**Il protocollo di comunicazione utilizzato**

**Funzionamento generale:**

Il client seleziona l’opzione di un menù fornito dal server e il server restituisce una stringa rielaborata includente il dato richiesto.

**Scelte progettuali:**

**Server:**

* Utilizzo del protocollo TCP per il trasporto dati in quanto questo è più semplice da gestire con il linguaggio java.
* Nel main del server è presente la creazione di un ServerSocket che ci consentirà la creazione di una connessione.
* Il codice del server è in grado gestire più connessioni grazie all’utilizzo dei thread. Dunque quando viene fatta una richiesta di comunicazione questa richiesta sarà gestita da un nuovo thread il quale sarà in grado di creare un nuovo socket. Ogni Thread implementa un metodo menù che servirà permettere al client connesso al server di decidere l'operazione che vuole eseguire.
* Nel progetto del server viene implementata una classe “GestoreDati” nella quale viene letto il file csv tramite l’utilizzo di uno scanner; il dato ottenuto,con i suoi campi, viene memorizzato in un ArrayList. La classe implementa tutti i metodi che permettono la rielaborazione dei dati. Questi metodi devono essere accessibili da tutti i thread e quindi da tutti i client connessi ai server.
* La struttura dati ArrayList permette una gestione dei dati memorizzati piuttosto semplice ma non garantisce che i dati memorizzati non siano duplicati. L’ArrayList è la struttura dati perfetta in quanto non ci serve garantire che i dati non siano duplicati.
* Il server riceverà la richiesta di eseguire un’operzione e il risultato dei dati rielaborati deve essere inviato al client tramite una stringa rielaborata contenete il dato.
* La richiesta da parte del client viene effettuata tramite la selezione di un'opzione di un menù mostrato tramite interfaccia grafica, ogni opzione selezionata corrisponde ad un intero che verrà inviato al server il quale capirà l’opzione di ricerca nel file deve eseguire. L’interfaccia grafica permette anche l’inserimento dei filtri per la ricerca nel file.

**Client:**

* Il codice del client garantisce la connessione al server tramite la creazione del socket
* Il client mostra tramite interfaccia grafica il menu con le operazioni possibili da eseguire.
* Quando l’utente sceglie l’opzione il client invierà al server un intero che servirà ad identificare univocamente il metodo che permette di eseguire tale operazione.
* In base all'opzione scelta verrà richiesto l’inserimento di alcuni filtri che permettono la rielaborazione dei dati in base all'operazione richiesta.
* Il server invierà una stringa rielaborata contenente il dato richiesto e il client dovrà visualizzarla sempre tramite interfaccia grafica.

**Possibili richieste che può fare il client verso il server:**

* Riga del flie csv: Il client permette l’inserimento di un intero corrispondente al numero della riga, el server restituisce il toString() del dato corrispondente alla riga selezionata. (tramite metodo “getRiga()” istanziato nella classe “GestoreDati”) **Codice 1.**
* Provincia che ha prodotto più rifiuti in un determinato anno: Il server richiede l’anno, il client permette l’inserimento, il server analizza i dati memorizzati (tramite metodo “provinciaRifiutiAnno()” istanziato nella classe “GestoreDati”) e invia la stringa “La provincia che a prodotto più rifiuti nell’anno x è y”. **Codice 2.**
* Quantità di rifiuti prodotti per tipo dalla regione: Dopo la richiesta del client il server invia la stringa “La regione Veneto ha prodotto x rifiuti pericolosi e y rifiuti speciali” (tramite metodo “rifiutiProdotti()” istanziato nella classe “GestoreDati”). **Codice 3.**
* Quantità di rifiuti prodotti per tipo in un anno:Il server richiede l’anno, il client permette l’inserimento, il server analizza i dati memorizzati (tramite metodo “rifiutiProdottiAnno()” istanziato nella classe “GestoreDati”) e invia la stringa “Nell’anno x sono stati prodotti y rifiuti pericolosi e z rifiuti speciali”. **Codice 4.**
* Quantità di rifiuti prodotti per tipo da una provincia:Il server richiede l’anno e la provincia, il client permette l’inserimento, il server analizza i dati memorizzati (tramite metodo “rifiutiProdottiProvincia()” istanziato nella classe “GestoreDati”) e invia la stringa “La provincia x nell’anno y ha prodotto z rifiuti pericolosi e k rifiuti speciali ”. **Codice 5.**
* Anno in cui sono stati prodotti più rifiuti: Dopo la richiesta del client il server invia la Stringa “La regione Veneto ha prodotto più rifiuti nel x” (tramite metodo “annoRifiuti()” istanziato nella classe “GestoreDati”). **Codice 6.**
* Provincia che ha prodotto più rifiuti: Dopo la richiesta del client il server invia la Stringa “La provincia che ha prodotto più rifiuti è : x” (tramite metodo “provinciaRifiuti() istanziato nella classe “GestoreDati”). **Codice 7.**