



**UNIVERSITÀ  
DEGLI STUDI DI BARI  
ALDO MORO**

DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA IN INFORMATICA (L-31)

---

**Metodi Avanzati di Programmazione**

**Hierarchical Clustering**

---

Anno Accademico **2023–2024**

# Indice

<b>1</b>	<b>Introduzione Hierarchical Clustering</b>	<b>3</b>
1.1	Tipologie di Hierarchical Clustering . . . . .	3
1.2	Calcolo della distanza . . . . .	3
1.2.1	Distanza tra esempi . . . . .	3
1.2.2	Distanza tra cluster . . . . .	4
1.3	Dendrogramma . . . . .	4
1.4	Applicazioni . . . . .	4
<b>2</b>	<b>Descrizione dei progetti</b>	<b>5</b>
2.1	Progetto base . . . . .	5
2.2	Progetto esteso . . . . .	5
<b>3</b>	<b>Utilizzo</b>	<b>7</b>
3.1	Progetto Base . . . . .	7
3.2	Progetto Esteso . . . . .	7
<b>4</b>	<b>Test progetto base</b>	<b>8</b>
4.1	Server . . . . .	8
4.1.1	Porta corretta . . . . .	8
4.1.2	Porta errata (valore fuori dal range) . . . . .	8
4.1.3	Porta errata (valore non numerico) . . . . .	8
4.1.4	Porta già in uso . . . . .	8
4.2	Client . . . . .	9
4.2.1	Connessione ad un server avviato . . . . .	9
4.2.2	Connessione ad un server non avviato . . . . .	9
4.2.3	Porta del server errata . . . . .	9
4.3	Indirizzo IP del server errato . . . . .	10
4.3.1	Indirizzo IP e porta del server errati . . . . .	10
4.3.2	Connessione al database con credenziali corrette . . . . .	10
4.3.3	Connessione al database con credenziali errate . . . . .	11
4.3.4	Connessione al database con servizio MySQL non attivo . . . . .	11
4.3.5	Connessione al database con database richiesto inesistente . . . . .	11
4.3.6	Tabella con dati corretti . . . . .	12
4.3.7	Tabella con dati non omogenei . . . . .	12
4.3.8	Tabella non esistente . . . . .	13
4.3.9	Tabella vuota . . . . .	13
4.3.10	Scelta corretta nel menu . . . . .	13
4.3.11	Scelta sbagliata nel menu . . . . .	14
4.3.12	Carica da file serializzato corretto . . . . .	14
4.3.13	Carica da file serializzato utilizzando percorso non valido . . . . .	15
4.3.14	Carica da file serializzato oggetto di istanza errata . . . . .	16
4.3.15	Carica da file vuoto . . . . .	16
4.3.16	Carica da file inesistente . . . . .	16
4.3.17	Genera con profondità corretta . . . . .	17
4.3.18	Genera con profondità non valida . . . . .	17
4.3.19	Genera con profondità non numerica . . . . .	18
4.3.20	Genera con percorso non valido per il salvataggio . . . . .	18
4.3.21	Chiusura prematura del server durante l'esecuzione del client . . . . .	18
4.3.22	Chiusura prematura del client durante l'esecuzione del server . . . . .	19

<b>5</b>	<b>Test Progetto Esteso</b>	<b>20</b>
5.1	Server . . . . .	20
5.1.1	Bottone di avvio premuto senza aver inserito testo . . . . .	20
5.1.2	Porta corretta . . . . .	20
5.1.3	Porta errata (valore fuori dal range) . . . . .	21
5.1.4	Porta errata (valore non numerico) . . . . .	21
5.1.5	Porta già in uso . . . . .	21
5.1.6	Chiusura del server dal pulsante "Stop Server" . . . . .	22
5.1.7	Chiusura del server dal pulsante [X] della finestra . . . . .	22
5.2	Client . . . . .	22
5.2.1	Bottone di avvio premuto senza aver inserito testo . . . . .	22
5.2.2	Connessione al server avviato . . . . .	23
5.2.3	Connessione al server non avviato . . . . .	23
5.2.4	Porta del server errata . . . . .	24
5.2.5	Indirizzo IP del server errato . . . . .	24
5.2.6	Indirizzo IP e porta del server errati . . . . .	24
5.2.7	Bottone di accesso premuto senza aver inserito testo . . . . .	25
5.2.8	Connessione al database con credenziali corrette . . . . .	25
5.2.9	Connessione al database con credenziali errate . . . . .	26
5.2.10	Connessione al database con servizio MySQL non attivo . . . . .	27
5.2.11	Connessione al database con database richiesto inesistente . . . . .	27
5.2.12	Tabella con dati non omogenei . . . . .	28
5.2.13	Tabella non esistente . . . . .	28
5.2.14	Tabella vuota . . . . .	29
5.2.15	Bottone "Carica" premuto senza aver inserito testo . . . . .	29
5.2.16	Carica da file serializzato corretto . . . . .	30
5.2.17	Carica da file serializzato utilizzando percorso non valido . . . . .	30
5.2.18	Carica da file serializzato oggetto di istanza errata . . . . .	31
5.2.19	Carica da file vuoto . . . . .	31
5.2.20	Carica da file inesistente . . . . .	32
5.2.21	Bottone "Genera" premuto senza aver inserito testo . . . . .	32
5.2.22	Genera con profondità corretta . . . . .	33
5.2.23	Genera con profondità non valida . . . . .	33
5.2.24	Genera con profondità non numerica . . . . .	34
5.2.25	Genera con percorso non valido per il salvataggio . . . . .	34
5.2.26	Chiusura prematura del server durante l'esecuzione del client . . . . .	35
5.2.27	Chiusura prematura del client durante l'esecuzione del server . . . . .	35

# 1 Introduzione Hierarchical Clustering

Il progetto si propone di implementare un algoritmo in grado di effettuare **Hierarchical Clustering analysis** su un set di dati ottenuto dalla lettura di una tabella di un database. Tale tecnica è utilizzata per raggruppare un insieme di dati in cluster, insiemi di elementi simili tra loro, sulla base delle loro caratteristiche.

## 1.1 Tipologie di Hierarchical Clustering

Tale processo prevede la costruzione di una gerarchia di cluster secondo due principali strategie.

1. **Clustering agglomerativo (bottom-up)**: Approccio utilizzato nel progetto. Si inizia considerando ogni esempio (nel progetto rappresentato da vettori di numeri reali) come un singolo cluster. Successivamente, questi cluster vengono uniti due alla volta, basandosi sul calcolo della distanza, fino a ottenere un unico grande cluster che contiene tutti gli esempi del dataset.

**Passi principali nel clustering agglomerativo:**

- **Calcolo delle distanze**: Si misura la distanza tra i cluster (al primo passo ognuno composto da un solo esempio).
  - **Unione di cluster**: Si selezionano i due cluster più vicini e li si unisce.
  - **Aggiornamento delle distanze**: Dopo ogni unione, si ricalcolano le distanze.
  - **Ripetizione**: Si ripetono i passi precedenti fino a che tutti gli esempi non sono stati uniti in un unico cluster.
2. **Clustering divisivo (top-down)**: Parte da un unico cluster che contiene tutti gli esempi e, progressivamente, lo divide in sotto-cluster fino a quando ogni cluster contiene un singolo esempio.

## 1.2 Calcolo della distanza

Il calcolo della distanza tra esempi (nel progetto, vettori di numeri reali) e cluster è un passaggio cruciale per determinare quali cluster fondere durante il processo di clustering gerarchico. Di seguito, vengono descritti i principali metodi utilizzati.

### 1.2.1 Distanza tra esempi

Per misurare la distanza tra due esempi  $x = [x_1, x_2, \dots, x_n]$  ed  $y = [y_1, y_2, \dots, y_n]$ , si è utilizzata la **distanza euclidea**.

$$dist(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Si supponga di avere i due esempi:

$$e_1 = [1, 3, 0] \quad \text{e} \quad e_2 = [0, 1, 0]$$

La distanza euclidea  $D(e_1, e_2)$  è calcolata come segue:

$$\begin{aligned} D(e_1, e_2) &= \sqrt{(1-0)^2 + (3-1)^2 + (0-0)^2} \\ &= \sqrt{1+4+0} \\ &= \sqrt{5} \end{aligned}$$

### 1.2.2 Distanza tra cluster

Nel clustering gerarchico, la distanza tra due cluster  $C_1$  e  $C_2$  può essere definita in diversi modi. Si riportano quelli utilizzati nel progetto.

- **Single Link:** La distanza tra due cluster è definita come la distanza minima tra qualsiasi coppia di esempi appartenenti ai due cluster:

$$D(C_1, C_2) = \min_{t_1 \in C_1, t_2 \in C_2} \text{dist}(t_1, t_2)$$

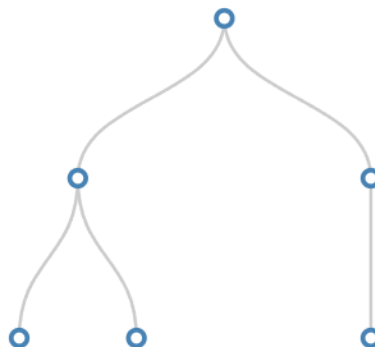
- **Average Link:** La distanza tra due cluster è la media delle distanze tra tutte le coppie di esempi appartenenti ai due cluster:

$$D(C_1, C_2) = \frac{\sum_{t_1 \in C_1, t_2 \in C_2} \text{dist}(t_1, t_2)}{|C_1| \times |C_2|}$$

dove  $|C_1|$  e  $|C_2|$  sono il numero di esempi nei cluster  $C_1$  e  $C_2$ .

## 1.3 Dendrogramma

Lo strumento di visualizzazione del risultato mediato dall'algoritmo di analisi è il **dendrogramma**, una struttura ad albero che mostra come i cluster vengono agglomerati.



## 1.4 Applicazioni

*Hierarchical Clustering* è ampiamente utilizzato in applicazioni come il riconoscimento di pattern e la bioinformatica (per classificare geni o proteine).

## 2 Descrizione dei progetti

### 2.1 Progetto base

La versione base del progetto consiste in un'architettura client/server pensata per essere utilizzata da riga di comando. Tale architettura permette di disaccoppiare le due componenti, ottenendo così due applicazioni indipendenti.

Il server dovrà essere eseguito su una macchina con un database MySQL in esecuzione, alla porta 3306. Prima di eseguire l'applicazione è opportuno che sulla macchina sulla quale verrà eseguito il server sia presente un database che contenga almeno una tabella che abbia unicamente attributi di tipo numerico, quelli che verranno utilizzati per eseguire l'operazione di clustering. A tale fine è fornito lo script di esempio *tabelle\_test.sql* che permette di ottenere un database che comprenda tabelle d'esempio per coprire alcuni casi di test sull'applicazione. All'apertura del server andrà inoltre specificata la porta (es. 8080) su cui il server dovrà attendere le richieste del client.

Quest'ultimo dovrà specificare l'indirizzo IP del server, che se eseguito localmente di default è 127.0.0.1, e la porta su cui il server attende le richieste di connessione.

I servizi offerti dal server all'utente tramite il client sono i seguenti:

- Consultazione di un dendrogramma precedentemente serializzato in un file.
- Generazione ex-novo di un dendrogramma a partire da un set di dati, letto da un database MySQL, a cui si è precedentemente connessi. Tale operazione è personalizzabile dall'utente che ha la facoltà di specificare la profondità del dendrogramma e il tipo di calcolo della distanza tra cluster. La generazione del dendrogramma è sempre seguita dal suo salvataggio in un file, il cui percorso viene specificato dall'utente.

### 2.2 Progetto esteso

La versione estesa del progetto prevede l'implementazione di interfacce grafiche che integrano l'architettura precedentemente discussa.

Nello specifico la realizzazione delle due interfacce, una per il client ed una per il server, è stata effettuata utilizzando il framework *JavaFX*.

Oltre a quanto già implementato nella versione base, la versione estesa prevede la visualizzazione **interattiva** del dendrogramma generato o caricato, permettendo all'utente di scegliere attraverso uno slider il livello di profondità da visualizzare, di default impostato alla profondità scelta dall'utente stesso al momento della generazione. Si precisa che le etichette dei nodi nella rappresentazione grafica dell'albero si riferiscono agli indici che identificano la posizione all'interno del dataset caricato dal database degli esempi contenuti in ogni cluster.

Ad esempio, considerando il dataset che contiene i seguenti esempi nel rispettivo ordine

0. 1.0 2.0 0.0

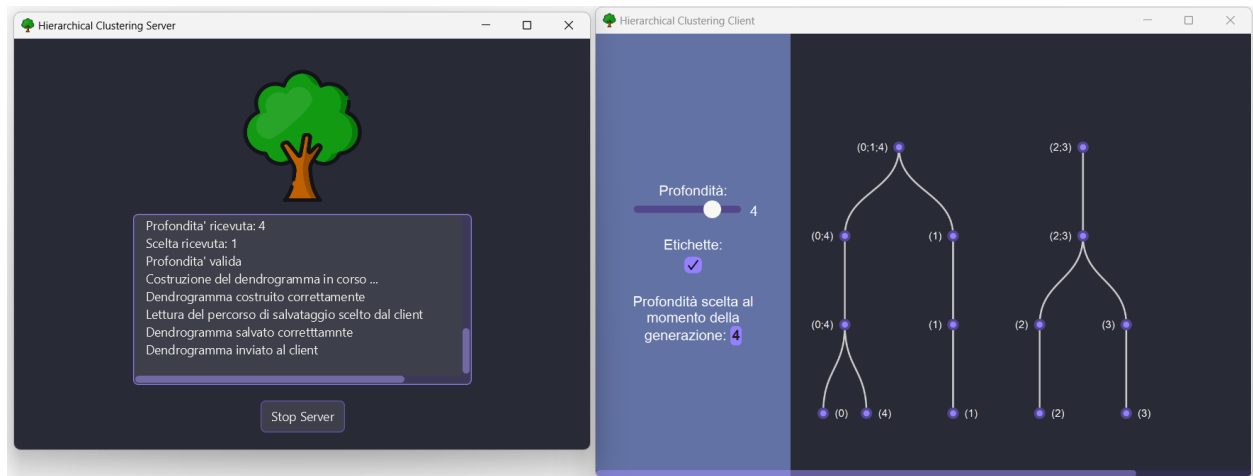
1. 0.0 1.0 -1.0

2. 1.0 3.0 5.0

3. 1.0 3.0 4.0

4. 2.0 2.0 0.0

gli indici che verranno associati ad ognuno degli esempi risultano essere quelli specificati dall'elenco puntato. Dunque, nella visualizzazione del dendrogramma, l'etichetta  $(0;4)$  indicherà che all'interno di tale cluster sono contenuti gli esempi  $1.0\ 2.0\ 0.0, 2.0\ 2.0\ 0.0$ .



## 3 Utilizzo

### 3.1 Progetto Base

Utilizzando windows, posizionarsi nelle rispettive directory e fare doppio click rispettivamente su

- `AVVIA_CLIENT.bat` per avviare il client;
- `AVVIA_SERVER.bat` per avviare il server.

In alternativa, su due distinti terminali, si può eseguire il comando `java -jar <name.jar>`, ove name va opportunamente sostituito con `H_CLUS_Client.jar` e `H_CLUS_Server.jar`.

### 3.2 Progetto Esteso

#### Opzionale: Build del codice sorgente

Indipendentemente dal sistema operativo, basta posizionarsi nelle rispettive directory, aprire un terminale in ciascuna cartella, e eseguire il comando `gradlew clean jar` per compilare il progetto.

Verrà così utilizzato il wrapper di gradle, già presente, per scaricare automaticamente le dipendenze necessarie e compilare il progetto.

Il file `.jar` verrà generato nella cartella `build/libs/`.

#### Esecuzione del jar

Per eseguire la versione del client e del server già buildata, presente in allegato al progetto, risulta sufficiente fare doppio click rispettivamente su

- `AVVIA_CLIENT.bat` per avviare il client;
- `AVVIA_SERVER.bat` per avviare il server.

In alternativa eseguire il comando su terminale `java -jar <name.jar>`, ove name va opportunamente sostituito con `hclus_client.jar` e `hclus_server.jar`.



## 4 Test progetto base

### 4.1 Server

#### 4.1.1 Porta corretta

Test effettuato eseguendo il server specificando una porta valida (nell'esempio, 8080). Una porta è ritenuta valida se rispetta le seguenti caratteristiche:

- è un valore numerico;
- è compreso tra 1024 e 65535;
- non è già in uso da un altro software.

Il risultato atteso è l'avvio del server e la visualizzazione di un messaggio di conferma dell'operazione appena eseguita.

```
Inserire la porta del server: 8080
Server socket aperto correttamente
```

#### 4.1.2 Porta errata (valore fuori dal range)

Si tenta di avviare il server su una porta non valida, specificando per quest'ultima un valore numerico non compreso tra 1024 e 65535 (nell'esempio, 9999999). Il risultato atteso è un messaggio di errore che permetta successivamente di specificare una nuova porta sulla quale aprire il server.

```
Inserire la porta del server: 999999
Errore: inserire un numero di porta compreso tra 1024 e 65535.
Inserire la porta del server: 8080
Server socket aperto correttamente
|
```

#### 4.1.3 Porta errata (valore non numerico)

Si specifica come porta sulla quale eseguire il server un valore non numerico (nell'esempio, una stringa di testo). Il risultato atteso è un messaggio di errore che permetta successivamente di specificare una nuova porta sulla quale eseguire il server.

```
Inserire la porta del server: asdfasdfasd
Error reading Integer data, MIN_VALUE value returned.
Errore: inserire un numero di porta compreso tra 1024 e 65535.
Inserire la porta del server: |
```

#### 4.1.4 Porta già in uso

Sulla medesima macchina possono essere in esecuzione molteplici server contemporaneamente. Si procede allora ad avviare quest'ultimo su di una porta già in uso da un altro software. Il risultato atteso è un messaggio di errore che permetta successivamente di specificare una nuova porta sulla quale eseguire il server.

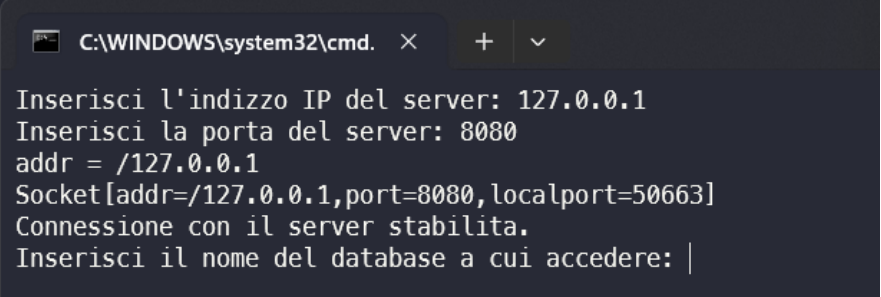
```
Inserire la porta del server: 8080
Errore nell'apertura del server socket. Riprovare inserendo una porta differente.
Inserire la porta del server: |
```

## 4.2 Client

### 4.2.1 Connessione ad un server avviato

Si avvia il server dell'applicazione, specificando per questo una porta valida (nell'esempio, 8080). Successivamente si procede ad eseguire il client, specificando l'indirizzo IP e la porta corretti per il server. Nel caso specifico, essendo il server in esecuzione sulla medesima macchina, si specifica come indirizzo IP "localhost" (127.0.0.1). Il risultato atteso è la corretta apertura della comunicazione tra server e client.

```
Inserire la porta del server: 8080
Server socket aperto correttamente
Connessione con il client stabilita: Socket[addr=/127.0.0.1,port=50663,localport=8080]
Stream di input aperto
Stream di output aperto
|
```



```
C:\WINDOWS\system32\cmd. x + v
Inserisci l'indirizzo IP del server: 127.0.0.1
Inserisci la porta del server: 8080
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=50663]
Connessione con il server stabilita.
Inserisci il nome del database a cui accedere: |
```

### 4.2.2 Connessione ad un server non avviato

Si tenta di avviare il client specificando l'indirizzo IP e la porta di un server, il quale non risulta però essere in esecuzione. Il risultato atteso è un messaggio di errore che permetta successivamente di specificare un nuovo indirizzo IP e una nuova porta, per procedere alla ricerca di un ulteriore server al quale connettersi.

```
Inserisci l'indirizzo IP del server: 127.0.0.1
Inserisci la porta del server: 8080
addr = /127.0.0.1
Errore nella connessione con il server.
Inserisci l'indirizzo IP del server: |
```

### 4.2.3 Porta del server errata

Si avvia il server dell'applicazione, specificando per questo una porta valida (nell'esempio, 8080). Successivamente si procede ad eseguire il client, inserendo l'indirizzo IP corretto per il server (essendo server e client in esecuzione sulla medesima macchina, 127.0.0.1), ed un valore numerico valido per la porta ma differente da quella su cui è in attesa il server (nell'esempio, 7878). Il risultato atteso è un messaggio di errore, seguito dalla richiesta del nuovo indirizzo di un server al quale connettersi.

```
Inserire la porta del server: 8080
Server socket aperto correttamente
```

```
C:\WINDOWS\system32\cmd.  ×  +  v
Inserisci l'indirizzo IP del server: 127.0.0.1
Inserisci la porta del server: 7878
addr = /127.0.0.1
Errore nella connessione con il server.
Inserisci l'indirizzo IP del server: |
```

### 4.3 Indirizzo IP del server errato

Si avvia il server dell'applicazione, specificando per questo una porta valida (nell'esempio, 8080). Successivamente si procede ad avviare il client, specificando per la porta il valore della porta sulla quale è in attesa il server (nell'esempio, 8080) ed un indirizzo IP in formato non valido (nell'esempio una stringa di testo). Il risultato atteso è un messaggio di errore, seguito dalla richiesta del nuovo indirizzo di un server al quale connettersi.

```
Inserisci l'indirizzo IP del server: asdfasdf
Inserisci la porta del server: 8080
Errore nella connessione con il server.
Inserisci l'indirizzo IP del server: |
```

#### 4.3.1 Indirizzo IP e porta del server errati

Si procede ad avviare il client, specificando un valore non numerico (nell'esempio, una stringa di testo) sia per la porta che per l'indirizzo IP del server. Il risultato atteso è un messaggio di errore, seguito dalla richiesta del nuovo indirizzo di un server al quale connettersi.

```
Inserisci l'indirizzo IP del server: asdfasdf
Inserisci la porta del server: asdfasdf
Error reading Integer data, MIN_VALUE value returned.
Errore nella connessione con il server.
Inserisci l'indirizzo IP del server: |
```

#### 4.3.2 Connessione al database con credenziali corrette

Si avvia il server su una porta valida e libera (nell'esempio, 8080). Successivamente si esegue il client e si stabilisce la connessione al server (nell'esempio, essendo client e server in esecuzione sulla medesima macchina, si specificano rispettivamente come indirizzo IP e porta i valori 127.0.0.1 e 8080). Si procede quindi ad avviare la connessione al database, specificando le credenziali corrette per l'accesso (nell'esempio, nome database: *MapDB*, username: *MapUser*, password: *map*). Il risultato atteso è la corretta apertura della comunicazione tra client e database.

```
Inserisci il nome del database a cui accedere: mapdb
Inserisci il nome utente con il quale accedere al DB: MapUser
Inserisci la password: map
Accesso al DB effettuato con successo.
Nome tabella:
|
```

#### 4.3.3 Connessione al database con credenziali errate

Si avvia il server su una porta valida e libera (nell'esempio, 8080). Successivamente si esegue il client e si stabilisce la connessione al server (nell'esempio, essendo client e server in esecuzione sulla medesima macchina, si specificano rispettivamente come indirizzo IP e porta i valori 127.0.0.1 e 8080). Si procede quindi ad avviare la connessione al database, specificando credenziali errate (differenti da quella indicate nel paragrafo 4.3.2). Il risultato atteso è un messaggio di errore che permetta successivamente di specificare nuovamente le credenziali per l'accesso al database al fine di effettuare un nuovo tentativo di connessione.

```
Inserisci il nome del database a cui accedere: mapdb
Inserisci il nome utente con il quale accedere al DB: MapUser
Inserisci la password: asdf
Errore: Connessione al database non riuscita. Riprova.
Inserisci il nome del database a cui accedere: |
```

#### 4.3.4 Connessione al database con servizio MySQL non attivo

Si avvia il server su una porta corretta (nell'esempio, 8080). Successivamente si esegue il client e si stabilisce la connessione al server (nell'esempio, essendo client e server in esecuzione sulla medesima macchina, si specificano rispettivamente come indirizzo IP e porta i valori 127.0.0.1 e 8080). Si procede quindi ad effettuare un tentativo di connessione al database specificando le credenziali corrette (riportate in 4.3.2) assicurandosi però che il servizio MySQL non sia in esecuzione. Il risultato atteso è un messaggio di errore, seguito dalla possibilità di inserire nuovamente le credenziali.

```
Inserisci il nome del database a cui accedere: mapdb
Inserisci il nome utente con il quale accedere al DB: MapUser
Inserisci la password: map
Errore: Connessione al database non riuscita. Riprova.
Inserisci il nome del database a cui accedere: |
```

#### 4.3.5 Connessione al database con database richiesto inesistente

Si avvia il server su una porta corretta (nell'esempio, 8080). Successivamente si esegue il client e si stabilisce la connessione al server (nell'esempio, essendo client e server in esecuzione sulla medesima macchina, si specificano rispettivamente come indirizzo IP e porta i valori 127.0.0.1 e 8080). Si procede quindi ad effettuare un tentativo di connessione al database, specificando tra le credenziali un nome di database inesistente. Il risultato atteso è un messaggio di errore che permetta successivamente di riprovare a connettersi ad un database esistente.

```
Inserisci il nome del database a cui accedere: asdf
Inserisci il nome utente con il quale accedere al DB: MapUser
Inserisci la password: map
Errore: Connessione al database non riuscita. Riprova.
Inserisci il nome del database a cui accedere: |
```

#### 4.3.6 Tabella con dati corretti

Si avvia il server su una porta corretta (nell'esempio, 8080). Successivamente si procede a stabilire la connessione tra client e server (nell'esempio, utilizzando gli stessi parametri dei test precedenti) e tra client e database (nell'esempio, specificando le credenziali riportate in 4.3.2). Si procede quindi ad inserire dal client un nome di tabella corretto, contenente dati validi. Per dati validi si intende una tabella che abbia attributi di tipo numerico. Il risultato atteso è la stampa a terminale del server di un messaggio che segnali il corretto caricamento dei dati dalla tabella specificata; nel client verrà invece visualizzato il menù.

```
Nome tabella:
exampletab
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:|
```

```
Inserire la porta del server: 8080
Server socket aperto correttamente
Connessione con il client stabilita: Socket[addr=/127.0.0.1,port=51366,localport=8080]
Stream di input aperto
Stream di output aperto
Tentativo di accesso al database di: MapUser
Connessione al database...
Connessione al database riuscita
Nome della tabella ricevuto: exampletab
Dati tabella caricati correttamente.
|
```

#### 4.3.7 Tabella con dati non omogenei

Si avvia il server su una porta corretta (nell'esempio, 8080). Successivamente si procede a stabilire la connessione tra client e server (nell'esempio, utilizzando i parametri 127.0.0.1 e 8080 essendo entrambe le applicazioni in esecuzione sulla medesima macchina) e tra client e database (credenziali nel paragrafo 4.3.2). Si procede quindi ad inserire dal client un nome di tabella che contenga dati non omogenei (nell'esempio si alternano colonne della tabella contenenti attributi di tipo *float* ed altri di tipo *char*). Il risultato atteso risulta essere un messaggio che segnali l'errore nella lettura delle entry della tabella seguito dalla possibilità di specificarne un'altra del medesimo database.

```
Nome tabella:
caratteri
Errore: presenza di attributi non numerici in tabella.
Nome tabella:
|
```

Nell'esempio, il contenuto della tabella *caratteri* risulta essere quello di seguito riportato.

C1	C2	C3	C4
a	2	0	2
f	1	-1	3
a	3	5	1.2
n	3	4	6
f	2	0	2.1

5 rows in set (0.00 sec)

#### 4.3.8 Tabella non esistente

Si avvia il server su una porta corretta (nell'esempio, 8080). Successivamente si procede a stabilire la connessione tra client e server (nell'esempio, utilizzando gli stessi parametri dei test precedenti) e tra client e database (nell'esempio, specificando le credenziali di 4.3.2). Si procede quindi ad inserire dal client un nome di tabella inesistente nel database. Il risultato atteso è un messaggio di errore che permetta successivamente di specificare un nuovo nome di tabella del database a cui si è eseguito l'accesso.

```
Nome tabella:
asdf
Tabella non trovata.
Nome tabella:
|
```

#### 4.3.9 Tabella vuota

Si avvia il server su una porta corretta (nell'esempio, 8080). Successivamente si procede a stabilire la connessione tra client e server (nell'esempio, utilizzando gli stessi parametri dei test precedenti) e tra client e database (dati di accesso in 4.3.2). Si procede quindi ad inserire dal client un nome di tabella che non contenga alcun dato al suo interno. Il risultato atteso è un messaggio di errore seguito dalla possibilità di specificare un nuovo nome di tabella del database a cui si è eseguito l'accesso.

```
Nome tabella:
vuota
Tabella vuota.
Nome tabella:
```

```
mysql> select * from vuota;
Empty set (0.00 sec)
```

#### 4.3.10 Scelta corretta nel menu

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database. Si procede quindi ad inserire dal client un nome di tabella contenente dati validi. Si effettua ora il tentativo di scelta di ognuna delle operazioni che risulta essere possibile effettuare per mezzo dell'applicazione. Il risultato atteso è la corretta

esecuzione dell'operazione di scelta, seguita dalla visualizzazione delle informazioni utili per il proseguimento dell'esecuzione dell'applicazione.

```
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:1
Inserire il nome dell'archivio (comprensivo di estensione):
|
```

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:2
Introdurre la profondita' del dendrogramma
|
```

#### 4.3.11 Scelta sbagliata nel menu

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database. Si procede quindi ad inserire dal client un nome di tabella contenente dati validi. Si effettua ora il tentativo di scelta di una operazione non valida (nell'esempio si inserisce il valore numerico 4 - al di fuori del range di valori ammessi - e una stringa di testo). Il risultato atteso è un messaggio di errore che permetta successivamente di specificare una nuova scelta nel menu.

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:4
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:asdf
Error reading Integer data, MIN_VALUE value returned.
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:|
```

#### 4.3.12 Carica da file serializzato corretto

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database, procedendo al caricamento dei dati da una tabella. Si procede ad effettuare la prima scelta dal menu, e quindi a specificare il percorso di un file contenente la serializzazione di un dendrogramma (nell'esempio, il file *Serializzazione.ser*). Il risultato atteso è la corretta visualizzazione del dendrogramma serializzato.



```

Inserire il nome dell'archivio (comprensivo di estensione):
Serializzazione.ser
level0:
cluster0:<[1.0,2.0,0.0,0.0]>
cluster1:<[0.0,1.0,-1.0,0.0]>
cluster2:<[1.0,3.0,5.0,0.0]>
cluster3:<[1.0,3.0,4.0,0.0]>
cluster4:<[2.0,2.0,0.0,0.0]>

level1:
cluster0:<[1.0,2.0,0.0,0.0]><[2.0,2.0,0.0,0.0]>
cluster1:<[0.0,1.0,-1.0,0.0]>
cluster2:<[1.0,3.0,5.0,0.0]>
cluster3:<[1.0,3.0,4.0,0.0]>

level2:
cluster0:<[1.0,2.0,0.0,0.0]><[2.0,2.0,0.0,0.0]>
cluster1:<[0.0,1.0,-1.0,0.0]>
cluster2:<[1.0,3.0,5.0,0.0]><[1.0,3.0,4.0,0.0]>

level3:
cluster0:<[1.0,2.0,0.0,0.0]><[0.0,1.0,-1.0,0.0]><[2.0,2.0,0.0,0.0]>
cluster1:<[1.0,3.0,5.0,0.0]><[1.0,3.0,4.0,0.0]>

```

#### 4.3.13 Carica da file serializzato utilizzando percorso non valido

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database, procedendo al caricamento dei dati da una tabella. Si procede ad effettuare la prima scelta dal menu, e quindi a specificare un percorso non valido. Si ritiene non valido un percorso che contenga al cui interno la sottostringa .., al fine di evitare l'accesso a directory non autorizzate. Il risultato atteso è un messaggio di errore che permetta successivamente di specificare un nuovo percorso per il file da caricare.

```

level2:
cluster0:<[1.0,2.0,0.0,0.0]><[2.0,2.0,0.0,0.0]>
cluster1:<[0.0,1.0,-1.0,0.0]>
cluster2:<[1.0,3.0,5.0,0.0]><[1.0,3.0,4.0,0.0]>

Inserire il nome dell'archivio (comprensivo di estensione):
../../password
Errore: percorso del file non valido.
Introdurre la profondita' del dendrogramma
|

```



#### 4.3.14 Carica da file serializzato oggetto di istanza errata

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database, procedendo al caricamento dei dati da una tabella. Si procede ad effettuare la prima scelta dal menu, e quindi a specificare il percorso di un file che risulta essere il frutto della serializzazione di un oggetto non istanza della classe *HierarchicalClusterMiner*, così come ci si aspetta invece che sia. Il risultato atteso è un messaggio di errore che segnali il tentativo di deserializzare il contenuto del file e la successiva possibilità di specificare un nuovo percorso per il file da caricare.

```
Inserire il nome dell'archivio (comprensivo di estensione):  
person.ser  
Errore: riferimento a classi non trovate.  
Inserire il nome dell'archivio (comprensivo di estensione):
```

#### 4.3.15 Carica da file vuoto

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database, procedendo al caricamento dei dati da una tabella. Si specifica poi un file vuoto presente all'interno della directory del server denominato *vuoto.txt* dopo aver effettuato la prima scelta nel menu. Il risultato atteso è un messaggio di errore che segnali il tentativo di deserializzare il contenuto del file e la successiva possibilità di specificare un nuovo percorso per il file da caricare.

```
Inserire il nome dell'archivio (comprensivo di estensione):  
vuoto.txt  
Errore nell'operazione di lettura dal file.  
Inserire il nome dell'archivio (comprensivo di estensione):  
|
```

#### 4.3.16 Carica da file inesistente

Si avvia il server su una porta corretta. Successivamente si stabilisce la connessione tra client e server e tra client e database, procedendo al caricamento dei dati da una tabella. Si effettua la prima scelta nel menu e successivamente si specifica il percorso di un file inesistente all'interno della directory del server. Il risultato atteso è un messaggio di errore che segnali l'avvenuto tentativo di accesso ad un file inesistente e che permetta successivamente di specificare un nuovo percorso per il file da leggere.

```
(1) Carica Dendrogramma da File  
(2) Apprendi Dendrogramma da Database  
Risposta:1  
Inserire il nome dell'archivio (comprensivo di estensione):  
asdf  
Errore: file non trovato.  
Inserire il nome dell'archivio (comprensivo di estensione):
```

#### 4.3.17 Genera con profondità corretta

Si avvia il server su una porta corretta. Si stabilisce la connessione tra client e server e tra client e database, si specifica il nome di una tabella valida e si procede ad effettuare la seconda scelta nel menu. Dal client si procede ad introdurre una profondità valida (nell'esempio, 1) e una distanza corretta (nell'esempio l'*Average-link Distance*) per la generazione del dendrogramma; relativamente alla serializzazione del dendrogramma, si specifica un percorso valido (nell'esempio, *serializzazione.ser*). Il risultato atteso è la corretta visualizzazione del dendrogramma generato.

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta: 2
Introdurre la profondita' del dendrogramma:
1
Distanza: single-link (1), average-link (2):
2
level0:
cluster0:<[1.0,2.0,0.0,0.0]>
cluster1:<[0.0,1.0,-1.0,0.0]>
cluster2:<[1.0,3.0,5.0,0.0]>
cluster3:<[1.0,3.0,4.0,0.0]>
cluster4:<[2.0,2.0,0.0,0.0]>

Inserire il nome dell'archivio (comprensivo di estensione):
serializzazione.ser
Press any key to continue . . . |
```

#### 4.3.18 Genera con profondità non valida

Si avvia il server su una porta corretta. Si stabilisce la connessione tra client e server e tra client e database, si specifica il nome di una tabella valida e si procede ad effettuare la seconda scelta nel menu. Dal client si procede ad introdurre una profondità non valida (nell'esempio, 999999) e una distanza corretta (nell'esempio la *Single-link distance*). Il risultato atteso è un messaggio di errore che segnali l'inserimento di un valore di profondità non valido e la possibilità di specificarne uno nuovo.

```
Scegli una opzione
(1) Carica Dendrogramma da File
(2) Apprendi Dendrogramma da Database
Risposta:2
Introdurre la profondita' del dendrogramma
999999
Distanza: single-link (1), average-link (2):
1
Errore: dimensione del dendrogramma non valida. Inserire un intero compreso tra 1 e 5
Introdurre la profondita' del dendrogramma
|
```

#### 4.3.19 Genera con profondità non numerica

Si avvia il server su una porta corretta. Si stabilisce la connessione tra client e server e tra client e database, si specifica il nome di una tabella valida e si procede ad effettuare la seconda scelta nel menu. Dal client si procede ad introdurre una profondità non numerica (nell'esempio, una stringa di testo) e una distanza corretta (nell' esempio la *Single-link distance*). Il risultato atteso è un messaggio di errore che segnali l'inserimento di un valore per la profondità non valido e la possibilità di specificarne uno nuovo.

```
Introdurre la profondita' del dendrogramma
asdf
Error reading Integer data, MIN_VALUE value returned.
Distanza: single-link (1), average-link (2):
1
Errore: dimensione del dendrogramma non valida. Inserire un intero compreso tra 1 e 5
Introdurre la profondita' del dendrogramma
|
```

#### 4.3.20 Genera con percorso non valido per il salvataggio

Si avvia il server su una porta corretta. Si stabilisce la connessione tra client e server e tra client e database, si specifica il nome di una tabella valida e si procede ad effettuare la seconda scelta nel menu. Dal client si procede ad introdurre una profondità e una distanza corrette (nell'esempio rispettivamente e 3 e *Single-link distance*). Successivamente si inserisce un path non valido per il salvataggio del dendrogramma generato (nell'esempio, una stringa di testo che contiene la sottostringa ".."). Il risultato atteso è un messaggio di errore che segnali l'inserimento del percorso per il salvataggio del dendrogramma non valido e la possibilità di specificarne uno nuovo.

```
level2:
cluster0:<[1.0,2.0,0.0,0.0]><[2.0,2.0,0.0,0.0]>
cluster1:<[0.0,1.0,-1.0,0.0]>
cluster2:<[1.0,3.0,5.0,0.0]><[1.0,3.0,4.0,0.0]>

Inserire il nome dell'archivio (comprensivo di estensione):
../../password
Errore: percorso del file non valido.
Introdurre la profondita' del dendrogramma
|
```

#### 4.3.21 Chiusura prematura del server durante l'esecuzione del client

Si avvia il server su una porta corretta. Si stabilisce la connessione tra client e server e tra client e database. Si procede quindi ad arrestare il server, inserendo poi il nome della tabella del database da consultare lato client. Al momento dell'invio del nome della tabella dal client al server dovrà verificarsi un errore, visualizzato lato client, per segnalare l'interruzione della comunicazione.

```
Inserisci l'indirizzo IP del server: 127.0.0.1
Inserisci la porta del server: 8080
addr = /127.0.0.1
Socket[addr=/127.0.0.1,port=8080,localport=63417]
Connessione con il server stabilita.
Inserisci il nome del database a cui accedere: mapdb
Inserisci il nome utente con il quale accedere al DB: MapUser
Inserisci la password: map
Accesso al DB effettuato con successo.
Nome tabella: exampletab
java.net.SocketException: Connection reset by peer
Press any key to continue . . . |
```

#### 4.3.22 Chiusura prematura del client durante l'esecuzione del server

Si avvia il server su una porta corretta. Si stabilisce la connessione tra client e server. Si procede quindi ad arrestare il client prematuramente. Il risultato atteso è un messaggio di errore che segnali sul server la prematura disconnessione del client.

```
Dati tabella caricati correttamente.
Errore nella comunicazione con il client.
Chiuso ServerSocket[addr=0.0.0.0/0.0.0.0,localport=8080]
Press any key to continue . . . |
```

## 5 Test Progetto Esteso

### 5.1 Server

#### 5.1.1 Bottone di avvio premuto senza aver inserito testo

Si avvia il server e, senza aver inserito alcun testo all'interno del campo relativo alla porta sulla quale avviare il server, si preme il pulsante "Start Server". Il risultato atteso risulta essere il corretto avvio del server: nel caso in cui non venga specificato alcun testo, alla porta viene assegnato il valore di default *8080* (visibile come prompt all'interno della casella di testo).



#### 5.1.2 Porta corretta

Si avvia il server e si inserisce all'interno del campo relativo alla porta un valore valido (nell'esempio, *8080*), procedendo quindi a premere il pulsante "Start server". Si ricorda che il valore assegnato ad una porta è ritenuto corretto se rispetta le specifiche precedentemente introdotte 4.1.1. Il risultato atteso risulta essere il corretto avvio del server sulla porta specificata.



### 5.1.3 Porta errata (valore fuori dal range)

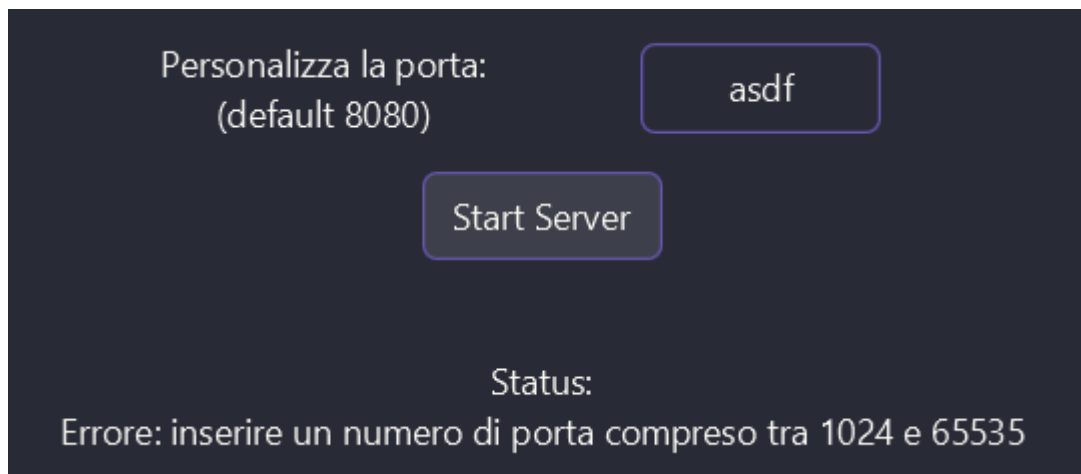
Si avvia il server e si inserisce all'interno del campo relativo alla porta un valore fuori dal range ammesso (nell'esempio, 9999999) e si preme il pulsante "Start Server". Il risultato atteso è un messaggio di errore che permetta successivamente di specificare una nuova porta sulla quale eseguire il server.



The screenshot shows a dark-themed interface with the text "Personalizza la porta: (default 8080)" at the top left. To its right is a text input field containing "9999999". Below the input field is a button labeled "Start Server". At the bottom, the "Status:" section displays the error message: "Errore: inserire un numero di porta compreso tra 1024 e 65535".

### 5.1.4 Porta errata (valore non numerico)

Si avvia il server e si inserisce all'interno del campo relativo alla porta un valore non numerico (nell'esempio, una stringa di testo) e si preme il pulsante "Start Server". Il risultato atteso è un messaggio di errore, che permetta poi di specificare una nuova porta sulla quale eseguire il server.



The screenshot shows the same dark-themed interface. The text "Personalizza la porta: (default 8080)" is at the top left. The text input field to its right now contains the string "asdf". Below the input field is the "Start Server" button. The "Status:" section at the bottom displays the error message: "Errore: inserire un numero di porta compreso tra 1024 e 65535".

### 5.1.5 Porta già in uso

Si avvia il servizio MySQL utilizzando la porta 3306. Successivamente si esegue il server, specificando nell'apposita casella di testo il valore corrispondente ad una porta occupata (nell'esempio 3306) e si preme il pulsante "Start Server". Il risultato atteso è un messaggio di errore che segnali il tentativo di utilizzo di una porta già occupata, seguito dalla possibilità di specificare una nuova porta.



### 5.1.6 Chiusura del server dal pulsante "Stop Server"

Si avvia il server. Nella schermata di log, premendo il pulsante "Stop Server" questo assicurerà una chiusura corretta del programma, in qualsiasi momento della eventuale comunicazione con il client questo si trovi. Per chiusura corretta, si intende l'arresto delle connessioni avviate e degli eventuali thread secondari.

### 5.1.7 Chiusura del server dal pulsante [X] della finestra

Si avvia il server. Premendo il pulsante [X] della *title bar* si assicura una corretta chiusura del programma, eseguendo le medesime operazioni previste nel caso della chiusura dell'applicazione per mezzo del pulsante "Stop Server".

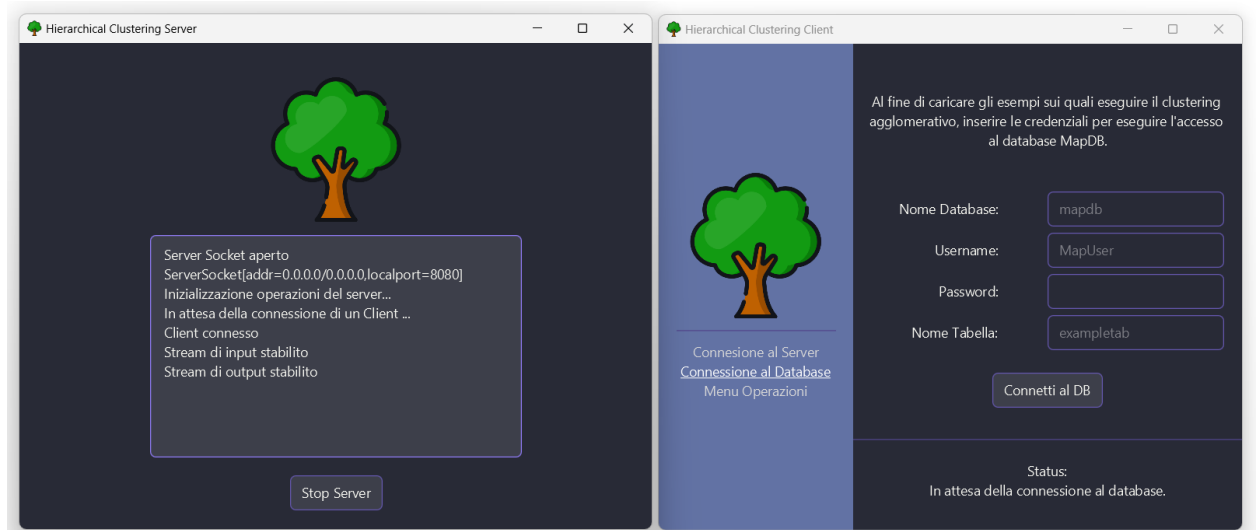
## 5.2 Client

### 5.2.1 Bottone di avvio premuto senza aver inserito testo

Si avvia il server, specificando per questo una porta valida (nell'esempio, 8080) e premendo il pulsante "Start server". Successivamente si avvia il client sulla medesima macchina sulla quale è in esecuzione il server e si clicca il pulsante "Inizia", senza specificare alcun valore all'interno delle apposite caselle di testo. Ci si aspetta che la connessione tra client e server avvenga correttamente, in quanto si prevede che nel caso in cui non venga inserito alcun testo nei campi previsti per specificare IP e porta vengano considerati i valori di default *127.0.0.1* e *8080*.<sup>1</sup>

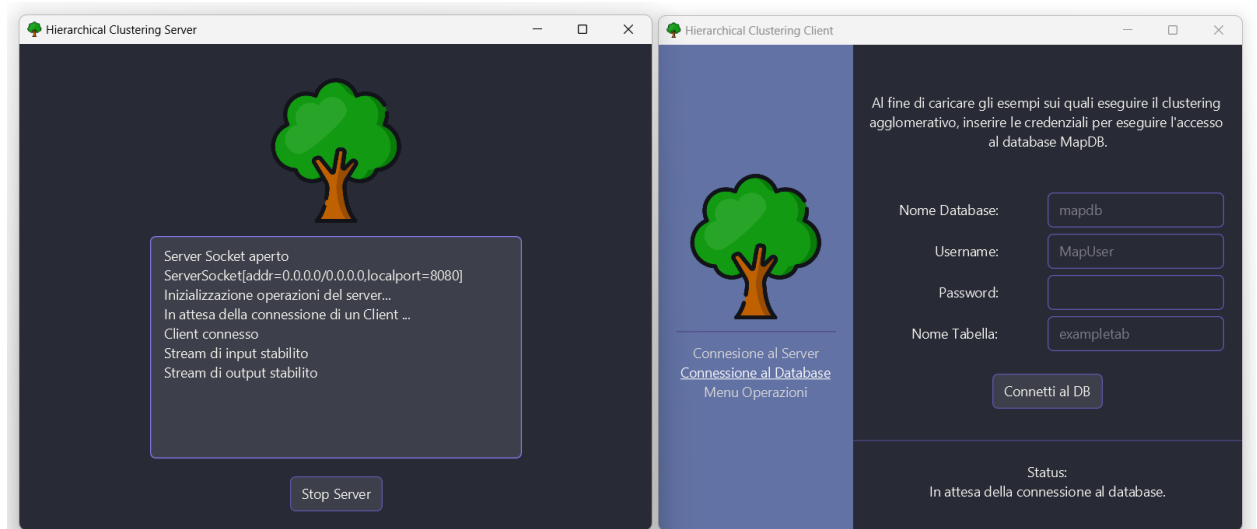
---

<sup>1</sup>Si precisa che la schermata del client visualizzata all'interno della precedente immagine risulta essere quella mostrata dopo aver premuto il pulsante "Inizia", in quanto il client vi ci giunge non appena ha stabilito la connessione. Resta tuttavia possibile verificare che quest'ultima sia stata stabilita grazie al *log* disponibile nel server.



### 5.2.2 Connessione al server avviato

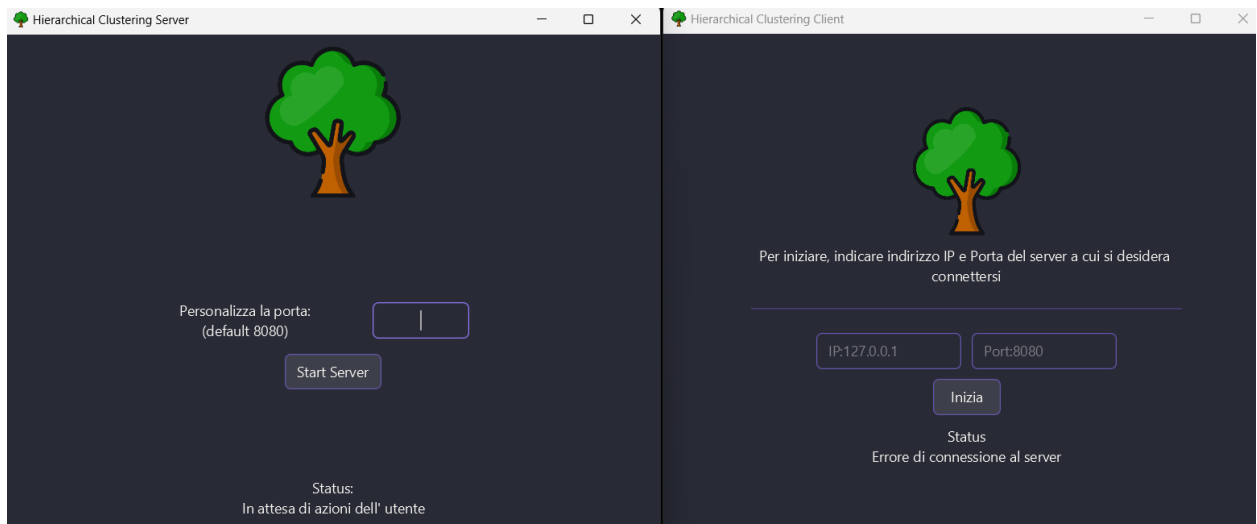
Si avvia il server, specificando per questo una porta valida (nell'esempio, 8080) e premendo il pulsante "Start server". Successivamente si avvia il client sulla medesima macchina sulla quale è in esecuzione il server e si specificano rispettivamente l'indirizzo ip del *localhost* (ossia 127.0.0.1) e la porta 8080, premendo poi il pulsante "Inizia". Ci si aspetta che venga quindi correttamente stabilita la connessione tra client e server al termine delle operazioni descritte.<sup>1</sup>



### 5.2.3 Connessione al server non avviato

A server non in esecuzione su alcuna porta, si avvia il client e si clicca il pulsante "Inizia", utilizzando quindi come indirizzo IP e porta rispettivamente i prompt 127.0.0.1 e 8080. Ciò che ci si aspetta è che il client mostri un messaggio di errore attendendo che venga nuovamente premuto il pulsante "Inizia" per tentare di stabilire una nuova connessione con il server all'indirizzo specificato.





#### 5.2.4 Porta del server errata

#### 5.2.5 Indirizzo IP del server errato

Si avvia il server specificando una porta valida e libera (nell'esempio, 8080). Si esegue il client specificando nei rispettivi campi di testo un indirizzo IP differente da quello del server (nell'esempio, 999.999.999.999) e la porta corretta; si preme il pulsante "Inizia". Ci si aspetta che il client segnali l'impossibilità di stabilire una connessione con il server dall'indirizzo specificato, fornendo la possibilità di inserire un nuovo indirizzo IP e una nuova porta per tentare una nuova connessione.



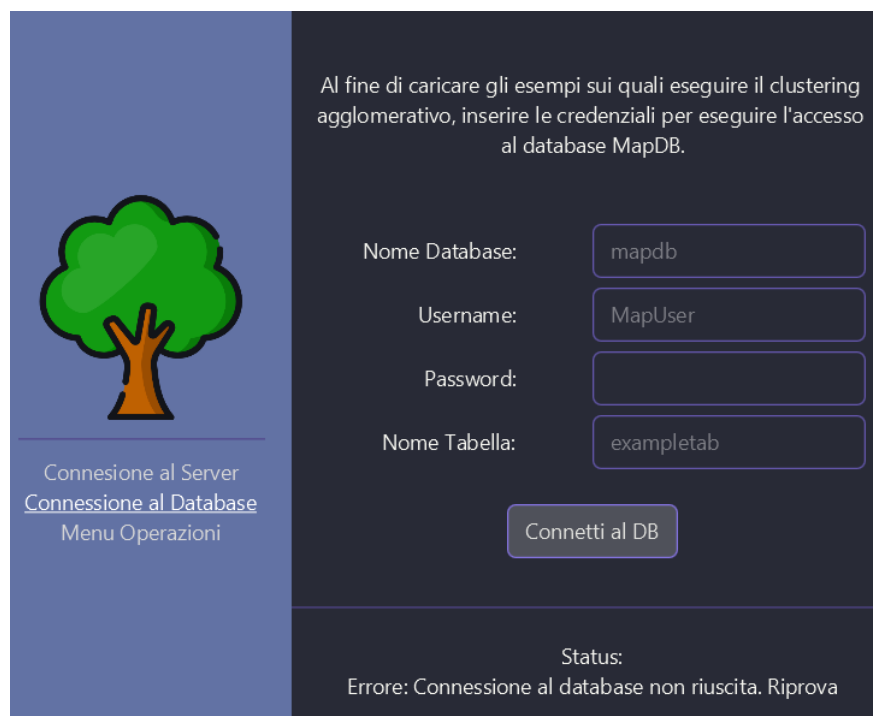
#### 5.2.6 Indirizzo IP e porta del server errati

Si avvia il server specificando una porta valida e libera (nell'esempio, 8080). Si esegue il client specificando nei rispettivi campi di testo un indirizzo IP e una porta in formato non numerico (nell'esempio, delle stringhe di testo). Si procede allora a premere il pulsante "Inizia". Ci si aspetta che il client segnali l'errore relativo all'inserimento dell'indirizzo IP e della porta e che attenda che venga nuovamente cliccato il pulsante "Inizia" per ripetere il tentativo di connessione.



### 5.2.7 Bottone di accesso premuto senza aver inserito testo

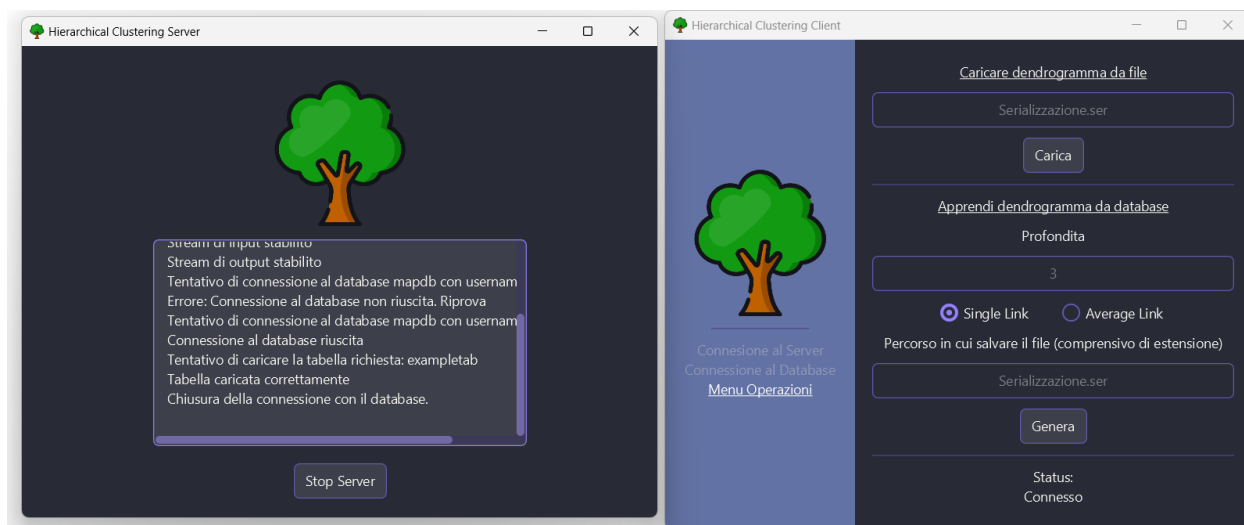
Si eseguono il server e il client mantenendo nei campi di testo i prompt previsti. Giunti nella schermata del client che richiede l'inserimento delle credenziali del database al quale si desidera effettuare la connessione, si preme il pulsante "Connetti al DB" senza specificare alcun valore nei campi di testo della schermata. Il risultato atteso risulta essere un messaggio di errore che segnali l'impossibilità di stabilire la connessione al database richiesto. Questo accade in quanto si è previsto l'inserimento di valori default, nel caso in cui non vengano specificati dall'utente, per i campi *nome database* (valore di default: *mapdb*), *username* (valore di default: *MapUser*) e *nome tabella* (valore di default: *exampletab*). Si è invece preferito non attribuire un valore default al campo *password*.



### 5.2.8 Connessione al database con credenziali corrette

Si eseguono il server e il client mantenendo nei campi di testo i prompt previsti; si attiva inoltre il servizio MySQL e ci si assicura che esista un database di nome *mapdb* con una

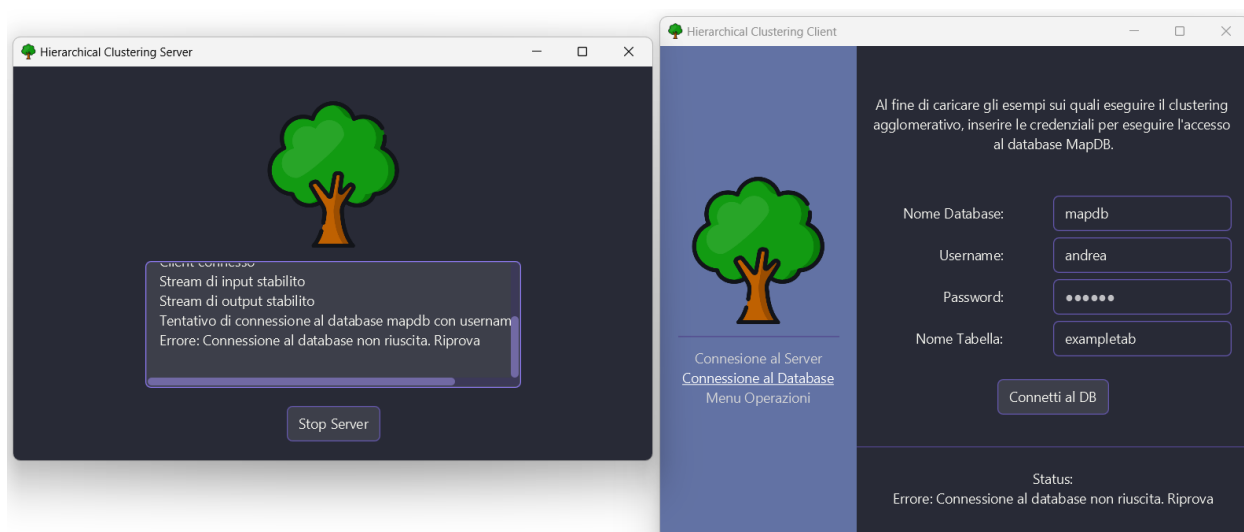
tabella di nome *exampletab* a cui sia previsto l'accesso da un utente con username *MapUser* e password *map*. Nella successiva schermata del client, quella relativa alla connessione con il database, si specifica solo il valore della password (nell'esempio, *map*), e si clicca il pulsante "Connetti al DB". Ci si aspetta che venga correttamente stabilita la connessione con il database richiesto.<sup>2</sup>



Si noti che la scelta di chiudere la connessione con il database a seguito della consultazione della tabella trova giustificazione nel fatto che non sarà possibile tornare indietro per scegliere una tabella differente da cui prendere gli esempi da computare.

### 5.2.9 Connessione al database con credenziali errate

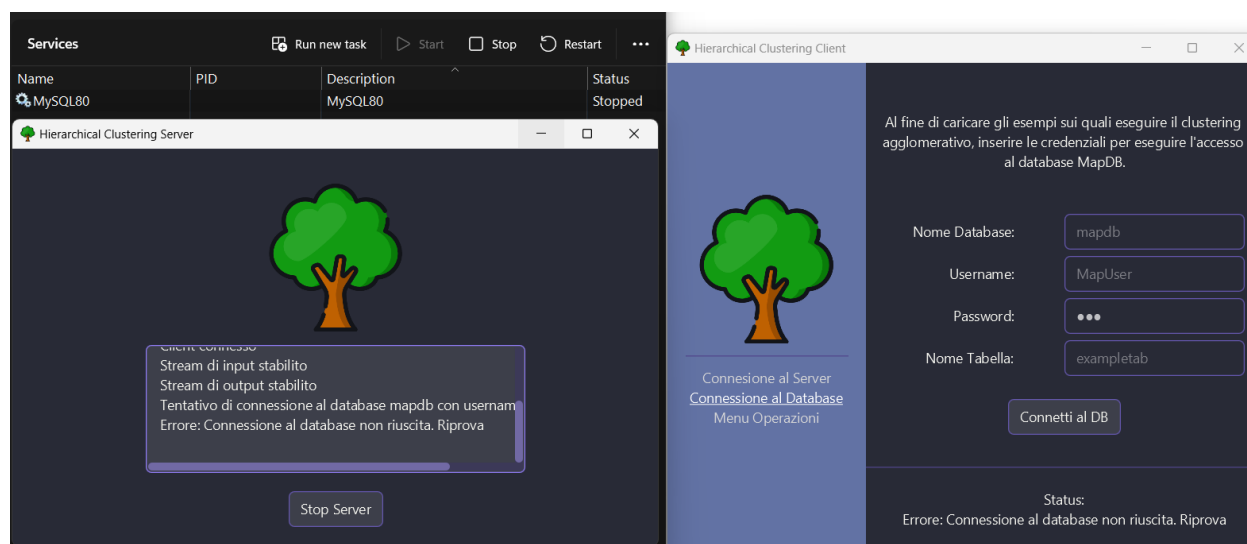
Si eseguono il server e il client mantenendo nei campi di testo i prompt previsti. Nella successiva schermata del client, quella relativa alla connessione con il database, si specifica uno username e una password differenti da quelli previsti per l'accesso al database (si prende come riferimento il database di 5.2.8). Ci si aspetta che il client segnali l'impossibilità di connettersi al database con le credenziali specificate e che attenda che venga nuovamente cliccato il pulsante "Connetti al DB" per riprovare l'operazione.



<sup>2</sup>Nell'immagine, dal lato client, risulta visualizzata la schermata che questo raggiunge nel momento in cui riesce a stabilire la connessione con il database. E' comunque possibile verificare che la connessione sia stata correttamente stabilita dal log del server.

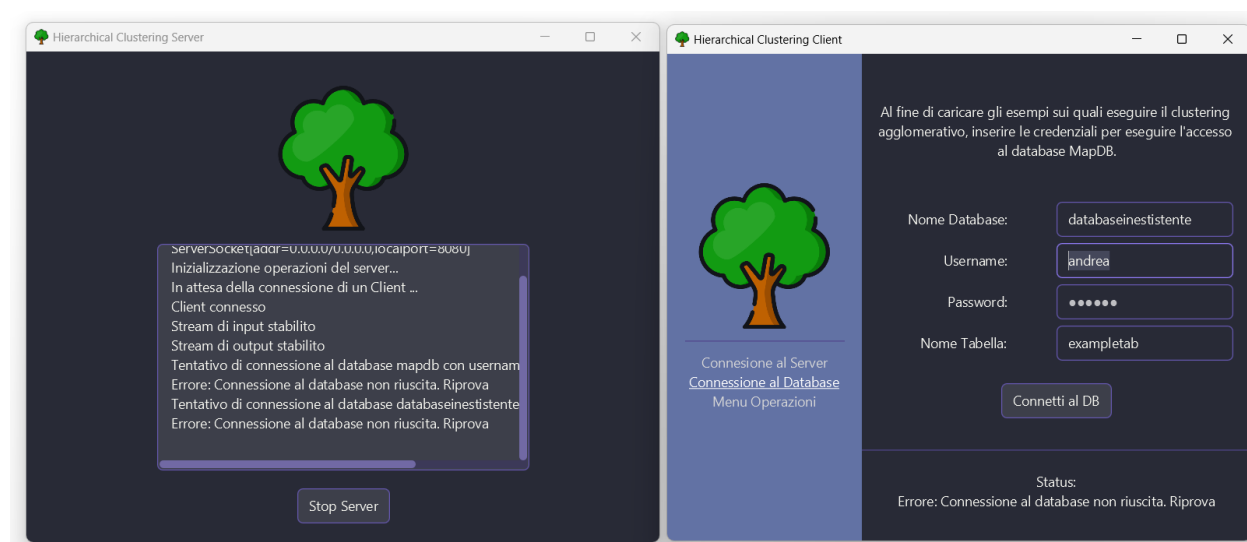
### 5.2.10 Connessione al database con servizio MySQL non attivo

Si eseguono il server ed il client su un medesimo elaboratore, sul quale si mantiene il servizio relativo a MySQL spento. Si procede successivamente ad avviare il server ed il client, mantenendo nelle caselle di testo i prompt previsti dall'applicazione. Giunti nella schermata del client che richiede le credenziali per l'accesso al database, si procede ad inserire la password *map* nell'apposito campo, e a sfruttare i prompt previsti per tutte le altre informazioni richieste (si fa riferimento al database specificato in 5.2.8). Ci si aspetta che il client segnali l'errore nello stabilire la connessione al database e che attenda che sia nuovamente premuto il pulsante per riprovare.



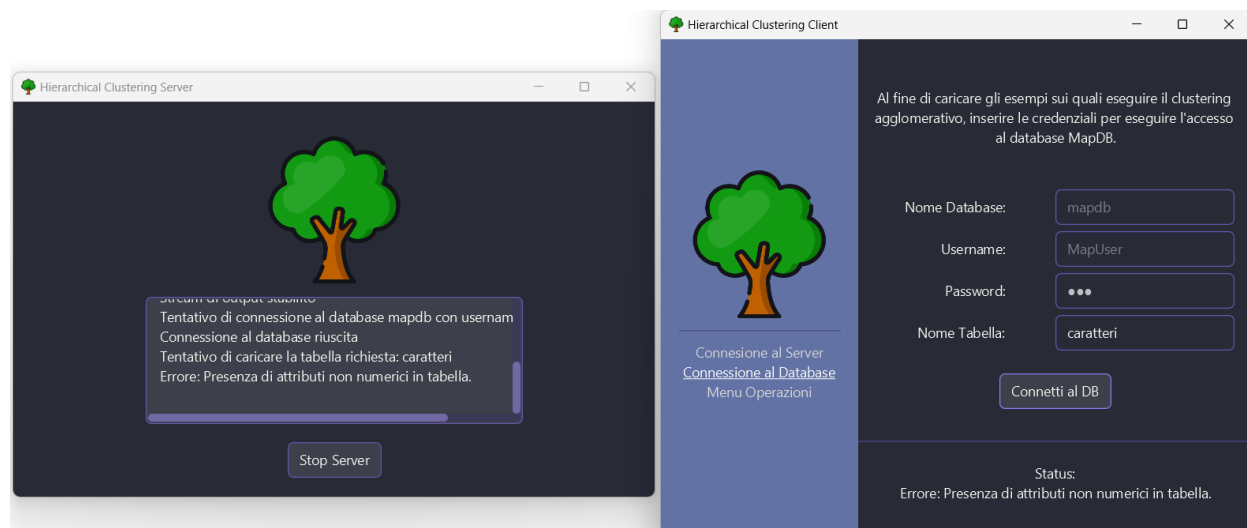
### 5.2.11 Connessione al database con database richiesto inesistente

Si eseguono il server ed il client su un medesimo elaboratore, sul quale si mantiene il servizio relativo a MySQL attivo. Si procede successivamente ad avviare il server ed il client, mantenendo nelle caselle di testo i prompt previsti dall'applicazione. Giunti nella schermata del client che richiede le credenziali per l'accesso al database, si procede ad inserire il nome database *databaseinesistente*, compilando gli altri campi con valori arbitrari (si assume che sulla macchina non si abbia un database chiamato *databaseinesistente*). Il risultato atteso risulta essere un messaggio di errore del client che segnali l'impossibilità di connettersi al database richiesto, rimanendo in attesa che venga nuovamente premuto il pulsante "Connetti al DB" per riprovare a stabilire una connessione.



### 5.2.12 Tabella con dati non omogenei

Si eseguono il server ed il client su un medesimo elaboratore, sul quale si mantiene attivo il servizio relativo a MySQL. Si procede successivamente, all'interno della schermata del client che permette di effettuare la connessione al database, ad inserire come *nomedb*, *username* e *password* quelli di 5.2.8 e come nome tabella *caratteri*. Il risultato atteso risulta essere la visualizzazione nel client di un errore relativo ai dati contenuti nella tabella; successivamente deve rimanere in attesa che venga nuovamente premuto il pulsante "Connetti al DB" per riprovare la connessione.



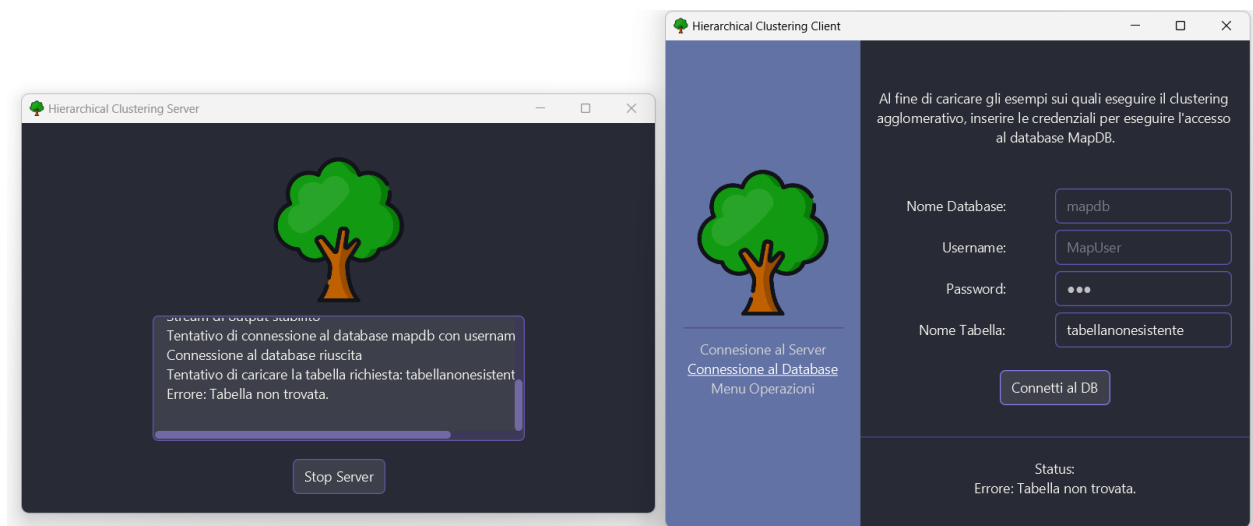
Si precisa che il contenuto della tabella caricata nel caso di test corrente risulta essere quello riportato di seguito:

C1	C2	C3	C4
a	2	0	2
f	1	-1	3
a	3	5	1.2
n	3	4	6
f	2	0	2.1

5 rows in set (0.00 sec)

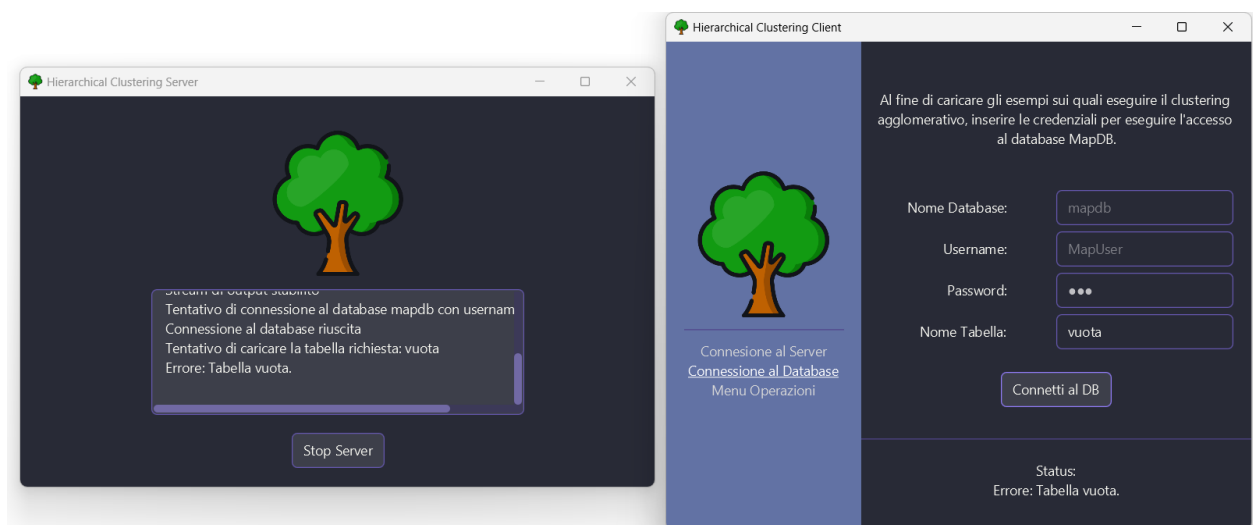
### 5.2.13 Tabella non esistente

Si eseguono il server ed il client su un medesimo elaboratore, sul quale si mantiene attivo il servizio relativo a MySQL. Si procede successivamente, all'interno della schermata del client che permette di effettuare la connessione al database, ad inserire come *nomedb*, *username* e *password* quelli di 5.2.8 e come nome tabella *tabella inesistente*. Il risultato atteso risulta essere la visualizzazione nel client di un errore relativo all'inesistenza della tabella richiesta; successivamente deve rimanere in attesa che venga nuovamente premuto il pulsante "Connetti al DB" per riprovare la connessione.



#### 5.2.14 Tabella vuota

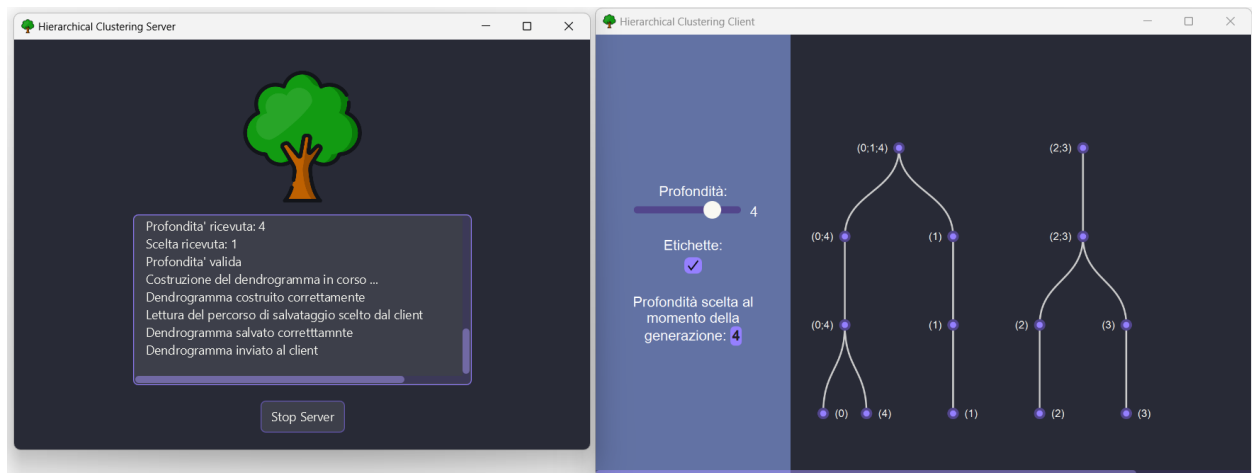
Si eseguono il server ed il client su un medesimo elaboratore, sul quale si mantiene attivo il servizio relativo a MySQL. Si procede successivamente, all'interno della schermata del client che permette di effettuare la connessione al database, ad inserire come *namedb*, *username* e *password* quelli di 5.2.8 e come nome tabella *vuota*. Il risultato atteso risulta essere la visualizzazione nel client di un errore relativo alla mancanza di dati all'interno della tabella; successivamente deve rimanere in attesa che venga premuto il pulsante "Connetti al DB" per riprovare la connessione.



#### 5.2.15 Bottone "Carica" premuto senza aver inserito testo

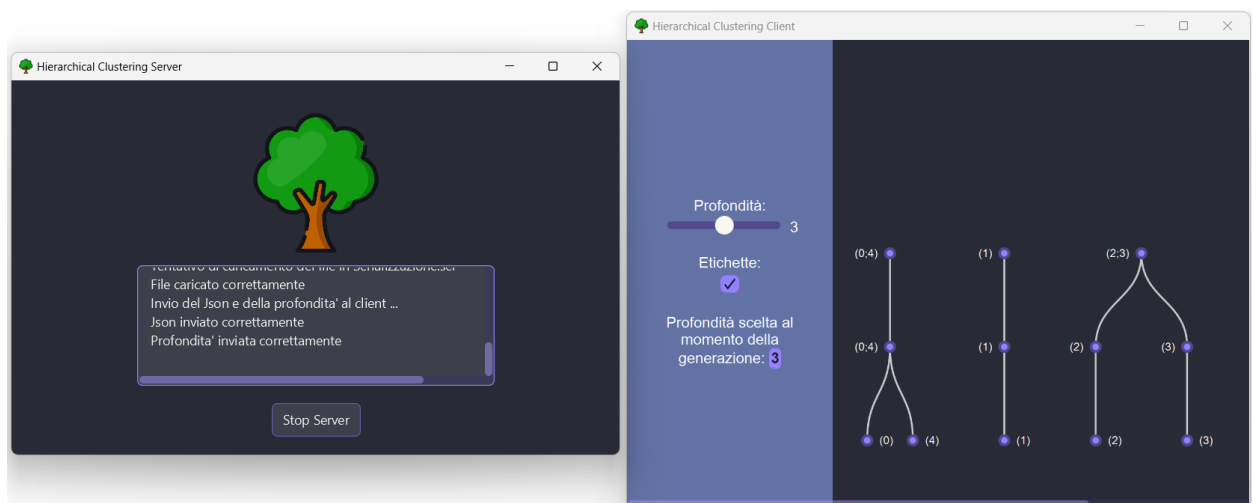
Si eseguono il server e il client su un medesimo elaboratore, procedendo successivamente dal lato client ad effettuare l'accesso al database (utilizzando le credenziali del database introdotte in 5.2.8). Si procede, nella schermata del menu, a premere il pulsante "Carica", riferito alla scelta di caricare un dendrogramma da un file in cui è stato precedentemente serializzato, senza aver specificato alcun percorso nell'apposito campo di testo. Ci si aspetta che l'applicazione carichi e visualizzi il dendrogramma contenuto all'interno del file *Serializzazione.ser*, già disponibile nel progetto. Tale comportamento risulta essere dovuto alla

scelta di compensare la mancanza di informazioni inserite dall'utente nel predetto campo di testo con un valore di default fissato al nome del file appena caricato.<sup>3</sup>



### 5.2.16 Carica da file serializzato corretto

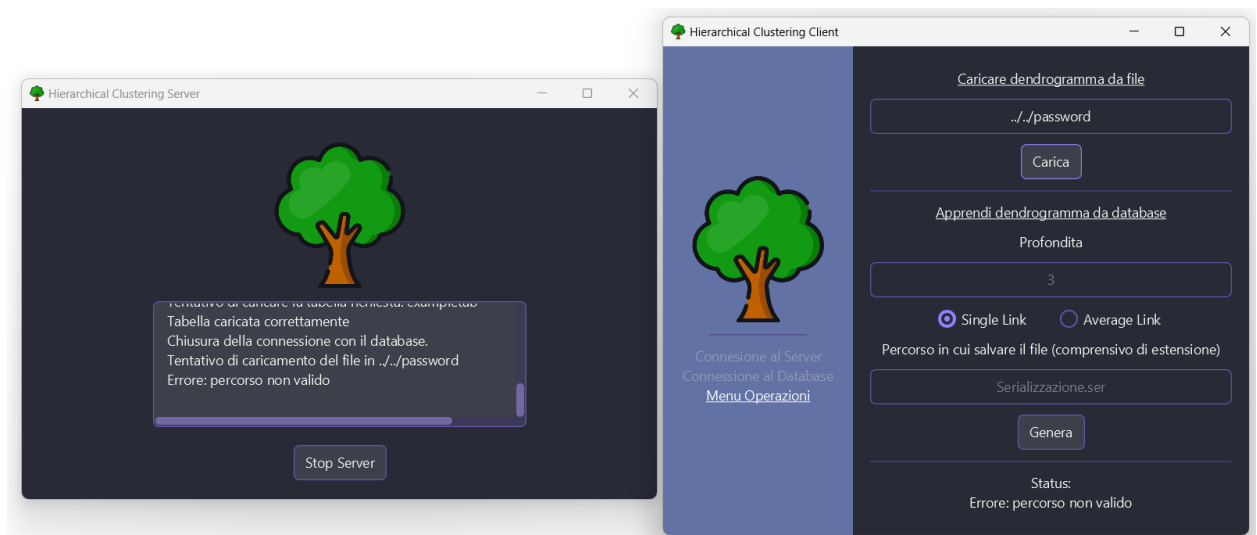
Si eseguono il server e il client su un medesimo elaboratore, procedendo successivamente dal lato client ad effettuare l'accesso al database (utilizzando le credenziali di 5.2.8). Si procede, nella schermata del menu, a premere il pulsante "Carica" specificando nella casella di testo riservata al *path* del file da cui caricare il dendrogramma il percorso di un file esistente nel progetto (nell'esempio, *Serializzazione.ser*). Ci si aspetta che il file venga correttamente letto e che sia visualizzato a schermo il dendrogramma in esso precedentemente serializzato.



### 5.2.17 Carica da file serializzato utilizzando percorso non valido

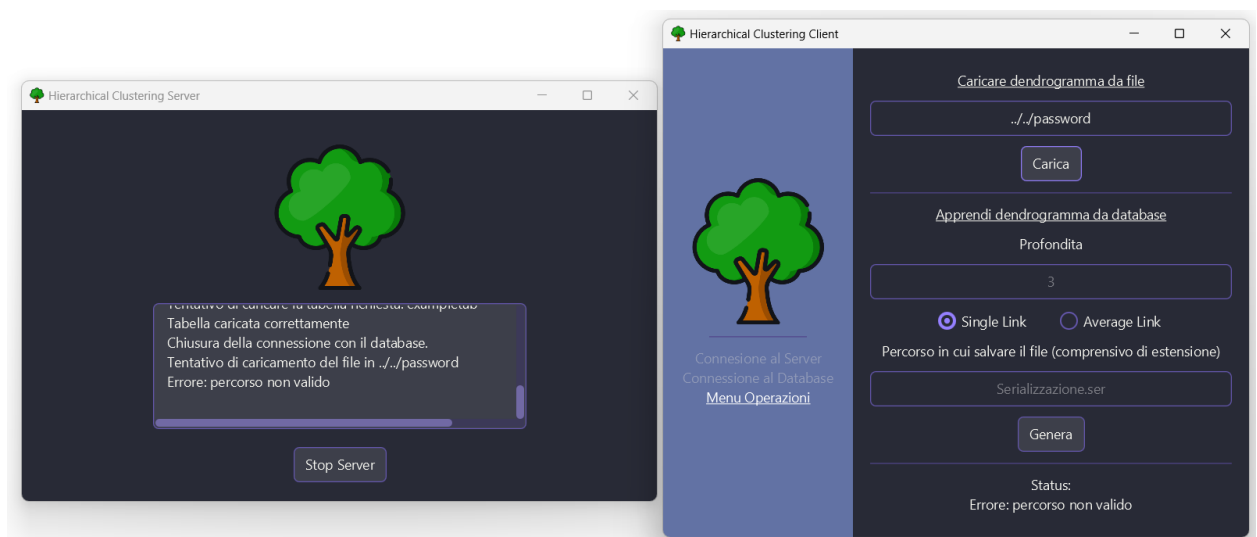
Si eseguono il server e il client su un medesimo elaboratore, procedendo dal lato client ad eseguire l'accesso al database (con le credenziali di 5.2.8). Si inserisce nella schermata successiva un percorso non valido (e quindi contenente al suo interno la sottostringa "..") nella casella di testo riservata al percorso del file da caricare, per poi premere il bottone "carica". Ci si aspetta che il client segnali l'errore di inserimento del percorso file e che attenda che il pulsante venga premuto nuovamente per ritentare l'operazione di caricamento.

<sup>3</sup>Nel client, non appena cliccato il pulsante "Carica", si passa alla schermata successiva nella quale viene visualizzato il dendrogramma. Nell'immagine che rappresentasi riporta quindi la schermata nella quale giunge il client; resta tuttavia traccia dell'operazione di caricamento del dendrogramma andata a buon fine nel log del server.



### 5.2.18 Carica da file serializzato oggetto di istanza errata

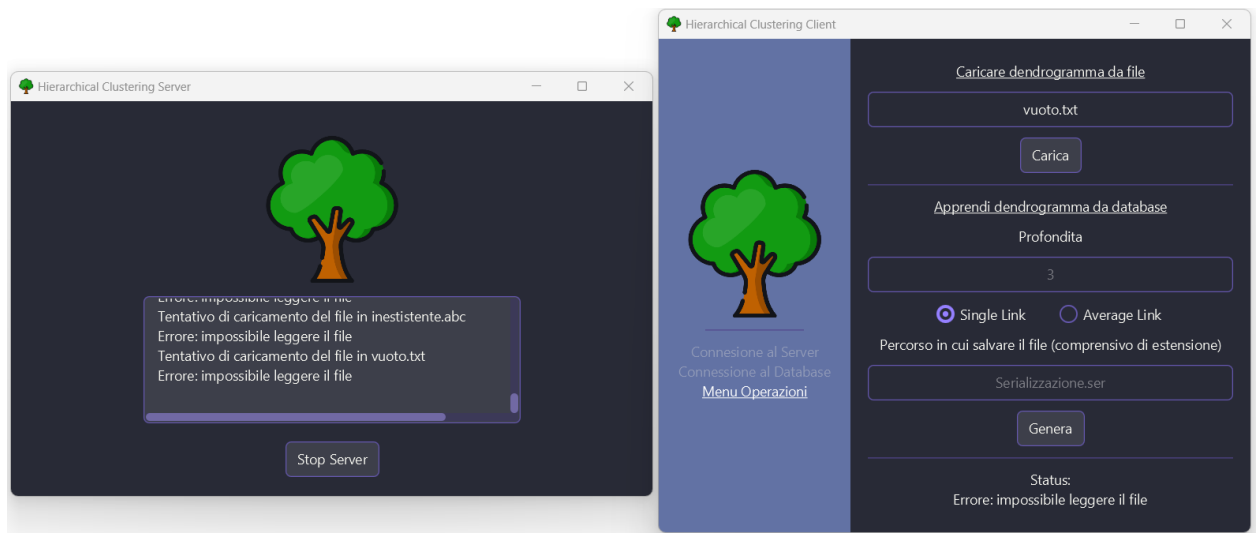
Si eseguono il server e il client su un medesimo elaboratore, procedendo successivamente dal lato client ad effettuare l'accesso al database (utilizzando le credenziali di 5.2.8). Si procede successivamente, nella schermata del menu, a premere il pulsante "Carica" specificando nella casella di testo riservata al *path* del file da cui caricare il dendrogramma il percorso di un file in cui è serializzato un oggetto non istanza di *HierarchicalClusterMiner* (nell'esempio, *pearson.ser*). Ci si aspetta che il server segnali l'errore all'utente e attenda che il pulsante "Carica" venga ripremuto, tentando una nuova operazione di caricamento.



### 5.2.19 Carica da file vuoto

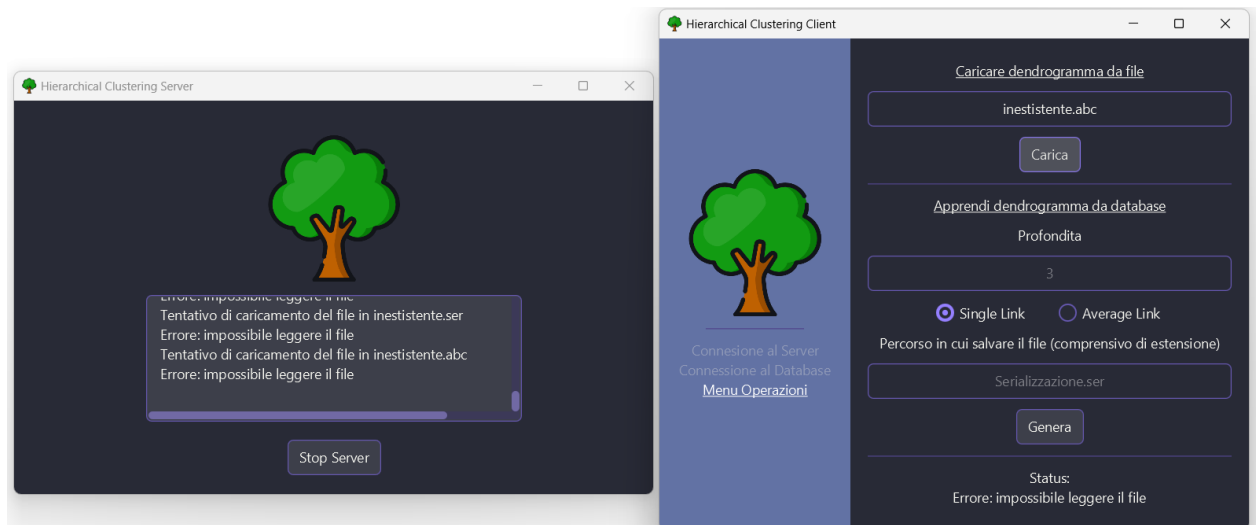
Si eseguono il server e il client su un medesimo elaboratore, procedendo ad effettuare dal client la connessione al server (credenziali nel paragrafo 5.2.8). Successivamente si inserisce all'interno dell'apposita casella di testo il percorso di un file vuoto (nell'esempio, *vuoto.txt*), e si preme il pulsante "Carica". Ci si aspetta che il client restituisca un errore, attendendo che il pulsante venga nuovamente premuto per riprovare l'operazione di caricamento di un file.





### 5.2.20 Carica da file inesistente

Si eseguono il server e il client su un medesimo elaboratore, procedendo ad avviare nel client la connessione al server (database del paragrafo 5.2.8). Successivamente si inserisce all'interno dell'apposita casella di testo il percorso di un file inesistente (nell'esempio, *inesistente.abc*), e si preme il pulsante "Carica". Ci si aspetta che il client visualizzi un errore, e che attenda che il pulsante venga nuovamente premuto per riprovare l'operazione di caricamento di un file.



### 5.2.21 Bottone "Genera" premuto senza aver inserito testo

Si esegue la connessione tra il server e il client. Si effettua all'interno del client la connessione e l'autenticazione nel database (credenziali nel paragrafo 5.2.8). Si preme, nella schermata del menu, il pulsante "Genera", senza aver specificato alcun valore all'interno delle caselle di testo relative alla profondità del dendrogramma, alla distanza da utilizzare e al percorso in cui salvare il file che serializzerà il dendrogramma generato. Ci si aspetta che il server restituisca un errore: sebbene siano presenti i prompt relativi al tipo di distanza (di default risulta essere selezionata la *Single-link distance*) e al percorso in cui serializzare il file (di default *Serializzazione.ser*), nessun prompt risulta essere previsto relativamente alla profondità del dendrogramma.

Apprendi dendrogramma da database

Profondità

3

☒ Single Link ☐ Average Link

Percorso in cui salvare il file (comprensivo di estensione)

Serializzazione.ser

Genera

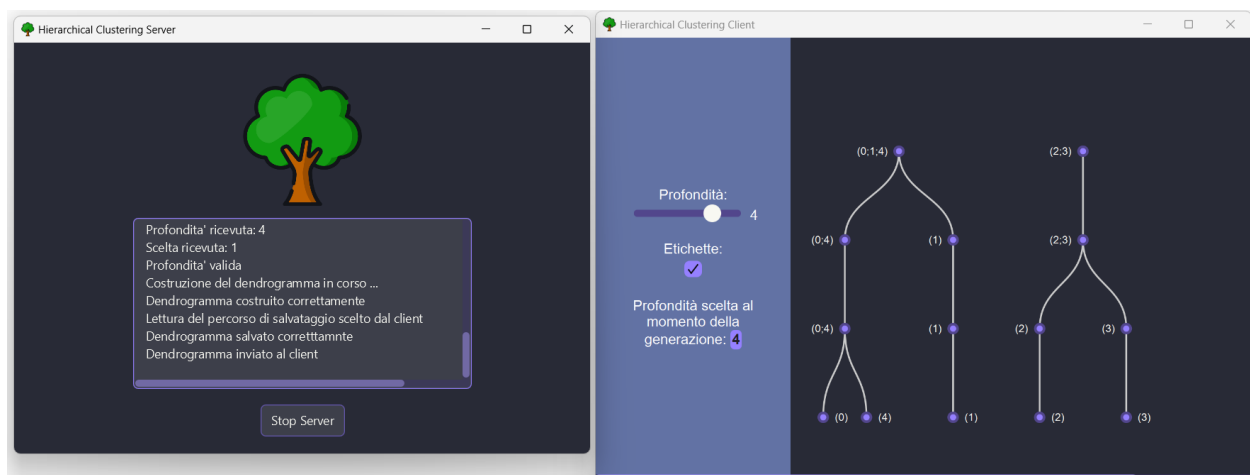
---

Status:

Inserire un valore numerico per la profondità'.

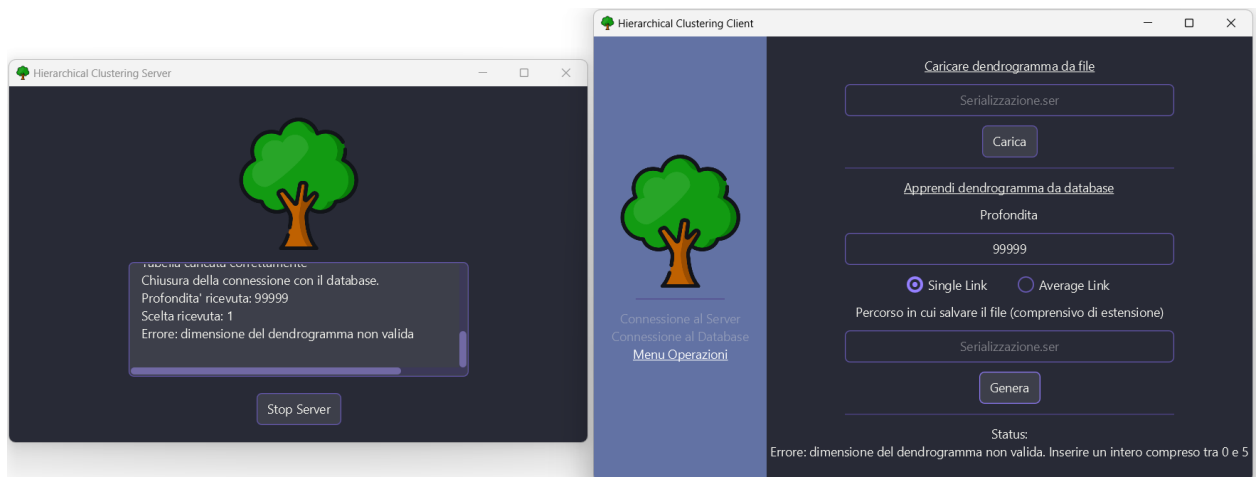
### 5.2.22 Genera con profondità corretta

Si esegue la connessione tra il server e il client e successivamente la connessione e l'autenticazione nel database (dati d'accesso al paragrafo 5.2.8). Si preme, nella schermata del menu, il pulsante "Genera", dopo aver specificato un valore valido per la profondità (nell'esempio, 4), e aver lasciato i valori default previsti per la tipologia di distanza da utilizzare nella visualizzazione del dendrogramma e il nome del file da utilizzare per salvare il dendrogramma serializzato. La profondità del dendrogramma si ritiene valida nel caso in cui sia compresa tra 1 e il numero di esempi contenuti nel dataset caricato dal database. L'esito atteso risulta essere la visualizzazione grafica del dendrogramma generato.



### 5.2.23 Genera con profondità non valida

Si esegue la connessione tra server e client e tra client e database, con le credenziali riportate al paragrafo 5.2.8. Successivamente si inserisce una profondità del dendrogramma, nell'apposita casella di testo, fuori dal range previsto, e si preme il pulsante "Genera". Ci si aspetta che il client mostri un errore e che attenda che sia premuto nuovamente il bottone per poter tentare nuovamente la generazione e visualizzazione del dendrogramma.



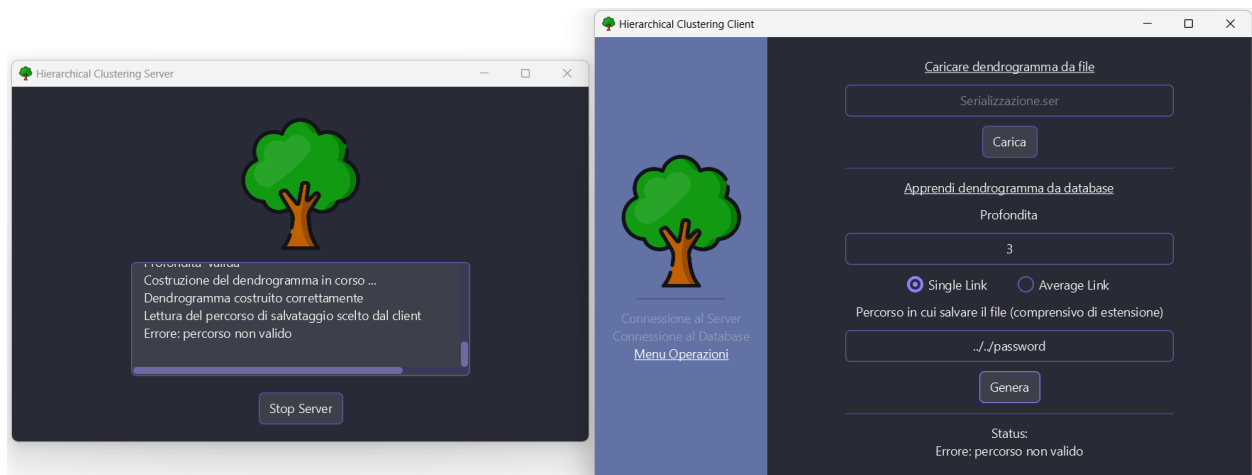
#### 5.2.24 Genera con profondità non numerica

Si apre la connessione tra server e client e tra client e database, con le informazioni riportate a 5.2.8. Successivamente si inserisce una profondità del dendrogramma non valida (nell'esempio, non numerica). Si procede a premere il pulsante "Genera". Ci si aspetta che il client segnali l'errore nella generazione del dendrogramma e che attenda che il bottone sia ripremuto per poter ritentare nuovamente l'operazione richiesta.



#### 5.2.25 Genera con percorso non valido per il salvataggio

Si avvia la connessione tra server e client e tra client e database, sempre sfruttando quanto riportato nel paragrafo 5.2.8. Successivamente si inserisce una profondità corretta per il dendrogramma (nell'esempio, 3) ed un percorso per il salvataggio non corretto. Si ricorda che per percorso non corretto si intende un *path* nel quale sia presente la sottostringa "...". Si procede a premere il pulsante "Genera". Ci si aspetta che il client segnali per mezzo di un messaggio di errore che il percorso di salvataggio fornito non risulti essere corretto; attende che il bottone sia premuto nuovamente per riprovare l'operazione di generazione del dendrogramma.



### 5.2.26 Chiusura prematura del server durante l'esecuzione del client

Si esegue la connessione del client al server, dopo aver opportunamente avviato il server su una porta valida e libera. Si procede successivamente ad eseguire la connessione al database, giungendo nella schermata del menù. Si arresta prematuramente il server. Ci si aspetta che il client, nel momento in cui tenterà di inviare informazioni al server dopo aver cliccato il pulsante "Genera" o "Carica" mostri un errore relativo alla connessione con il server, ormai interrotta.



### 5.2.27 Chiusura prematura del client durante l'esecuzione del server

Si avvia la connessione tra client e server, dopo aver avviato il server su una porta valida e libera. Successivamente, nel momento in cui lato client si visualizzerà la schermata per effettuare la connessione al database, si procederà ad arrestare quest'ultimo. Ci si aspetta che il server visualizzi un messaggio indicante l'interruzione della connessione con il client, rimanendo comunque in esecuzione.

