

Metodi avanzati di programmazione 2020-2021

Caso di studio :
Scoperta di Pattern Frequenti e Pattern Emergenti

Presentazione di:
Nicola Francavilla matricola: 709153
Email: n.francavilla5@studenti.uniba.it

Indice

Indice degli argomenti

- Breve descrizione del Software.
- Differenze tra il nuovo Software e l'originale.
- Guida di installazione
- Guida utente
- Esempi di test

Breve descrizione del Software

Breve descrizione del Software

Pattern Frequenti

- In letteratura possiamo trovare due tipi particolari di pattern. **Pattern Frequenti** i quali sono composti da una collezione D di transizioni dove ognuna di esse è un vettore di Item composto dalle coppie (attributo-valore)

Obiettivo: Identificare gli insiemi di Item che occorrono con una frequenza minima ($\geq \text{minS}$) in D.

Pattern Emergenti

- Un **pattern emergente** è un **pattern frequente** che raggiunge un minimo grow rate

Obiettivo: Identificare tutti i **pattern emergenti** che siano **frequenti** (ossia con un supporto $\geq \text{minS}$) calcolato nel DataSet Dtarget ed **emergenti** (ossia con un grow rate $\geq \text{minGr}$) calcolato nel DataSet Dbackground.

Dati a nostra disposizione:

- **Dtarget** composto da transizioni target
- Un **Dbackground** composto da transizioni di background.
- Valore minimo di supporto ($0 < \text{minS} \leq 1$)
- Valore minimo di growRate ($\text{minGr} \geq 1$)

Breve descrizione del Software

Passo 1

Individuare i pattern Frequenti

- Scoprire tutti i pattern di lunghezza pari a k a partire da **pattern frequenti** con una lunghezza pari a $k-1$.
- Testare i pattern «Candidati» ad essere **pattern frequenti**, calcolati nel dataset **Dtarget**.

Passo 2

Individuare i pattern Emergenti

- Per ciascun Pattern Frequente **P**, si calcola il **growRate** di **P** rispetto al dataset **Dbackground**.
- Alla fine verranno selezionati i soli Pattern con un **grow rate** che sia superiore alla soglia prestabilita.

Algoritmo utilizzato: Apriori

- Classico algoritmo di associazioni utilizzato nell'ambito del **data mining**. Tale algoritmo viene generalmente utilizzato per la generazioni di itemSet frequenti, per approssimazioni successive, partendo da itemSet con un solo elemento.
- **The Apriori algorithm** in pratica si basa sul fatto che se un insieme di oggetti è frequente, allora anche tutti i suoi sottoinsiemi sono frequenti, ma, se un itemSet non è frequente, allora per certo si può dire che nessun insieme che lo contiene è **frequente**.

Breve descrizione del Software

- Il progetto base consiste in un'applicazione distribuita client/server
- Prima di eseguire il server, bisogna definire come argomenti del main del progetto Server una porta e un indirizzo Ip.
- Qualora non lo si specificasse, il Server di default verrà eseguito rimanendo in ascolto sulla **Porta 8080** con **Indirizzo Ip 120.0.0.1**
- Il server si occupa di ricevere le richieste di uno o più Client connessi simultaneamente, i quali possono effettuare le seguenti operazioni:
 1. Definire un minimo supporto.
 2. Definire un minimo GrowRate
 3. Richiedere di effettuare i calcoli specificando il nome di una targetTable e BackGroundTable.
 4. Salvare il risultato all'interno di un file.
 5. Leggere un file precedentemente creato.
 6. Chiudere la connessione con il server.

Differenze tra i due progetti

Estensione VS Progetto originale

Software originale - Before

- Inizialmente il software era stato ideato per avere un lato Client e un lato Server, entrambi sotto forma di console.
- Per tale motivo, l'inserimento dei dati (targetTableName, BackgroundTableName ecc..) da parte dell'utente avveniva nella stessa console.
- Inoltre il server rispondeva alle risposte del client , solo una volta che il server aveva ricevuto tutti i dati che si aspettava di ricevere.

Software con estensione - After

Il cambiamento più radicale del nuovo software rispetto al vecchio è quello di astrarre il concetto di comunicazione tra più client (come in questo caso) connessi simultaneamente ad un unico server che riceve e invia le risposte a ciascuna richiesta del client.

In questa versione rivisitata è stato sostituito l'inserimento dei dati nella console con l'utilizzo di una chat stile «telegram/messenger» con cui l'utente può interfacciarsi con il server.

Tale interfaccia è stata realizzata con l'utilizzo di librerie di **JavaFX Sdk 18**, rendendo più fluido e naturale l'utilizzo del software.

Infine è stato implementata:

- Finestra di login (Il login button rimane disabilitato fino a quando non si inserisce un username e indirizzo IP)
 - il software accetta una qualsiasi password inserita.
- Possibilità di scegliere una lingua (italiana/inglese) per tradurre la finestra di login.

Per maggiori informazioni su JavaFx: Seguire questo link!

Guida installazione

Installazione server e Client

Guida installazione Server



Per poter utilizzare il server è necessario:

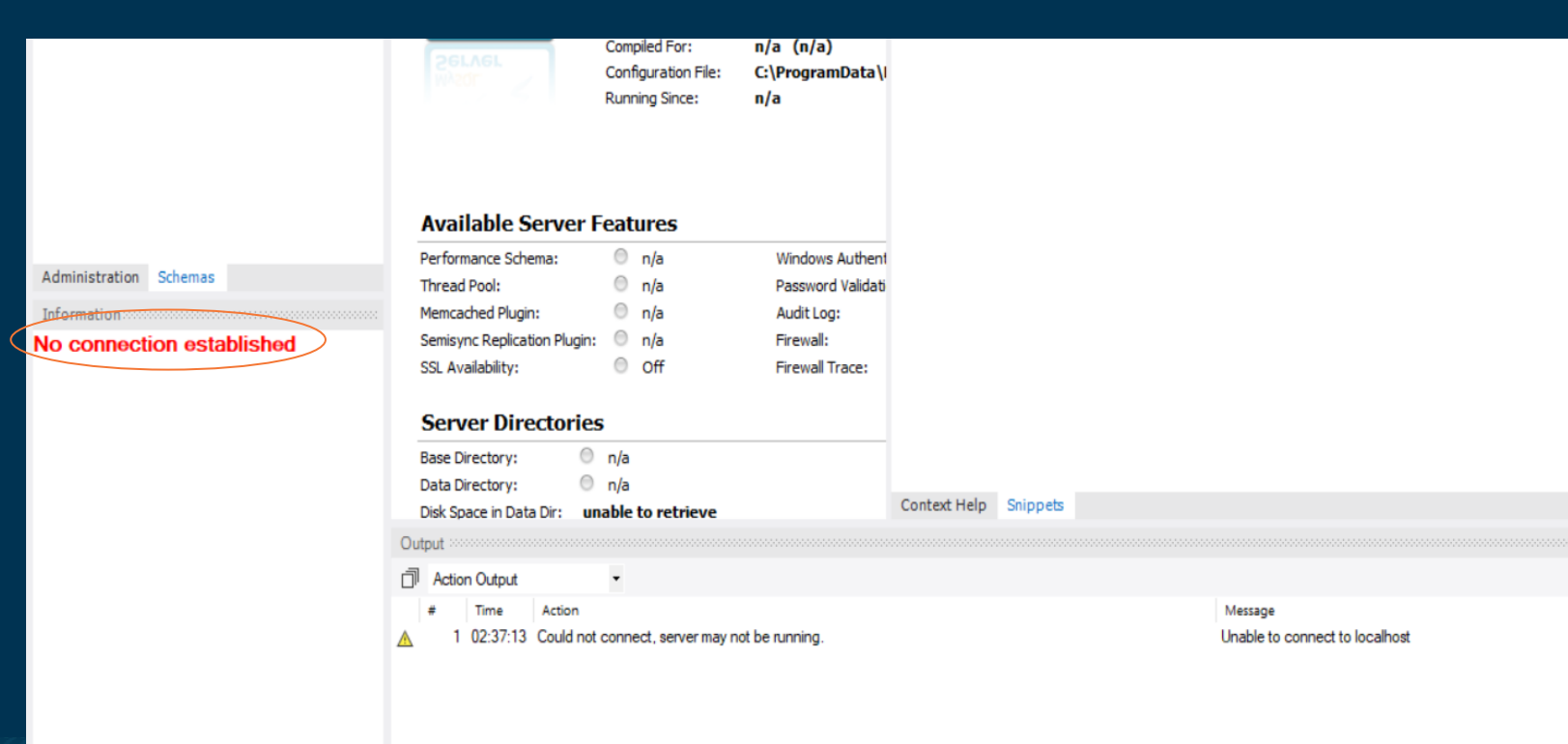
- Installare MySql sul proprio Computer.
- Eseguire lo script SQL (script.sql) presente nel percorso «\EpMiner\sql».
 - Eseguendo tale script nella shell di MySQL verrà creato il database e le tabelle necessarie.
- Installare JRE 8
- E' stata utilizzata la JDK Oracle OpenJDK version 17.0.2

Guida installazione Client

- Installare JRE 8
- Server in ascolto : Attenzione, nel caso in cui si dovesse aprire il client ancora prima di aver eseguito il server, anche se indirizzo Ip e Porta siano corretti chiaramente il client darà errore, ossia che non è in ascolto alcun server sulla porta e indirizzo specificato.

Possibili Problemi di connessione

- Nel caso in cui, aprendo la Workbench di MySql (in cui bisogna creare un server in localhost) dovesse apparire questo messaggio di errore di connessione:



Bisogna:

- Cercare nella barra di ricerca di windows «servizi»
- Trovare il servizio «MYSQL80»
- Tasto destro e cliccare su «Avvia servizio.»

Guida Utente

Progetto EpMiner originale

Guida utente - EpMiner Originale

Avvio Client e server

- All'interno della cartella «EpMinerClient\bin» è possibile trovare un file con estensione .jar, ovvero «EpMinerClient.jar». Tale file .jar è associato al rispettivo file .bat chiamato «EpMinerClient.bat» che contiene la stringa «java -jar "EpMinerClient.jar"», necessaria per avviare il Client (digitarla nel CMD). Di default il progetto originale imposta come parametri l'indirizzo Ip 127.0.0.1 e la porta 8080.
- Per poter avviare correttamente il Client bisogna prima eseguire il server utilizzando lo stesso approccio. Ossia nella cartella «EpMinerServer\bin» è presente il file con estensione .jar «EpMinerServer.jar» il quale per eseguire tale file bisogna digitare all'interno del CMD la stringa «java -jar EpMinerServer.jar» (ATTENZIONE essere nella cartella corretta bin del progetto per poter eseguire tale file.)

Guida utente - EpMiner Originale - Funzionamento Client

Avvio eseguito

- Una volta avviato il client e stabilita la connessione con il server, l'utente potrà selezionare tra due opzioni:
 - 1) Creare un file contenente tutti i pattern frequenti ed emergenti trovati all'interno del database
 - 2) Caricare un file già esistente, ossia creato in precedenza

In entrambi i casi, l'utente dovrà inserire:

- minimo supporto ($\text{minsup} > 0$ e $\text{minsup} \leq 1$) = Nel caso in cui si inserisca un parametro che non rientra in tale condizione verrà visualizzato nuovamente il messaggio di inserimento di tale parametro.

```
addr = /127.0.0.1
port=8080
Socket[addr=/127.0.0.1,port=8080,localport=56937]
Scegli una opzione:
1:Nuova scoperta
2: Risultati in archivio
1
Inserire valore minimo supporto (minsup>0 e minsup<=1):
3
Inserire valore minimo supporto (minsup>0 e minsup<=1):
4
Inserire valore minimo supporto (minsup>0 e minsup<=1):
```

Mentre nel caso in cui si provi ad inserire una stringa, l'applicazione setterà con un valore di default pari a Nan il minimo Supporto.

```
Inserire valore minimo supporto (minsup>0 e minsup<=1):
s
Error reading float data, NaN value returned.
```

Guida utente - EpMiner Originale - Funzionamento Server

Avvio eseguito

- Una volta avviato il Server, verrà stampato a console il messaggio «Server Avviato», qualora la sua creazione fosse avvenuta senza alcun tipo di problema.
- Per ogni Client che instaura una connessione con tale server, verrà stampato a video la socket di connessione del client connesso (ossia il suo indirizzo Ip e porta)

```
Server avviato  
Connessione di Socket[addr=/127.0.0.1,port=56937,localport=8080]  
Nuovo client connesso
```

Guida utente - EpMiner Originale - Funzionamento Client

- Minimo Grow Rate (minGr che sia maggiore di 0) = Nel caso in cui l'utente dovesse inserire un parametro che non rientri in tale condizione, il messaggio che richiede l'inserimento di tale parametro verrà visualizzato nuovamente (stesso funzionamento di minSup).

```
Inserire valore minimo grow rate (minGr>0):  
-1  
Inserire valore minimo grow rate (minGr>0):  
-2  
Inserire valore minimo grow rate (minGr>0):  
-3  
Inserire valore minimo grow rate (minGr>0):
```

- Anche in questo caso, se l'utente dovesse inserire una stringa, verrà impostato per il minGr un valore di default pari a **Nan**.

```
Inserire valore minimo grow rate (minGr>0):  
s  
Error reading float data, NaN value returned.
```

Guida utente - EpMiner Originale - Funzionamento Client

- Nome della tabella di Target (**ATTENZIONE:** Nel Database sono presenti solo due dataset chiamati «playtennistarget» e «playtennisbackground»).
- Nome della tabella di Background:

```
1:Nuova scoperta
2: Risultati in archivio
1
Inserire valore minimo supporto (minsup>0 e minsup<=1):
0.3
Inserire valore minimo grow rate (minGr>0):
1
Tabella target:
playtennistarget
Tabella background:
playtennisbackground
```

Guida utente - EpMiner Originale - Funzionamento Client

Una volta inseriti tutti i dati corretti possiamo avere un risultato come nel seguente esempio:

Dati di input:

- minSup = 0.3
- minGr = 1
- targetTable = playtennistarget
- backgroundTable = playtennisbackground

Dati di Output:

- TargetTable
- BackgroundTable

```
Frequent patterns
1:(outlook=rain)[0.375]
2:(outlook=sunny)[0.375]
3:(temperature in [0.0,6.1205997])[0.375]
4:(temperature in [24.24,30.3606])[0.375]
5:(umidity=normal)[0.375]
6:(wind=strong)[0.375]
7:(outlook=sunny) AND (umidity=high)[0.375]
8:(outlook=sunny) AND (play=no)[0.375]
9:(temperature in [0.0,6.1205997]) AND (umidity=normal)[0.375]
10:(temperature in [24.24,30.3606]) AND (umidity=high)[0.375]
11:(umidity=high) AND (outlook=sunny)[0.375]
12:(umidity=high) AND (temperature in [24.24,30.3606])[0.375]
13:(umidity=high) AND (play=no)[0.375]
14:(umidity=normal) AND (temperature in [0.0,6.1205997])[0.375]
15:(wind=weak) AND (play=yes)[0.375]
16:(play=no) AND (outlook=sunny)[0.375]
17:(play=no) AND (umidity=high)[0.375]
18:(play=yes) AND (wind=weak)[0.375]
19:(outlook=sunny) AND (umidity=high) AND (play=no)[0.375]
20:(outlook=sunny) AND (play=no) AND (umidity=high)[0.375]
21:(umidity=high) AND (outlook=sunny) AND (play=no)[0.375]
22:(umidity=high) AND (play=no) AND (outlook=sunny)[0.375]
23:(play=no) AND (outlook=sunny) AND (umidity=high)[0.375]
24:(play=no) AND (umidity=high) AND (outlook=sunny)[0.375]
25:(play=no)[0.5]
26:(play=yes)[0.5]
27:(umidity=high) AND (wind=weak)[0.5]
28:(wind=weak) AND (umidity=high)[0.5]
29:(umidity=high)[0.625]
30:(wind=weak)[0.625]
```

```
Emerging patterns
1:(outlook=rain)[0.375][1.125]
2:(outlook=sunny)[0.375][1.125]
3:(wind=weak)[0.625][1.25]
4:(umidity=high)[0.625][1.875]
5:(temperature in [0.0,6.1205997])[0.375][2.25]
6:(temperature in [24.24,30.3606])[0.375][2.25]
7:(temperature in [0.0,6.1205997]) AND (umidity=normal)[0.375][2.25]
8:(umidity=high) AND (play=no)[0.375][2.25]
9:(umidity=normal) AND (temperature in [0.0,6.1205997])[0.375][2.25]
10:(play=no) AND (umidity=high)[0.375][2.25]
11:(play=no)[0.5][3.0]
12:(outlook=sunny) AND (umidity=high)[0.375][Infinity]
13:(outlook=sunny) AND (play=no)[0.375][Infinity]
14:(temperature in [24.24,30.3606]) AND (umidity=high)[0.375][Infinity]
15:(umidity=high) AND (outlook=sunny)[0.375][Infinity]
16:(umidity=high) AND (temperature in [24.24,30.3606])[0.375][Infinity]
17:(play=no) AND (outlook=sunny)[0.375][Infinity]
18:(outlook=sunny) AND (umidity=high) AND (play=no)[0.375][Infinity]
19:(outlook=sunny) AND (play=no) AND (umidity=high)[0.375][Infinity]
20:(umidity=high) AND (outlook=sunny) AND (play=no)[0.375][Infinity]
21:(umidity=high) AND (play=no) AND (outlook=sunny)[0.375][Infinity]
22:(play=no) AND (outlook=sunny) AND (umidity=high)[0.375][Infinity]
23:(play=no) AND (umidity=high) AND (outlook=sunny)[0.375][Infinity]
24:(umidity=high) AND (wind=weak)[0.5][Infinity]
25:(wind=weak) AND (umidity=high)[0.5][Infinity]
```

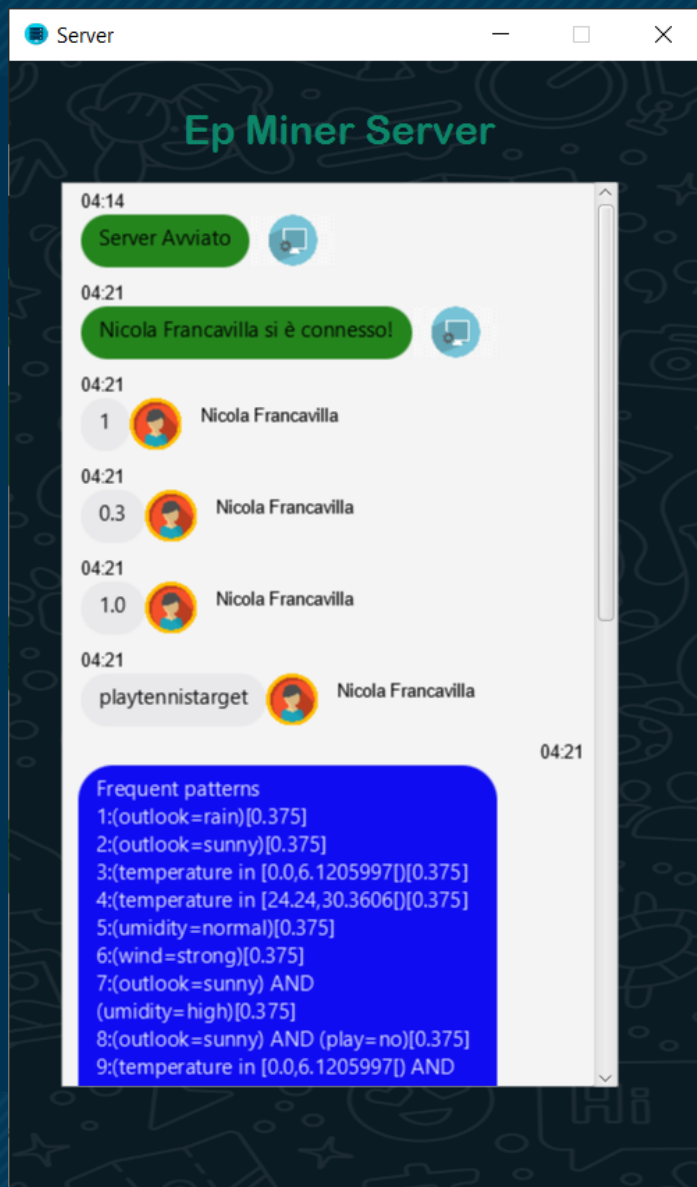
Vuoi ripetere?(s/n)




Alla fine della computazione scrivendo «s» è possibile fare un'altra richiesta al server mentre in caso negativo il Client si chiude

Guida utente

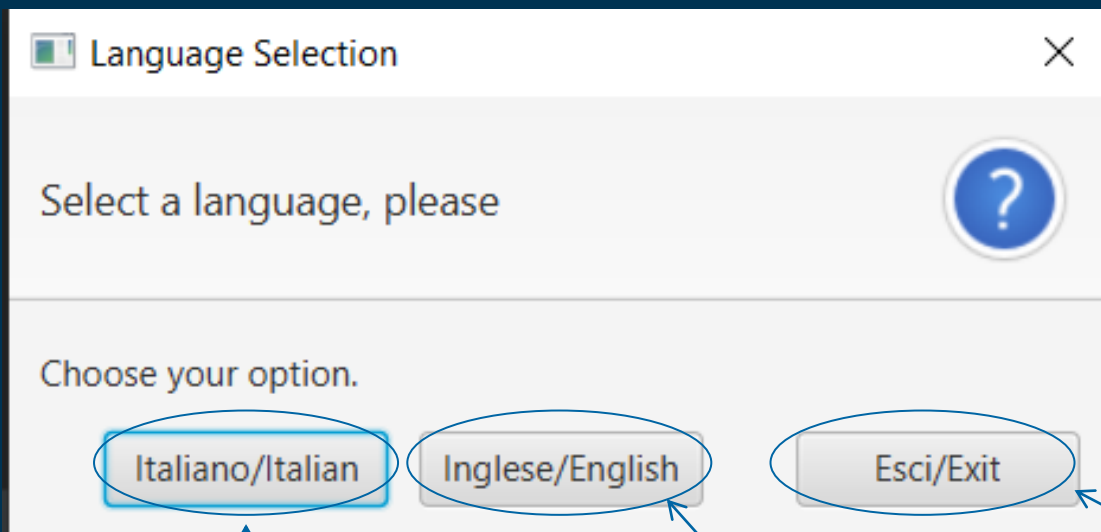
Progetto EpMiner Estensione

Guida utente - EpMiner Estensione - Server



- Una volta eseguito il file .jar (folder «EpMinerApp/EpMinerAppServer/bin») l'avvio del server si presenta nel seguente modo.
- L'applicazione (sia Client che Server) ha 3 tipi di messaggi:
 -  Messaggi di Sistema = errore o azione terminata con successo
 -  Messaggi che il Server invia al Client
 -  Messaggi che il Client invia al Server
 - Specificando
 - Username dell'utente connesso e che fa l'attuale richiesta
 - Orario del ricevimento e invio dei messaggi

Guida utente - EpMiner Estensione - Client



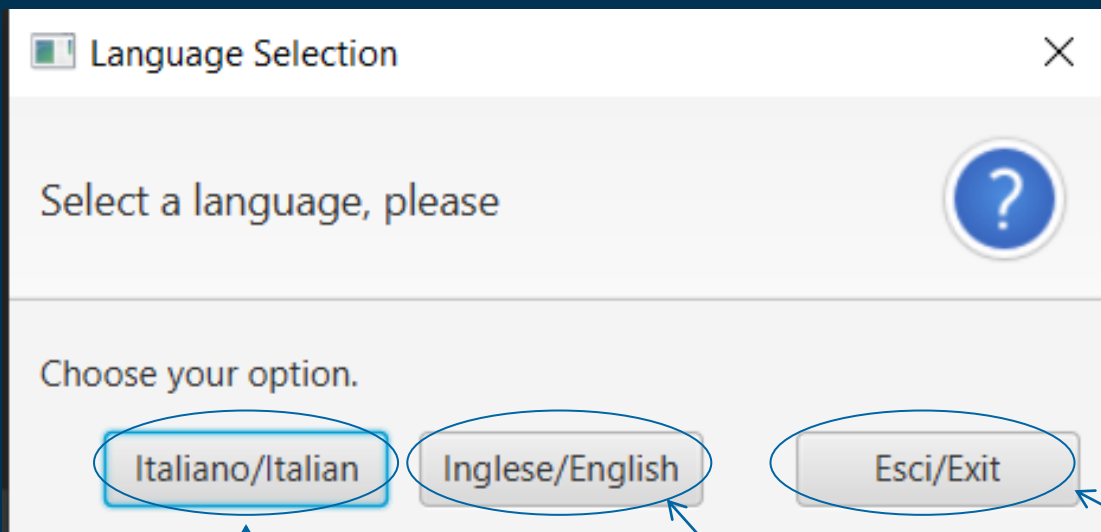
- Una volta eseguito il file .jar (folder «EpMinerApp/EpMinerAppClient/bin») viene visualizzata una finestra di dialogo che presenta 3 tipi di pulsanti..

Per impostare le Label e pulsanti (della finestra di dialogo di login) in lingua italiana

Per impostare le Label e pulsanti (della finestra di dialogo di login) in lingua Inglese

Chiude il client

Guida utente - EpMiner Estensione - Client



Per impostare le Label e pulsanti (della finestra di dialogo di login) in lingua italiana

Per impostare le Label e pulsanti (della finestra di dialogo di login) in lingua Inglese

Chiude il client

- Una volta eseguito il file .jar (folder «EpMinerApp/EpMinerAppClient/bin») viene visualizzata una finestra di dialogo che presenta 3 tipi di pulsanti..

Guida utente - EpMiner Estensione - Client

- Dopo che l'utente ha scelto una lingua, l'applicazione mostra la seguente finestra di login:
- Utilizzando dei fogli di stile, l'utente può chiaramente vedere quali textField siano obbligatori (username e indirizzo Ip per abilitare il button del login).
- L'applicazione accetta una qualsiasi password

L'utente può caricare un file (jpgge/png)
come immagine del profilo

L'utente può selezionare una
delle 3 foto già preimpostate.

Guida utente - EpMiner Estensione - Client

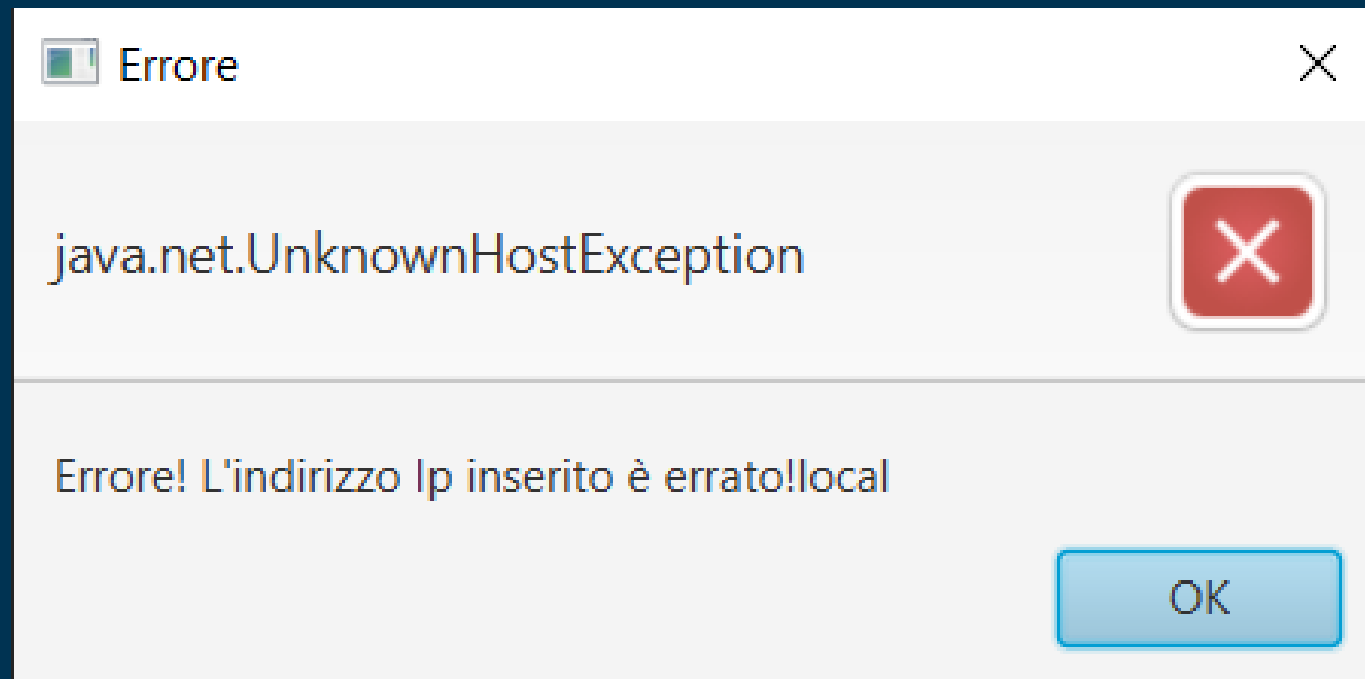
- L'applicazione aggiorna dinamicamente l'immagine del profilo dell'utente.
- Qualora l'utente inserisse un username e indirizzo Ip, l'interfaccia mostra la convalida dei rispettivi textField (Fogli di stile Css).

The screenshot shows the 'EPMIner Client Login' window. The title bar includes a user icon and the text 'EPMIner Client Login' with a close button. The main content area has a welcome message: 'Benvenuto! Inserisci i tuoi dati di accesso, Indirizzo Ip e Porta del server!'. Below this are four input fields: 'Username:' with the value 'Nicola Francavilla', 'Password:' with masked characters '•••••', 'Indirizzo IP:' with the value '127.0.0.1', and 'Porta:' with a dropdown menu showing '8080'. To the right of these fields is a circular profile picture placeholder showing a woman with blonde hair. Below the 'Porta:' field is a 'Cambia Immagine' button. A dropdown menu is open below this button, titled 'Seleziona immagine profilo', showing three options: 'User image1' (with a user icon), 'User image2' (with a user icon), and 'User image3' (with a generic user icon). At the bottom right are 'Login' and 'Esci' buttons.

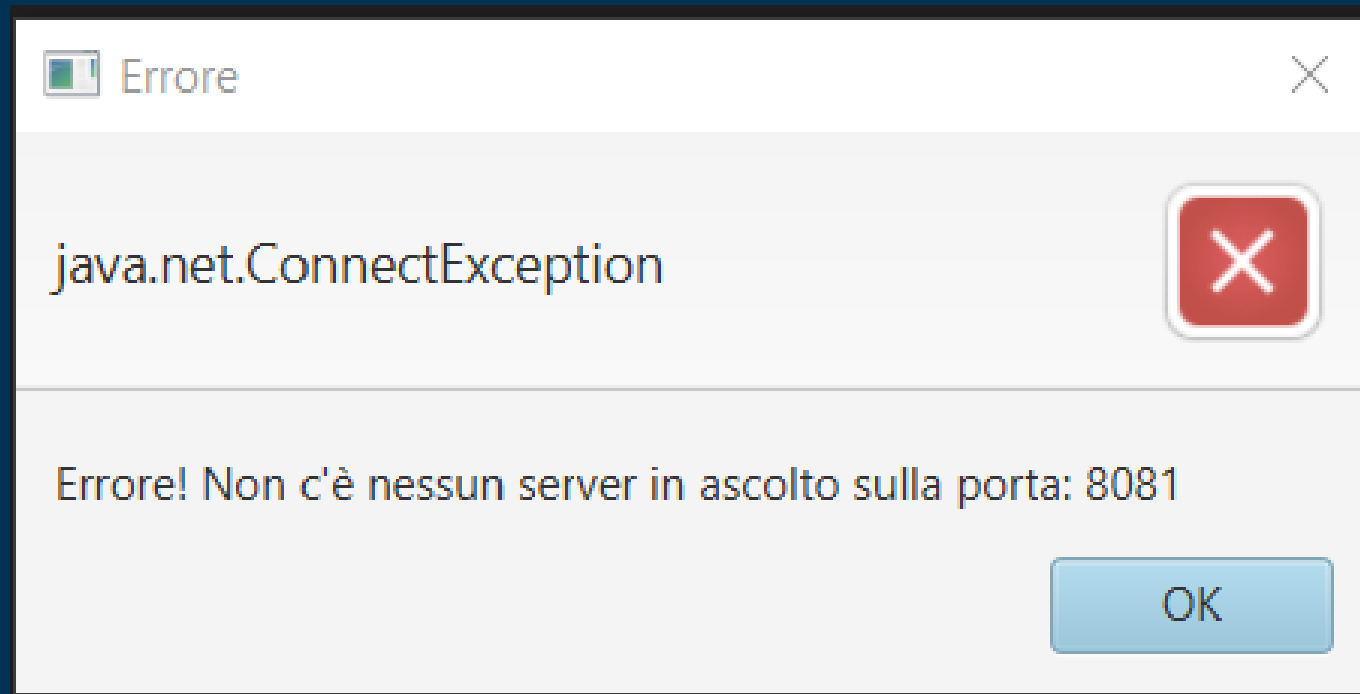
Esempi di Test

Progetto EpMiner Estensione

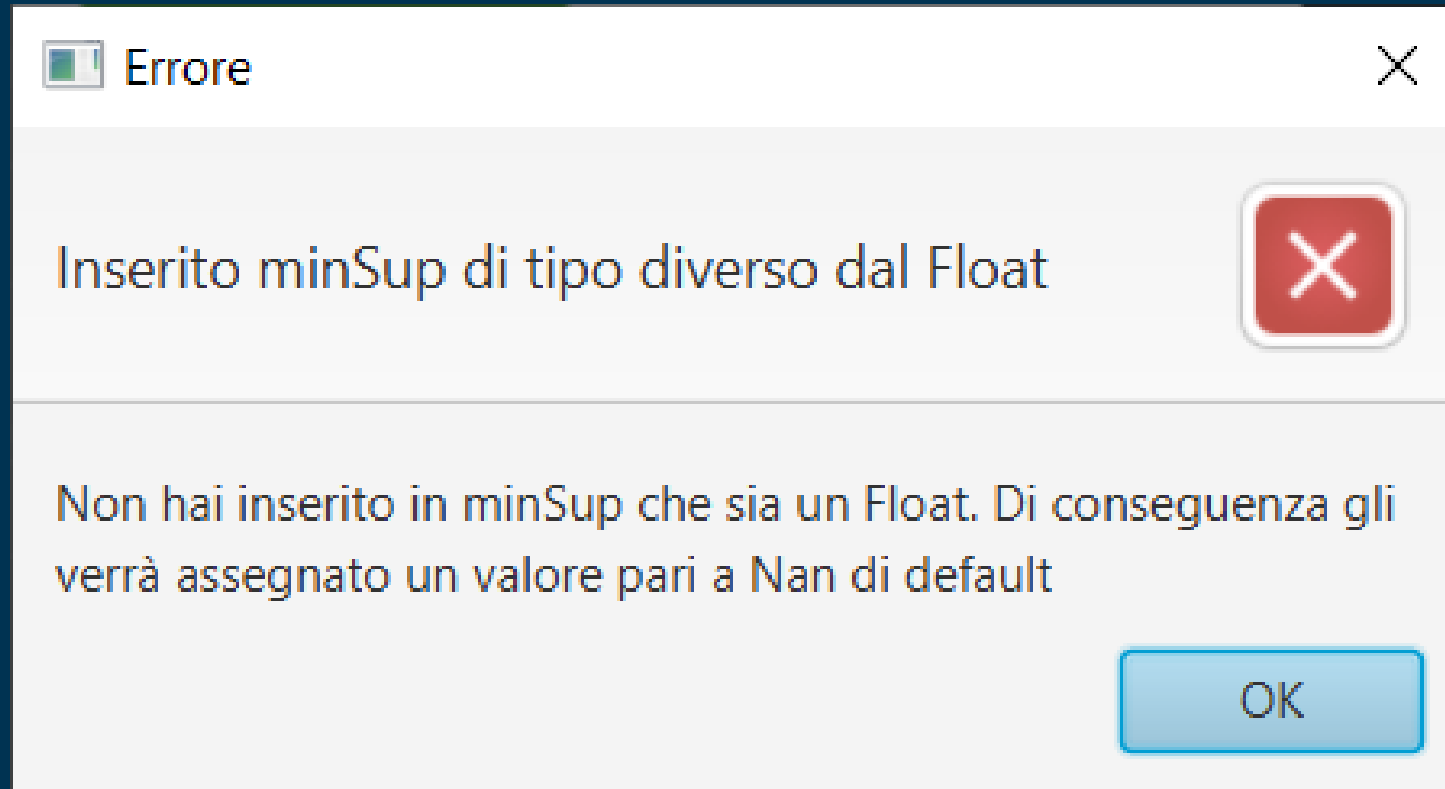
Esempi di Test - Indirizzo Ip Errato



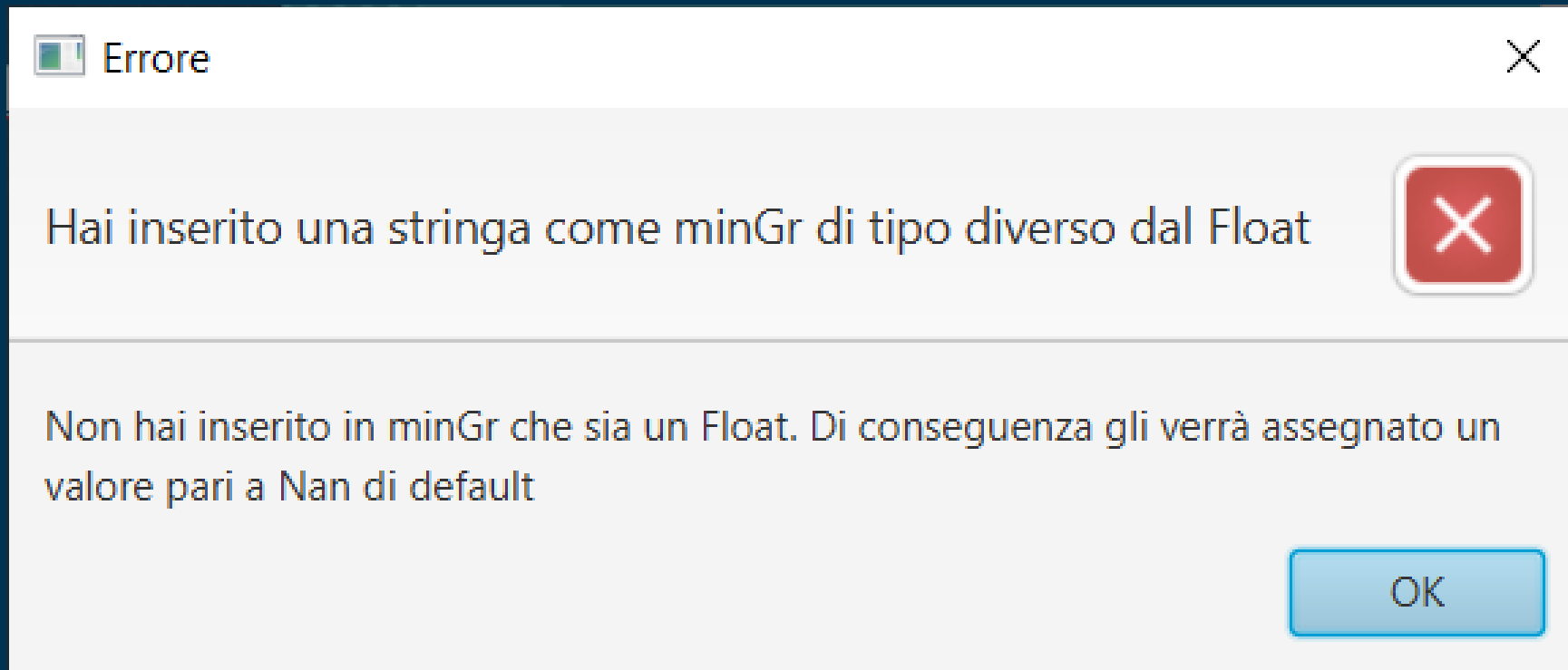
Esempi di Test - Porta inserita errata / Nessun server in ascolto



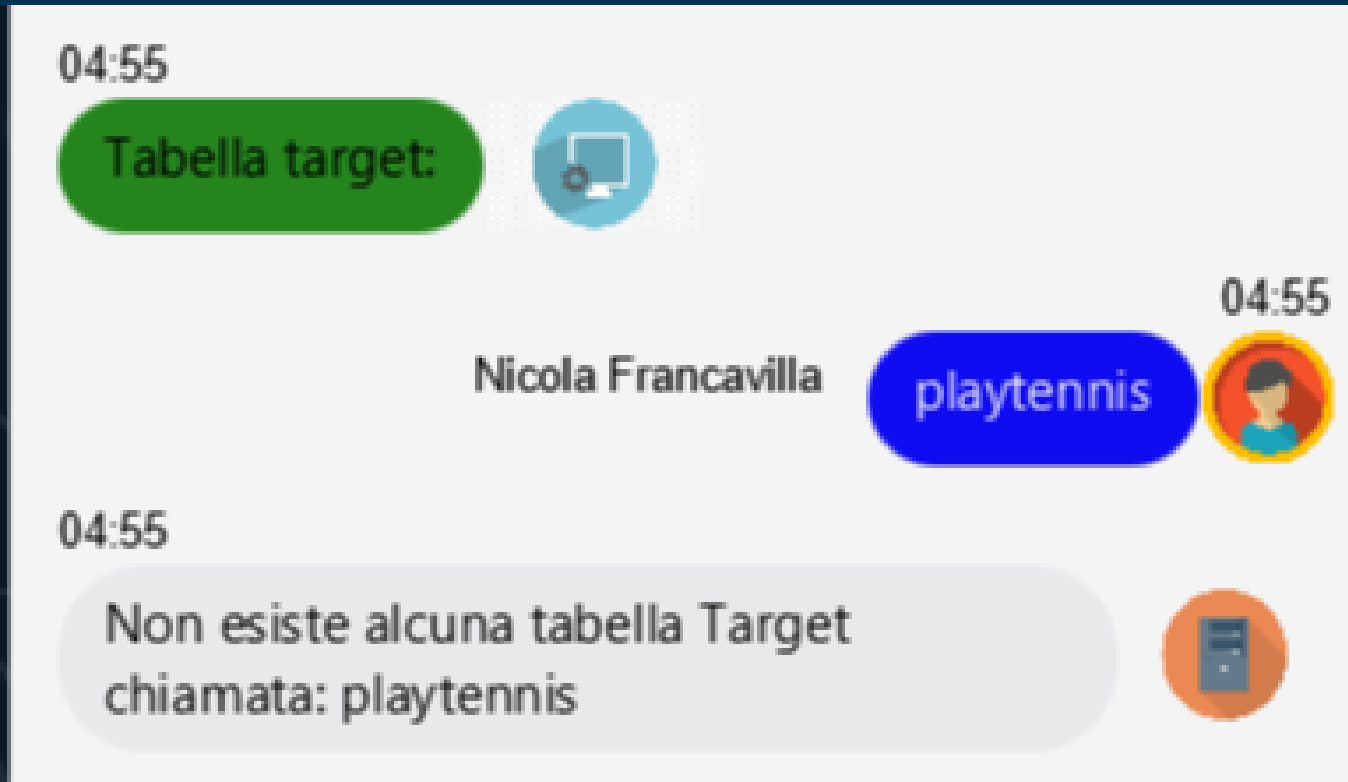
Esempi di Test - Inserimento minSup errato



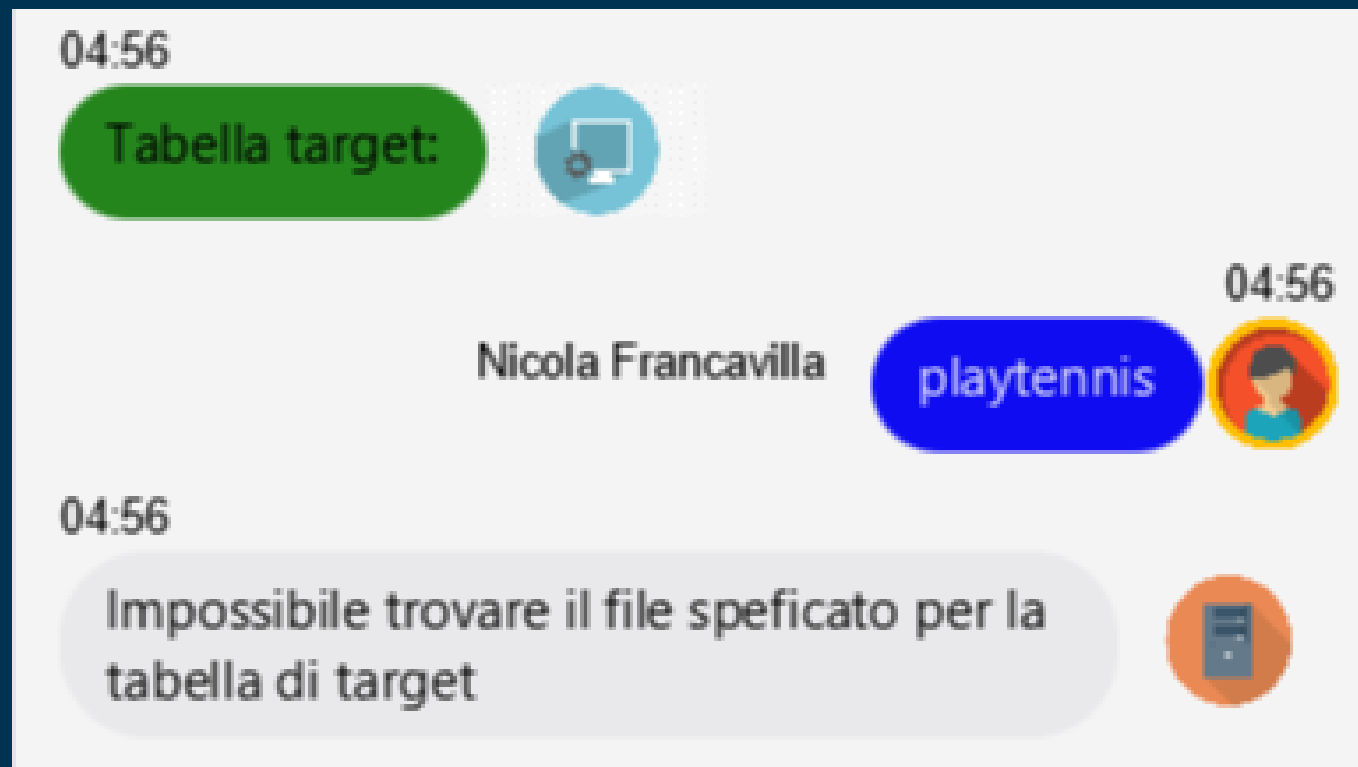
Esempi di Test - Inserimento minGr errato



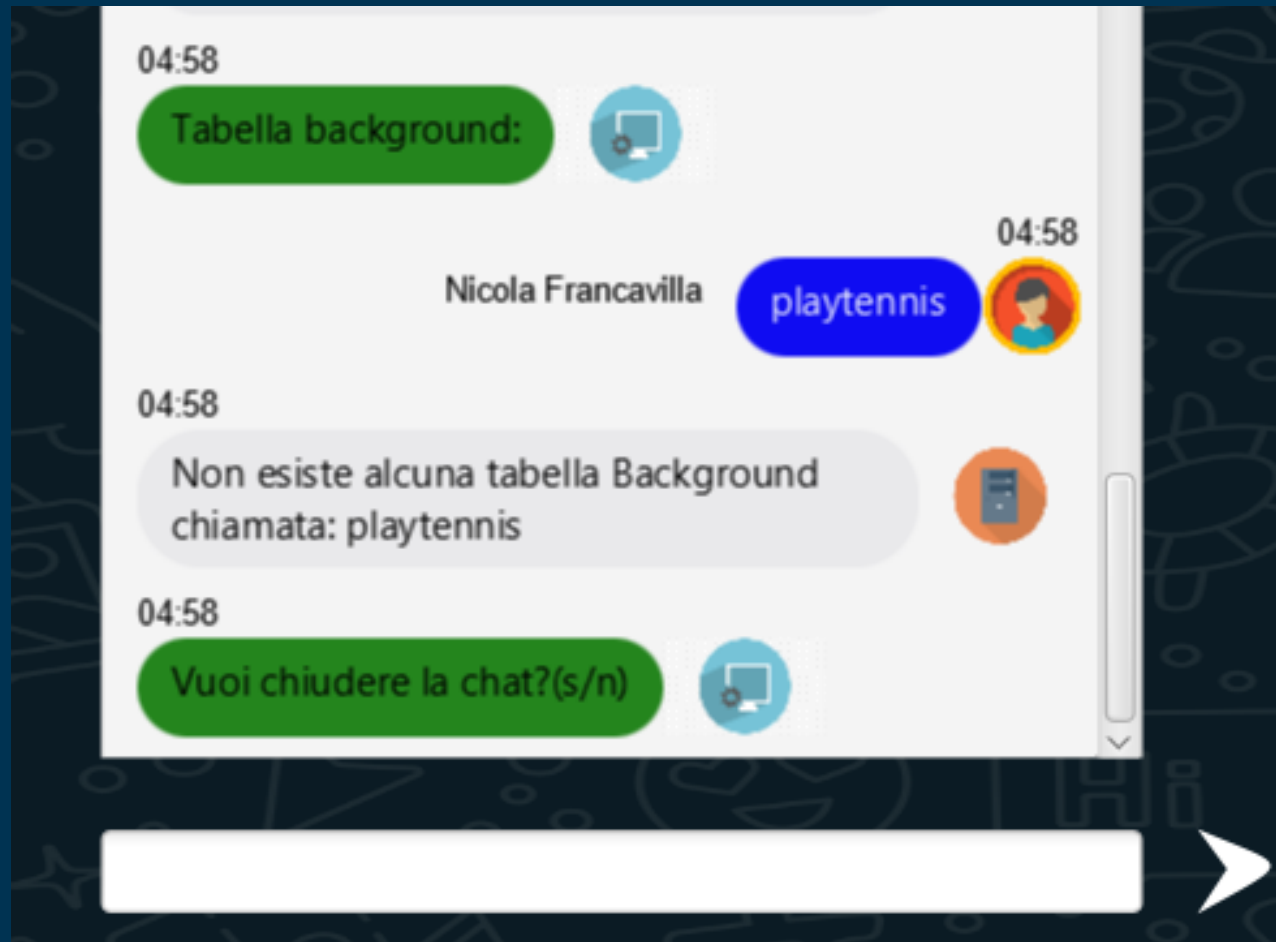
Esempi di Test - nome tabella Target errato (opzione 1)



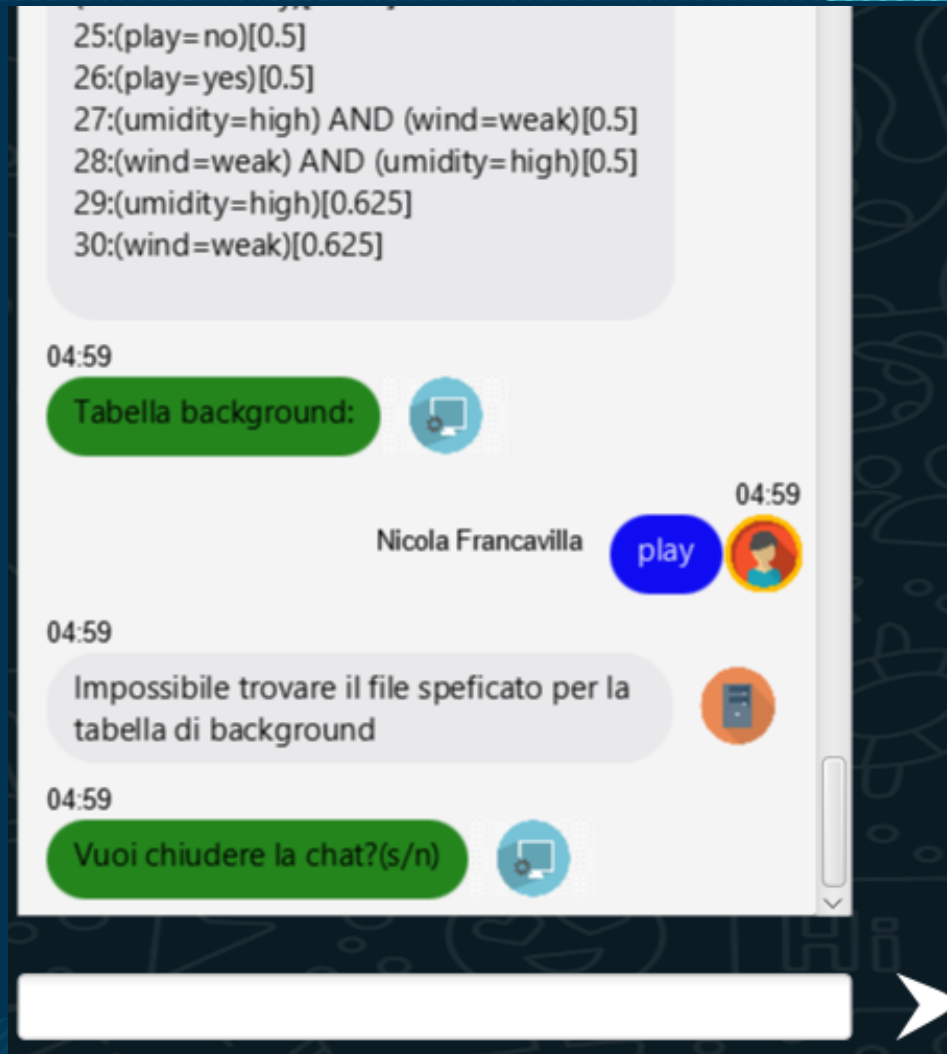
Esempi di Test - nome tabella Target errato (opzione 2)



Esempi di Test - nome tabella Background errato (opzione 1)



Esempi di Test - nome tabella Background errato (opzione 2)



Esempi di Test

- 1) Indirizzo Ip errato
- 2) Porta Inserita errata / Nessun Server in ascolto
- 3) Inserire un MinSup errato (stringa)
- 4) Inserire un MinGr errato (stringa)
- 5) Nome targetTable errato (opzione 1/2)
- 6) Nome backgroundTable errato (opzione 1/2)



**Fine
presentazione
Grazie**