25/03/2025

**AUTORE CLASSE**

Leonardo Fortin 4IB

**PROGETTO DI FINE ANNO**

**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** |
| Realizzazione di un’applicazione client-server per la consultazione remota di 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.

Usare una struttura a mappa…

* Offra un’interfaccia di comunicazione remota tramite socket per ricevere richieste e inviare risposte al client.

Usare il protocollo TCP…

* Gestisca in modo robusto eventuali errori, come richieste non valide o malformate.

Creare apposite eccezioni personalizzate in Java, per aumentare la chiarezza del codice.

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

**PROGETTAZIONE E IMPLEMENTAZIONE DEL SERVER**

L

**PROGETTAZIONE E IMPLEMENTAZIONE DEL CLIENT**

L

**PROTOCOLLO DI COMUNICAZIONE TRA CLIENT E SERVER**

|  |  |  |  |
| --- | --- | --- | --- |
| RICHIESTA DAL CLIENT | SPIEGAZIONE RICHIESTA | RISPOSTA DAL SERVER | EVENTUALI MESSAGGI DI ERRORE |
| GET\_ROW r | Ottenere la riga r del file CSV | Il monumento corrispondente | ERROR: Riga inesistente |
| GET\_PER\_COMUNE c | Ottenere tutti i monumenti del comune c | I dati corrispondenti | ERROR: Non esiste nessun monumento nel comune c |
| GET\_PER\_PROVINCIA p | Ottenere tutti i monumenti della provincia p | I dati corrispondenti | ERROR: Non esiste nessun monumento nella provincia p |
| GET\_PER\_REGIONE r | Ottenere tutti i monumenti della regione r | I dati corrispondenti | ERROR: Non esiste nessun monumento nella regione r |
| GET\_PER\_NOME n | Ottenere un monumento con il nome n | Il monumento corrispondente | ERROR: Non esiste nessun monumento con il nome n |
| GET\_PER\_NOME\_PARZIALE n | Ottenere tutti i monumenti che contengono n nel proprio nome | I dati corrispondenti | ERROR: Nessun monumento contiene n nel proprio nome |
| GET\_PER\_ANNO a | Ottenere tutti i monumenti inseriti in un certo anno | I dati corrispondenti | ERROR: L’anno a deve ancora arrivare  ERROR: Nessun monumento è stato inserito nell’anno a |
| GET\_PER\_ANNI a1, a2 | Ottenere tutti i monumenti inseriti tra l’anno a1 e a2 (a2 > a1) | I dati corrispondenti | ERROR: L’anno a2 deve ancora arrivare  ERROR: Nessun monumento è stato inserito in questi anni |

Il client fa in modo che le richieste possano essere combinate filtrando più campi uno per volta.

Ovviamente i filtri non possono essere applicati a seguito di comandi come GET\_PER\_NOME n e GET\_ROW r, poiché essi forniscono un unico risultato.

Questi due comandi possono essere invece dei filtri applicabili a un precedente GET\_PER\_REGIONE r.