

Sommario

INTRODUZIONE	1
REQUISITI	1
INSTALLAZIONE	2
<i>Installare Java 17.0.2.....</i>	<i>2</i>
<i>Installare IntelliJ o Eclipse</i>	<i>2</i>
<i>Installare MySql 8.0.17.0.....</i>	<i>2</i>
<i>Caricare il database.....</i>	<i>2</i>
<i>Impostare le dipendenze sull'ide per le librerie su IntelliJ.....</i>	<i>3</i>
<i>Impostare le dipendenze sull'ide per le librerie su Eclipse.....</i>	<i>5</i>
GUIDA ALL'UTILIZZO DEL PROGRAMMA.....	7
<i>Modalità di avvio del client e del server</i>	<i>7</i>
<i>Specificare un server personalizzato per la connessione.....</i>	<i>7</i>
<i>Funzionamento programma.....</i>	<i>8</i>
<i>Esempi utilizzo programma</i>	<i>9</i>

INTRODUZIONE

Il progetto è stato realizzato da:

- Domenico Mattia Latrofa (Matricola 703009)
- Natasha Fabrizio (Matricola 717446)
- Francesco Saverio Cassano (Matricola 716133)

Per l'anno accademico 2020/2021.

Il progetto, denominato "**EPMiner**" consiste nella realizzazione di un sistema client-server per il data mining. Il **client** consente di usufruire del servizio di scoperta remoto e visualizza la conoscenza (pattern) scoperta. Il **server** include funzionalità di data mining per la scoperta di pattern frequenti e pattern emergenti. Il progetto segue le indicazioni descritte dalla professoressa con la sola aggiunta, di stampe nel server, riguardanti le attività del client. L'UML, il JavaDoc, i file jar ed il sorgente del progetto, sono presenti nelle loro corrispettive cartelle.

REQUISITI

Il progetto è stato realizzato con le seguenti dipendenze:

- [Java 17.0.2 \(LTS\)](#)
- [MySql 8.0.17.0](#)
- Java Connector (Presente nella stessa cartella)
- Script Database MySql (Presente nella stessa cartella)
- Un IDE per lo sviluppo come [IntelliJ IDEA](#) o [Eclipse](#).

INSTALLAZIONE

Il programma è stato realizzato e testato con il sistema operativo Windows 10 64 bit. La guida per l'installazione è basata su quest'ultimo.

Installare Java 17.0.2

- Bisogna scaricare l'installer (il link è presente nella sezione "**Requisiti**").
- L'installazione è guidata e spiegata nel relativo installer.
- [Video guidato](#).

Installare IntelliJ o Eclipse

- Bisogna scaricare l'installer (il link è presente nella sezione "**Requisiti**").
- L'installazione è guidata e spiegata nel relativo installer.

Installare MySql 8.0.17.0

- Bisogna scaricare l'installer (il link è presente nella sezione "**Requisiti**").
- L'installazione è guidata e spiegata nel relativo installer.
- Durante il processo di installazione, si può optare per installare solo i componenti realmente necessari per il caricamento e l'utilizzo del database. (come il router, la shell ed il server).

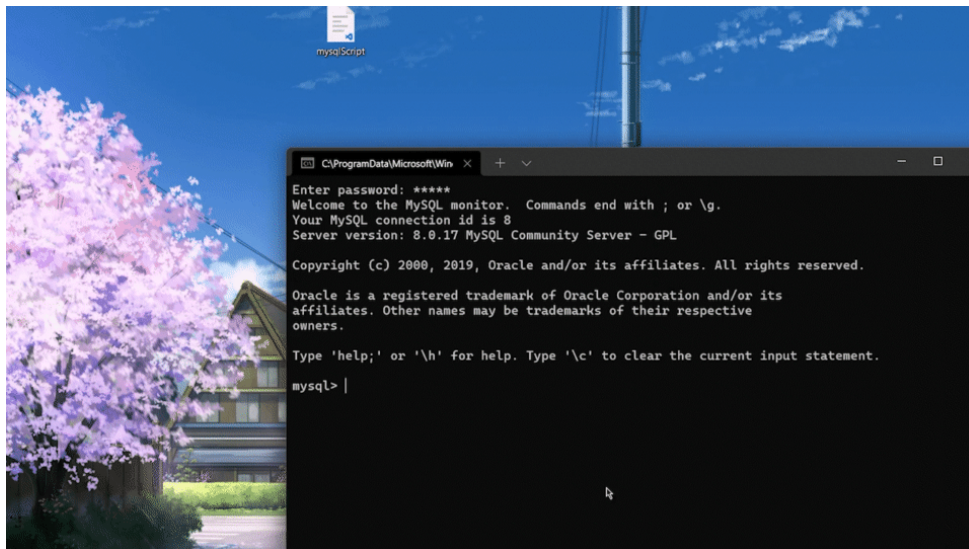
Caricare il database

Caricare il database con il comando "*source mysqlScript.sql*" dalla **shell mysql**.

- Basterà aprire la shell di **MySQL** ed inserire la password (Per aprire la shell, cercare sulla barra di ricerca di Windows "MySQL 8.0 Command Line Client" ed avviarlo).



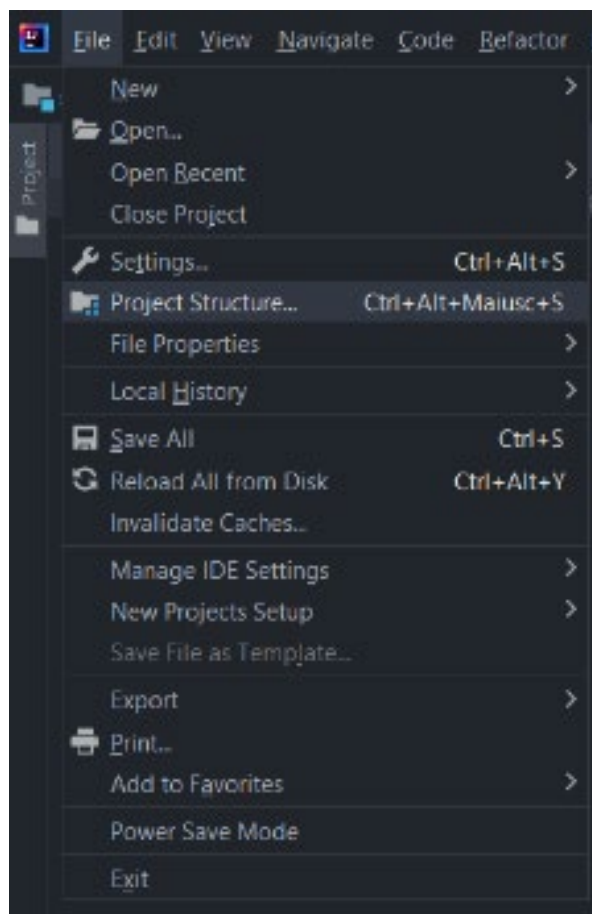
- Una volta inserita la password bisogna utilizzare il comando "*source*" per caricare il database.
- Digitare il comando "*source*" sulla shell, inserendo uno spazio e trascinando il file "*mysqlScript.sql*" all'interno della shell ed infine premere invio (il file sql è presente nella cartella "*Parte_Base*").

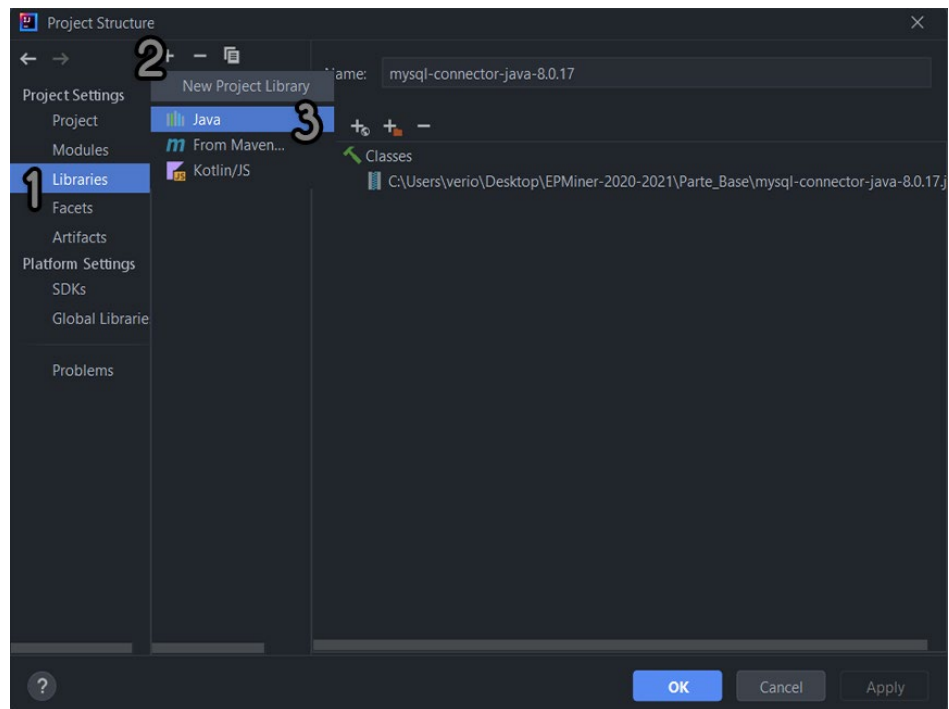


- È possibile visualizzare la gif animata del processo appena descritto, la gif si trova nella stessa cartella con il nome "SQL2.gif".
- In caso di errori nel caricamento del database:
 - Spostare il file *mysqlScript.sql* all'interno della cartella dove è presente la shell di **MySQL**.
 - Effettuare il comando "*source mysqlScript.sql*".

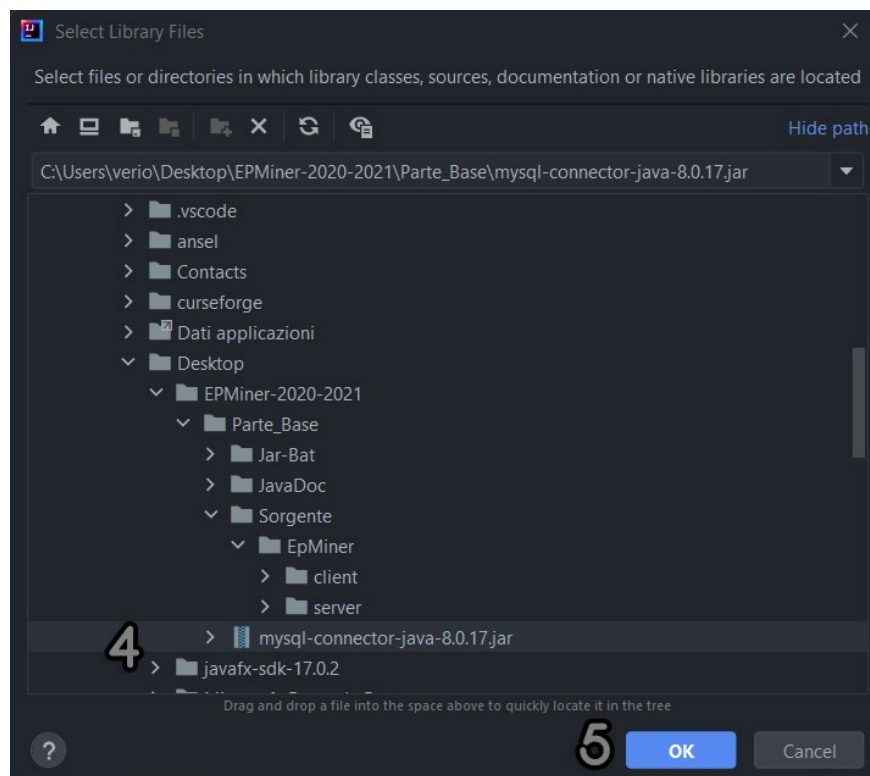
Impostare le dipendenze sull'ide per le librerie su IntelliJ

Assicurarsi che le dipendenze del progetto siano attive, in particolare quelle del server. Per far ciò aprire il progetto su **IntelliJ**, cliccare sul menu in alto a sinistra la voce "File" e poi "Project Structure".

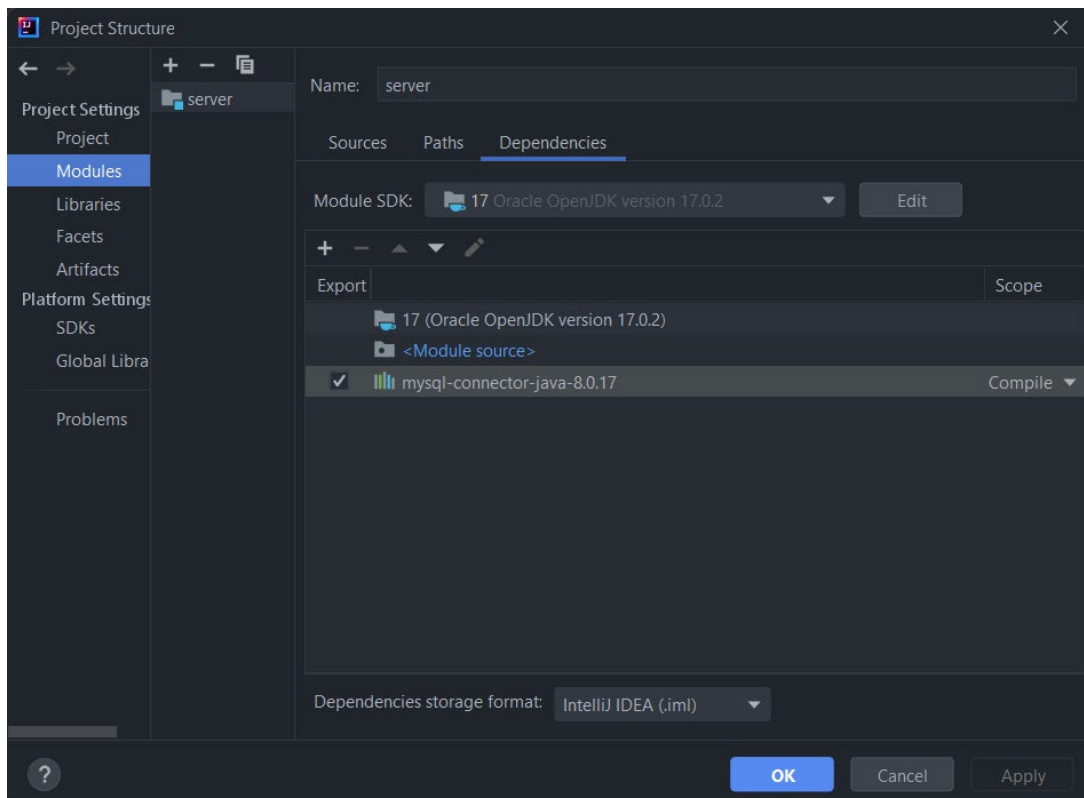




1. Cliccare su "Libraries". Controllare che il file jar di sql sia selezionato correttamente. In caso contrario, leggere i punti successivi.
2. Cliccare sul "+".
3. Cliccare su "Java".

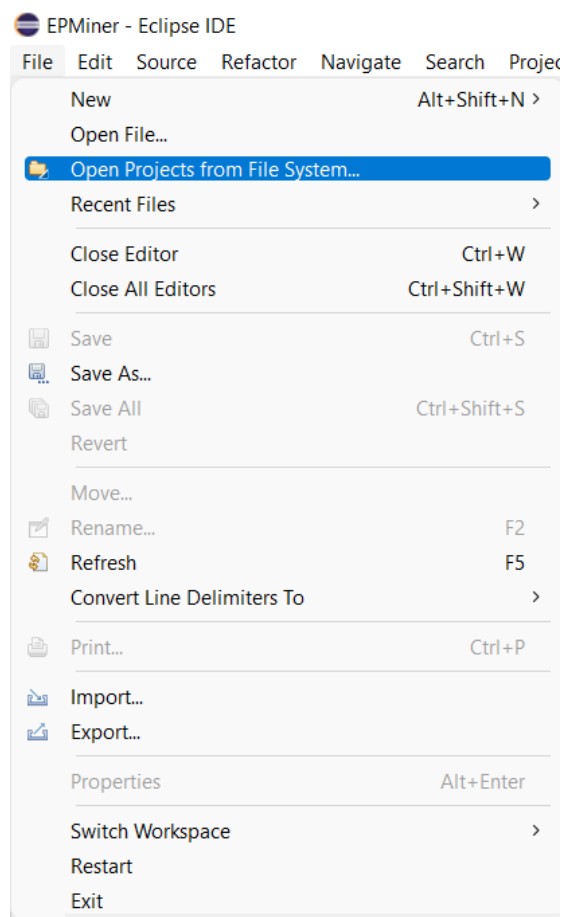


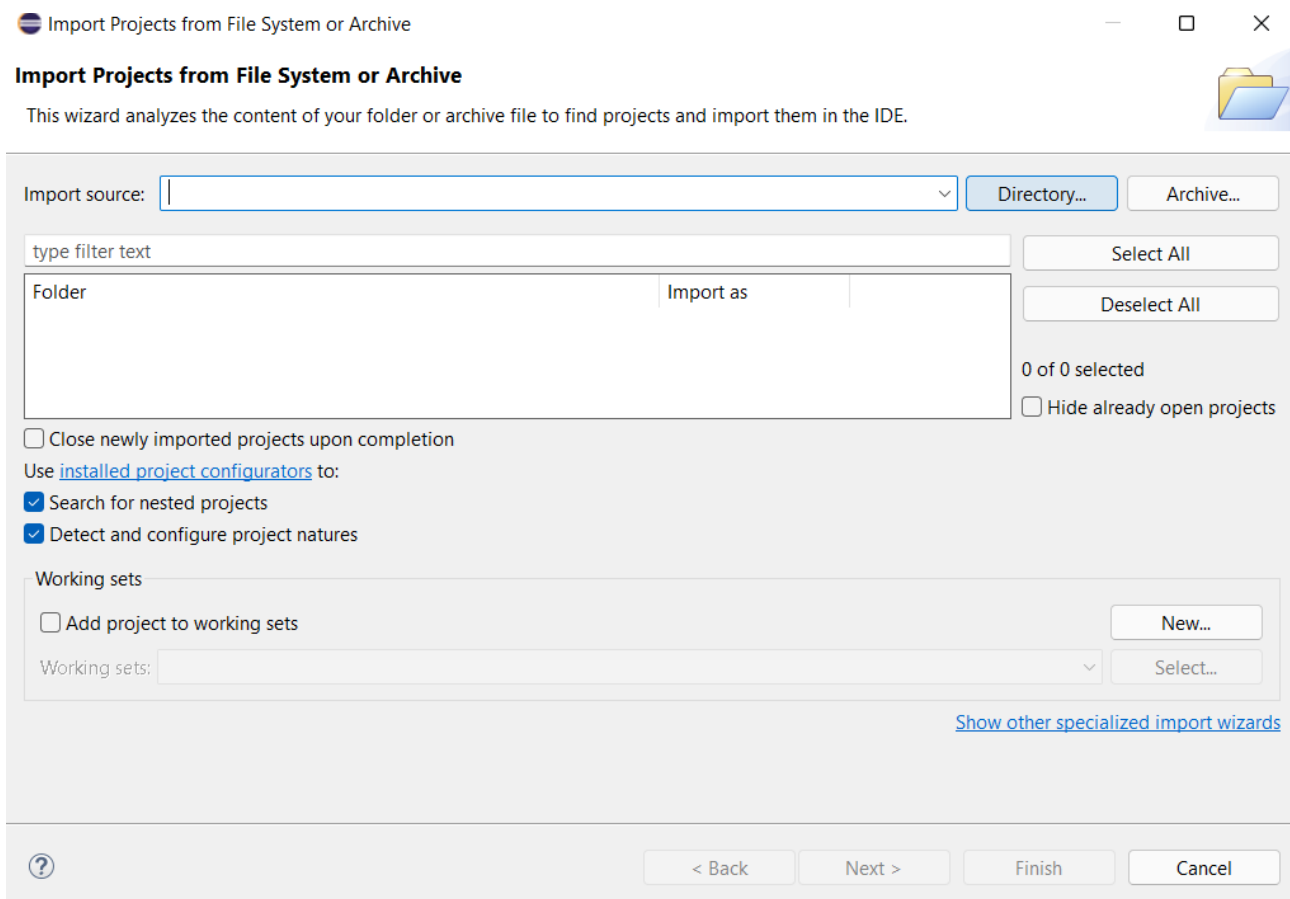
4. Selezionare il file jar *"mysql-connector-java-8.0.17.jar"*, sempre presente nella cartella del progetto denominata *"Parte_Base"*.
5. Cliccare su ok e poi su applica.
6. Controllare nella sezione *Modules* che la spunta sul jar è presente, in caso contrario, inserirla.



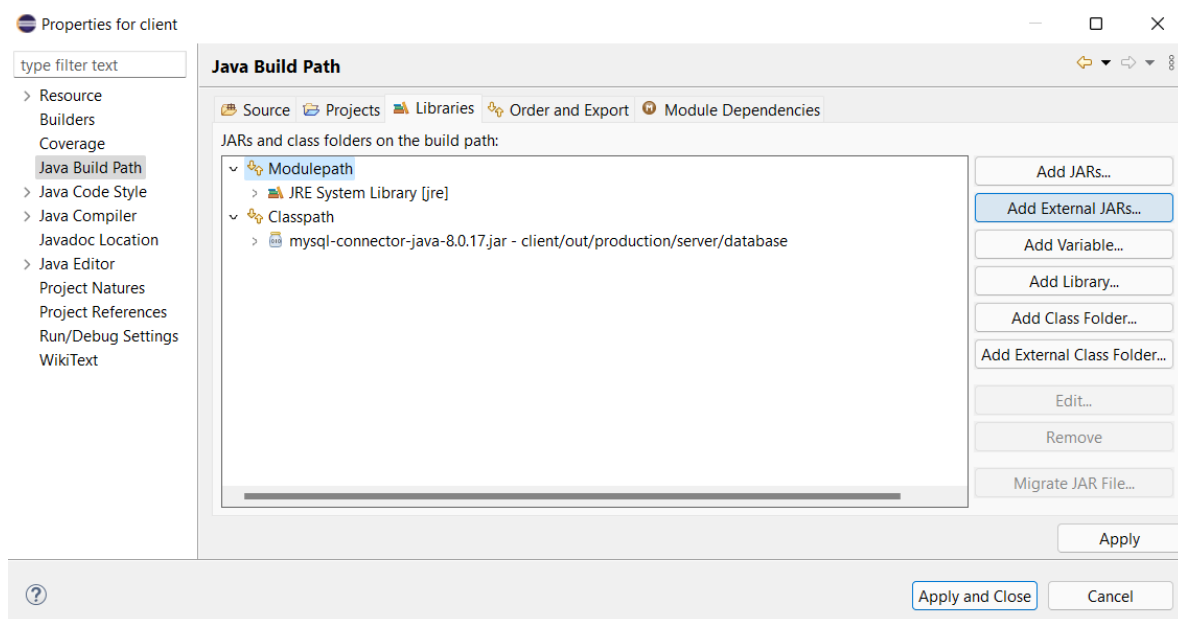
Impostare le dipendenze sull'ide per le librerie su Eclipse

Per prima cosa aprire **Eclipse**. Successivamente, cliccare sulla voce del menù *“File”* e poi su *“Open Projects from File System”*.





Selezionare la voce “Directory” e poi selezionare la cartella “server” che è ubicata nel percorso “EPMIner-2020-2021\Parte_Base\Sorgente\EPMIner”. Stesso identico discorso anche per il “client” (presente sempre nella stessa cartella). Una volta aperto il client, bisogna inserire le librerie per il collegamento con il database. Per far ciò, fare tasto destro sul progetto “client” da Eclipse e cliccare su “Properties”.



Selezionare la voce “Java Build Path”, successivamente cliccare su “Modulepath” e poi sul bottone sulla destra “Add External JARs”. Selezionare il file jar “mysql-connector-java-8.0.17”, sempre presente nella cartella del progetto “EPMIner-2020-2021\Parte_Base”. Una volta selezionato cliccare su “Apply and Close”.

GUIDA ALL'UTILIZZO DEL PROGRAMMA

Modalità di avvio del client e del server

Per prima cosa bisogna avviare il *server* e successivamente il *client*. È possibile avviare il client ed il server in 3 differenti modi:

1. Direttamente **dall'IDE**.
2. Avviando il file **server.bat** e **client.bat** (aprendo il file *server_client.bat*, si apriranno in automatico sia server che client).
3. Avviando i due file jar da linea di comando.

Per avviare il **client** o il **server** tramite linea di comando, bisogna aprire il **Prompt dei Comandi** ed assicurarsi che il percorso della shell sia nella stessa cartella dove abbiamo posizionati i file da eseguire. È possibile navigare nelle directory della shell utilizzando:

- **cd nome_cartella** per entrare dentro una cartella.
- **cd ..** per uscire da una cartella.

Una volta posizionati nella cartella corretta, bisogna utilizzare il seguente comando per avviare il client:

```
java -jar client.jar
```

per avviare il server:

```
java -jar server.jar
```

In caso di errori nell'esecuzione, assicurarsi che tra le variabili d'ambiente di Windows, sia presente java.

Per avviare il programma dall'IDE, basterà aprire il progetto del client e del server e cliccare sul tastino play verde (in caso di problemi con l'esecuzione del codice nell'IDE, visionare la sezione "*Impostare le dipendenze sull'ide per le librerie*").

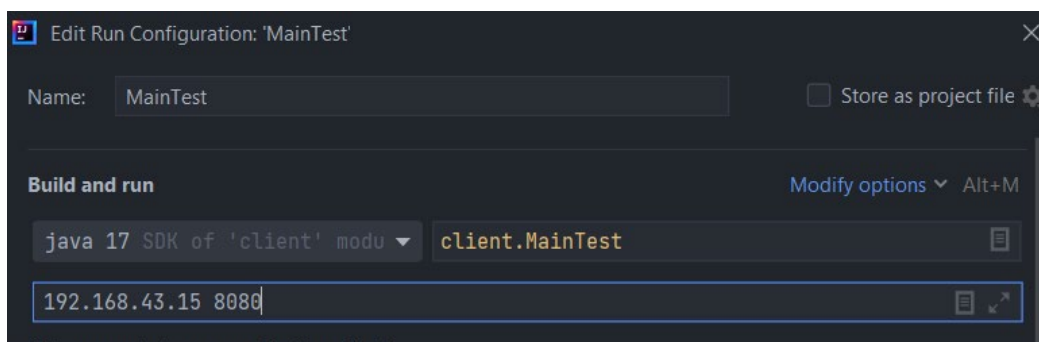
Specificare un server personalizzato per la connessione

Il client se avviato senza parametri aggiuntivi, proverà a collegarsi ad un server hostato in localhost (quindi 127.0.0.1) sulla porta 8080. Se si vorrà cambiare il server a cui collegarsi, basterà aggiungere come argomento per l'avvio **indirizzo ip** del server e la **porta**. Esempio:

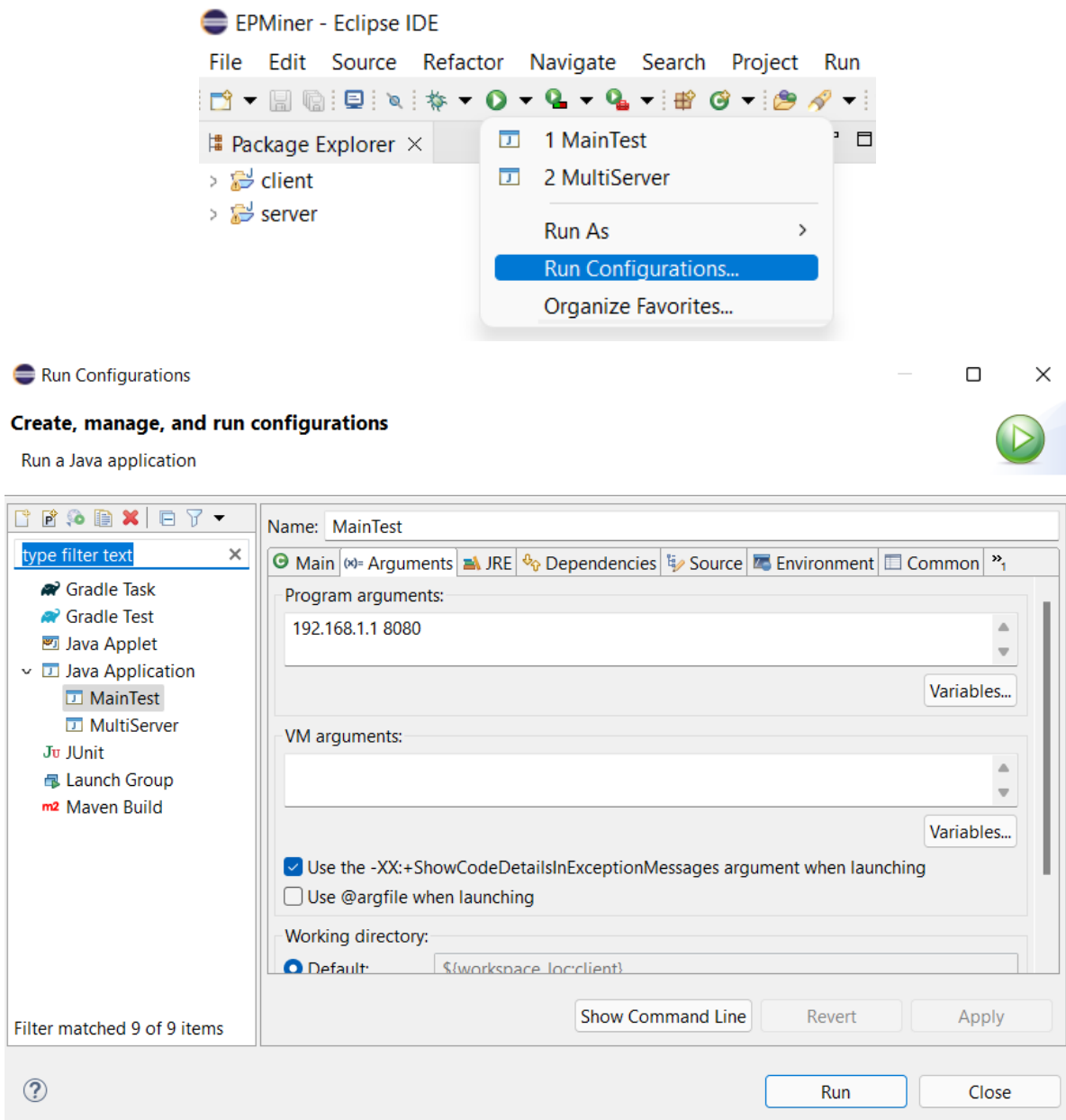
```
java -jar client.jar 192.168.43.15 8080
```

È possibile modificare il file "*client.bat*", per personalizzare il server di connessione, adoperando lo stesso funzionamento appena descritto. Per modificare un file bat, bisogna fare tasto destro sul file e cliccare sulla voce "*Modifica*" del menu contestuale e poi modificare il testo scritto aggiungendo l'indirizzo IP e la porta (separate da spazi).

Per modificare gli argomenti dall'IDE **IntelliJ**, basterà cliccare sull'icona verde del play e cliccare sulla voce "*Modify Run Configuration*" e inserire indirizzo IP e la porta del server (come mostrato nell'esempio qui sotto) e poi cliccare su ok per la conferma.



Per modificare il server di collegamento da **Eclipse**, ci basterà cliccare sull'icona del play verde e poi successivamente cliccare su **"Run Configurations"** e poi nella scheda **"Arguments"** inseriamo l'indirizzo IP e poi la porta, clicchiamo su **"Apply"** o **"Run"** per avviare subito.



Funzionamento programma

Nell'interfaccia del client, possiamo effettuare 2 operazioni:

1. Nuova scoperta.
2. Risultati archivio.

Nuova scoperta permette di effettuare una nuova scoperta analizzando i dati presenti nel database, per trovare Pattern Frequenti e Pattern Emergenti.

Risultati archivio permette di visionare i dati presenti nell'archivio (ovvero le ricerche effettuate in precedenza).

Per selezionare una delle due possibili scelte, basterà inserire nella console "1" per la prima scelta, "2" per la seconda e premere invio.

Dopo la scelta, dovremo inserire:

- Un valore minimo supporto (minsup>0 e minsup<=1).
- Un valore minimo grow rate (minGr>0).
- La Tabella target
- La Tabella background

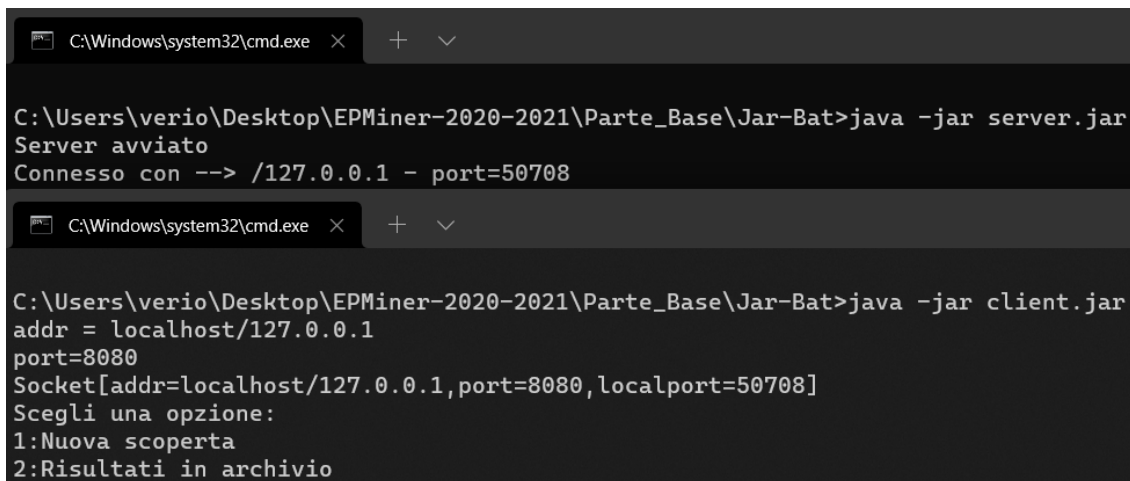
Se i valori inseriti saranno corretti e/o presenti, potremo visualizzare i dati estratti selezionati nel terminale, altrimenti verrà stampato un messaggio di errore. Il server tiene traccia di tutte le operazioni effettuate dal client, ovvero i valori inseriti e stamperà a video eventuali errori di ricerca in caso di dati incompleti o mancanti.

Una volta stampati i risultati, ci ritroveremo davanti ad una scelta:

- Premendo s, possiamo ri-effettuare la ricerca.
- Premendo n, chiuderemo il programma.

Esempi utilizzo programma

Qui sotto sono riportati alcuni esempi pratici dell'esecuzione del programma.



```
C:\Windows\system32\cmd.exe x + v
C:\Users\verio\Desktop\EPMiner-2020-2021\Parte_Base\Jar-Bat>java -jar server.jar
Server avviato
Connesso con --> /127.0.0.1 - port=50708

C:\Windows\system32\cmd.exe x + v
C:\Users\verio\Desktop\EPMiner-2020-2021\Parte_Base\Jar-Bat>java -jar client.jar
addr = localhost/127.0.0.1
port=8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=50708]
Scegli una opzione:
1:Nuova scoperta
2:Risultati in archivio
```

Esempio di Server e Client appena avviati.

```
Server avviato
Connesso con --> /127.0.0.1 - port=59598
Connessione interrotta con il client: Socket[addr=/127.0.0.1,port=59598,localport=8080]
Chiusura socket.
```

Esempio lato server di una connessione e disconnessione di un client.

```
Scegli una opzione:
1:Nuova scoperta
2:Risultati in archivio
1
Inserire valore minimo supporto (minsup>0 e minsup<=1):
1
Inserire valore minimo grow rate (minGr>0):
1
Tabella target:
1
Tabella background:
1
Errore
Errore
Vuoi ripetere?(s/n)
```

Esempio di errore nella ricerca nel client.

```

addr = localhost/127.0.0.1
port=8080
Socket[addr=localhost/127.0.0.1,port=8080,localport=50944]
Scegli una opzione:
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
Frequent patterns
1:(outlook=rain)[0.375]
2:(outlook=sunny)[0.375]
3:(temperature in [0.0,6.1812()][0.375]
4:(temperature in [24.7248,30.906()][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 [0.0,6.1812() AND (umidity=normal)[0.375]
10:(temperature in [24.7248,30.906() AND (umidity=high)[0.375]
11:(umidity=high) AND (outlook=sunny)[0.375]
12:(umidity=high) AND (temperature in [24.7248,30.906()][0.375]

```

Inserimento dei dati per la ricerca nel client.

```

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 [0.0,6.1812()][0.375][2.25]
6:(temperature in [24.7248,30.906()][0.375][2.25]
7:(temperature in [0.0,6.1812() AND (umidity=normal)[0.375][2.25]
8:(umidity=high) AND (play=no)[0.375][2.25]
9:(umidity=normal) AND (temperature in [0.0,6.1812()][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 [24.7248,30.906() AND (umidity=high)[0.375][Infinity]
15:(umidity=high) AND (outlook=sunny)[0.375][Infinity]
16:(umidity=high) AND (temperature in [24.7248,30.906()][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)

```

Esempio di richiesta di ripetizione dopo aver terminato di visionare i risultati.

```
Server avviato
Connesso con --> /127.0.0.1 - port=50944
Il client ha inserito i seguenti dati:
MinSup: 0.3 MinGr: 1.0 TargetName: playtennistarget TargetBackground: playtennisbackground
```

Esempio log shell server.

```
Il client ha inserito i seguenti dati:
MinSup: 1.0 MinGr: 1.0 TargetName: 1 TargetBackground: 1
Errore ricerca dati.
Il client ha inserito i seguenti dati:
MinSup: 0.3 MinGr: 1.0 TargetName: playtennis TargetBackground: playtennisbackground
Errore ricerca dati.
Il client ha inserito i seguenti dati:
MinSup: 1.0 MinGr: 1.0 TargetName: playtennisbackground TargetBackground: playtennistarget
Errore, i dati inseriti dall'client + Socket[addr=/127.0.0.1,port=50944,localport=8080] non hanno prodotto risultati.
```

Ulteriore esempio di log della shell del server.