

A.A. 2023-2024
ALLIEVI DEL III ANNO IN INGEGNERIA INFORMATICA

PROGETTO DA PRESENTARE
OBBLIGATORIAMENTE
PER LA PROVA ORALE D'ESAME DELL'INSEGNAMENTO
INGEGNERIA DEL SOFTWARE (9 CFU)

N.B. Una opportuna ulteriore attività proposta dai docenti, tesa a estendere o approfondire il progetto (prima e seconda parte) realizzato nell'ambito dell'insegnamento di “Ingegneria del Software” e svolta autonomamente dal singolo studente, con produzione di un elaborato individuale, può essere l'oggetto della PROVA FINALE (3 CFU) per il conseguimento della LAUREA IN INGEGNERIA INFORMATICA.

TEMA

Si desidera realizzare, secondo un processo di sviluppo incrementale/iterativo, un sistema software di supporto allo scambio, senza passaggi di denaro, di prestazioni d'opera riconducibili a un insieme prefissato (non vuoto) di categorie di attività. L'applicazione può essere adottata da varie organizzazioni, che presiedono allo scambio di ore di attività afferenti a insiemi diversi di categorie e/o operano in luoghi diversi.

L'applicazione prevede due tipologie di utente, il *configuratore* e il *fruitore*. Il primo è un esponente dell'organizzazione che, attraverso l'applicazione software, sovrintende agli scambi di prestazioni d'opera senza uso di denaro; egli è deputato alla descrizione delle categorie di attività scambiabili e può ottenere informazioni relative agli attuali scambi potenziali o a quelli già avvenuti. Il secondo tipo di utente è una persona che si rivolge all'applicazione con l'intento di effettuare degli scambi, se questi risulteranno possibili. Se il sistema ravvisa la possibilità di concretizzare delle proposte di scambio, lo notifica al configuratore, il quale lo comunica (al di fuori del sistema) a ciascun fruitore coinvolto.

GENERALITÀ

L'applicazione è tesa a supportare lo scambio di ore di prestazioni d'opera relative a diverse attività afferenti ad alcune categorie, dette *categorie foglia*, stabilite dal configuratore.

Il configuratore definisce anche delle *categorie* non foglia, ciascuna delle quali è dotata di un *campo* caratteristico che può assumere un valore appartenente a un dominio finito (non vuoto) di valori discreti. Il *nome* del campo dipende dalla categoria di appartenenza, così come il *dominio* di suddetto campo. A ciascun valore che cade nel dominio di un campo può essere associata una *descrizione*. Una categoria si articola in una o più (sotto)categorie, una per ciascun valore assumibile dal suo campo caratteristico. Ciascuna (sotto)categoria è a sua volta suddividibile ricorsivamente, e così via, secondo una *gerarchia* ad albero. Le categorie foglia dell'albero non sono dotate di alcun campo.

La figura mostra graficamente un esempio di gerarchia, la cui radice è la categoria Lezioni di musica, dotata del campo *tipo*, il cui dominio consiste in {teoria, strumento}. Essa si articola nelle (sotto)categorie Lezioni di teoria musicale, corrispondente al primo valore del campo *tipo*, e Lezioni di uno strumento musicale, corrispondente al secondo valore dello stesso campo. La (sotto)categoria Lezioni di teoria musicale è dotata del campo *materia*, il

cui dominio consiste in {solfeggio, storia della musica}; le sue sottocategorie sono Lezioni di solfeggio, corrispondente al primo valore del campo materia, e Lezioni di storia della musica, corrispondente al secondo valore dello stesso campo. Esse sono entrambe delle categorie foglia.



La categoria Lezioni di chitarra in figura ha una sola sottocategoria, Lezioni di chitarra per principianti. L'aver dotato la categoria Lezioni di

chitarra del campo livello sembra inutile dal momento che esso può assumere un solo valore (principianti). Il campo livello però è stato utilizzato per creare una gerarchia aperta (in qualche misura) a estensioni future. Infatti, in futuro si potrebbero aggiungere nuovi valori al dominio di livello e, quindi, nuove sottocategorie alla categoria Lezioni di chitarra. Al valore principianti del campo livello della categoria Lezioni di chitarra il configuratore può associare una descrizione testuale, come, ad esempio, “sono richieste conoscenze di solfeggio”.

Il nome di ciascuna categoria è unico all'interno della gerarchia di appartenenza. Non si deve erroneamente pensare che una gerarchia definisca ontologicamente un concetto (quello relativo alla radice), declinandolo in tutte le sfumature possibili. La gerarchia comprende invece solo le categorie che sono di interesse ai fini delle operazioni di scambio di prestazioni d'opera che l'organizzazione che si è dotata dell'applicazione intende sostenere.

La medesima applicazione può considerare più gerarchie. Vigge il vincolo di unicità del nome di ciascuna *categoria radice* entro la totalità delle categorie radice di tali gerarchie. Come già accennato, le ore di prestazione d'opera da scambiare devono riferirsi solo alle attività di cui alle categorie foglia delle gerarchie considerate. Per ogni coppia ordinata distinta (c_1, c_2) di categorie foglia diverse fra loro, appartenenti alla stessa gerarchia o a gerarchie diverse, è fissato un *fattore di conversione*. Esso è il fattore per cui

moltiplicare una quantità di ore di prestazione d'opera relative a c_1 per ottenere la corrispondente quantità di ore di prestazione d'opera relative a c_2 . Sussistono i seguenti vincoli:

- per ogni coppia ordinata di categorie foglia, il fattore di conversione assume un valore reale compreso fra 0.5 e 2.0;
- se f_{12} è il valore del fattore di conversione inerente alla coppia ordinata (c_1, c_2) , allora il valore del fattore di conversione inerente alla coppia ordinata (c_2, c_1) , è $1/f_{12}$;
- se f_{12} è il valore del fattore di conversione inerente alla coppia ordinata (c_1, c_2) e f_{23} è il valore del fattore di conversione inerente alla coppia ordinata (c_2, c_3) , allora il valore del fattore di conversione inerente alla coppia ordinata (c_1, c_3) , è il prodotto di f_{12} per f_{23} .

Gli scambi che coinvolgono un fruitore possono avvenire solo entro il *comprensorio* geografico di appartenenza dello stesso (per semplicità, si assume che le prestazioni d'opera non siano erogate a distanza). Ciascun fruitore appartiene a un singolo comprensorio geografico. La *composizione* di ciascun comprensorio considerato dall'applicazione è fissata dal configuratore: un comprensorio è un insieme contenente uno o più comuni limitrofi.

VERSIONE 1 – REQUISITI FUNZIONALI

La prima versione dell'applicazione consente l'accesso del solo configuratore. Ogni configuratore effettua il primo accesso sfruttando credenziali predefinite (comunicate a ciascun nuovo configuratore autorizzato a registrarsi), che lo qualificano come appartenente al gruppo (non vuoto, eventualmente individuale) dei configuratori dell'applicazione. Nell'ambito della sessione aperta al primo accesso, egli dovrà immediatamente scegliere credenziali personali, da usare in tutti gli accessi successivi: solo a valle di tale scelta egli potrà operare sul *back-end* dell'applicazione. Lo *username* di ciascun configuratore lo individua univocamente.

La prima versione dell'applicazione consente al configuratore di

- introdurre un nuovo comprensorio geografico,
- introdurre una nuova gerarchia di categorie,
- dotare ogni categoria non foglia di tale gerarchia di nome, campo caratteristico e dominio dello stesso, nonché di una eventuale descrizione per ciascun valore appartenente a suddetto dominio,

- stabilire il valore dei fattori di conversione relativi a tutte le coppie distinte di categorie foglia, dove tali categorie foglia appartengono entrambe alla gerarchia appena creata oppure una appartiene alla gerarchia appena creata e l'altra appartiene a una gerarchia preesistente;
- salvare in modo persistente tutte le informazioni di cui ai punti precedenti,
- visualizzare ciascun comprensorio geografico presente,
- visualizzare ciascuna gerarchia presente e tutte le informazioni a corredo della stessa,
- visualizzare tutti i fattori di conversione che coinvolgono come prima categoria una categoria foglia assegnata.

Si noti che, l'assegnamento di un valore al fattore di conversione relativo a una coppia di categorie foglia, di cui almeno una appartenente alla gerarchia appena creata, comporta l'assegnamento implicito da parte del sistema di un valore anche ad altri fattori di conversione. Ad esempio, supponiamo che il configuratore abbia assegnato il valore 1 al fattore di conversione inerente alla coppia ordinata di categorie $(c1, c2)$, con $c1$ che appartiene alla gerarchia $G1$ appena creata e $c2$ che appartiene alla gerarchia preesistente $G2$. Dal momento che $G2$ è preesistente, tutti i fattori di conversione da una sua qualsivoglia categoria foglia a qualsiasi altra categoria foglia di $G2$ e delle altre gerarchie preesistenti sono già fissati. Supponiamo che il fattore di conversione relativo alla coppia $(c2, c3)$, dove $c3$ è un'altra categoria foglia di $G2$, sia 0.7. Pertanto, il fattore di

conversione relativo alla coppia $(c1, c3)$ è $1 * 0.7 = 0.7$: questo valore deve essere calcolato dal sistema e assegnato automaticamente al fattore a cui si riferisce. Il sistema deve sollevare il configuratore dall'onere di fissare i valori (dei fattori di conversione) derivabili da altri.

Si noti altresì che, se il configuratore assegna un valore al fattore di conversione inerente a una coppia ordinata, risulta implicitamente assegnato anche il valore del fattore di conversione relativo alla coppia ordinata in senso opposto.

Si noti, infine, che i valori che possono essere derivati da altri dipendono dall'ordine con cui vengono assegnati i valori dei fattori di conversione alle coppie ordinate di categorie di cui almeno una appartiene alla gerarchia appena creata. Supponiamo, ad esempio, che $c4$, $c5$ e $c6$ siano tutte categorie foglia della gerarchia $G1$ appena creata. Se il configuratore assegna prima il valore del fattore di conversione relativo alla coppia $(c4, c5)$ e poi quello relativo alla coppia $(c5, c6)$, sarà derivabile il valore del fattore di conversione relativo alla coppia $(c4, c6)$. Se, invece, il configuratore assegna prima il valore del fattore di conversione relativo alla coppia $(c4, c6)$ e poi quello relativo alla coppia $(c6, c5)$, sarà derivabile il valore del fattore di conversione relativo alla coppia $(c4, c5)$.

VERSIONE 2 – REQUISITI FUNZIONALI

La seconda versione dell'applicazione consente l'accesso anche al fruitore. Al primo accesso il fruitore deve indicare il suo comprensorio di appartenenza, scegliendolo fra quelli disponibili. Se non effettua tale scelta, il fruitore non potrà continuare a interagire con l'applicazione. Sempre nell'ambito del primo accesso, a valle dell'indicazione del comprensorio, ogni fruitore deve fornire le sue credenziali individuali, che gli consentiranno di operare sul *front-end* dell'applicazione. Lo *username* di ciascun fruitore individua univocamente il fruitore stesso (esso non deve coincidere con lo username di alcun altro fruitore né di alcun configuratore). Inoltre, al primo accesso il fruitore deve fornire un suo *indirizzo* di posta elettronica, che deve essere salvato.

La seconda versione dell'applicazione dà la facoltà al fruitore di visualizzare le categorie radice delle diverse gerarchie di categorie e di navigare entro ciascuna gerarchia, scendendo dalla categoria radice della stessa a una categoria foglia grazie all'impostazione progressiva dei valori dei campi delle categorie attraversate.

VERSIONE 3 – REQUISITI FUNZIONALI

La terza versione consente al fruitore di formulare delle proposte di scambio di prestazioni d'opera. Ogni *proposta* di scambio si articola in due parti, *richiesta* e *offerta*, dove la prima si riferisce alla prestazione d'opera di cui il proponente necessita mentre la seconda si riferisce alla prestazione d'opera che il proponente offre in cambio. Ciascuna delle due parti specifica la categoria foglia di appartenenza. Per quanto riguarda la richiesta, il proponente deve indicare la *durata* della prestazione d'opera desiderata, espressa in un numero intero di ore. La durata relativa all'offerta viene invece calcolata dal sistema; essa è pari al numero di ore indicate nella richiesta moltiplicato per il fattore di conversione relativo (approssimando il risultato all'intero più vicino). A valle della visualizzazione della durata dell'offerta, il fruitore può confermare (o meno) l'offerta compilata.

Assumiamo, ad esempio, che l'applicazione, oltre alla gerarchia già presentata in figura, disponga di una gerarchia relativa ad attività didattiche di ripetizione, una categoria foglia della quale è Ripetizioni di matematica per scuola superiore. Un fruitore A può compilare la seguente proposta:

<pre>richiesta: [Lezioni di pianoforte per principianti, 10 ore] offerta: [Ripetizioni di matematica per scuola superiore, 10 ore]</pre>
--

La durata dell'offerta è evidenziata in azzurro perché essa è stata calcolata dall'applicazione. Assumendo che il fattore di conversione fra le categorie foglia Lezioni di pianoforte per principianti e Ripetizioni di matematica per scuola superiore sia pari a 1, il sistema ha stabilito che il numero di ore relative all'offerta è pari a 10 e lo ha mostrato al fruitore. Se il fruitore conferma l'offerta compilata sopra esemplificata, egli accetta di usufruire di 10 ore di lezioni di pianoforte per principianti in cambio della fornitura di un identico numero di ore di ripetizione di argomenti di matematica trattati nella scuola superiore.

Una proposta, una volta che è stata confermata dal suo autore, si trova nello stato di proposta *aperta*. Ogni proposta aperta viene salvata in forma persistente, associandola al fruitore che l'ha formulata.

VERSIONE 4 – REQUISITI FUNZIONALI

La quarta versione dell'applicazione determina quali scambi possano essere realizzati. Si ricorda che gli scambi da attuare possono coinvolgere solo fruitori appartenenti tutti al medesimo comprensorio.

Non appena una proposta diventa aperta, il sistema valuta se essa possa essere immediatamente soddisfatta.

Continuando l'esempio di cui ai requisiti della terza versione, se il sistema ha precedentemente archiviato la seguente proposta aperta associata al fruitore B, appartenente allo stesso comprensorio del fruitore A,

<pre>richiesta: [Ripetizioni di matematica per scuola superiore, 10 ore] offerta: [Lezioni di pianoforte per principianti, 10 ore]</pre>
--

allora la proposta di A potrà essere soddisfatta, soddisfacendo nel contempo quella di B.

Quando il sistema appura che una proposta aperta può essere soddisfatta, soddisfacendo nel contempo almeno una seconda proposta aperta, esso cambierà lo stato delle proposte aperte coinvolte, trasformando ciascuna di esse in una proposta *chiusa*. Tali proposte

formano un cosiddetto *insieme chiuso*. Il sistema comunicherà al configuratore tutti gli estremi delle proposte appartenenti all'insieme chiuso e del fruitore associato a ciascuna di esse (compreso l'indirizzo di posta elettronica di quest'ultimo). Il configuratore provvederà, al di fuori del sistema, a inviare un messaggio di posta elettronica ai fruitori coinvolti, cosicché essi possano accordarsi per gli scambi, agendo sempre al di fuori del sistema.

Si noti che il soddisfacimento di una (nuova) proposta aperta può avvenire grazie all'apporto non necessariamente di una singola proposta aperta preesistente ma anche di più proposte aperte (preesistenti).

Supponiamo, ad esempio, che, oltre alle due gerarchie di categorie di prestazioni d'opera già citate, l'applicazione consideri una terza gerarchia, relativa ad attività di collaborazione domestica, una categoria foglia della quale è *Stiratura*. Supponiamo che la nuova proposta aperta, formulata dal fruitore H, sia la seguente:

<pre>richiesta: [Stiratura, 15 ore] offerta: [Ripetizioni di matematica per scuola superiore, 10 ore]</pre>
--

Per questa proposta, si è ipotizzato che il sistema calcoli una durata dell'offerta pari a 10 ore poiché il valore del fattore di conversione per la coppia (Ripetizioni di matematica per scuola superiore, Stiratura) è 1.5. Supponiamo altresì che esistano le seguenti due proposte aperte preesistenti, formulate rispettivamente dai fruitori J e K, appartenenti entrambi allo stesso comprensorio del fruitore H:

richiesta: [Ripetizioni di matematica per scuola superiore, 10 ore]
offerta: [Lezioni di pianoforte per principianti, 10 ore]

richiesta: [Lezioni di pianoforte per principianti, 10 ore]
offerta: [Stiratura, 15 ore]

La durata dell'offerta calcolata per la seconda proposta è di 15 ore, poiché (in base ai valori dei fattori di conversione già impostati sopra) il fattore di conversione relativo a (Lezioni di pianoforte per principianti, Stiratura) ha valore pari a 1,5.

Il sistema appura che la nuova proposta può essere soddisfatta in questo modo: H fornisce 10 ore di ripetizioni di matematica per scuola superiore a J, il quale fornisce 10 ore di lezioni di pianoforte per principianti a K, il quale fornisce 15 ore di stiratura a H.

Una proposta rimane aperta fino a quando non viene chiusa dal sistema o non viene *ritirata* dal fruitore che l'aveva formulata. L'applicazione deve mantenere traccia dei passaggi di stato subiti da una proposta, salvando le informazioni relative a tali passaggi in modo persistente.

In ogni momento, il configuratore può indicare una categoria (foglia) di una qualsiasi gerarchia e visualizzare tutte le Proposte aperte, le Proposte chiuse e le Proposte ritirate relative (nella richiesta o nell'offerta) a tale categoria.

Il fruitore può invece visualizzare tutte le Proposte aperte, le Proposte chiuse e le Proposte ritirate di cui è autore, indipendentemente dalla categoria di appartenenza.

REQUISITI NON FUNZIONALI

Il modello di processo da adottare è incrementale/iterativo. Ogni versione da produrre estende le funzionalità della precedente, senza modificarle.

Il linguaggio di programmazione da utilizzare è Java.

L'architettura esterna da realizzare per l'applicazione è stand alone.

Requisito non prescrittivo ma importante in sede di valutazione è l'impiego di precondizioni, postcondizioni e invarianti di classe entro il codice Java.

Non è richiesta la creazione di una interfaccia utente grafica (tuttavia è bene che l'architettura interna sia progettata in modo da ridurre gli effetti collaterali e lo sforzo connesso al cambiamento se in futuro il sistema di interazione testuale fosse sostituito da una GUI).

Non è richiesto l'impiego di alcun DBMS (Data Base Management System).

NOTE

1. Nei requisiti sopra enunciati si sono ignorati i possibili aspetti legali connessi alla gestione di un'applicazione volta a supportare lo scambio di prestazioni d'opera, all'uso della stessa e agli scambi veri e propri. È evidente che ogni applicazione che operi nel mondo reale deve rispettare la legislazione del Paese in cui viene utilizzata. Tale legislazione potrebbe imporre, per esempio, che ciascun fruitore fornisca le sue generalità e/o che dichiari di essere maggiorenne, ecc. Inoltre, dal momento che ogni fruitore mette a disposizione del configuratore un proprio indirizzo email, che verrà conservato dall'applicazione, la legislazione potrebbe richiedere che il fruitore esprima un consenso al trattamento dei dati e che il configuratore si impegni al rispetto delle leggi sulla privacy. Secondo le specifiche delle pagine precedenti, un fruitore offre dei servizi senza che venga esercitata alcuna forma di controllo circa le sue effettive abilità. La ragione per cui le questioni legali e i controlli menzionati sono stati ignorati è che l'applicazione che sarà realizzata è da intendere come il frutto di un esercizio, necessariamente di dimensioni limitate, e non è destinata a essere effettivamente installata e usata nella realtà quotidiana, come sottolineato dal requisito di adottare un'architettura esterna stand alone (mentre una applicazione usata nel mondo reale dovrebbe operare in rete). Per la stessa ragione non si è mai parlato del ritorno atteso da parte dell'organizzazione che, attraverso l'applicazione, sovrintende alle operazioni di

scambio di prestazioni d'opera. Tale ritorno esula completamente dagli scopi didattici del progetto.

2. Nel mondo reale il requisito di sicurezza gioca un ruolo di rilievo. Tale requisito, sempre più importante in un lavoro professionale, non è stato considerato nei limiti dell'esercizio proposto né il gruppo di lavoro deve tenerne conto. L'enfasi del progetto non è infatti sulla sicurezza, pertanto anche l'attribuzione di credenziali a configuratore e fruitore non è ritenuta un'operazione critica.

L'interpretazione relativa alle “credenziali predefinite” comunicate a ciascun nuovo configuratore che intende registrarsi, di cui ai requisiti funzionali della prima versione, non è univoca. Secondo l'interpretazione più semplice – e meno sicura – tali credenziali sono uniche e immutabili, stabilite al momento dell'installazione dell'applicazione. Un'interpretazione più complessa è quella secondo cui le credenziali predefinite sono diverse per ciascun configuratore che intende registrarsi. In questo caso, al primo configuratore che intende registrarsi vengono comunicate credenziali predefinite stabilite al momento dell'installazione dell'applicazione mentre a ciascuno di quelli successivi vengono comunicate nuove credenziali, fissate di volta in volta da un configuratore già registrato. Questa soluzione è più sicura ma non ne è richiesta la realizzazione.

3. La funzionalità più complessa dell'applicazione da sviluppare è quella svolta dalla quarta versione, che, ogni volta che viene aperta una nuova proposta, verifica se la stessa possa essere soddisfatta e in che modo. In un'applicazione professionale, l'algoritmo che svolge questo compito è importante. Tuttavia, ai fini dell'elaborato, la scelta di tale algoritmo non è ritenuta critica. Per gli obiettivi didattici dell'insegnamento è infatti rilevante effettuare una buona progettazione dell'applicazione, non già trovare algoritmi più efficienti. Se il progetto è di buona qualità, l'algoritmo che verrà codificato potrà infatti in ogni momento essere sostituito con un altro migliore senza dovere cambiare nessuna altra parte del codice. Sempre in considerazione del fatto che la scelta dell'algoritmo di cui sopra non è ritenuta critica, nelle pagine precedenti non si è sottolineato che, data una proposta appena aperta, possono esistere più insiemi chiusi atti a soddisfarla. Pertanto, un'applicazione professionale dovrebbe effettuare una scelta fra le varie soluzioni possibili (ad esempio, potrebbe scegliere l'insieme chiuso di cardinalità maggiore). Questi aspetti non sono tuttavia reputati interessanti dal punto di vista dell'esercizio proposto; pertanto, l'algoritmo che sarà codificato può terminare la sua esecuzione dopo avere trovato una soluzione (se esiste), senza ricercarne eventuali altre.

4. L'algoritmo che, ogni volta che viene aperta una nuova proposta, verifica se la stessa possa essere soddisfatta, trova soluzioni (insiemi chiusi) in cui, per ogni proposta soddisfatta, vengono fornite e ottenute esattamente lo stesso numero di ore indicate rispettivamente nell'offerta e nella richiesta della proposta stessa. Nel caso non esistano soluzioni esatte, una versione più realistica dell'applicazione potrebbe cercare anche soluzioni che non garantiscono una identità perfetta col numero di ore indicate in ciascuna proposta soddisfatta ma che garantiscono che il numero di ore di prestazioni d'opera erogate e ottenute rientrano entro delle tolleranze prefissate rispetto a quelle indicate.
5. I requisiti non funzionali non impongono alcuna tecnologia da utilizzare per la memorizzazione persistente dei dati (credenziali, indirizzi di posta elettronica, categorie, compensori, proposte, ciascuna associata al fruitore autore della stessa, e relativi passaggi di stato). La scelta dei progettisti potrebbe cadere banalmente sulla serializzazione di oggetti. In alternativa si potrebbe ricorrere all'impiego di file di testo aventi una sintassi definita appositamente dai progettisti stessi (ad esempio, file in formato CSV o JSON o XML ...). Si noti che tali file potrebbero essere adottati anche per importare gli ingressi dell'applicazione in modalità batch, in aggiunta alla modalità interattiva già descritta nei requisiti.

6. In una applicazione reale ogni proposta aperta dovrebbe avere un tempo massimo di latenza, trascorso il quale essa viene ritirata automaticamente, senza l'intervento del fruitore che l'ha avanzata. I requisiti espressi delle pagine precedenti non pretendono tale realizzazione.
7. Secondo le pagine precedenti, è il configuratore che comunica, mediante messaggi email, la possibilità di attuare uno scambio ai fruitori autori della proposte appartenenti a un insieme chiuso determinato dall'applicazione. In un'applicazione più avanzata, tali messaggi email potrebbero essere generati automaticamente ed eventualmente essere direttamente inviati ai destinatari, senza richiedere alcun intervento da parte del configuratore.

8. Nella sezione GENERALITÀ si è accennato al fatto che le gerarchie di categorie create dal configuratore possono essere, in una certa misura, aperte a cambiamenti futuri. Per quanto riguarda le quattro versioni previste ai fini dell'elaborato non sono però stati enunciati requisiti relativi a tali cambiamenti. In tutte le versioni si assume implicitamente che ciascuna gerarchia di categorie, una volta introdotta dal configuratore, sia immutabile (e pertanto è molto importante che tale gerarchia sia stata congegnata con attenzione). In futuro si potrebbe permettere l'estensione di una gerarchia attraverso l'aggiunta di sottoalberi, come descritto nella sezione GENERALITÀ. Altri tipi di modifiche comporterebbero effetti collaterali; ad esempio, la trasformazione di una categoria foglia in una categoria intermedia (mediante l'aggiunta alla stessa di un campo caratteristico) richiede una gestione particolare delle proposte non ancora chiuse né ritirate che cadono in tale categoria.
9. I requisiti (funzionali e non) delle quattro versioni dell'applicazione da realizzare sono deliberatamente espressi a un alto livello di astrazione (ad esempio, non si è stabilita la lunghezza massima, in termini di numero di caratteri, delle stringhe che rappresentano il nome di una categoria o la descrizione del valore di un campo) al fine di consentire agli ingegneri del software di fornire un'interpretazione personale, che comporta sempre l'aggiunta di ulteriori requisiti. Tali aggiunte devono essere chiaramente documentate.

ESTENSIONI FUTURE

Si elencano di seguito alcuni possibili punti di estensione o modifica dell'applicazione, non affinché i requisiti a essi relativi siano soddisfatti ma perché anticipare i cambiamenti è un importante principio di progettazione (teso a rendere l'applicazione insensibile agli stessi).

- Il sistema di interazione potrebbe divenire grafico.
- L'architettura esterna potrebbe diventare distribuita (il back-end sarebbe installato sul server, mentre il front-end sarebbe installato sul client).
- L'archivio dei dati potrebbe essere gestito attraverso un DBMS.
- Si potrebbe consentire il soddisfacimento in forma approssimata (anziché esatta) delle proposte aperte, ammettendo delle tolleranze rispetto alle durate indicate nelle stesse (vedi NOTA 4).
- Gli ingressi del back-end dell'applicazione, ovvero gerarchie delle categorie e composizione dei compensori, e/o gli ingressi del front-end, ovvero le proposte avanzate da ciascun fruitore, potrebbero essere importate in modalità batch (cioè, attraverso uno o più file di input), in aggiunta alla modalità interattiva (vedi NOTA 5).
- Si potrebbe ritirare automaticamente ciascuna proposta aperta allo scadere di un tempo di latenza prefissato (vedi NOTA 6).

- Si potrebbero generare (e inviare) automaticamente i messaggi di posta elettronica destinati a tutti i fruitori coinvolti nelle proposte appartenenti a un insieme chiuso (vedi NOTA 7).
- Si potrebbe abilitare l'estensione delle gerarchie di categorie presenti (vedi NOTA 8).
- L'utilizzo di un DBMS agevolerebbe il configuratore nell'effettuare analisi relative agli attuali scambi potenziali o a quelli già avvenuti. Ad esempio, il configuratore potrebbe ottenere risposte alle seguenti domande (e/o ad altre simili):
 - Quante sono le attuali Proposte aperte/Proposte chiuse/Proposte ritirate?
 - Quali sono le attuali Proposte aperte/Proposte chiuse/Proposte ritirate di cui è autore un certo fruitore (individuato attraverso il suo username)?
 - Quale categoria foglia compare il maggior numero di volte nell'offerta entro le Proposte aperte?
 - Quale categoria foglia compare il maggior numero di volte nella richiesta entro le Proposte aperte?
 - Qual è globalmente il numero massimo e medio di Proposte aperte/Proposte chiuse/Proposte ritirate per fruitore?

Richieste relative alla PRIMA PARTE DEL PROGETTO

Agli studenti è richiesto di realizzare evolutivamente quattro versioni software che soddisfino i requisiti sopra esposti. Ogni gruppo (costituito al più da tre persone), dovrà:

- 1) per ogni versione, produrre la documentazione di progetto, contenente
 - casi d'uso (comprensivi dell'espressione di eventuali requisiti aggiuntivi), sia in forma testuale, sia in forma di diagramma UML; si invita a rendere evidenti a colpo l'occhio le integrazioni/modifiche apportate ai casi d'uso (testuali e grafici) della versione precedente per ottenere quelli della versione corrente;
 - diagramma UML delle classi,
 - diagrammi UML comportamentali (opzionali),e qualsiasi altra specifica ritenuta opportuna;
- 2) per l'ultima versione, redigere un unico manuale di installazione e uso (il cui contenuto potrebbe eventualmente divenire parte dell'help in linea dell'applicazione);
- 3) raccogliere in un unico file la documentazione, relativa alle quattro versioni, di cui al punto 1 e il manuale di cui al punto 2; consegnare tale file in formato elettronico;
- 4) per ogni versione, consegnare codice sorgente + codice interpretabile + (preferibilmente) codice eseguibile;

5) preparare una dimostrazione del funzionamento dell'applicazione, che avverrà durante la prova orale. Al fine di evitare lunghi tempi di digitazione degli ingressi durante la dimostrazione, sfruttare il salvataggio in forma persistente dei dati che dovranno essere letti (cioè, recuperati dalla memoria di massa) e usati dall'applicazione durante la sessione dimostrativa. In particolare, salvare:

- le credenziali di almeno 2 configuratori;
- le credenziali (e gli indirizzi di posta elettronica) di almeno 3 fruitori;
- le composizioni di almeno 2 comprensori;
- le descrizioni di almeno 3 gerarchie di categorie, aventi globalmente almeno 6 categorie foglia, nonché il valore dei fattori di conversione relativi alle stesse;
- le informazioni inerenti ad almeno 10 proposte aperte, 3 proposte chiuse e 2 proposte ritirate.

MODALITÀ DI CONSEGNA E ALTRO

Come riportato dalla guida in linea relativa all'insegnamento, ai fini del superamento dell'esame, ogni studente deve sostenere **una sola prova orale** (discussione di entrambe le parti dell'elaborato in un'unica data).

Il materiale relativo alla PRIMA PARTE del progetto, di cui ai precedenti punti 3 e 4, deve essere consegnato attraverso la **piattaforma Moodle**, in anticipo rispetto al materiale relativo alla SECONDA PARTE e rispetto alla prova orale. Più precisamente, il materiale relativo alla PRIMA PARTE deve essere consegnato (al più tardi) entro la data in cui è fissata la prova orale dell'appello precedente rispetto a quello in cui si intende sostenere la prova orale. Le date delle prove orali sono indicate nella *pagina di appelli e prove parziali del portale di Ateneo*. La consegna del materiale relativo alla PRIMA PARTE dell'elaborato deve avvenire senza alcuna iscrizione ad alcuna prova da parte dei componenti del gruppo.

La consegna (attraverso la piattaforma Moodle) della seconda parte dell'elaborato deve invece essere accompagnata dall'iscrizione su ESSE3 alla prova orale (la prova è corredata da una nota che la indica come “Discussione dell'elaborato”) da parte di tutti i componenti del gruppo.

Per la prima come per la seconda parte, si raccomanda di riportare la composizione del gruppo (comprensiva del numero di matricola di ciascun componente) entro il materiale consegnato (in particolare, per quanto riguarda la PRIMA PARTE, entro il file di cui al precedente punto 3) e possibilmente anche in una nota Moodle, compilabile al momento del caricamento di suddetto materiale. La consegna deve essere effettuata da un solo componente, per conto dell'intero gruppo.

Al momento della discussione della PRIMA PARTE del progetto, il gruppo dovrà rispondere a domande relative alla documentazione prodotta ed effettuare una dimostrazione del funzionamento dell'applicazione sviluppata (si veda il precedente punto 5). A tal fine è bene che ogni gruppo si presenti alla prova orale munito di un portatile, da connettere al sistema di proiezione disponibile in aula. Si raccomanda di non dimenticare il cavo di alimentazione; se il portatile non dispone di un'uscita HDMI, serve anche un adattatore a femmina HDMI, da collegare a un'uscita USB del portatile.

Si rammenta che il progetto corrente, assegnato nell'a.a. 2023-2024, potrà essere discusso (in ciascuna delle sue due parti) solo **fino alla sessione d'esame di Giugno – Luglio 2025**. Gli studenti che non avessero effettuato la discussione entro tale sessione dovranno svolgere il progetto assegnato nell'anno accademico successivo.