

RELAZIONE DI SISTEMI E RETI

5/5/2025

AUTORE: STUDENTE DELLA CLASSE – 4IB

Jacopo Olivo

RELAZIONE DI SISTEMI

Relazione di Sistemi e Reti - CLASSE 4IB

Obiettivo dell'esperienza

Realizzare un'applicazione Client-Server in linguaggio Java, che utilizzi i socket TCP e che consenta agli utenti di consultare in remoto le informazioni di interesse relative a Strutture ricettive (Regione Veneto) presenti in un file in formato CSV.

Sviluppo applicazione

1. Progettare un server che:
 - Legga i dati dal file CSV allegato e li inserisca in una struttura interna (in questo caso un ArrayList).
 - Si implementi un'interfaccia di comunicazione per effettuare richieste e ricevere risposte.
 - Si gestiscano in maniera corretta gli errori.
2. Progettare un client che:
 - Permetta l'invio di richieste al server tramite l'interfaccia comunicativa.
 - Si mostrino in modo chiaro le risposte del server.
 -

Estensioni al programma

- ✓ Implementazione di ricerche filtrate dei dati.
- ✓ Implementazione di un'interfaccia grafica per il client (Parzialmente corretta, bug spiegato nella sezione 'Note' del file README.md).
- X Implementazione del protocollo UDP.

Struttura del programma e funzionamento

Qui di seguito la struttura delle varie classi presenti nelle cartelle del progetto e i relativi scopi:

Cartella ParteServer:

- MainServer.java: classe che avvia il server e gestisce i socket.
- CSVReader.java: classe che si occupa di leggere i dati dal file CSV e implementa i vari metodi di ricerca.
- ServerThreads.java: classe che gestisce il flusso di messaggi tra server e client.
- Dati.java: classe che comprende tutti gli attributi di un singolo dato del file CSV.

Cartella ParteClient:

- MainClient.java: classe che gestisce le richieste del client verso il server.
- Menur.java: classe che si occupa del menù e dei bottoni 'Connetti' ed 'Esci'.
- GUIClient.java: classe che comprende la finestra per la ricerca dei dati.

Funzionamento comunicazione server-client

La comunicazione tra il server e il client funziona tramite la logica dei socket TCP (la funzione estesa dei socket UDP non è stata implementata). Si inserisce l'indirizzo IP del server (di default è il local host 127.0.0.1) e il numero di porta scelto (di default: 1050). Il server successivamente legge le richieste di connessione del client dalla porta e l'accetta. Poi si procede con la lettura da parte del server delle richieste sui dati da parte del client e invia le risposte (ogni risposta termina con 'END_MESSAGE' per evitare che il client non termini più di leggere).

PROTOCOLLO DI COMUNICAZIONE CLIENT-SERVER

Qui di seguito viene illustrato il protocollo di comunicazione che viene utilizzati dalle due parti:

<i>Comando <u>parola chiave</u></i>	<i>Funzionamento</i>
<i>END</i>	<i>Termina la comunicazione tra le parti.</i>
<i>?</i>	<i>Mostra i comandi del protocollo.</i>
<i>nuova_ricerca</i>	<i>Inizializza una nuova ricerca rimuovendo i filtri precedenti.</i>
<i>ricerca_comune <u>comune</u></i>	<i>Mostra le strutture del comune inserito.</i>
<i>ricerca_provincia <u>provincia</u></i>	<i>Mostra le strutture della provincia inserita.</i>
<i>ricerca_tipologia <u>tipologia</u></i>	<i>Mostra le strutture della tipologia inserita.</i>
<i>ricerca_categoria <u>categoria</u></i>	<i>Mostra le strutture della categoria inserita.</i>
<i>ricerca_stelle <u>numeroStelle</u></i>	<i>Mostra le strutture con un determinato numero di stelle.</i>
<i>ricerca_denominazione <u>denominazione</u></i>	<i>Mostra la struttura con la denominazione inserita.</i>
<i>ricerca_indirizzo <u>indirizzo</u></i>	<i>Mostra la struttura presente all'indirizzo inserito.</i>
<i>ricerca_CAP <u>cap</u></i>	<i>Mostra le strutture presenti al CAP inserito.</i>
<i>ricerca_numeroTelefono <u>telefono</u></i>	<i>Mostra la struttura con il numero di telefono inserito.</i>
<i>ricerca_fax <u>fax</u></i>	<i>Mostra la struttura con il fax inserito.</i>
<i>ricerca_postaElettronica <u>postaElettronica</u></i>	<i>Mostra la struttura con la posta elettronica inserita.</i>
<i>ricerca_zona <u>zona</u></i>	<i>Mostra le strutture presenti in una zona inserita.</i>
<i>ricerca_feature <u>feature</u></i>	<i>Mostra le strutture che possiedono la feature inserita.</i>
<i>ricerca_ambiente <u>ambiente</u></i>	<i>Mostra le strutture che possiedono un determinato ambiente circostante.</i>
<i>ricerca_lingua <u>lingua</u></i>	<i>Mostra le strutture nelle quali si parla la lingua inserita.</i>
<i>ricerca_codice <u>codice</u></i>	<i>Mostra la struttura che possiede il codice inserito.</i>
<i>GET_ROW <u>n</u></i>	<i>Mostra la struttura che si trova nella riga <u>n</u> del file CSV.</i>
<i>ALL</i>	<i>Mostra tutte le strutture del file CSV.</i>

Errori che possono essere visualizzati nell'output:

- Errore nel comando! -> Indica un errore nel comando inserito (non appartenente al protocollo soprastante).
- Errore: nessun risultato trovato per la parola parola ! -> Indica un errore nella parola chiave inserita (non sono state trovate strutture con tale caratteristica).
- Errore: il numero è fuori dai limiti! -> Viene visualizzato se viene inserito un numero più piccolo di 0 o più piccolo del numero di righe totali durante il GET_ROW.