

# TURING: disTribUted collaboRative edItiNG Laboratorio di Reti, Corsi A e B Progetto di Fine Corso A.A. 2018/19

## 1 Descrizione del problema

Recentemente l'uso di strumenti on line di produzione collaborativa di documenti è enormemente cresciuto, in parallelo con la diffusione del web.

La scrittura collaborativa consiste nella creazione di testi da parte di un gruppo di persone, dove ognuno fornisce un contributo individuale alla produzione del documento finale. I testi prodotti possono essere di varia natura: da documenti scientifici e didattici a codice sorgente, ed anche testi letterari. Esempi di strumenti di questo tipo sono Google Documents, OverLeaf, BoomWriter.

Il progetto consiste nell'implementazione di TURING (disTribUted collaboRative edItiNG), uno strumento per l'editing collaborativo di documenti che offre un insieme di servizi minimale.

## 2 TURING: specifica delle operazioni

Di seguito vengono elencate le operazioni che un utente può eseguire in TURING e che dovranno essere implementate nel progetto.

**Registrazione al servizio** Ogni utente deve per prima cosa effettuare una registrazione, fornendo username e password e, solo successivamente, può iniziare ad usufruire del servizio. Dopo aver effettuato la registrazione, l'utente può autenticarsi ed accedere ai servizi effettuando il login. Lo username, in quanto identificativo dell'utente, deve essere univoco e non può essere cambiato.

**Creazione di un documento** Dopo essersi registrato, l'utente può creare un nuovo documento ed invitare altri utenti registrati al servizio a collaborare all'editing del documento creato. Ogni documento ha un solo creatore, che è l'unico che ha il diritto di invitare i collaboratori. Il creatore del documento può invitare nuovi collaboratori durante tutto il ciclo di vita del documento. Al

momento della creazione del documento, il creatore deve dividere il documento in un certo numero di sezioni. La sezione è l'unità base dell'editing. Non è possibile aggiungere sezioni, dopo la creazione del documento.

**Gestione Inviti.** Quando un utente  $U_1$  decide di invitare un altro utente  $U_2$ , TURING aggiunge  $U_2$  alla lista degli utenti autorizzati a lavorare sul documento. L'invito viene notificato immediatamente ad un utente, se online, oppure al momento del successivo login.

**Lista dei documenti.** Il server offre un servizio per richiedere la lista di documenti che un certo utente, precedentemente autenticato, è autorizzato ad accedere e modificare. L'elenco dei documenti può cambiare anche durante una sessione di login dell'utente (ad esempio a seguito di un invito).

**Editing di un documento** Ogni documento è strutturato in un insieme di sezioni, dove ogni sezione contiene una sequenza di linee di testo (non è previsto l'inserimento di foto o di immagini nel documento). Durante la fase di editing, ogni utente può interagire con TURING attraverso i seguenti comandi (si supponga che  $S$  indichi una sezione del documento  $D$ ):

- *Create* ( $D, numSezioni$ ). Crea il documento  $D$  diviso in `numsezioni` sezioni.
- *Edit* ( $S, D$ ). Verifica che l'utente sia il creatore di  $S$  oppure che abbia ricevuto un invito, e che nessun altro utente stia editando quella sezione. In questo caso, il testo di  $S$  viene inviato all'utente. Quindi in ogni istante di tempo, al massimo un utente, tra quelli abilitati, può modificare una sezione. Un utente in ogni istante di tempo può modificare al più una sezione di un documento. NOTA: non si richiede che il programma java gestisca anche le azioni di editing sul testo del documento. A questo scopo possono essere usati editor esterni.
- *End-Edit* ( $S, D$ ). Notifica la fine dell'editing di una sezione e spedisce la sezione modificata a TURING.
- *Show* ( $S, D$ ) Richiede la visualizzazione di  $S$ . TURING invia all'utente l'ultima versione memorizzata di  $S$ , che non include le modifiche eventualmente in corso su  $S$ , ed indica se un utente sta eventualmente editando, nello stesso momento, la stessa sezione.
- *Show* ( $D$ ) Richiede la visualizzazione di  $D$ . TURING invia all'utente l'ultima versione memorizzata di  $D$ , che non include le modifiche eventualmente in corso su  $D$ , ed indica quali sezioni sono eventualmente in fase di modifica al momento della richiesta.

**Chat** Tutti gli utenti che collaborano, in un certo istante, all'editing di un documento (su sezioni diverse) possono interagire sfruttando un servizio di chat. TURING crea la chat non appena un utente sta editando il documento e la chiude quando nessun utente sta lavorando sul documento.

**Memorizzazione dei documenti** Ogni sezione di un documento è memorizzata su un file ed un documento può essere quindi considerato un insieme di file. Non appena un utente ha terminato l'editing di una sezione, invia la sezione modificata a TURING che la sostituisce nel file corrispondente.

### 3 Specifiche per l'implementazione

Nella realizzazione del progetto devono essere utilizzate molte tecnologie illustrate durante il corso. In particolare:

- la fase di registrazione viene implementata mediante RMI.
- la fase di login viene implementata con TCP
- le richieste di apertura/modifica di un documento e le relative risposte sono inviate dall'utente al server mediante connessioni TCP.
- gli inviti vengono inviati mediante connessioni TCP
- il servizio di chat è implementato mediante multicast UDP
- il server può essere realizzato multithreaded oppure può effettuare il multiplexing dei canali mediante NIO. La gestione dei file è realizzata utilizzando NIO.
- *interazione con TURING*. L'utente può interagire con TURING mediante una semplice interfaccia grafica, oppure attraverso una CLI (Command Line Interface), definendo un insieme di comandi, presentati in un menù (vedi Appendice A).
- *implementazione della chat*: nel caso in cui si decida di implementare l'interfaccia grafica, essa prevederà due semplici aree di testo in cui rispettivamente inserire/ricevere i messaggi testuali inviati alla chat. In questo caso, i messaggi vengono immediatamente presentati all'utente, mano a mano che vengono ricevuti. Invece, nel caso si preferisca una interazione con TURING a linea di comando, saranno definiti due comandi per, rispettivamente, inviare nuovi messaggi alla chat/ricevere tutti i messaggi ricevuti a partire dall'ultima esecuzione del comando di visualizzazione messaggi. In questo caso, i messaggi vengono presentati all'utente in modo asincrono, su sua richiesta.

## 4 Modalità di svolgimento e consegna

Il materiale consegnato deve comprendere:

- il codice dell'applicazione e di eventuali programmi utilizzati per il test delle sue funzionalità. È importante che il codice segua le convenzioni JAVA e le linee guida date per gli assegnamenti;
- la relazione in formato pdf che deve contenere:
  - 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.

L'organizzazione e la chiarezza della relazione influiranno sul voto finale.

Relazione e codice sorgente devono essere consegnati su Moodle. La relazione deve anche essere consegnata presso la portineria del Dipartimento.

Per domande di chiarimento sul progetto verrà attivato un forum di discussione su moodle a comune dei moduli A e B nell'ambito del corso di Reti di Elaboratori (NOTA BENE: non del corso di Laboratorio).