



**Università degli Studi di Bologna
Scuola di Ingegneria**

Corso di Reti di Calcolatori T

**Esercitazione 8 (proposta)
Servizio di Gestione delle
Votazione FattoreX**

Antonio Corradi, Luca Foschini

Lorenzo Rosa, Giuseppe Martuscelli, Marco Torello

Anno accademico 2022/2023

SERVIZIO DI VOTAZIONE PER FATTOREX

Il progetto **VotaFattoreX** gestisce il sistema di votazione di un noto talent show: lo show è una competizione tra giudici, ognuno dei quali è a capo di un gruppo di candidati

Ogni candidato fa parte di una *categoria*:

Uomini (**U**), Donne (**D**), Over25 (**O**), Band (**B**);

e può essere in una delle seguenti *fasi*:

Audizioni (**A**), BootCamp (**B**), Live Show (**S**)

Il server mantiene **per ogni partecipante** le seguenti informazioni:
candidato (chiave della tabella che può comparire una sola volta),
giudice, **categoria**, **nome del file** che descrive il candidato (*nome unico all'interno di un direttorio sul server*), **fase** in cui si trova il candidato e un intero che riporta il **numero totale di voti ricevuti** da un candidato (come da tabella presentata più avanti)

Obiettivo è la realizzazione di alcune **funzionalità di supporto** usando **RPC di SUN**

SPECIFICA

Si realizzino le operazioni per:

Visualizzare la classifica dei giudici: questa operazione **non richiede parametri** di input e restituisce i **nomi dei giudici ordinati per punteggio totale**. Si ricordi che ogni giudice può avere più candidati, per i quali occorre sommare tutti i punteggi

Aggiungere un voto per un candidato: questa operazione richiede all'utente **il nome del candidato** e **aggiunge un voto** alla prestazione identificata

Sottrarre un voto per un candidato: questa operazione richiede all'utente **il nome del candidato** e **sottrae un voto**. Si assuma che il punteggio minimo possibile sia zero

STRUTTURA DATI

La **struttura dati** mantiene lo stato, fino ad un massimo di N candidati (L, per libero come valore di default), da implementare nei diversi ambienti richiesti

N {	Candidato	Giudice	Categoria	Nome file	Fase	Voto
	L	L	L	L	L	-1
	Brasco	Bowie	U	BrascoProfile.txt	A	100
	Viga	Winehouse	D	VigaProfile.txt	S	50

	L	L	L	L	L	-1

SPECIFICA: ULTERIORI DETTAGLI

Il **Client filtro** realizza l'interazione con l'utente, proponendo ciclicamente i servizi che utilizzano le due procedure remote e stampa a video gli esiti delle chiamate, fino alla fine del file di input da tastiera

Il **Server** implementa le procedure invocabili:

- La procedura **classifica_giudici** non richiede parametri d'ingresso e restituisce **la struttura Output che contiene i nomi di tutti i giudici, ordinati secondo il punteggio totale dei rispettivi candidati**
- La procedura **esprimi_voto** accetta come parametro d'ingresso **due stringhe**, la prima rappresenta **il candidato** mentre la seconda specifica il **tipo di operazione (aggiunta/sottrazione)** e aggiorna la struttura dati aggiungendo o sottraendo un voto al candidato scelto



PROPOSTA DI ESTENSIONE: GESTIONE LISTE



Si vuole estendere la soluzione proposta, in modo da gestire **i candidati** senza limitazioni statiche, realizzando e gestendo una **struttura dati dinamica**

Inoltre, si aggiungano alcune **nuove operazioni ritenute significative per rendere dinamico ogni aspetto della applicazione**, in particolare per la **modifica** e la **consultazione** di tale struttura

(es. per aggiungere e cancellare candidati o per ottenere l'elenco dei candidati con un punteggio maggiore di N, ottenere tutti i candidati di una categoria, ottenere tutti i candidati in una fase particolare dello show...)