



UNIVERSITÀ DI PISA

DIPARTIMENTO DI INFORMATICA

Relazione del progetto **TURING**
RCL a.a. 18-19

Autore: Michele Zoncheddu
Marzo 2019

Indice

| | | |
|----------|---|----------|
| 1 | Cosa è richiesto nella relazione | 2 |
| 2 | Scelte progettuali | 3 |
| 2.1 | Architettura | 3 |
| 2.2 | Gestione della concorrenza | 3 |

1 Cosa è richiesto nella relazione

- una descrizione generale dell'architettura complessiva del sistema, in cui sono motivate le scelte di progetto;
- uno schema generale dei threads attivati da ogni componente e delle strutture dati utilizzate, con particolare riferimento al controllo della concorrenza;
- una descrizione sintetica delle classi definite ed indicazioni precise sulle modalità di esecuzione.
- una sezione di istruzioni su come compilare ed eseguire il progetto (librerie esterne usate, argomenti da passare al codice, sintassi dei comandi per eseguire le varie operazioni...). Questa sezione deve essere un manuale di istruzioni semplice e chiaro per gli utilizzatori del sistema.

2 Scelte progettuali

Di seguito verranno illustrate le più importanti scelte progettuali effettuate durante la realizzazione del progetto.

2.1 Architettura

Per l'architettura del server erano possibili principalmente due soluzioni, che sono state trattate più in dettaglio durante il Laboratorio di Reti di Calcolatori:

1. multithread sincrona con I/O bloccante (**Sockets** di **Java IO**);
2. monothread sincrona con I/O non bloccante (**Selectors** di **Java NIO**).

Le due soluzioni hanno pregi e difetti complementari, ma il trade-off principale è tra **velocità** e **scalabilità**.

Per questo progetto didattico è stata ritenuta più importante la velocità, quindi è stata scelta la prima soluzione.

2.2 Gestione della concorrenza

Per ogni client, il server riserva un thread, che si occuperà di gestire l'intera connessione TCP.

I thread sono gestiti da un **cached thread pool**, scelto per la sua elasticità nella gestione delle risorse; sono inoltre indipendenti ed isolati, e per ogni richiesta esterna devono utilizzare uno dei seguenti manager:

- **userManager**: gestisce concorrentemente gli utenti registrati;
- **documentManager**: gestisce concorrentemente¹ i documenti salvati;
- **addressManager**: gestisce concorrentemente gli indirizzi IP multicast da assegnare alle chat dei gruppi di lavoro.

¹In questo caso la corretta gestione della concorrenza non è garantita da una struttura dati concorrente, ma dalle locks delle sezioni.