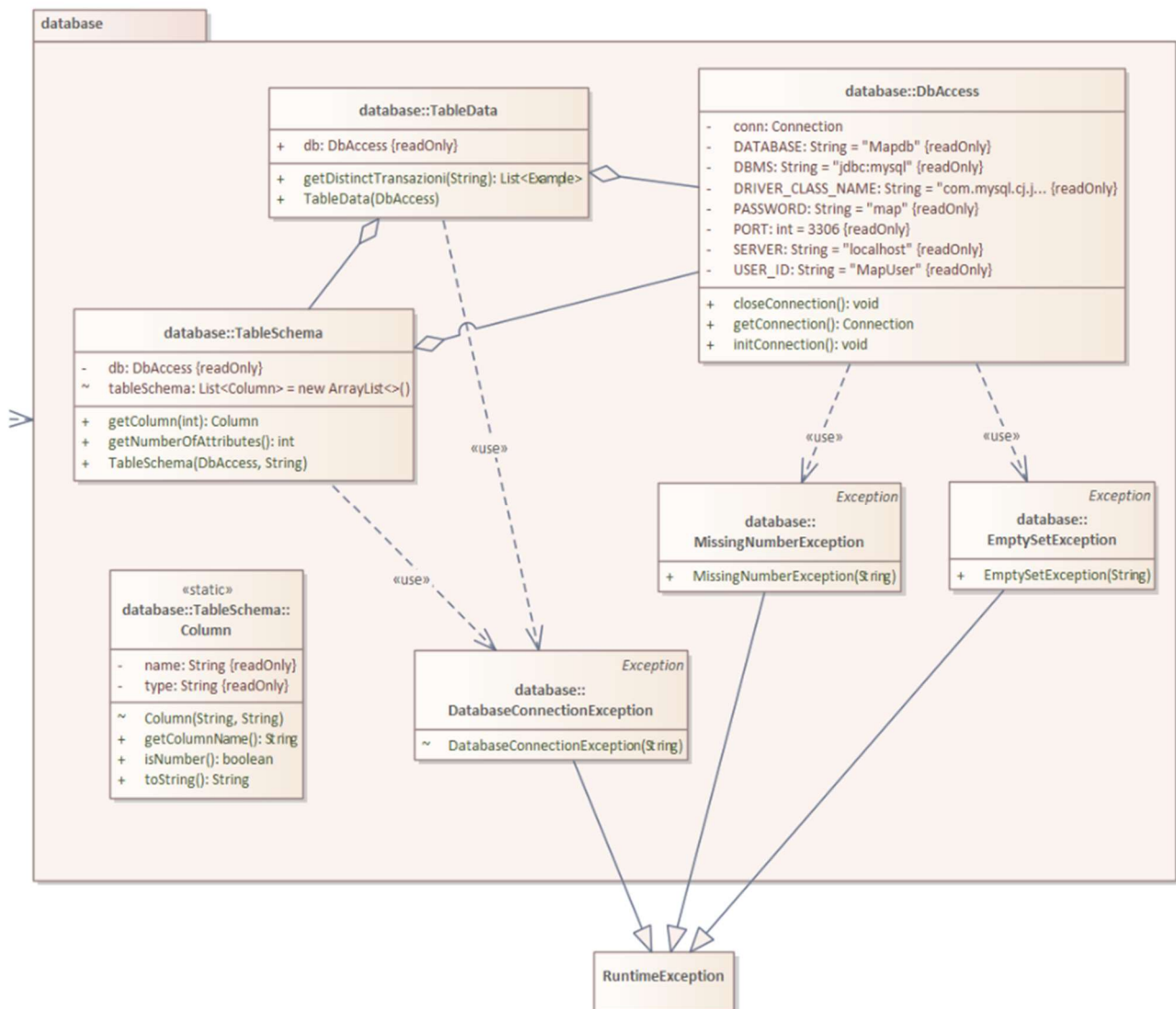
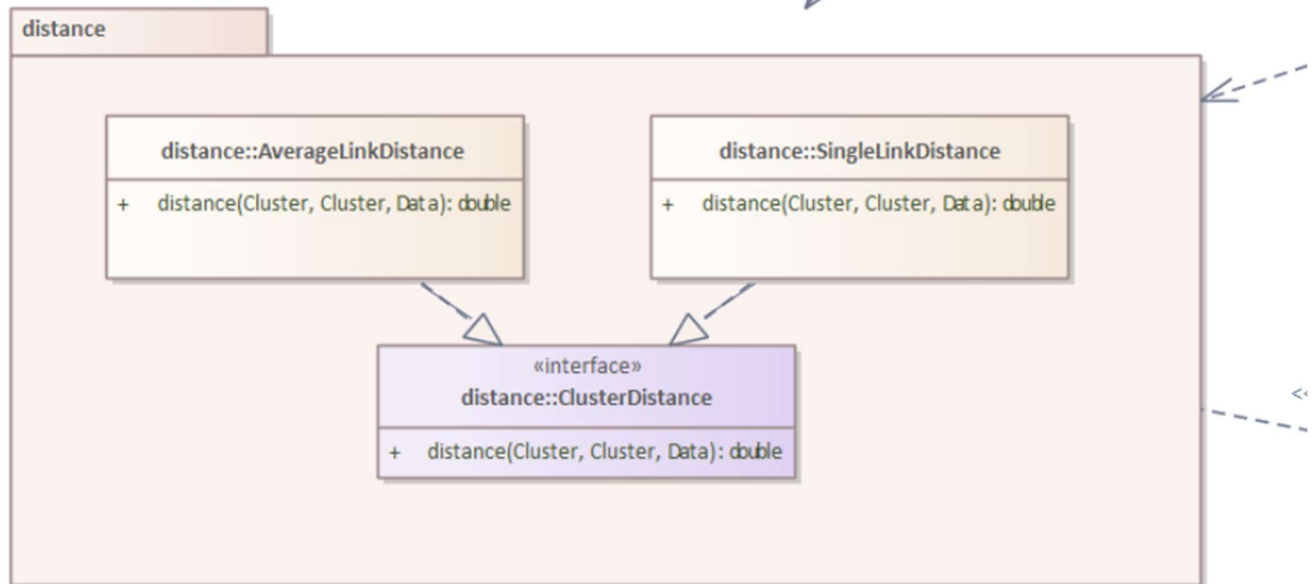
Una volta avviato il server, cercare su telegram **@HierarchicalClusBot** e scrivere **/start** in chat, da tastiera o selezionandolo nella lista dei comandi del menu nella chat. In questo caso, se si vuole chiudere la connessione al server si può scrivere **/end** in chat in qualsiasi momento.

Altrimenti, se si vuole utilizzare il progetto da terminale, utilizzare solo il progetto base e avviare prima il file “server.bat” e poi il file “client.bat”, con un doppio clic o da terminale, come già fatto in precedenza.

4. Diagrammi UML

4.1 Server





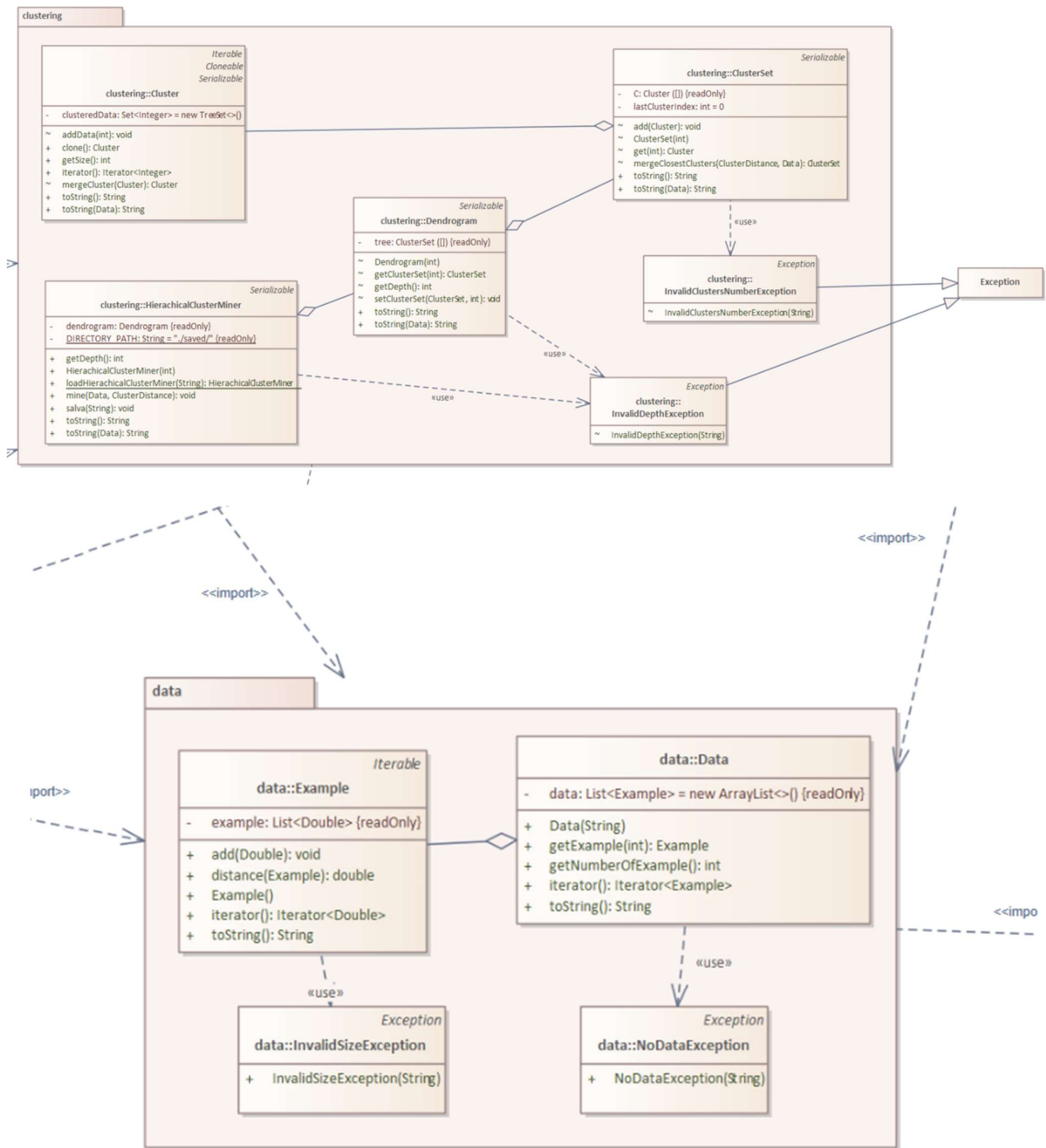
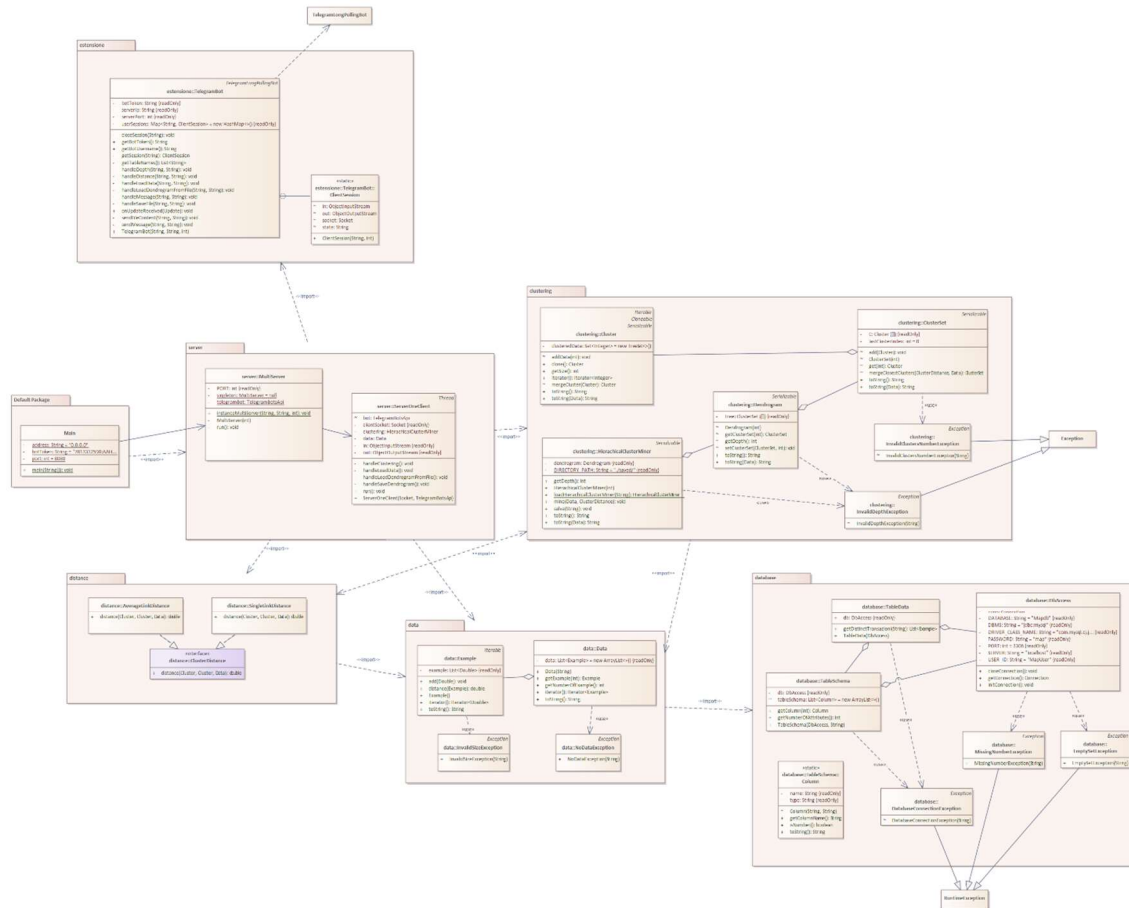
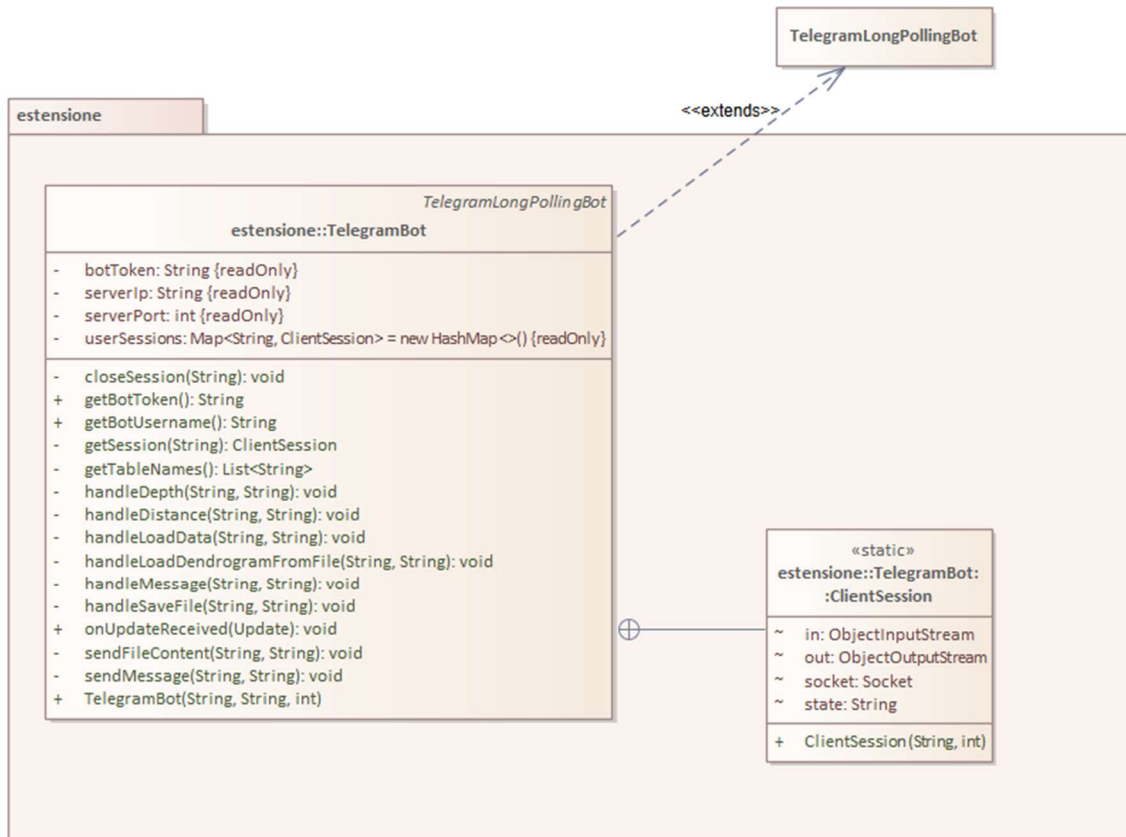


Diagramma delle classi Server completo

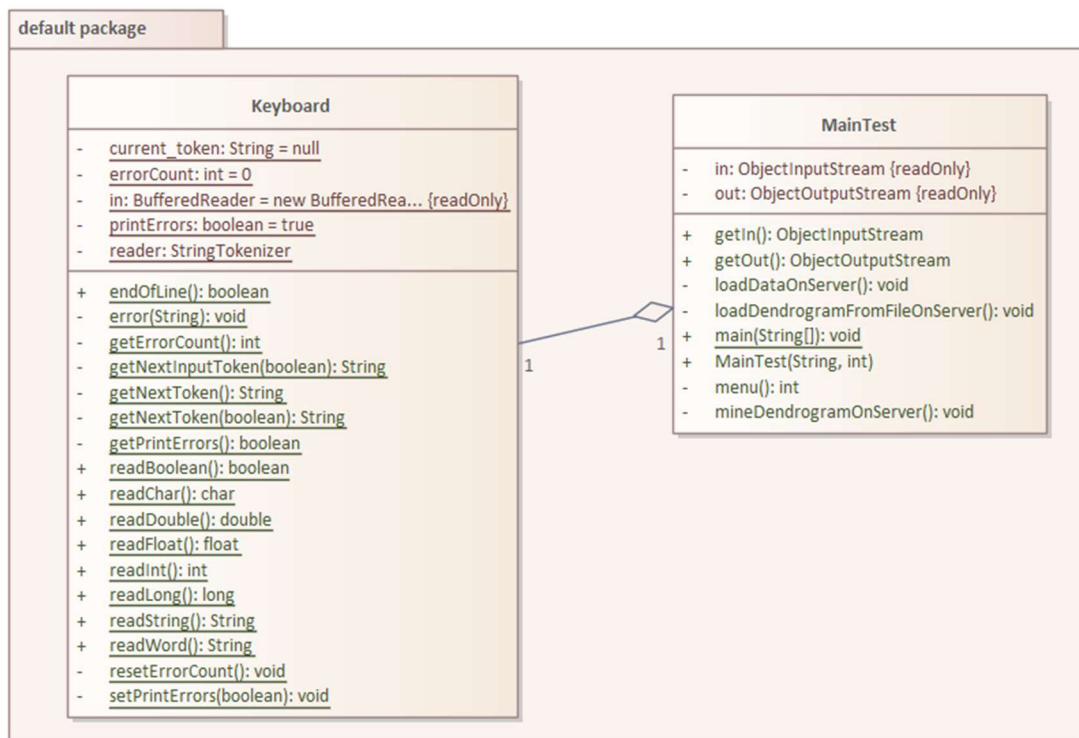


UML Bot



4.2 Client

Diagramma delle classi



5. Test

Esecuzione del client con server offline

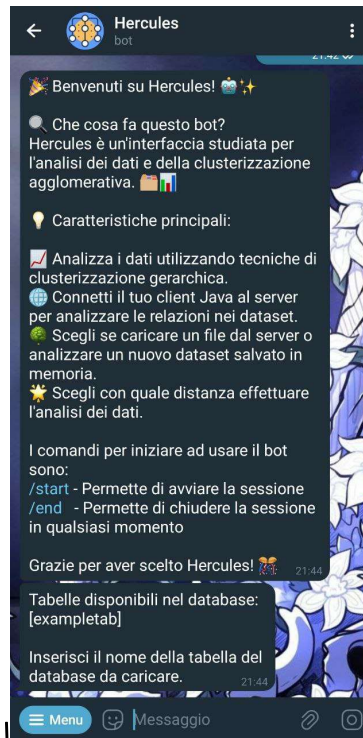


Il bot non partirà

Esecuzione del server con processo già in uso

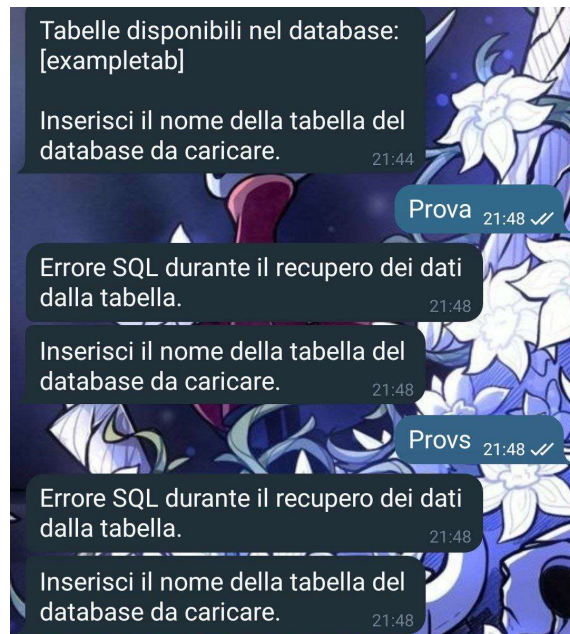
Il bot non riuscirà a connettersi

Esecuzione del client con server online



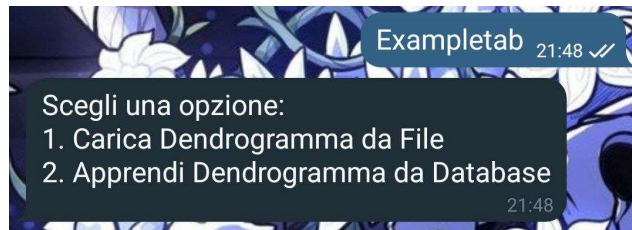
Il bot invia il messaggio di spiegazione e presenta le tabelle disponibili nel database

Inserimento del nome del database non valido



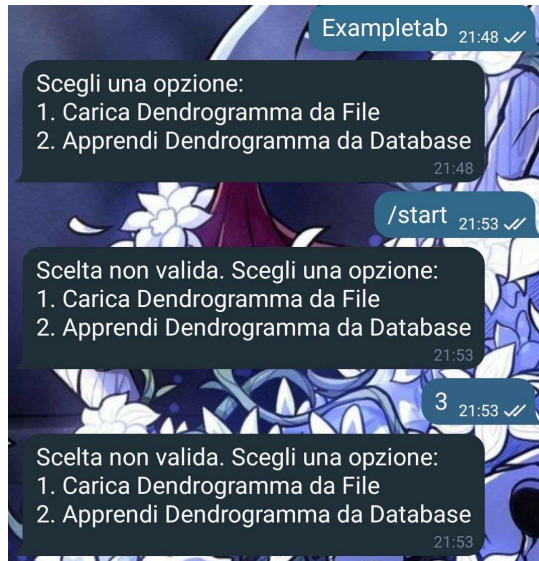
Il bot restituisce errore e chiede di reinserire il nome della tabella

Inserimento del nome della tabella presente nel database



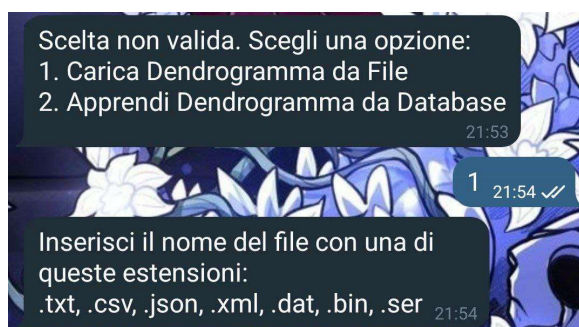
Il bot chiede di scegliere che tipo di operazione fare tra 1. Caricare il dendrogramma da file o 2. apprenderlo da database

Scelta diversa da 1 o 2 nel momento in cui si sceglie tra caricare il dendrogramma o apprenderlo da database



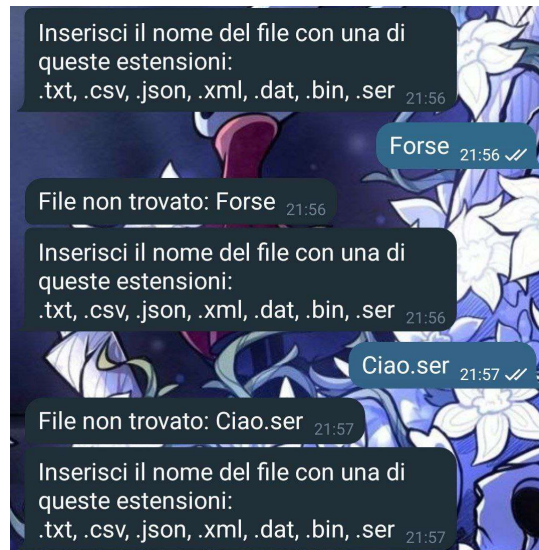
Il bot restituisce che la scelta fatta non è valida e richiede l'inserimento

Inserimento scelta 1 (Carica Dendrogramma da File)



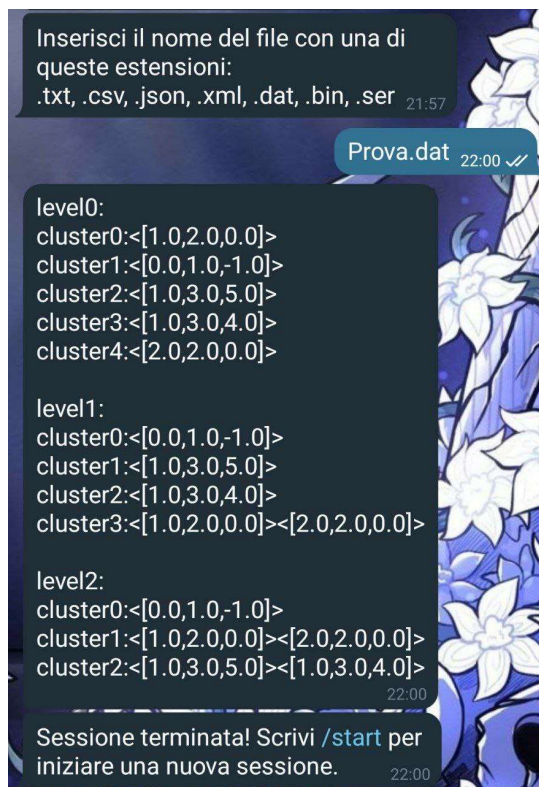
Viene richiesto di inserire il nome di un file già presente nella cartella "saved" del progetto

Inserimento errato del file



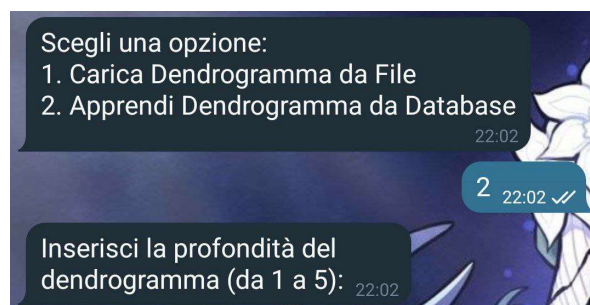
Il bot comunicherà che il file non è stato trovato e richiederà di inserirlo

Inserimento corretto del file



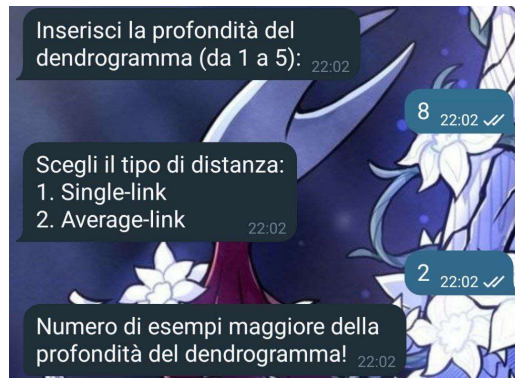
Il bot permette la visualizzazione del file e termina la sessione. C'è la possibilità di riaprire la sessione.

Inserimento scelta 2 (Apprendi Dendrogramma da Database)



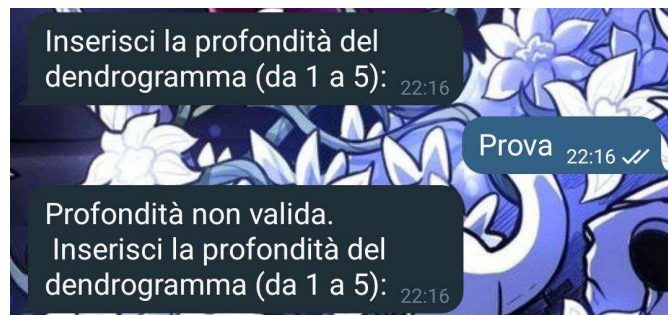
Il bot chiede di scegliere la profondità del dendrogramma

Inserimento della profondità maggiore di quella disponibile



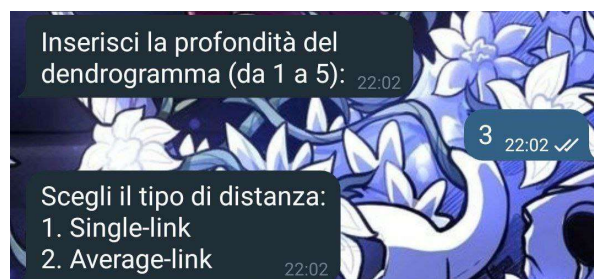
Il bot continuerà a chiedere di reinserire la profondità fino a quando non sarà valida

Inserimento della profondità diversa da un valore numerico



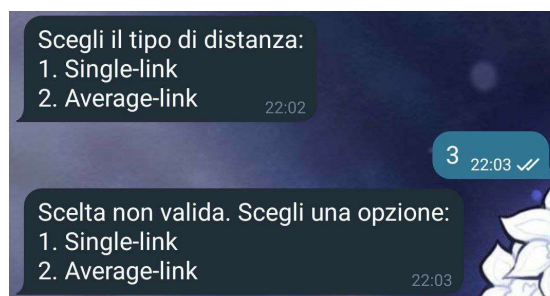
Viene richiesto l'inserimento fino a quando il valore non sarà valido

Inserimento scelta profondità valida



Viene richiesto di inserire il tipo di distanza da utilizzare

Scelta errata tra 1 e 2



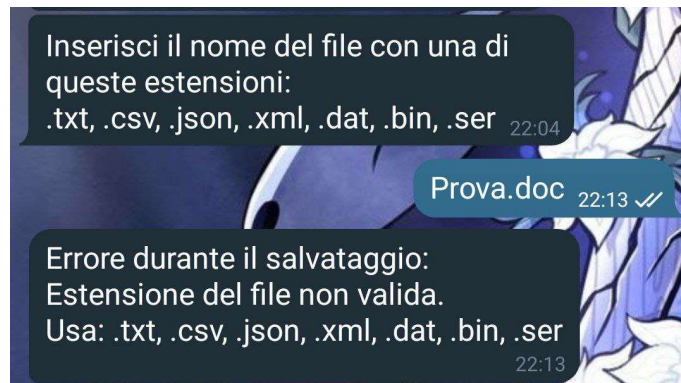
Il bot chiederà di reinserire il valore fino a quando non è corretto

Scelta corretta tra 1 e 2



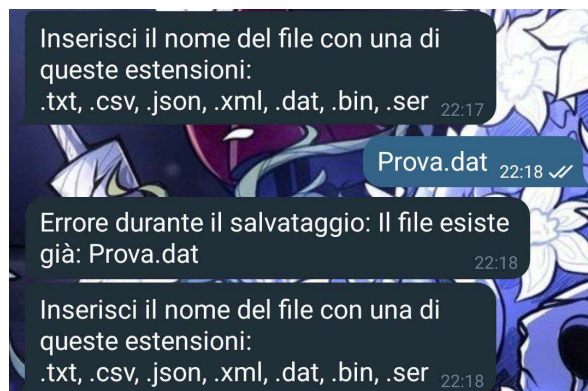
Viene visualizzato il dendrogramma e viene richiesto il salvataggio

Tentativo di salvare il file con estensione diversa da quanto richiesto



Il bot segnala l'errore e richiede di salvare il file fino a quando non viene inserita un'estensione valida

Tentativo di salvare il file con il nome di un file già presente



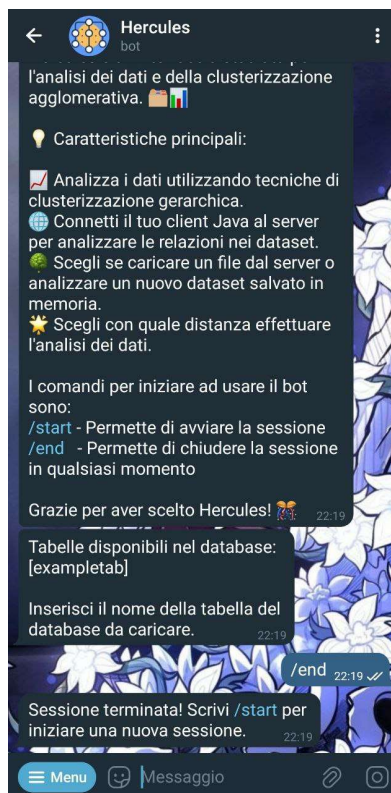
Segnala che il file è già presente e richiede l'inserimento

Salvataggio di un file con estensione corretta e non presente nell'archivio



Il file viene salvato e la sessione viene chiusa. C'è la possibilità di iniziare una nuova sessione

Inserimento di /end



Sessione terminata. Viene data la possibilità di avviarne un'altra con il comando /start

6. Javadoc

La documentazione Javadoc si può trovare al seguente link: [Javadoc](#)