25/03/2025

**PROGETTO DI FINE ANNO**

**AUTORE CLASSE**

Leonardo Fortin 4IB

**PRESENTAZIONE**

Progetto di fine anno di Sistemi e Reti

Assegnato il martedì 25 marzo 2025

Data di consegna: martedì 6 maggio 2025

|  |
| --- |
| **Descrizione del progetto** |
| Applicazione Client-Server per la consultazione da remoto di informazioni relative a Monumenti presenti in un file CSV |

|  |
| --- |
| **Obiettivo dell’esperienza** |
| Realizzare un’applicazione Client-Server in linguaggio Java che consenta agli utenti di consultare in remoto le informazioni di interesse relative a **Monumenti** presenti in un file in formato CSV.  Il file CSV contiene dati strutturati da utilizzare per implementare e testare l’applicazione. |

**SVILUPPO TEORICO**

Progettare e implementare un Server che:

* Legga e interpreti un file CSV, organizzando i dati in una struttura interna.
* Offra un’interfaccia di comunicazione remota tramite socket per ricevere richieste e inviare risposte al client.
* Gestisca in modo robusto eventuali errori, come richieste non valide o malformate.

Progettare e implementare un Client che:

* Permetta all’utente di inviare richieste specifiche al server, che sono documentate nel file README del progetto.
* Mostri in modo chiaro e leggibile i dati ricevuti dal server.

Sviluppare per questo un’apposita applicazione client avanzata dotata di interfaccia grafica.

Progettare un protocollo di comunicazione tra client e server che deve seguire un formato semplice e ben definito e documentato successivamente nella seguente relazione.

**PROTOCOLLO DI COMUNICAZIONE TRA CLIENT E SERVER**

|  |  |  |
| --- | --- | --- |
| RICHIESTA DAL CLIENT | SPIEGAZIONE RICHIESTA | EVENTUALI MESSAGGI DI ERRORE |
| GET\_ROW r | Ottenere la riga r del file CSV | ERROR: Riga inesistente |
| GET\_PER\_COMUNE c | Ottenere tutti i monumenti del comune c | ERROR: Non esiste nessun monumento in questo comune |
| GET\_PER\_PROVINCIA p | Ottenere tutti i monumenti della provincia p | ERROR: Non esiste nessun monumento in questa provincia |
| GET\_PER\_REGIONE r | Ottenere tutti i monumenti della regione r | ERROR: Non esiste nessun monumento in questa regione |
| GET\_PER\_NOME n | Ottenere un monumento con il nome n | ERROR: Non esiste nessun monumento con questo nome |
| GET\_PER\_NOME\_PARZIALE n | Ottenere tutti i monumenti che contengono n nel proprio nome | ERROR: Nessun monumento contiene questo nome |
| GET\_PER\_TIPO t | Ottenere tutti i monumenti di tipo t | ERROR: Non esiste nessun monumento di questo tipo |
| GET\_PER\_ANNO a | Ottenere tutti i monumenti inseriti in un certo anno | ERROR: Nessun monumento è stato inserito in quell’anno |
| GET\_PER\_ANNI a1, a2 | Ottenere tutti i monumenti inseriti tra l’anno a1 e a2 (a2 > a1) | ERROR: Non è stato inserito nessun monumento tra i due anni |
| GET\_TRA\_LONGITUDINI lon1, lon2 | Ottenere tutti i monumenti presenti tra le due longitudini lon1 e lon2 (lon2 > lon1) | ERROR: Non è presente nessun monumento tra le due longitudini |
| GET\_TRA\_LATITUDINI lat1, lat2 | Ottenere tutti i monumenti presenti tra le due latitudini lat1 e lat2 (lat2 > lat1) | ERROR: Non è presente nessun monumento tra le due latitudini |
| GET\_TRA\_LONGITUDINI\_E\_LATITUDINI lon1, lon2, lat1, lat2 | Ottenere tutti i monumenti presenti tra le due longitudini lon1 e lon2 (lon2 > lon1) e le due latitudini lat1 e lat2 (lat2 > lat1) | ERROR: Non è presente nessun monumento tra le due longitudini e le due latitudini |
| END | Chiude la connessione verso il Server | Il Server è stato chiuso o riavviato |

**DESCRIZIONE TECNICA**

**Struttura del progetto**

Il progetto prevede due applicazioni Client-Server, una che funziona con TCP e una con UDP.

* TCP: Il Server gestisce più connessioni utilizzando i sockets
* UDP: Il Server gestisce una connessione alla volta

**Funzionamento**

Il Server dispone del file CSV contenente tutti i Monumenti italiani

Il Client, munito di interfaccia grafica, comunica con il Server per consultare informazioni relative ai Monumenti.

**Installazione ed esecuzione**

Aprire i progetti del Server e del Client (dell'applicazione TCP o UDP) sull'ambiente di sviluppo.

1. Avviare il Server

2. Avviare il Client, o stabilire una connessione Telnet (solo TCP)

**Specifiche e Funzionalità**

Il Server mette a disposizione funzionalità di ricerca. Riceve comandi dal Client sottoforma di Stringhe JSON, li interpreta, e restituisce al Client le informazioni che ha richiesto, sempre sottoforma di Stringhe JSON.

Al momento della connessione, il Client chiede al Server la lista dei comandi disponibili e gli attributi dei Monumenti. In questo modo esso si rende pressoché indipendente ed è possibile modificare questi due elementi sul Server senza bisogno di modificare il Client.

I comandi possono essere inviati sulla console del Client così come sono riportati nella tabella, oppure tramite l’interfaccia grafica.

Se si sceglie invece di stabilire una connessione Telnet, i comandi dovranno essere inviati sul terminale con il seguente formato:

{comando:”<comando>”,parametro:”<parametro (o parametri separati da spazi)>”}