



UNIVERSITÀ DEGLI STUDI  
DI MILANO



# Laboratorio di basi di dati A.A. 2017-18

Prof. Alfio Ferrara  
Prof. Marco Frasca, Prof. Stefano Montanelli

## Traccia del progetto: *profiling-for-crowdsourcing* (p4c)

La presente traccia è valida per i progetti presentati negli appelli **da gennaio 2018 fino a gennaio 2019 inclusi**.

Si chiede di realizzare un'applicazione web di basi di dati per lo svolgimento di attività (*task*) in modalità crowdsourcing.

## 1 Descrizione del dominio applicativo

Il crowdsourcing è un modello di lavoro collettivo nel quale un insieme potenzialmente ampio di lavoratori (la cosiddetta *crowd*) si rende disponibile allo svolgimento di task a fronte di un compenso (denaro, bonus promozionali o altre forme di remunerazione).

Un task è da considerarsi come un'attività da svolgere espressa come una domanda a risposta chiusa dotata di un titolo, una descrizione contestuale e un elenco di possibili opzioni di risposta. Un task è inoltre associato a un elenco di parole chiave (*keyword*) che descrive le conoscenze/attitudini richieste ai lavoratori per lo svolgimento di quel task. Ciascun lavoratore, ricevendo un task da eseguire, considera la richiesta e seleziona l'opzione che ritiene più adeguata come risposta fra quelle disponibili.

Ciascun task viene assegnato a più lavoratori e ogni lavoratore risponde alla domanda contenuta nel task in modo autonomo, senza interagire con altri lavoratori. Un lavoratore non può eseguire uno specifico task più di una volta.

Il task ha un risultato *valido* quando esiste consenso sull'opzione scelta come risposta tra i lavoratori che hanno eseguito il task. Per la valutazione del consenso tra i lavoratori si utilizza un criterio a maggioranza.

Sono previste due tipologie di utenza: i promotori denominati *requester* e i lavoratori denominati *worker*.

**Requester.** Sono gli utenti responsabili delle campagne di lavoro. Una campagna di lavoro è promossa da un requester e ha un nome identificativo (il medesimo nome di campagna può essere utilizzato da requester diversi), una data di apertura, una data di terminazione e un periodo di registrazione nel quale i lavoratori possono scegliere di aderire alla campagna e farne parte. Alla creazione di una campagna di lavoro, il requester definisce:

- i task da eseguire;

- il numero di lavoratori da coinvolgere nello svolgimento di ciascun task;
- la soglia di maggioranza richiesta per considerare valido il risultato di un task.

Il risultato di ciascun task viene determinato quando è stato raccolto il numero di risposte previsto. Il risultato è definito quando i lavoratori che hanno eseguito il task concordano sulla risposta con una maggioranza uguale o superiore a quella richiesta. Il risultato è *null* qualora non sia stata raggiunta una maggioranza tra i lavoratori che hanno partecipato allo svolgimento del task.

**Worker.** Sono gli utenti coinvolti nello svolgimento dei task. I lavoratori sono associati a un profilo che definisce, tra le altre cose, sia le conoscenze sia le attitudini del lavoratore. A ciascuna voce del profilo è associato un valore che quantifica il grado di conoscenza/attitudine su quella specifica voce. Ad esempio, un possibile profilo di worker può contenere gli elementi *cinema* e *classificazione* per denotare che il lavoratore è competente di argomenti legati alla cinematografia e possiede l'attitudine di saper classificare oggetti. Il valore associato ai due elementi definisce rispettivamente il grado di competenza sulla cinematografia e l'abilità nel classificare gli oggetti in categorie. Ciascun lavoratore sceglie autonomamente le campagne di lavoro a cui aderire. Nell'ambito di una campagna, il lavoratore fa richiesta di un task da eseguire e il sistema sceglie il task più appropriato da assegnare tenendo conto di i) parole chiave del task e del profilo del lavoratore, ii) grado di competenza/attitudine del lavoratore rispetto alle varie voci del suo profilo. In questo senso, l'applicazione da realizzare può essere considerata un esempio di sistema di raccomandazione dotato di un meccanismo di scheduling dei task. Dopo aver svolto un task assegnato, il lavoratore può fare richiesta di un nuovo task da eseguire. E' dunque possibile che un lavoratore svolga più task per ciascuna campagna a cui partecipa. Per ogni campagna di lavoro a cui partecipa, ciascun lavoratore è associato a uno **score** che denota il numero di task validi correttamente eseguiti, cioè task per i quali valgono le seguenti condizioni: i) il task è stato eseguito dal lavoratore, ii) il task ha un risultato finale (è stata raggiunta la maggioranza richiesta), iii) eseguendo il task, la risposta fornita dal lavoratore coincide con il risultato del task, cioè il lavoratore appartiene alla maggioranza che ha determinato il risultato del task.

**Esempio (utile per comprendere il funzionamento del sistema).** Si consideri una campagna di lavoro denominata "music and games" basata sui task contenuti nel file json distribuito con la traccia del progetto (esempio\_task.json). In particolare, si consideri il seguente task:

Id:	9
Titolo:	Indica la categoria migliore per il seguente album musicale
Descrizione:	The dark side of moon
Parole chiave:	musica
Opzioni risposta:	Jazz, Rock, Pop

Il requester ha richiesto l'esecuzione di ciascun task da parte di 5 worker e una maggioranza del 60% affinché il risultato dei task sia valido. Tra i lavoratori, *steve* è molto competente di musica leggera e possiede buone attitudini di "originalità" e "ragionamento deduttivo". Il lavoratore *john* è competente di "giochi" e possiede l'attitudine "comprensione orale-scritta". Qualora *steve* e *john* decidano di partecipare a "music and games" e facciano richiesta di task da eseguire, il sistema assegnerà preferibilmente i task musicali a *steve* e i task sui giochi a *john*. Per il task con id=9, i lavoratori coinvolti hanno risposto nel seguente modo: a) *rock*, b) *pop*, c) *rock*, d) *rock*, e) *pop*. Una volta raccolte tutte le risposte, il sistema valuta se esiste una maggioranza del 60% e determina l'eventuale risposta del task. Esistendo una tale maggioranza, il risultato del

task con id=9 è *rock* essendo la risposta scelta da 3 lavoratori su 5 coinvolti. Si procede quindi all'incremento dello score per i tre lavoratori che hanno fornito la risposta *rock*.

## 2 Requisiti della base di dati

La base di dati deve contenere specifiche procedure e trigger per realizzare le seguenti funzionalità:

- **Calcolo della maggioranza e definizione del risultato dei task.** In base alle risposte fornite dai lavoratori, la base di dati determina il risultato del task verificando la presenza della maggioranza richiesta.
- **Aggiornamento dello score dei lavoratori.** La base di dati dovrà aggiornare automaticamente lo score dei lavoratori ogni volta che le condizioni precedentemente illustrate lo rendono necessario.
- **Assegnazione task a lavoratori.** Data la richiesta di un task da eseguire da parte di un lavoratore, la base di dati seleziona il task da assegnare considerando le keyword del task e il profilo del lavoratore. Si scelgano i criteri e l'algoritmo da utilizzare per effettuare l'assegnazione in modo automatico.
- **Report di una campagna di lavoro.** Per ogni campagna, il requester può richiedere un report che mostra i task contenuti e il risultato di ciascun task. Inoltre, per ogni campagna, è possibile visualizzare una sintesi relativa a quanti task sono stati completati, quanti sono in fase di esecuzione e informazioni statistiche rispetto ai task che hanno ottenuto un risultato valido (ad esempio percentuale di task validi sul totale di task complessivamente eseguiti).
- **Top-10 di una campagna di lavoro.** Per ogni campagna, il requester può richiedere la *top-10*, cioè l'elenco dei 10 lavoratori migliori con relativo posizionamento in classifica. Si scelga il criterio da utilizzare per definire la top-10.
- **Statistiche per lavoratore.** Ciascun lavoratore può ottenere un report di sintesi che mostra la posizione in classifica, i task eseguiti e i task validi da lui svolti in ciascuna campagna di lavoro. Per determinare la posizione in classifica, si usi il criterio definito al punto precedente.

Da finire  
implementand  
o lato web

Si ribadisce che tutte le funzionalità illustrate in questa sezione **devono essere realizzate mediante strutture interne alla base di dati** (e.g., viste, procedure, trigger).

## 3 Requisiti dell'applicazione Web

L'applicazione web dovrà fornire le seguenti funzionalità:

- registrazione di requester e worker. I requester devono essere approvati da un utente amministratore del sistema prima di poter essere attivati;
- creazione di campagne di lavoro, inclusa la definizione dei task da eseguire (solo per i requester);
- adesione a campagne di lavoro ed esecuzione di task (solo per i lavoratori);

- visualizzazione (e modifica se previsto) del proprio profilo (solo per i lavoratori).

Si raccomanda di verificare che l'applicazione web realizzata sia valida rispetto allo standard HTML5 (<http://validator.w3.org/>).

## 4 Specifiche aggiuntive per gli studenti che svolgono il progetto in gruppo

L'applicazione dovrà fornire una funzionalità di aggiornamento “intelligente” dei profili dei lavoratori. Questo significa che il grado di competenza/attitudine di un lavoratore rispetto a ogni conoscenza o abilità nel suo profilo può variare a seguito dell'esecuzione dei task. In particolare, la funzionalità da realizzare dovrà tenere conto delle seguenti indicazioni:

- quando il lavoratore riceve un incremento di score a seguito dell'esecuzione di un task (il lavoratore è nella maggioranza che ha determinato il risultato del task), il profilo del lavoratore viene aggiornato positivamente, cioè viene aumentato il grado di competenza/attitudine del lavoratore sulle voci di profilo legate al task svolto. Ad esempio, se il task svolto era legato all'ambito musicale (le keyword del task contenevano il termine “musica”), il grado di competenza del lavoratore sulla voce di profilo “musica” viene incrementata;
- viceversa, quando il lavoratore esegue un task, ma non riceve un incremento di score (non si costituisce una maggioranza o la maggioranza costituita non contiene il lavoratore), il profilo del lavoratore viene aggiornato negativamente.

## 5 Informazioni aggiuntive (e IMPORTANTI)

Affinché il progetto sia ritenuto valido è necessario che lo studente **progetti e realizzi TUTTE le specifiche riportate nelle Sezioni 2 e 3**. Gli studenti che svolgono il progetto in gruppo dovranno considerare in aggiunta anche le specifiche riportate in Sezione 4.

**Nel caso in cui la realizzazione di una funzionalità si prestasse a interpretazioni alternative ma compatibili con il testo delle specifiche, lo studente potrà adottare la soluzione che riterrà più opportuna, introducendo eventuali vincoli aggiuntivi e documentando la scelta nella relazione tecnica da allegare al progetto consegnato.** In particolare, i seguenti punti sono DELIBERATAMENTE lasciati alla libera interpretazione in modo che ciascuno studente possa proporre la propria soluzione:

- gestione delle keyword associate a task e worker. In particolare, si illustri come affrontare il problema del confronto fra parole chiave per trovare possibili corrispondenze (l'uso di un vocabolario controllato o il confronto sintattico/linguistico di keyword sono entrambi approcci ammissibili alla risoluzione del problema);
- definizione del profilo di un lavoratore. In particolare, si illustri i) chi, quando e come definisce il profilo di un lavoratore; ii) come rappresentare il grado di competenza/attitudine di un lavoratore rispetto a ogni voce di profilo; iii) se il profilo dei lavoratori prevede per tutti le stesse voci o se ciascun lavoratore ha voci specifiche relative alle proprie competenze/attitudini (e potenzialmente diverse da quelle degli altri lavoratori);
- algoritmo di scheduling utilizzato per assegnare i task ai lavoratori. In particolare, si illustri cosa accade quando un lavoratore richiede un task da eseguire, ma non esistono task compatibili con il suo profilo;

- meccanismo di aggiornamento del profilo dei lavoratori (per coloro che svolgono il progetto in gruppo). In particolare, si illustri i) come aggiornare il profilo se le voci possono differire da un lavoratore all'altro; ii) come aggiornare il profilo se il lavoratore esegue un task non-compatibile con il suo profilo, cioè come aggiornare voci relative a competenze/attitudini che il lavoratore non possiede nel suo profilo.

**E' importante documentare nella relazione tecnica le scelte adottate in corrispondenza dei suddetti punti e di ogni altro aspetto che si ritenga utile portare all'attenzione di chi valuta il progetto.**

In sede di discussione del progetto, la base di dati dovrà essere adeguatamente popolata per consentire la verifica di tutte le funzionalità richieste dalle specifiche.

Per informazioni relative alle modalità di consegna si faccia riferimento al documento `modalita__consegna.pdf` allegato al progetto.