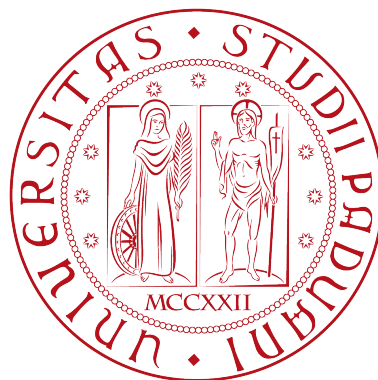


TESI DI LAUREA IN INFORMATICA

NICOLA MORETTO



Analisi e progettazione di un'interfaccia grafica per la  
consultazione dei contenuti informativi in una piattaforma  
web tematica

30 novembre 2012

*Tesi di Laurea in Informatica*

Analisi e progettazione di un'interfaccia grafica per la consultazione  
dei contenuti informativi in una piattaforma web tematica

© 30 novembre 2012

STUDENTE:

Nicola Moretto (matr. 578258)

RELATORE:

Prof. Tullio Vardanega

LUOGO:

Padova

DATA:

30 novembre 2012

*Ad Alessandro, Christian, Ester, Marta e Romeo con cui tutto è iniziato.*

*Al dott. Bovo per la passione e l'entusiasmo.*

*Al prof. Vardanega per la cortesia e la disponibilità.*

*Agli amici di sempre.*

*Alla mia famiglia per aver sempre creduto in me.*

## SOMMARIO

---

L'attività di stage si è svolta presso l'azienda *Sintesi Sas*, che opera nel settore ICT (*Information and Communication Technology*) realizzando software ERP e piattaforme Web per aziende, in particolare attive nel settore turistico, e fornendo servizi di consulenza e di formazione di imprenditori nell'ambito del marketing strategico, operativo e del controllo di gestione.

Il prodotto di punta dell'azienda - *Planet Hotel* - costituisce un sistema software per la gestione alberghiera tra i più flessibili, ampi e completi presenti nel panorama italiano, in grado di coprire la maggior parte delle necessità aziendali: oltre alla gestione delle prenotazioni e dei conti, esso offre un insieme di moduli integrati per supportare il controllo di gestione e degli interventi di marketing.

Si tratta di una realtà imprenditoriale a clientela nazionale con sede unica a Mestre (VE), la cui direzione e amministrazione è affidata al solo fondatore, che ha assunto il ruolo di tutor esterno e referente aziendale per l'intera durata dello stage.

Le attività svolte si inseriscono nell'ambito di un progetto esterno rispetto al business dell'azienda, finalizzato alla realizzazione di una piattaforma web tematica per la condivisione di informazioni e la vendita diretta di prodotti alla clientela e affidato ad un team costituito da differenti figure professionali (sociologi, informatici, ingegneri, ...).

## CONTENUTI

Il presente documento costituisce una relazione dettagliata in merito all'attività di stage svolta dallo studente Nicola Moretto presso l'azienda *Sintesi Sas*. I contenuti sono organizzati nei seguenti capitoli:

### PROGETTO

Il primo capitolo illustra le strategie dell'azienda e gli obiettivi, i requisiti e i vincoli del progetto in cui si inseriscono le attività di stage.

### STAGE

Il secondo capitolo illustra gli obiettivi, i requisiti e l'organizzazione (piano e norme di lavoro) delle attività di stage. A seguire vengono presentate le scelte più rilevanti effettuate e i risultati conseguiti.

## VALUTAZIONI FINALI

Il terzo capitolo presenta un'analisi critica a posteriori dell'attività di stage: raggiungimento degli obiettivi prefissati, competenze professionali acquisite, ... .

## CONVENZIONI TIPOGRAFICHE

Al fine di agevolare la consultazione del documento sono state adottate alcune convenzioni tipografiche illustrate di seguito.

**GLOSSARIO** Gli acronimi, le abbreviazioni, i nomi propri e i termini specialistici contenuti nel presente documento sono illustrati nel **GLOSSARIO**, consultabile in appendice, al fine di agevolare la lettura e la comprensione degli argomenti trattati. La prima occorrenza di ciascun termine o espressione presente nel glossario appare sottolineata.

**TERMINOLOGIA** I termini propri o di provenienza straniera divenuti di uso corrente nella lingua italiana sono evidenziati in *corsivo*, mentre la prima occorrenza di parole o espressioni che assumono particolare significato nel presente contesto è riportata in **MAIUSCOLETTO**.

**CODICE E FORMULE** I nomi di tabelle, classi, package, ... impiegano uno stile di carattere sans serif, mentre i frammenti di codice impiegano un carattere a spaziatura fissa.

## INDICE

---

1	PROGETTO	9
1.1	Genesi	9
1.2	Reti sociali	9
1.3	Architettura	10
1.3.1	Contenuti informativi	11
1.3.2	Sistema di classificazione	14
1.4	Requisiti e vincoli	14
1.4.1	Criterio di classificazione	15
1.4.2	Interfaccia grafica	15
2	STAGE	17
2.1	Piano di stage	17
2.1.1	Obiettivi e requisiti	17
2.1.2	Pianificazione	19
2.2	Norme di stage	19
2.3	Criterio di classificazione	22
2.3.1	Enciclopedia del sapere	23
2.3.2	Ambiguità sintattica	24
2.3.3	Ambiguità semantica	27
2.3.4	Modello relazionale	29
2.4	Filtri di ricerca	31
2.4.1	Introduzione	31
2.4.2	Parametri	32
2.4.3	Classi di filtri	33
2.4.4	Progettazione	34
2.5	Risultati di ricerca	38
2.5.1	Visualizzazione	38
2.5.2	Aggiornamento	38
2.5.3	Interazione	39
2.6	Navigazione dei contenuti	40
2.6.1	Contenuti informativi	40
2.6.2	Cronologia	43
2.6.3	Raggruppamento	43
2.6.4	Ordinamento	44
2.6.5	Navigazione	45
2.7	Risultati	45
3	VALUTAZIONI FINALI	48
3.1	Obiettivi	48
3.2	Competenze professionali	50
3.3	Stage e università	51
A	GLOSSARIO	53
B	CRITERIO DI CLASSIFICAZIONE	56

## INDICE

C	INTERFACCIA GRAFICA	60
---	---------------------	----

## ELENCO DELLE FIGURE

---

Figura 1	Gerarchia degli utenti	11
Figura 2	Diagramma di Gantt	20
Figura 3	Valore informativo di un contenuto	22
Figura 4	Relazioni tra le entità del dominio	24
Figura 5	Ambiguità sintattica di un'entità	25
Figura 6	Etichetta primaria e secondarie	27
Figura 7	Accezioni di un'etichetta	27
Figura 8	Pagina di disambiguazione [2]	28
Figura 9	Accezioni chiave e sinonimiche	28
Figura 10	Modello relazionale	29
Figura 11	Modello ad oggetti delle accezioni	30
Figura 12	Diagramma delle classi per il package <i>model.filter</i>	35
Figura 13	Diagramma delle classi per il package <i>model.criteria</i>	36
Figura 14	Diagramma delle classi per il package <i>view.filter</i>	37
Figura 15	Asse temporale	45
Figura 16	UC.1 - Ricerca di contenuti informativi	61
Figura 17	UC.2 - Raffinamento dei criteri di ricerca	62
Figura 18	UC.3 - Consultazione dei risultati di ricerca	63
Figura 19	UC.4 - Gestione dei filtri utente	63

## ELENCO DELLE TABELLE

---

Tabella 1	Pianificazione settimanale delle attività	19
Tabella 2	Configurazione dell'ambiente di lavoro	20
Tabella 3	Lista dei parametri per i filtri di ricerca	33
Tabella 4	Elenco dei parametri per i filtri di ricerca, suddivisi per tipo.	34
Tabella 5	Riepilogo delle classi gestite dal <i>design pattern Abstract Factory</i>	37
Tabella 6	Proprietà di un contenuto informativo	41



## PROGETTO

---

### 1.1 GENESI

L'idea della piattaforma *Social (Life) Shuttle* nasce nel 2010 da un progetto concepito per dar vita ad una comunità virtuale destinata agli artisti sconosciuti e accessibile in mobilità mediante un'applicazione dedicata, *ArtYR*.

Nello stesso periodo una consulenza nell'ambito dei sistemi informativi territoriali ad un'azienda di Bolzano conduce allo sviluppo di un'innovativa piattaforma software: un sistema informativo territoriale in cui l'erogazione di informazioni turistiche è integrata con la vendita di servizi collegati.

Il progetto evolve - grazie alla partecipazione di Comuni, Province e Regioni - in una rete tematica di agenzie di viaggio con un'identità comune e finalizzata alla fusione dei sistemi informativi distrettuali e di vendita.

L'architettura di *Social (Life) Shuttle* trae profonda ispirazione, integrando tre componenti differenti:

#### BUSINESS

Vendita diretta di prodotti alla clientela.

#### SOCIALE

Creazione e sviluppo delle relazioni sociali attraverso la condivisione di informazioni e conoscenza.

#### TERRITORIO

Sistema di erogazione di informazioni turistiche e territoriali.

### 1.2 RETI SOCIALI

Il modello sociologico di rete sociale non ha attualmente riscontro presso le piattaforme web di condivisione dei contenuti (*blog, forum, ...*) o i *social network* (*Facebook, Twitter, ...*), che si limitano a considerare e concretizzarne singoli aspetti.

Nelle moderne reti sociali è infatti assente l'incentivo alla condivisione e distribuzione della conoscenza, fattore cruciale per l'aggregazione fisica dei membri delle comunità, da intendersi a sua volta

come aggregazioni formatesi intorno ed attraverso la manifestazione di interesse nei confronti di uno specifico tema di dialogo o discussione, che attraversa la sfera individuale, intima e personale dei suoi membri.

Il progetto *Social (Life) Shuttle* rappresenta una nuova generazione di piattaforma di socializzazione, in cui il web diventa solamente un canale di condivisione e un serbatoio della conoscenza generata dalla dialettica tra persone e dove vengono integrati i canoni classici di *blog*, *forum*, *social network* e *media*.

Una relazione sociale nata e costruita su un interesse comune stravolge l'attuale paradigma delle reti sociali virtuali, in cui il legame nasce a prescindere dalla presenza di interessi comuni o informazioni da condividere, e favorisce l'incontro tra persone aventi esperienze simili frutto di tali interessi condivisi.

Ove l'esperienza riguarda anche beni o prodotti, la componente *business* intende offrire ai membri la possibilità di interagire con i produttori, anch'essi attori della comunità.

L'architettura di *Social (Life) Shuttle* consente di declinare la piattaforma in innumerevoli varianti, applicabili ai temi più svariati: al momento sono in fase di sperimentazione per il mondo del vino, il cibo biologico, l'arte commercializzabile e l'attività di ricerca e progettazione collaborativa.

### 1.3 ARCHITETTURA

Tale piattaforma presenta numerose aspetti che la differenziano dalla concorrenza attuale:

- profonda integrazione degli aspetti *social* e *business*;
- nessuna distinzione tra creatori e fruitori dei contenuti (ciascun membro può condividere le proprie esperienze, segnalare eventi, pubblicare articoli critici, ...);
- l'autorevolezza di ciascun membro della comunità si rafforza o si indebolisce a seconda della qualità dei contenuti pubblicati, dei giudizi degli altri membri e di altri criteri di valutazione;
- lo sfruttamento di tecnologie e dispositivi mobili per favorire la crescita di relazioni al di fuori dell'ambito virtuale della piattaforma (partecipazione ad eventi, raccolta e condivisione di informazioni geolocalizzate, ...).

Per quanto concerne le attività di stage, due aspetti della piattaforma assumono particolare rilevanza: i contenuti informativi e i relativi criteri di classificazione.

### 1.3 ARCHITETTURA

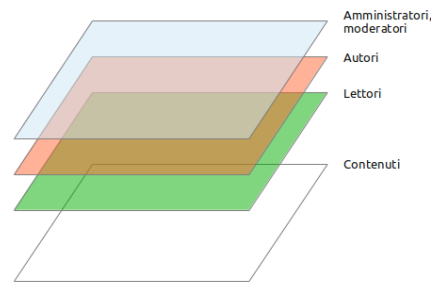


Figura 1: Gerarchia degli utenti nelle piattaforme web tradizionali

#### 1.3.1 *Contenuti informativi*

I contenuti informativi rappresentano lo strumento essenziale per la condivisione delle esperienze e della conoscenza intorno al tema specifico della piattaforma.

Per individuare le classi di contenuti adatte a esprimere in una forma strutturata le informazioni si è tratta ispirazione dalle forme espressive e comunicative tipiche della dialettica quotidiana, poiché immediatamente e intuitivamente comprensibili agli utenti, a prescindere dal loro livello di esperienza.

In particolare, si distinguono la natura della comunicazione, connessa allo scopo e al tono con cui ci esprimiamo, e il formato delle informazioni, che dipendono strettamente dai sensi e dai canali di comunicazione a disposizione per scambiare informazioni con l'interlocutore, sia esso un individuo singolo o un gruppo.

**CLASSI** I tipi di contenuto pubblicabili nella piattaforma dovrebbero essere in numero adeguato a coprire il maggior numero possibile di esigenze comunicative pur rimanendo facilmente e intuitivamente distinguibili, ossia l'utente non dovrebbe nutrire dubbi circa il più adatto a formalizzare di volta in volta l'informazione che desidera condividere.

#### **DOMANDA**

La domanda classica rende particolarmente esplicito lo scopo della comunicazione, ossia la richiesta di informazioni di varia natura agli altri utenti della piattaforma. Si distingue in pubblica o privata, a seconda che l'utente desideri rivolgerla ad un particolare sottoinsieme di utenti.

#### **RISPOSTA**

Duale della domanda, la risposta è anch'essa in forma pubblica o privata per consentire all'utente di renderla accessibile e consultabile solo a certi utenti, spesso l'autore della domanda a cui risponde.

### PENSIERINO

Il pensierino rappresenta una forma di comunicazione adatta ad esprimere un contenuto di lunghezza breve e prevalentemente superficiale (considerazioni, stati d'animo, freddure, ...).

### EVENTO

L'evento aiuta a promuovere qualsiasi iniziativa che rientri nell'ambito tematico della piattaforma e cui possano prender parte altre persone (incontro pubblico, concerto, fiera, ...).

### DISCORSO

Il discorso identifica un contenuto articolato, sia nella forma sia nei contenuti, destinato alla condivisione di informazioni dettagliate e approfondite.

### RECENSIONE

La recensione esprime un giudizio critico nei confronti di un prodotto specifico.

### COMUNICAZIONE PRIVATA

La comunicazione privata è l'unica forma di contatto diretto e riservato tra due utenti.

**ELEMENTI** Ciascun tipo di contenuto esprime un intento comunicativo ben preciso, ma non è vincolato ad una struttura e ad un formato predefiniti: la classe, che esprime l'intento della comunicazione, si colloca in un piano distinto rispetto al formato, ossia la struttura e le caratteristiche specifiche del contenuto informativo condiviso.

Ove la dialettica quotidiana dispone infatti di cinque sensi e può esprimersi in forma non solo verbale, nel web gli utenti sperimentano differenti forme di comunicazione: contenuti testuali e grafici, flussi audio e video, documenti elettronici, messaggistica istantanea, ... .

I contenuti informativi non presentano dunque una struttura fissa a seconda della classe, ma possono essere liberamente redatti a partire da una serie di elementi predefiniti, frutto di una ricerca tra le principali e più diffuse piattaforme web disponibili (*blog, forum, social network, chat, ...*) e di una successiva analisi e rielaborazione dei risultati ottenuti:

### AUDIO

Contenuto audio statico o in tempo reale (*live streaming, ...*).

### IMMAGINI

Contenuto grafico statico.

### VIDEO

Contenuto video statico o in tempo reale (*live streaming, ...*).

#### SONDAGGIO

Domanda a risposta multipla.

#### DOCUMENTO

File di testo o binario caricato nella piattaforma.

#### STRINGA

Contenuto testuale avanzato (intestazioni, formattazione dei caratteri, collegamenti ipertestuali, ...).

#### CITAZIONE

Citazioni o riferimenti ad altri elementi di un contenuto, ad un contenuto informativo o a prodotti presenti nella piattaforma.

La struttura modulare dei contenuti informativi consente di riusare, riferire o citare gli elementi costituenti e di catalogarli con maggior facilità e precisione, riuscendo a classificare ciascun frammento di informazione presente al loro interno.

**PROPRIETÀ** I contenuti informativi - a prescindere dalla classe e dalla struttura - presentano un insieme di proprietà comuni, alcune delle quali assumono particolare rilevanza per l'attività di stage, rappresentando utili criteri addizionali per filtrare i contenuti informativi durante una ricerca: si tratta di AUTORE, DATA DI PUBBLICAZIONE e TIPO del contenuto.

**RELAZIONI** Ove tradizionalmente ci si affida ai commenti per consentire agli utenti di esprimere un'opinione rispetto alle informazioni riportate o alle posizioni espresse in un contenuto, in *Social (Life) Shuttle* si permette di rispondere ad un contenuto pubblicato nella piattaforma direttamente con altri contenuti, in numero arbitrario.

La relazione di dipendenza tra i contenuti prescinde dalla classe specifica, non ponendo vincoli di alcun genere circa la classe ed il formato della risposta ad un contenuto informativo.

Ciò consente maggiore libertà all'utente nello scegliere la forma espressiva più adeguata per condividere il proprio messaggio, ne facilita la catalogazione e allo stesso tempo rispecchia il principio di uguaglianza tra gli utenti espresso in precedenza ed elemento cardine della piattaforma.

Nel corso del tempo a partire da ciascun contenuto informativo possono così svilupparsi e ramificarsi diverse **DISCUSSIONI**, senza limiti di ampiezza o profondità.

### 1.3.2 *Sistema di classificazione*

Il sistema di classificazione consiste in un insieme di criteri che associano a ciascun contenuto alcuni metadati, in grado di fornire agli utenti della piattaforma informazioni utili a contestualizzarlo, ad interpretarlo e a valutarne l'interesse soggettivo:

#### ARGOMENTO

L'argomento di un contenuto rappresenta la branca del sapere - agnostica rispetto al tema specifico della piattaforma - cui appartiene.

#### EMOZIONI

Le emozioni indicano lo stato d'animo con cui un contenuto sia stato pubblicato dall'autore.

#### GIUDIZI

I giudizi forniscono una valutazione qualitativa sul contenuto e sono espressi dagli utenti.

#### INTENZIONI

Le intenzioni indicano lo spirito con cui l'autore redige il contenuto (opinione, critica, ...).

#### INTERESSI

Gli interessi rappresentano temi specifici della piattaforma nei confronti dei quali ciascun utente registrato dichiara di nutrire passione.

Queste meta-informazioni assumono particolare rilevanza nel processo di ricerca di informazioni all'interno della piattaforma, poiché consentono di escludere o meno determinati contenuti dai risultati.

Tra quelli evidenziati, tuttavia, spicca l'assenza di un criterio in grado di catalogare ordinatamente le informazioni presenti nei contenuti per facilitarne la ricerca, il reperimento e la consultazione: il primo obiettivo dell'attività di stage consiste nell'individuare un meccanismo efficiente per rendere più agevole la consultazione della conoscenza custodita nella piattaforma, fornendo un livello di astrazione rispetto alla semplice enumerazione dei contenuti.

## 1.4 REQUISITI E VINCOLI

Durante gli incontri preliminari all'attività di stage sono stati fissati gli obiettivi, i requisiti ed i vincoli concernenti le attività previste ed i prodotti attesi.

### 1.4.1 *Criterio di classificazione*

La progettazione del criterio di classificazione deve tenere conto di alcuni vincoli e requisiti riguardanti l'architettura della piattaforma e l'integrazione con il sistema di classificazione:

#### INDIPENDENZA DAI CRITERI ESISTENTI

Il criterio deve minimizzare il grado di accoppiamento per risultare facilmente mantenibile e aggiornabile senza intaccare lo stato, l'integrità e le funzionalità dei rimanenti e deve tenere conto di possibili evoluzioni della piattaforma, che comportino l'aggiornamento o la rimozione dei criteri esistenti o l'aggiunta di nuovi.

#### INDIPENDENZA DALLE CLASSI DI CONTENUTI

Il criterio non deve distinguere tra contenuti di classi diverse, ma deve considerare esclusivamente le proprietà e le relazioni definite o definibili sul contenuto generico.

#### INDIPENDENZA DALLE COMPONENTI DEL SISTEMA

Il criterio deve minimizzare le dipendenze e l'accoppiamento con le altre componenti del sistema, che possono essere soggette ad aggiornamenti sostanziali (in particolare quelle di terze parti) o interventi di manutenzione evolutiva.

#### MODULARITÀ

Il criterio dev'essere progettato in modo tale da potersi avvantaggiare - in futuro - di soluzioni tecniche o tecnologiche in grado di automatizzare (in parte o del tutto) le operazioni di classificazione dei contenuti.

Le potenziali criticità tecniche, legate all'implementazione del criterio di classificazione, devono essere raccolte e condivise con il team di progetto, che provvederà a valutarle, a fornire eventuali indicazioni e ad individuare le soluzioni ritenute appropriate e compatibili con l'architettura e le specifiche della piattaforma.

### 1.4.2 *Interfaccia grafica*

L'interfaccia grafica per la consultazione dei risultati di una ricerca sui contenuti informativi deve soddisfare alcuni requisiti essenziali:

- deve consentire all'utente di inserire dei termini di ricerca e selezionare un ambito;
- deve potersi interfacciare a componenti terze per ottenere i risultati di ricerca;

- deve mostrare i contenuti con forme geometriche elementari, che permettano di distinguerne intuitivamente la classe di appartenenza;
- deve permettere la consultazione della DISCUSSIONE associata ad un contenuto, evidenziando il flusso informativo (le sequenze di risposte ad un contenuto);
- dovrebbe gestire dei filtri basati sugli interessi ed il livello di esperienza di un utente registrato;
- dev'essere in grado di visualizzare ordinatamente un numero elevato di risultati di ricerca, evitando un sovraccarico cognitivo;
- dev'essere utilizzabile agevolmente da utenti con differenti livelli di esperienza;
- dev'essere adeguatamente fruibile su dispositivi mobili.



## STAGE

---

### 2.1 PIANO DI STAGE

#### 2.1.1 *Obiettivi e requisiti*

L'attività di stage si colloca nell'ambito del progetto presentato nel capitolo 1 e persegue due obiettivi distinti ma correlati, focalizzandosi sulla componente *social* della piattaforma.

**CRITERIO DI CLASSIFICAZIONE** Il primo obiettivo consiste nell'estendere l'attuale sistema di classificazione (v. sezione 1.3.2), integrandovi un criterio aggiuntivo per la catalogazione dei contenuti pubblicati dagli utenti e la costruzione di un'enciclopedia della conoscenza, al fine di rendere il reperimento e la consultazione delle informazioni desiderate il più efficiente ed agevole possibile.

L'ideazione e concezione del suddetto criterio deve tener conto della natura tematica della piattaforma, conciliando due esigenze distinte:

- dev'essere sufficientemente astratto e flessibile per adattarsi alla molteplicità di varianti tematiche in cui la piattaforma stessa può essere declinata;
- deve cercare di avvantaggiarsi delle peculiarità di una piattaforma tematica, ad esempio la maggior correlazione degli argomenti trattati.

La soluzione individuata deve inoltre prescindere da assunzioni legate alla tecnologia utilizzata.

Alla luce di possibili evoluzioni nello sviluppo della piattaforma, si desidera infine che la classificazione di un contenuto informativo (assegnazione di metadati, individuazione di correlazioni, ...) possa essere - in futuro - demandata a componenti software integrate nella piattaforma.

**INTERFACCIA GRAFICA** Il secondo obiettivo consiste nel progettare un'interfaccia grafica per la consultazione dei contenuti informativi, che sfrutti il criterio di classificazione aggiuntivo per facilitare la ricerca ed il reperimento delle informazioni di interesse per l'utente all'interno del patrimonio enciclopedico della piattaforma.

La sfida principale consiste nel progettare un'interfaccia in grado di visualizzare in maniera chiara e ordinata un numero arbitrario di contenuti, a prescindere dalla classe del dispositivo impiegato (*smartphone, tablet, notebook, ...*).

Il primo passo consiste nell'individuare le informazioni essenziali per una rapida e precisa identificazione dei contenuti (titolo, autore, data di pubblicazione, ...) e nel valutare la notazione (grafica o testuale) più adatta per esprimerle, al fine di renderle accessibili al maggior numero possibile di utenti; le informazioni aggiuntive devono risultare comunque accessibili, ma solo su esplicita richiesta dell'utente.

In questo frangente si inseriscono alcune analisi e valutazioni di carattere sociologico, svolte da altri membri del team di progetto, per individuare le soluzioni più idonee a comunicare tali informazioni in modo da renderne la comprensione chiara e intuitiva a qualsiasi utente.

Il secondo passo richiede di definire le specifiche per un'interfaccia facilmente navigabile, che sia in grado di mostrare in modo ordinato e intuitivo i contenuti e le reciproche relazioni. Occorre individuare opportuni criteri di raggruppamento, ordinamento e collocazione dei contenuti visualizzati per favorirne la consultazione, evitando un sovraccarico cognitivo e garantendo un livello adeguato di leggibilità.

Con il terzo ed ultimo passo si mira ad aggiungere la possibilità per l'utente di filtrare i contenuti visualizzati in accordo a proprietà (argomento, autore, data di pubblicazione, tipo) o metadati associati (emozioni, giudizi, intenzioni, ...).

Per gli utenti autenticati si desidera offrire un livello aggiuntivo di personalizzazione, che consenta di filtrare automaticamente i contenuti secondo le preferenze associate al profilo (interessi, livello di esperienza).

Per individuare i requisiti essenziali si prendono in considerazione alcuni casi d'uso:

1. l'utente naviga liberamente tra i contenuti (più recenti, più letti, più discussi, ...);
2. l'utente consulta la discussione generata da un singolo contenuto;
3. l'utente cerca le informazioni riguardanti un certo tema;
4. l'utente esplora gli argomenti trattati e le reciproche relazioni.

## 2.1.2 Pianificazione

L'attività di stage viene suddivisa in due fasi distinte per semplificarne la pianificazione:

1. l'estensione del sistema di classificazione;
2. l'analisi e la progettazione dell'interfaccia grafica.

Per ciascuna fase sono fissati gli obiettivi generali, sono individuate e organizzate su base settimanale le attività da svolgere, cercando di garantire un carico di lavoro equilibrato, e sono indicati i prodotti attesi.

La durata complessiva dello stage si attesta su 8 settimane a tempo pieno, corrispondenti a 320 ore di lavoro.

ATTIVITÀ	ORE
<i>Fase 1: estensione del sistema di classificazione</i>	
Analisi delle specifiche del sistema di classificazione	40
Analisi comparativa dei principali sistemi di classificazione della conoscenza	40
Progettazione del sistema di classificazione	40
Implementazione del sistema di classificazione nel modello relazionale	40
<i>Fase 2: analisi e progettazione dell'interfaccia grafica</i>	
Analisi dei requisiti dell'interfaccia grafica	40
Progettazione dell'interfaccia grafica: visualizzazione dei contenuti	40
Progettazione dell'interfaccia grafica: filtraggio dei contenuti	40
Progettazione dell'interfaccia grafica: navigazione dei contenuti	40

Tabella 1: Pianificazione settimanale delle attività

## 2.2 NORME DI STAGE

Nel corso dello stage sono stati impiegati diversi strumenti per gestire le attività di progetto e produrre la documentazione prevista.

## 2.2 NORME DI STAGE

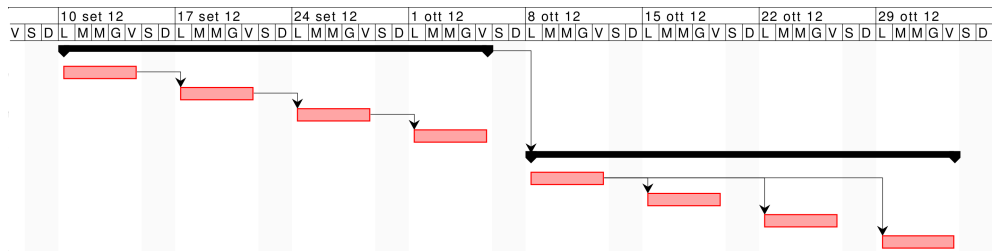


Figura 2: Diagramma di Gantt

CONTROLLO DI VERSIONE	<u>Mercurial</u> 2.0.2
EDITOR $\text{\LaTeX}$	<u>LaTeXila</u> 2.4.0 - <u>gedit</u> 3.4.0
EDITOR UML	<u>UMLet</u> 11.5.1
FOGLIO ELETTRONICO	<u>LibreOffice Calc</u> 3.6
GESTIONE DATABASE	<u>MySQL Workbench</u> 5.2.42
MOCKUP	<u>Pencil</u> 2.0.2
PIANIFICAZIONE	<u>ProjectLibre</u> 1.5.1
REPOSITORY	<u>Bitbucket</u>
SISTEMA OPERATIVO	<u>Ubuntu</u> 12.04

Tabella 2: Configurazione dell'ambiente di lavoro

**DOCUMENTAZIONE** La documentazione è redatta in  $\text{\LaTeX}$  e pubblicata in formato PDF. A ciascun documento è assegnato un numero di versione  $x.y$ , ove  $x$  rappresenta l'ultima **VERSIONE FORMALE**, rivista e approvata dal referente aziendale e disponibile a terze parti interessate (membri del team di progetto, tutor interno), mentre il numero  $y$  si riferisce ad una **VERSIONE PRELIMINARE** per uso interno, eventualmente consultabile dal referente aziendale.

**MODELLO RELAZIONALE** Il modello relazionale del database è stato realizzato mediante lo strumento adottato dal team di progetto, ossia l'editor *MySQL Workbench*, per facilitare la condivisione e l'integrazione delle informazioni.

I nomi delle tabelle sono espressi in lingua italiana e contengono solo caratteri alfabetici minuscoli e non accentati, eventualmente separati mediante il simbolo '\_' (trattino basso). I nomi degli attributi, preceduti dal nome della tabella e dal carattere '.' (punto), sono espressi

in lingua italiana e contengono solo caratteri alfabetici in formato CamelCase, ove la lettera iniziale è sempre in minuscolo.

**DIGRAMMI UML** Durante l'attività di stage sono stati redatti e inclusi nella documentazione diversi diagrammi UML dei casi d'uso, dei package e delle classi, di cui sono presenti svariati esempi nel prosieguo del documento.

I nomi delle sottoclassi riportano - per esteso o in forma abbreviata - l'identificatore della superclasse diretta: nel secondo caso sono presenti - come prefisso - le sole lettere maiuscole, nel medesimo ordine di apparizione.

**CASI D'USO** La notazione utilizzata per identificare un caso d'uso è così definita:

$$UC.x.y$$

ove:

- UC è l'abbreviazione di *Use Case* (Caso d'uso);
- $x \in \{1, 2, \dots\}$  è il numero identificativo del diagramma cui appartiene il caso d'uso;
- $y \in \{1, 2, \dots\}$  è il numero associato al caso d'uso.

**REQUISITI FUNZIONALI** I requisiti del sistema software sono univocamente identificati mediante la seguente notazione:

$$Rf.x.y$$

ove:

- Rf è l'abbreviazione di *requisito funzionale*;
- $x \in \{ob, de\}$  rappresenta il tipo di requisito funzionale (*ob* per obbligatorio, *de* per desiderabile);
- $y \in \{1, 2, \dots\}$  è il numero associato ad un requisito.

**TRACCIAMENTO DEI CASI D'USO** Il tracciamento delle dipendenze tra casi d'uso e requisiti software è realizzato mediante un foglio elettronico, ove:

- ciascuna riga rappresenta un requisito del sistema software;
- ciascuna colonna rappresenta un caso d'uso;
- ciascuna cella contiene il carattere 'X' se esiste una relazione di dipendenza tra il caso d'uso e il requisito, altrimenti è vuota.

## 2.3 CRITERIO DI CLASSIFICAZIONE

Per ciascuna riga e colonna viene impiegata una semplice formula per asserire la completezza e la necessità della matrice dei requisiti:

`CONTA.SE(A:Z;"X")`

ove:

- A:Z corrisponde all'intervallo di celle di una singola riga o colonna;
- X rappresenta la stringa da cercare;
- `CONTA.SE` è una funzione che accetta due argomenti (l'intervallo di celle ed il *pattern*) e restituisce il numero di celle appartenenti all'intervallo contenenti una o più occorrenze del *pattern* specificato.

**COMPLETEZZA** Per ogni colonna, se la formula restituisce un valore pari a o (zero) sta ad indicare che il requisito utente non è soddisfatto da alcun requisito software.

**NECESSITÀ** Per ogni riga, se la formula restituisce un valore pari a o (zero) sta ad indicare che il requisito software corrispondente è superfluo.

## 2.3 CRITERIO DI CLASSIFICAZIONE

Il patrimonio di conoscenza della piattaforma è garantito essenzialmente dai contenuti pubblicati dagli utenti ed arricchito dal loro valore informativo: ciascuno di essi, a prescindere dalla forma (testo, immagini, audio, video, ...) o dalla classe (domanda, discorso, evento, recensione, ...), condivide delle informazioni inerenti uno o più elementi del dominio tematico della piattaforma.

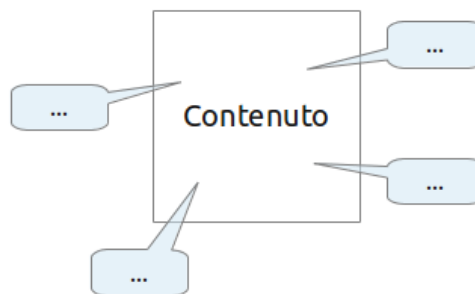


Figura 3: Valore informativo di un contenuto

2.3.1 *Enciclopedia del sapere*

Attualmente, il limite della piattaforma consiste nell'essere un serbatoio di CONTENUTI INFORMATIVI disaggregati e priva degli strumenti per classificare e catalogare il sapere custodito conferendovi una struttura ordinata, una sorta di indice in grado di facilitarne la ricerca, il reperimento e la consultazione.

L'obiettivo del criterio di classificazione consiste essenzialmente nel costruire un'enciclopedia del sapere a partire dalle informazioni presenti nei contenuti informativi.

Un'enciclopedia è un'opera letteraria che raccoglie e ordina la sintesi della conoscenza umana in tutti i campi o in un determinato settore. Le enciclopedie sono divise in voci, o lemmi, cui si accede di solito in ordine alfabetico.  
[1]

Il modello concettuale dell'enciclopedia diventa un riferimento utile ad identificare:

1. i casi d'uso essenziali concernenti la consultazione del sapere custodito nella piattaforma (ricerca per lemma, argomento o affinità);
2. gli elementi strutturali del sistema di classificazione (lemmi, accezioni, ...);
3. i requisiti e le specifiche più rilevanti per il criterio di classificazione;
4. le principali criticità rispetto alla coerenza e consistenza dell'enciclopedia.

**CONTENUTO INFORMATIVO** Ciascun contenuto può essere considerato una collezione di frammenti di informazioni (v. figura 3), che contribuiscono ad arricchire la conoscenza relativa ad uno o più lemmi dell'enciclopedia. Il criterio di classificazione deve quindi tenere traccia della relazione tra contenuti informativi e lemmi per essere in grado di catalogare e ricostruire l'intero sapere disponibile riguardo un certo tema.

**ENTITÀ** Nell'ambito della piattaforma, i lemmi vengono definiti entità ( $d_i \in D$ ) e rappresentano elementi concreti (luoghi, persone, eventi, ...) o astratti (concetti, temi, ...) a cui afferiscono i contenuti.

L'insieme delle entità definite - in un certo istante - all'interno della piattaforma rappresenta il **DOMINIO DELLA CONOSCENZA** (di seguito per brevità **DOMINIO**).

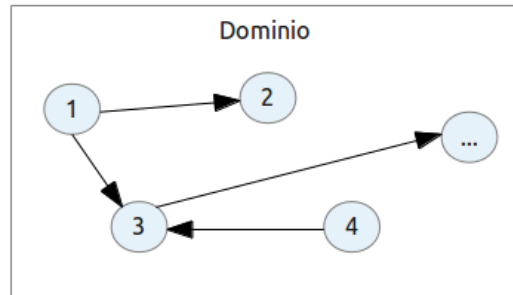


Figura 4: Relazioni tra le entità del dominio

Così come ciascun lemma di un'enciclopedia contiene spesso riferimenti ad altre voci, che trattano temi specifici o attinenti, nella piattaforma ciascuna entità può riferire o essere riferita da un numero arbitrario di entità distinte e possono esistere riferimenti incrociati, ossia coppie di entità che si citano a vicenda.

Ciò si traduce nel modello relazionale con un vincolo referenziale di tipo molti-a-molti tra le entità, che distingue referenti e riferite e permette di interpretare il dominio  $D$  mediante una struttura a grafo orientato ove:

- ciascun nodo rappresenta un entità;
- ciascun arco uscente identifica un'ENTITÀ RIFERITA;
- ciascun arco entrante identifica un'ENTITÀ REFERENTE.

**ETICHETTE** Ciascuna entità ha un valore semantico preciso ed univoco, ma dev'essere identificata anche sul piano sintattico mediante un'etichetta ( $e_j \in E$ ), ossia una stringa di lunghezza variabile che consenta agli utenti di riferirla all'interno di ciascun contenuto.

L'insieme di etichette definite in un certo istante costituisce il **DIZIONARIO** della piattaforma.

Il modello illustrato presenta tuttavia due notevoli inconvenienti, che devono essere risolti per soddisfare i requisiti e gli obiettivi fissati: l'ambiguità sintattica e quella semantica.

### 2.3.2 Ambiguità sintattica

Nel linguaggio comune, è possibile riferirsi ad una certa entità ( $d_i$ ) con termini o espressioni differenti ( $e_{i,j} \in E_i$ ): la presenza di sinonimi, aventi il medesimo valore semantico ma diversa sintassi, rappresenta un fattore di ambiguità intrinseco e non trascurabile, che trasforma la relazione uno-a-uno tra entità ed etichette in una di tipo uno-a-molti.



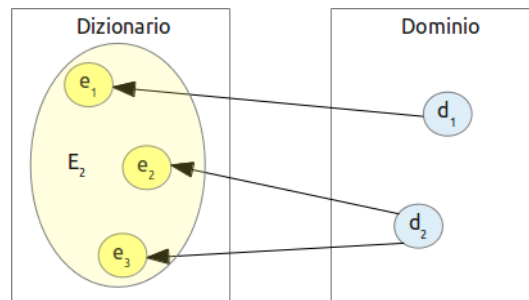


Figura 5: Ambiguità sintattica di un'entità

Il criterio di classificazione è dunque chiamato a conciliare la possibilità per l'utente di riferirsi ad una certa entità con etichette differenti e l'esigenza di identificarla univocamente nei contenuti informativi.

**SINONIMI** La gestione dei sinonimi di un'entità rappresenta un aspetto cruciale, poiché scegliere tra i possibili sinonimi un'etichetta arbitraria con cui identificare un'entità ed ignorare i rimanenti, pur semplificando il modello, costringerebbe gli utenti a conoscere e ad utilizzare solo quell'etichetta, imponendo una scelta del tutto arbitraria e - in quanto tale - fortemente soggettiva.

Per facilitare la ricerca di un'entità è opportuno includere e conservare nel dizionario tutte le etichette note o utilizzate dagli utenti, così da rendere il sistema sempre più accurato nell'individuare e riconoscere l'entità cui un utente fa riferimento ogni qualvolta utilizza una certa etichetta. Con il passare del tempo, il numero dei sinonimi di ciascuna entità è destinato a crescere grazie al contributo degli utenti, garantendo così una migliore copertura sintattica.

La proliferazione di sinonimi, ossia etichette equivalenti sul piano semantico ma sintatticamente differenti, rischia tuttavia di avere pesanti ripercussioni sull'efficacia del criterio di classificazione e particolarmente sull'efficienza della ricerca: se ai contenuti vengono assegnate delle etichette, reperire tutte e sole le informazioni inerenti una certa entità richiederebbe infatti di cercare riscontri nei contenuti pubblicati per ogni etichetta con cui possa venir riferita.

**CONTENUTI** A tale inconveniente si pone rimedio facilmente definendo una relazione molti-a-molti tra le entità e i contenuti, che ignori le etichette utilizzate per indicarle.

Così facendo si preserva l'identificazione univoca di ciascuna entità nei contenuti informativi, almeno sul piano semantico, e si guadagna in termini di efficienza.

Si considerino il dizionario  $E$  ed il dominio  $D$ . Il numero medio  $\alpha$  di etichette associate a ciascuna entità risulta pari a:<sup>1</sup>

$$\alpha = \frac{|E_0| + |E_1| + \dots}{|D|} = \frac{\sum |E_i|}{|D|} \quad (1)$$

ove

$$\sum |E_i| = |E| \quad (2)$$

Ciò significa intuitivamente che assegnare ai contenuti un'etichetta arbitraria aumenterebbe di una costante moltiplicativa  $\alpha$  la complessità dell'operazione di ricerca, che dovrebbe essere ripetuta per ciascuna delle  $\alpha$  etichette anziché per la sola entità corrispondente.<sup>2</sup>

**ETICHETTE** Tuttavia, per riferire un'entità nella piattaforma web occorre caratterizzarla anche sul piano sintattico. Sebbene non vi siano particolari controindicazioni nel permettere all'utente di scegliere l'etichetta da utilizzare nei contenuti pubblicati, si rischia di:

- generare ambiguità e confusione tra gli utenti stessi, rendendo poco intuitivo stabilire se due entità citate in due contenuti diversi e con etichette differenti rappresentino effettivamente la stessa;
- dover aggiungere alla relazione tra entità e contenuti un'informazione aggiuntiva, ossia l'etichetta scelta per indicarla.

Per tale ragione, si individua arbitrariamente - per ciascuna entità  $d_i$  - un'ETICHETTA PRIMARIA ( $e_{i,0} \in E_i$ ), che la identifica univocamente nell'ambito della piattaforma, mentre le restanti (ETICHETTE SECONDARIE) ne vengono considerate sinonimi.

Così facendo si introducono tuttavia alcuni vincoli aggiuntivi sulla relazione, che risulta scissa in due distinte:

1. una relazione di tipo uno-a-uno tra l'entità e la corrispondente etichetta primaria;
2. una relazione uno-a-molti tra l'entità e le etichette secondarie.

In conclusione, il nodo cruciale della soluzione individuata consiste nel mantenere separato l'aspetto semantico (le entità) da quello sintattico (le etichette): ogni qualvolta l'utente ricerca o assegna un'etichetta ad un contenuto, il sistema traduce il suo ingresso sintattico (l'etichetta) in un'uscita semantica (l'entità corrispondente).

<sup>1</sup> Poiché la relazione tra etichette ed entità è di tipo uno-a-molti, ciascuna etichetta è associata ad una ed una sola entità. Ne consegue che  $P = \{E_0, E_1, \dots\}$  è una partizione di  $E$ , ossia vale  $\bigcup E_i = E$  e  $\forall A \in P, B \in P : A \neq B \implies A \cap B = \emptyset$ .

<sup>2</sup> Si assuma per semplicità che la ricerca verifichi - per ciascun contenuto - quali entità o etichette cercate siano presenti, una per volta, e che la complessità computazionale sia equivalente in entrambi i casi.

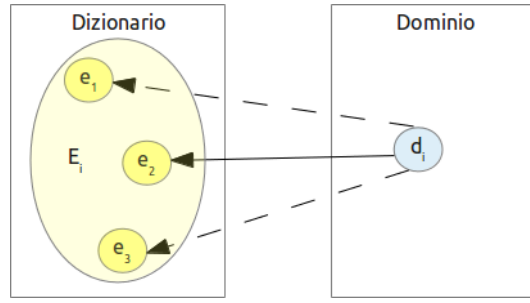


Figura 6: Etichetta primaria e secondarie

Tale meccanismo è tuttavia soggetto ad alcune complicazioni, dovute all'ambiguità semantica di ciascuna etichetta.

### 2.3.3 Ambiguità semantica

Riprendendo il modello dell'enciclopedia, si osserva come ciascuna etichetta  $e_j$  possa avere svariate ACCEZIONI ( $a_{j,k} \in A_j$ ), ciascuna delle quali ne individua un significato differente e si riferisce ad un'entità distinta.

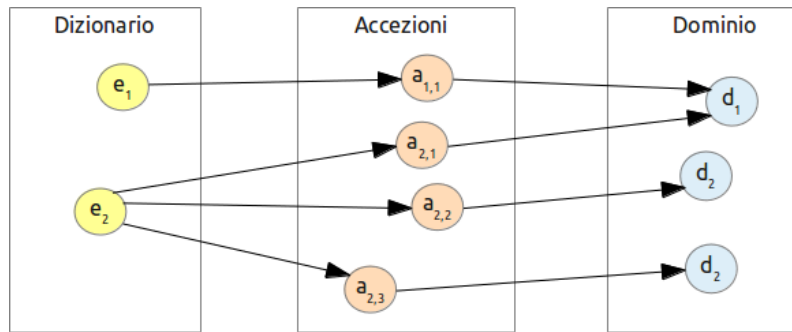


Figura 7: Accezioni di un'etichetta

Sebbene la natura tematica della piattaforma tenda a mitigare la presenza di etichette aventi accezioni multiple, tale eventualità non può essere esclusa e dev'essere dunque opportunamente gestita.

Occorre innanzi tutto rivedere la relazione uno-a-molti tra entità ed etichette, che diventa di tipo molti-a-molti e per cui vale la proprietà illustrata di seguito.

Siano  $a_{j,k}$  e  $a_{j,k'}$  accezioni distinte di un'etichetta  $e_j$  e sia definita la funzione

$$f(a_{j,k}) : A_j \rightarrow D \quad (3)$$

che restituisce l'entità  $d_i$  riferita dall'accezione  $a_{j,k}$ . Allora

$$a_{j,k} \neq a_{j,k'} \implies f(a_{j,k}) \neq f(a_{j,k'}) \quad (4)$$

## The Newsroom

Da Wikipedia, l'enciclopedia libera.

**A questo titolo corrispondono più voci, di seguito elencate.**

Questa è una [pagina di disambiguazione](#); se sei giunto qui cliccando un [collegamento](#), puoi tornare indietro e correggerlo, indirizzandolo direttamente alla voce giusta.

- **The Newsroom** - serie televisiva canadese, andata in onda dal 1996 al 2005
- **The Newsroom** - serie televisiva statunitense del 2012

Figura 8: Pagina di disambiguazione [2]

L'equazione 1 conserva la propria validità, mentre la 2 dev'essere modificata per riflettere la differente molteplicità della suddetta relazione, ossia che ciascuna etichetta riferisce una o più entità:

$$\sum |E_i| = \sum |B_i| = |A| = \sum |A_j| \geq |E| \quad (5)$$

A ciascuna entità continuano ad essere associate una e una sola etichetta primaria e un numero arbitrario (anche nullo) di etichette secondarie. Ove ciascuna etichetta può riferirsi a diverse entità, tale distinzione varia a seconda dell'accezione considerata, poiché la medesima etichetta può essere primaria rispetto ad un'entità e secondaria rispetto ad un'altra.

Per preservare i suddetti vincoli sulle relazioni tra etichette ed entità, si classificano le accezioni in CHIAVE ( $a_{j,0}$ ) o SINONIMICHE ( $a_{j,1}, \dots$ ), a seconda che l'etichetta risulti rispettivamente primaria o secondaria per l'entità corrispondente. In questo modo si riesce a trasferire tale distinzione dalle etichette alle accezioni, preservando la percezione della relazione dal punto di vista dell'entità.

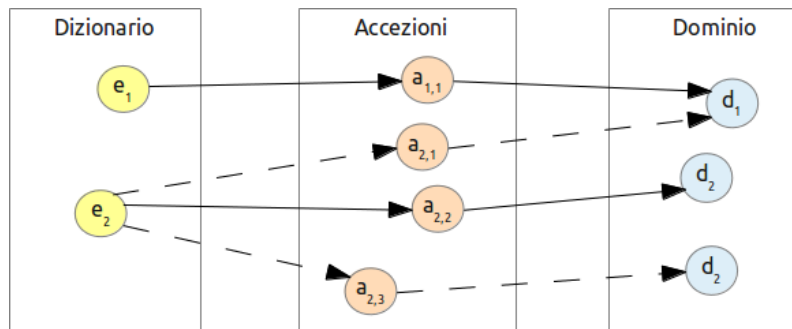


Figura 9: Accezioni chiave e sinonimiche

A risentire maggiormente dell'esistenza delle accezioni è il processo di ricerca di un'entità  $d_i$  a partire da un'etichetta  $e_j$ : se  $|A_j| \geq 2$  l'identificazione dell'entità richiede - da parte dell'utente - la selezione di un'accezione  $a_{j,k} \in A_j$  tra le  $|A_j|$  disponibili per indicare esplicitamente l'entità cui fa riferimento.

## 2.3.4 Modello relazionale

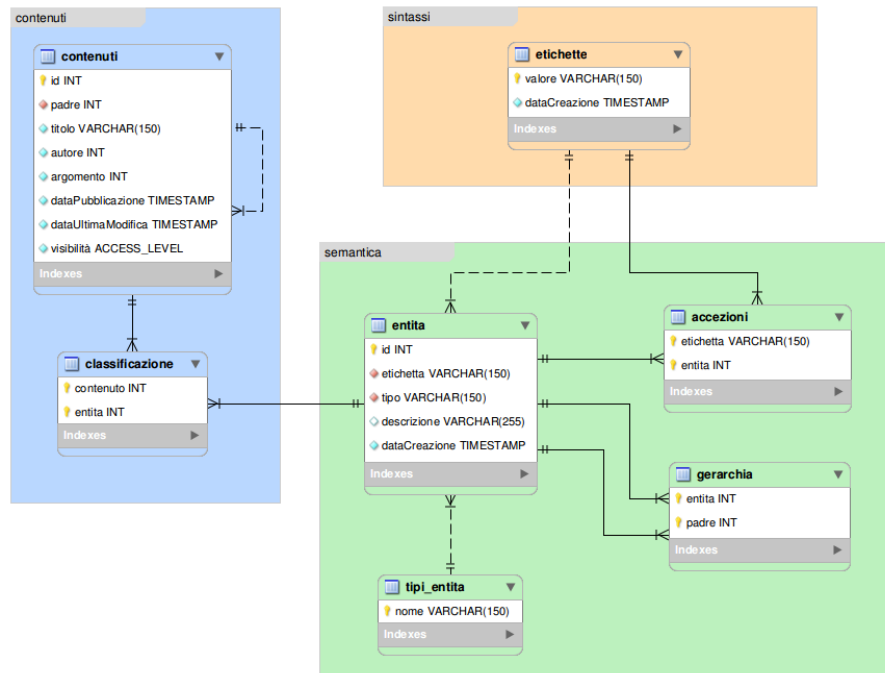


Figura 10: Modello relazionale del criterio di classificazione

Al termine della fase progettuale, si rende necessario modificare il modello relazionale della piattaforma per integrare le informazioni aggiuntive legate al nuovo criterio di classificazione.

Rispetto all'immagine 10, la sola tabella contenuti risulta importata dal modello relazionale della piattaforma per evidenziare alcuni vincoli fondamentali. Le rimanenti sono organizzate in tre *layer* (contenuti, sintassi e semantica) per chiarirne il ruolo all'interno del sistema di classificazione.

**ETICHETTE** Le etichette sono rappresentate mediante la tabella etichette e sono identificate univocamente dalla stringa associata, che rappresenta l'attributo etichette.valore. Tale scelta risponde alla naturale identificazione dell'etichetta nella sequenza di caratteri corrispondente, costituisce una garanzia contro la presenza di duplicati e consente di recuperare il valore dell'etichetta primaria di un'entità senza dover effettuare un'operazione di *join* tra le tabelle entita ed etichette.

**ACCEZIONI** Le accezioni rappresentano un legame univoco tra le etichette e le entità e possono essere di tipo chiave o sinonimico. Esse non presentano attributi propri significativi, ma prevedono tre vincoli referenziali:

1. a ciascuna entità è associata una e una sola accezione chiave (relazione uno-a-uno), che ne identifica l'etichetta primaria;
2. a ciascuna entità sono associate  $0 \dots n$  accezioni sinonimiche (relazione uno-a-molti), che rappresentano i sinonimi dell'etichetta primaria;
3. ciascuna etichetta possiede  $0 \dots n$  accezioni (relazione uno-a-molti).

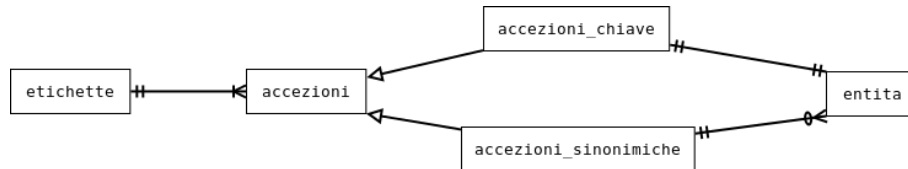


Figura 11: Modello ad oggetti delle accezioni

Ne consegue che sia la superclasse (accezioni) sia le sottoclassi (accezioni\_chiave e accezioni\_sinonimiche) presentano dei vincoli referenziali; in particolare, la presenza delle prime due relazioni costringe a distinguere - dal punto di vista dell'entità - tra accezioni chiave e sinonimiche.

Per modellare tale scenario vengono presi in considerazione tre possibili approcci:

#### TABELLA UNICA

La tabella unica ben si adatta a gestire l'assenza di attributi propri per le sottoclassi e ad esprimere i vincoli referenziali che coinvolgono la superclasse, ma non è in grado di esprimere e adeguatamente rappresentare quelli coinvolgenti le sottoclassi.

#### PARTIZIONAMENTO ORIZZONTALE

Il partizionamento orizzontale riesce a modellare i vincoli referenziali delle sottoclassi, ma non quello della superclasse, e genera due classi aventi i medesimi attributi.

#### PARTIZIONAMENTO VERTICALE

Il partizionamento verticale consente di modellare correttamente tutti e tre i vincoli referenziali, relativi sia alla superclasse sia alle sottoclassi. Tuttavia si rende più complesso modificare il tipo di un'accezione e si introduce l'esigenza di un'operazione *join* per recuperare la lista completa delle accezioni, pur non possedendo le sottoclassi attributi propri.

A seguito di alcune osservazioni si decide di adottare la soluzione della tabella unica:

- la distinzione tra etichette chiave e sinonimiche ha rilevanza essenzialmente dal punto di vista della classe entita;
- il vincolo referenziale tra etichette ed accezioni suggerisce che la distinzione di cui al punto precedente sia irrilevante dal punto di vista delle etichette, ragion per cui risulta utile mantenere tutte le accezioni nella medesima tabella.<sup>3</sup>
- le sottoclassi non hanno attributi propri, per cui il partizionamento verticale e orizzontale sono ritenute soluzioni inadeguate.

Il soddisfacimento delle condizioni richieste viene raggiunto eliminando qualsiasi riferimento al tipo dell'accezione nella classe accezioni e modellando direttamente la relazione uno-a-uno tra le entità e le relative etichette primarie mediante un vincolo referenziale di chiave esterna nella classe entita (entita.etichetta), che identifica la corrispondente etichetta primaria (etichette.valore). Così facendo si riescono ad esprimere tutti i vincoli referenziali senza la necessità di definire le sottoclassi.

## 2.4 FILTRI DI RICERCA

### 2.4.1 Introduzione

La seconda fase dell'attività di stage consiste nell'analisi e nella progettazione di un'interfaccia grafica per la consultazione dei risultati di una ricerca sui contenuti informativi, che sfrutti il criterio di classificazione definito in precedenza e permetta all'utente di:

1. impostare i parametri iniziali di ricerca (parole chiave, ambito);
2. modificare la lista delle entità cercate mediante sostituzione o eliminazione;
3. filtrare i risultati di ricerca in accordo a criteri di classificazione o proprietà dei contenuti;
4. consultare - in forma grafica o testuale - le proprietà fondamentali, i metadati di classificazione e i legami tra i contenuti;
5. mostrare la DISCUSSIONE associata ad un contenuto informativo;
6. impostare i filtri personalizzati (solo per utenti autenticati).

<sup>3</sup> Si consideri ad esempio il caso d'uso della ricerca di un'entità a partire da un'etichetta, ove occorre recuperare la lista completa delle relative accezioni.

Nella fase di progettazione dell'interfaccia grafica devono essere tenuti in debita considerazione alcuni requisiti di qualità desiderabili:

- deve risultare intuitiva e facilmente utilizzabile da qualsiasi categoria di utenti, a prescindere dal livello di esperienza e dalla familiarità con piattaforme web esistenti (*chat, forum, social network, ...*);
- dev'essere fruibile dal maggior numero possibile di dispositivi (*computer, tablet, smartphone, ...*), ciascuno secondo le peculiari modalità di interazione;
- deve rappresentare in maniera ordinata ed efficace le informazioni, a prescindere dal numero di contenuti caricati.

#### 2.4.2 Parametri

La progettazione dei filtri di ricerca richiede innanzi tutto di scegliere i criteri di classificazione e le proprietà dei contenuti informativi (di seguito PARAMETRI) utilizzabili in combinazione ad un filtro.

Il requisito obbligatorio consiste nell'essere definiti su un insieme  $U$  di possibili valori  $u \in U$ , ciascuno dei quali - in un dato istante - dev'essere ammissibile ( $u \in U_A$ ) o bloccato ( $u \in U_B$ ). Deve quindi valere in ogni istante che  $U_A \cap U_B = \emptyset$  e  $U_A \cup U_B = U$ .

Inizialmente tutti i valori relativi ad un certo parametro sono ammessi, ossia  $U_A = U$  e  $U_B = \emptyset$ : l'azione dell'utente altera tale suddivisione dell'insieme  $U$  bloccando un valore, autorizzandolo o azzerando il filtro, ossia rendendo ammissibili tutti i valori ( $U_A = U$ ).

**SODDISFACIBILITÀ** Un contenuto informativo, incluso tra i risultati di ricerca, viene quindi visualizzato se e solo se il valore di ciascun parametro soddisfa il filtro associato.

A seconda che i contenuti assumano - per un dato parametro - un singolo valore (autore, data di pubblicazione, ...) o una lista (emozioni, etichette, intenzioni, ...), il filtro è soddisfatto se e solo se:

- il valore assunto è ammissibile (valore singolo);
- almeno uno dei valori assunti è ammissibile (valori multipli).

Al termine dell'analisi viene redatta una lista completa dei parametri soddisfacenti il requisito iniziale, visibile nella tabella 3 (tra parentesi si riporta il nome della classe corrispondente, ove disponibile).<sup>4</sup>

<sup>4</sup> Il termine ETICHETTE riportato nella tabella designa in seguito il criterio di classificazione illustrato nella sezione 2.3. L'espressione *Data di pub.* è un'abbreviazione usata di seguito per indicare la data di pubblicazione.



PROPRIETÀ	CRITERI DI CLASSIFICAZIONE
Autore (User)	Argomento (Topic)
Data di pub. (Date)	Emozioni (Emotion)
Tipo (Type)	Etichette (Entity)
Giudizi (Rating)	Interessi (Interest)
	Intenzioni (Intention)

Tabella 3: Lista dei parametri per i filtri di ricerca

### 2.4.3 Classi di filtri

Per identificare i requisiti e i casi d'uso correlati ai filtri di ricerca risulta conveniente classificarli in relazione a:

- la natura dell'insieme  $U$  (ordinato, finito, ...);
- le modalità d'interazione dell'utente (selezione di un intervallo, impostazione di una soglia, ...), che determinano i sottoinsiemi  $U_A$  e  $U_B$ .

L'analisi si conclude con l'individuazione di quattro classi di filtri di ricerca, illustrate di seguito.

**A LISTA DI VALORI** Il filtro a lista di valori consente di autorizzare o bloccare selettivamente ciascun elemento  $u \in U$ . I casi d'uso includono:

- la consultazione della lista dei valori ammessi ( $U_A$ );
- la consultazione della lista dei valori bloccati ( $U_B$ );
- l'autorizzazione di un valore;
- il blocco di un valore;
- l'azzeramento del filtro ( $U_A = U$  e  $U_B = \emptyset$ ).

**A SOGLIA DI VALORE** Nel filtro a soglia di valore l'insieme  $U$  è ordinato e contiene un elemento minimo  $\min \in U$  ed uno massimo  $\max \in U$ .

L'utente può scegliere arbitrariamente un valore  $x$ , purché soddisfi la condizione  $\min \leq x \leq \max$ . L'insieme dei valori ammissibili risulta così definito:  $U_A = \{u : u \geq x, u \in U\}$ .

**AD INTERVALLO** Nel filtro ad intervallo l'insieme  $U$  è ordinato e contiene un elemento minimo  $\min \in U$  ed uno massimo  $\max \in U$ .

L'utente può liberamente selezionare l'estremo inferiore ( $\inf \in U$ ) e superiore ( $\sup \in U$ ) dell'intervallo, purché sia soddisfatta la seguente catena di disuguaglianze:  $\min \leq \inf \leq \sup \leq \max$ .

L'insieme dei valori ammissibili risulta ovviamente  $U_A = \{u \in U : \inf \leq u \leq \sup\}$ .

I casi d'uso annoverano:

- la selezione del valore minimo  $\inf$ ;
- la selezione del valore massimo  $\sup$ .
- la selezione di un valore esatto ( $\inf = \sup$ ).

**AD INTERRUETTORE** Il filtro ad interruttore consente all'utente di scegliere se applicare o meno ai risultati di ricerca il filtro corrispondente, che può essere di un tipo qualsiasi tra quelli descritti in precedenza.

I casi d'uso prevedono solamente:

- l'attivazione del filtro;
- la disattivazione del filtro.

LISTA DI VALORI	SOGLIA DI VALORE	INTERVALLO	INTERRUPTORE
Autore	Giudizi	Data di pub.	Interessi
Argomento			
Emozioni			
Etichette			
Intenzioni			
Tipo			

Tabella 4: Elenco dei parametri per i filtri di ricerca, suddivisi per tipo.

#### 2.4.4 Progettazione

La fase di progettazione deve integrare nell'architettura MVC della piattaforma le classi e i parametri dei filtri di ricerca; il risultato finale è mostrato nella tabella 5.

**MODEL** Ciascun tipo di filtro di ricerca è rappresentato dalla classe corrispondente (v. figura 12), che implementa un'interfaccia standard (Filter) per ridurre l'accoppiamento con le altre componenti e facilitare l'estendibilità del sistema (aggiunta di nuovi tipi di filtri, ...):

- FList (a lista di valori);
- FRange (ad intervallo);
- FSwitch (ad interruttore);
- FValue (a soglia di valore).

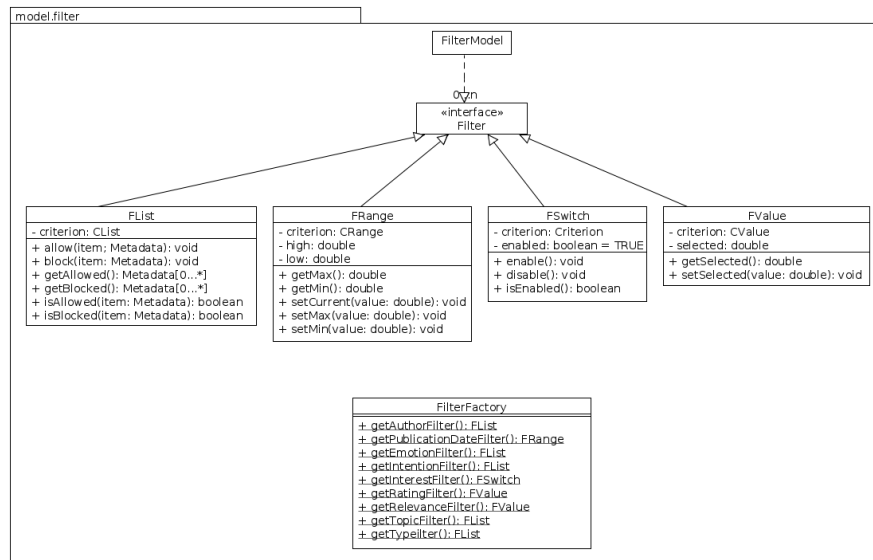


Figura 12: Diagramma delle classi per il package *model.filter*

A parità di classe del filtro, i parametri differiscono per le specifiche dell'insieme  $U$ , in particolare per la **SORGENTE** (statica o dinamica) ed il **TIPO** (emozioni, intenzioni, etichette, ...) di dati gestiti.

Ciascuna classe riportata nella tabella 3 rappresenta - a livello di tipo - il corrispondente parametro (un generico insieme  $U$ ) mentre - a livello di istanze - ciascuno dei possibili valori ( $u \in U$ ).

Per rendere le classi dei filtri polimorfe rispetto ai parametri, ossia per facilitare l'aggiunta, l'aggiornamento e la rimozione di questi ultimi, si definisce una classe astratta (Metadata), che tutte le suddette classi devono estendere. Essa include un metodo astratto (getValueList) da ridefinire per gestire opportunamente l'approvvigionamento dell'insieme dei valori  $u \in U$ , sia che si tratti di una lista statica (tipo, argomento, ...) o dinamica (autore, data di pubblicazione, ...), ossia ricavata dai risultati di ricerca. Nel primo caso la lista di valori può essere memorizzata direttamente in un campo dati statico della classe.

I filtri vengono quindi creati a partire da una lista di valori, collezione di istanze di una sottoclasse concreta di Metadata: per favorire il disaccoppiamento si introducono alcune classi aggiuntive (v. figura 13), che incapsulano tali valori e li rendono accessibili mediante un'interfaccia standard:

- CList (criterio a lista di valori);
- CRange (criterio ad intervallo);
- CValue (criterio a soglia di valore).

Questa soluzione offre la possibilità di inizializzare un filtro ad interruttore con qualsiasi tipo di criterio, nascondendone così le specifiche alle classi che ne gestiscono solamente lo stato di attivazione (FSwitch e FVSwitch).

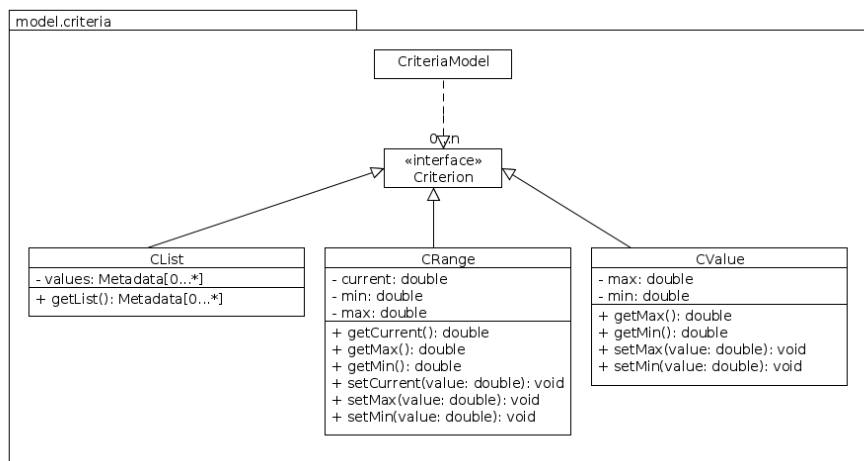


Figura 13: Diagramma delle classi per il package *model.criteria*

**VIEW** Ciascun tipo di filtro richiede un componente grafico differente, che metta a disposizione dell'utente gli strumenti e le opzioni adeguate per consentire un'interazione coerente rispetto a quanto previsto dai rispettivi casi d'uso:

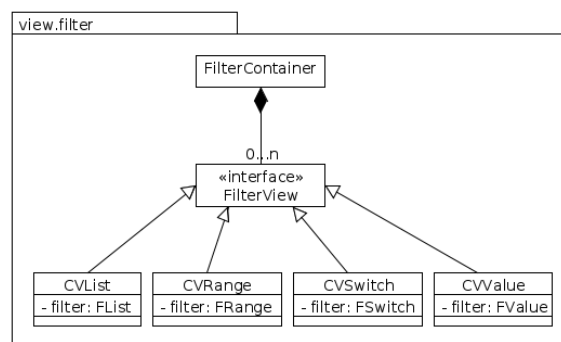
- FVList (filtro a lista di valori);
- FVRange (filtro ad intervallo);
- FVSwitch (filtro ad interruttore);
- FVValue (filtro a soglia di valore).

Ciascuna classe implementa un'interfaccia standard (ViewFilter) e viene istanziata con un corrispondente oggetto di tipo Filter. Per assicurare che il tipo dinamico dell'oggetto sia coerente rispetto al tipo statico del componente grafico (v. tabella 5) il costruttore di ciascuna

	Metadata	Criterion	Filter	FilterController	FilterView
Argomento	Topic	CList	FList	FCList	FVList
Autore	User	CList	FList	FCList	FVList
Data di pub.	Date	CRange	FRange	FCRange	FVRange
Emozioni	Emotion	CList	FList	FCList	FVList
Giudizi	Rating	CValue	FValue	FCValue	FVValue
Intenzioni	Intention	CList	FList	FCList	FVList
Interessi	Interest	CList	FSwitch	FCSwitch	FVSwitch
Tipo	Type	CList	FList	FCList	FVList

Tabella 5: Riepilogo delle classi gestite dal *design pattern Abstract Factory*

classe considerata si aspetta un argomento avente tipo statico corrispondente alla sottoclasse concreta di Filter, che modella il filtro corrispondente.

Figura 14: Diagramma delle classi per il package *view.filter*

**CONTROLLER** Ciascun tipo di filtro prevede diverse modalità d'interazione, che richiedono di essere gestite mediante altrettante classi del controller

**ABSTRACT FACTORY** Per assicurare la corretta combinazione di istanze delle classi (v. tabella 5) si ricorre al *design pattern Abstract Factory*, a cui sono note le corrette combinazioni di oggetti da utilizzare e che permette così di nascondere i dettagli relativi alle classi effettivamente istanziate durante il processo di creazione di un filtro.

## 2.5 RISULTATI DI RICERCA

I risultati di una ricerca sono un insieme  $S$  di contenuti informativi, all'interno del quale si individua un sottoinsieme  $S_v \subseteq S$  formato da tutti e soli quelli visualizzati, che soddisfano tutti i filtri di ricerca.

## 2.5.1 Visualizzazione

Per stabilire se un contenuto debba essere visualizzato ( $s \in U_s$ ) occorre associargli dei metadati, che permettano di asserire se soddisfi tutti i filtri e - a fronte dell'interazione dell'utente con questi - di aggiornare conseguentemente il sottoinsieme  $S_v$ .

Si individuano due possibili esempi di soluzione:

1. un CONTATORE, che tenga traccia del numero di filtri soddisfatti in un dato istante;<sup>5</sup>
2. un ARRAY di valori booleani, ciascuno dei quali esprima il soddisfacimento rispetto ad un singolo filtro.<sup>6</sup>

## 2.5.2 Aggiornamento

L'aggiornamento di un filtro da parte dell'utente presenta due principali fattori di differenziazione:

- la variazione di  $S_v$ , che esprime una diminuzione o un aumento dell'insieme dei valori ammissibili;
- la classe del filtro, che determina l'esatta natura della condizione da verificare per asserire se un contenuto lo soddisfi.

A seconda della variazione di  $S_v$ , per stabilire quali contenuti debbano essere visualizzati occorre:

- se  $\Delta |S_v| < 0$ , individuare gli elementi  $s \in S_v$  che non soddisfino la nuova condizione e spostarli in  $S$ ;
- se  $\Delta |S_v| > 0$ , individuare gli elementi  $s \in S$  che soddisfino il filtro corrente e spostare in  $S_v$  quelli che li soddisfano tutti.

La soluzione del contatore richiede, nel primo caso, un decremento unitario del valore corrispondente per tutti e soli gli elementi che non soddisfino il filtro, sapendo che la preconditione  $s \in S_v$  implica il soddisfacimento di tutti i filtri, incluso quello corrente.

<sup>5</sup> Un contenuto viene visualizzato se e solo se il valore del contatore coincide con il numero di filtri.

<sup>6</sup> Un contenuto viene visualizzato se e solo se tutti gli elementi sono veri.

Nel secondo caso, invece, emerge un limite legato alla necessità di verificare il soddisfacimento del filtro nella condizione precedente e successiva all'aggiornamento dello stesso, non potendo stabilire sulla base del semplice valore del contatore se il filtro in questione fosse tra quelli soddisfatti o meno.

Viceversa, la soluzione basata su un array di valori booleani offre un controllo granulare, che consente di controllare solamente la post-condizione, ma nel secondo caso emerge una complessità proporzionale al numero di filtri e dovuta all'esigenza di verificare che ciascuno di essi sia soddisfatto.

### 2.5.3 *Interazione*

Ciascuna classe di filtri prevede diverse modalità d'interazione per alterare l'insieme dei valori ammissibili  $S_v$ .

#### A LISTA DI VALORI

- Autorizzare un valore;
- bloccare un valore.

#### A SOGLIA DI VALORE

- Aumentare il valore di soglia;
- diminuire il valore di soglia.

#### AD INTERVALLO

- Aumentare il limite inferiore (inf);
- diminuire il limite inferiore (inf);
- aumentare il limite superiore (sup);
- diminuire il limite superiore (sup).

#### AD INTERRUETTORE

- Attivare il filtro;
- disattivare il filtro.

**PROGETTAZIONE** La soluzione consigliata per modellare tale scenario si basa sul *design pattern Command* e consente di:

- incapsulare le singole richieste di aggiornamento in oggetti di tipo `FilterUpdate`;
- gestire le richieste concorrenti mediante una coda (`FilterUpdateQueue`);
- processare serialmente le richieste grazie ad un *dispatcher* (`FilterUpdateDispatcher`);
- aggiungere nuovi comandi a fronte della definizione di nuovi tipi di filtri (`Filter`).

Si conclude definendo un'opportuna gerarchia di classi `FilterUpdate`, che incapsulano la logica di controllo e le informazioni necessarie per aggiornare l'insieme  $S_v$  a fronte della variazione di un filtro a causa di una specifica modalità d'interazione, tra quelle elencate in precedenza.

## 2.6 NAVIGAZIONE DEI CONTENUTI

La navigazione e la consultazione dei risultati di una ricerca rappresentano un'attività cardine degli utenti perciò è necessario presentare i contenuti e le relative informazioni in modo chiaro e ordinato, a prescindere dal numero di risultati e dalle specifiche del dispositivo utilizzato (dimensione, risoluzione dello schermo, modalità d'interazione, ...).

Seguendo un approccio *bottom-up* ci si concentra innanzi tutto sulla rappresentazione di un contenuto informativo (proprietà, metadati, relazioni) e in seguito sulla progettazione dell'interfaccia nel suo complesso.

### 2.6.1 Contenuti informativi

Ciascun contenuto presenta numerose informazioni sotto forma di proprietà o metadati, questi ultimi legati a criteri di classificazione o contestuali alla ricerca.

Il primo passo consiste nel distinguere le informazioni **ESSENZIALI**, ossia utili o indispensabili all'identificazione e alla contestualizzazione di un contenuto, da quelle **AGGIUNTIVE** (o accessorie), che rappresentano un valore aggiunto a fronte di un particolare interesse dell'utente: le prime devono essere immediatamente e chiaramente riconoscibili e consultabili, mentre le seconde devono risultare nascoste ma facilmente accessibili.



PROPRIETÀ ESSENZIALI	PROPRIETÀ AGGIUNTIVE
Autore	Argomento
Attinenza	Emozioni
Data di pub.	Etichette
Tipo	Giudizi
Titolo	Intenzioni

Tabella 6: Proprietà di un contenuto informativo

Tale classificazione consente di rendere più efficiente lo sfruttamento dello spazio disponibile, comunicando all'utente le sole informazioni rilevanti per stabilire la pertinenza del contenuto. Ove questo ne soddisfi le aspettative e ne susciti l'interesse, l'utente può consultarne le informazioni aggiuntive.

In seguito occorre scegliere - per ciascuna informazione - la modalità di rappresentazione (grafica o testuale) più adatta per esprimerla e comunicarla all'utente, in modo da risultare intuitivamente comprensibile e cercando di:

- mantenere un bilanciamento equilibrato tra informazioni testuali e grafiche;
- utilizzare in modo limitato e non esclusivo il colore;
- privilegiare le forme grafiche per esprimere informazioni quantitative o definite su un insieme limitato di valori, nell'intento di facilitare l'associazione rispettivamente tra la dimensione o il tipo di forma al corrispondente significato.

**ARGOMENTO** L'argomento di un contenuto è espresso sia in forma grafica, mediante l'uso del colore, sia testuale e viene collocato in posizione esterna rispetto alla forma per porre in risalto che si tratta di un'informazione aggiuntiva.

**AUTORE** L'autore viene indicato in forma testuale.

**ATTINENZA** Il grado di attinenza è espresso dalla dimensione proporzionale del componente grafico: in questo modo i contenuti maggiormente attinenti alla ricerca sono i primi a catturare l'attenzione dell'utente.

**DATA DI PUBBLICAZIONE** La data di pubblicazione è espressa in forma testuale.

**EMOZIONI** Le emozioni associate ad un contenuto sono riportate in forma testuale.

**ETICHETTE** La lista delle entità assegnate al contenuto è riportata in forma testuale, con particolare evidenza e rilevanza visiva (ordine di apparizione, formato, ...) per quelle corrispondenti ai termini di ricerca.

**GIUDIZI** I giudizi espressi dagli altri utenti sono riportati in forma testuale.

**INTENZIONI** Le intenzioni associate ad un contenuto sono riportate in forma testuale.

**TIPO** Il tipo di contenuto è identificato dalla forma del componente grafico: la scelta di impiegare forme elementari risponde all'esigenza di renderli immediatamente riconoscibili e comprensibili a qualsiasi categoria di utenti. La scelta della forma associata a ciascun tipo si avvale di analisi sociologiche svolte da altri membri del team di progetto.

Dal momento che la rappresentazione visiva delle classi di contenuti differisce essenzialmente per la forma elementare scelta, che a sua volta influenza il posizionamento ed il *rendering* delle informazioni, si definisce una gerarchia di classi che faciliti l'aggiornamento di tali informazioni:

- la classe base (ContentView) rappresenta il componente grafico relativo ad un generico contenuto;
- le sottoclassi (CVAnswer, CVEvent, ...) identificano le singole classi di un contenuto, a ciascuna delle quali sono associati una distinta forma elementare e un insieme di regole per la visualizzazione delle informazioni.

**TITOLO** Il titolo è riportato in forma testuale.

### 2.6.2 Cronologia

I risultati di una ricerca devono essere presentati secondo una struttura chiara ed ordinata, che ne agevoli la consultazione e sia adeguatamente fruibile a prescindere dal numero di risultati o dal dispositivo utilizzato.

Per assicurare la chiarezza espositiva delle informazioni in scenari variegati occorre tenere presente alcuni vincoli fisici, onde evitare un sovraccarico cognitivo dovuto all'eccessivo numero di informazioni mostrate. Una situazione simile pregiudicherebbe alle fondamenta l'esperienza d'uso dell'interfaccia, rendendo assai arduo individuare i contenuti d'interesse: date le previsioni di crescita della piattaforma, si ritiene opportuno includere sin da subito le potenziali criticità nelle valutazioni progettuali.

Una volta prese in considerazione alcune proposte di soluzione e vagliatine pro e contro, si giudica che l'organizzazione cronologica dei contenuti informativi rappresenti una soluzione estremamente intuitiva dal punto di vista dell'utente e allo stesso tempo in grado di offrire un'eccellente flessibilità rispetto ai vincoli e limiti espressi in precedenza.

### 2.6.3 Raggruppamento

Sia  $[t_{\min}, t_{\max}]$  l'arco di tempo entro il quale sono stati pubblicati i risultati di ricerca.

La soluzione individuata consiste nell'organizzare e suddividere i risultati di ricerca in UNITÀ DI TEMPO, corrispondenti ad intervalli di lunghezza fissa (espressa in giorni, mesi o anni), gestendo opportunamente i seguenti parametri:

- l'ampiezza della singola unità di tempo ( $r$ );
- il numero di unità di tempo visualizzate ( $t$ );
- il numero di contenuti visualizzati per unità di tempo ( $c$ ).

La variazione dinamica di tali parametri secondo le specifiche del dispositivo utilizzato consente di offrire un'esperienza di consultazione e navigazione dei risultati di ricerca su misura: è sufficiente definire alcune semplici regole o criteri che scelgano automaticamente i valori ottimali in base ad opportune informazioni (risoluzione orizzontale o verticale, tipo di dispositivo, ...).

Si potrebbero ad esempio variare  $t$  in misura proporzionale alla risoluzione orizzontale dello schermo e  $c$  rispetto a quella verticale,

fissando dei valori costanti (in pixel) per la larghezza di un'unità di tempo e l'altezza di un contenuto.

Il numero di risultati visualizzati corrisponde sempre e comunque a:

$$c_{\text{tot}} = c * t \quad (6)$$

Questi meccanismi consentirebbero inoltre di ottimizzare la visualizzazione delle unità di tempo (e i contenuti associati) a seconda dell'orientamento del dispositivo:

- se orizzontale, le unità di tempo vengono mostrate come colonne affiancate e i risultati come un elenco verticale;
- se verticale, le unità di tempo possono essere mostrate come righe e i risultati come una lista di elementi;

**ARCO TEMPORALE** Se si desidera infine visualizzare l'intero arco temporale  $[t_{\min}, t_{\max}]$ , sotto l'ipotesi che per garantire una sufficiente leggibilità dei contenuti il numero di unità di tempo  $t$  sia fissato, occorre impostare l'ampiezza  $r$  delle unità di tempo in modo che:

$$r = \frac{t_{\max} - t_{\min}}{t} \quad (7)$$

In caso contrario, si mostrano solo i risultati relativi ad un sottointervallo di  $[t_{\min}, t_{\max}]$ , con un evidente guadagno in termini di:

- efficienza, legata alla possibilità di caricare i soli contenuti da visualizzare;
- scalabilità rispetto al numero di contenuti e all'ampiezza dell'intervallo  $[t_{\min}, t_{\max}]$ .

Per ciascuna unità di tempo avente un numero di risultati di ricerca associati superiore al numero massimo di contenuti visualizzabili ( $c_i > c$ ) può essere sufficiente includere un collegamento, che consenta di visualizzare l'intera lista di risultati in uno spazio dedicato.

#### 2.6.4 Ordinamento

In uno scenario caratterizzato da un elevato numero di risultati di ricerca appare particolarmente opportuno definire uno o più criteri di ordinamento dei risultati all'interno di ciascuna unità temporale, cercando di favorire quelli aventi maggiore attinenza<sup>7</sup>, pubblicati più di recente, ... .

<sup>7</sup> Se  $E_s$  è l'insieme delle etichette cercate e  $E_c$  l'insieme delle etichette assegnate a ciascun contenuto si possono distinguere tre casi, a seconda del numero di etichette assegnate ad un contenuto tra quelle cercate: corrispondenza completa ( $E_s \subseteq E_c$ ), parziale ( $E_s \cap E_c \neq \emptyset$ ) o assente ( $E_s \cap E_c = \emptyset$ ).

D'altro canto, la probabilità  $P$  che un contenuto venga visualizzato in un'unità temporale è inversamente proporzionale al numero di risultati associati:

$$P = \frac{c}{c_i} \quad (8)$$

ove  $c_i$  rappresenta il numero di risultati di ricerca pubblicati nell'unità di tempo  $i$ -esima.

### 2.6.5 Navigazione

Infine occorre fornire agli utenti gli strumenti e le opzioni di navigazione temporale dei contenuti per consentire loro di:

- visualizzare i contenuti relativi al passato o al futuro (rispetto al periodo corrente);
- modificare la scala temporale, impostando un numero arbitrario di giorni, mesi o anni ( $r$ ).

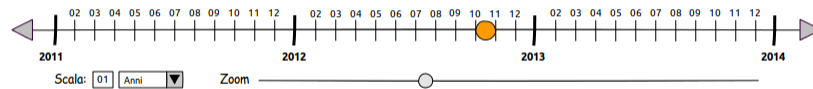


Figura 15: Asse temporale

## 2.7 RISULTATI

L'esito delle attività è stato nel complesso soddisfacente: gli obiettivi fissati sono stati raggiunti e tutti i vincoli o requisiti espressi dal committente sono stati esauditi.

La progettazione del criterio di classificazione si è conclusa con il soddisfacimento dei requisiti illustrati nella sezione 1.4.1, in particolare l'indipendenza da altri criteri di classificazione e dalle classi di contenuti (v. figura 10).

La soluzione individuata consente infatti di preservare la totale indipendenza del criterio aggiuntivo rispetto a quelli esistenti e definisce le proprietà o relazioni necessarie alla classificazione delle informazioni sul generico contenuto (v. figura 10), garantendo così la possibilità di catalogare il sapere disponibile in modo autonomo e indipendente rispetto alla forma comunicativa impiegata per condividerlo (v. sezione 1.3.1).

La semplicità del criterio di classificazione e la sua implementabilità a livello di DBMS (in termini di proprietà, vincoli e logica) dovrebbero inoltre favorire in futuro:

- l'automatizzazione (parziale o totale) dei processi di catalogazione e classificazione del sapere (l'assegnazione automatica delle entità ai contenuti, l'individuazione delle entità associabili ad una nuova etichetta, l'individuazione di legami di affinità o correlazioni tra i contenuti informativi, ...);
- l'integrazione con componenti software terze, che a vario titolo e in qualsiasi forma siano interessate o necessitino di accedere alle informazioni di classificazione.

Il criterio aggiuntivo beneficia in taluni frangenti, in particolare nella mitigazione del fenomeno dell'ambiguità semantica, della natura tematica della piattaforma, offrendo nel contempo un meccanismo di classificazione astratto e flessibile, in grado di adattarsi agevolmente alle varianti tematiche in cui possa essere declinata la piattaforma e anche a sue future evoluzioni.

Il problema dell'ambiguità sintattica viene invece risolto distinguendo le etichette - dal punto di vista dell'entità - in primarie e secondarie. Associare a ciascuna entità del dominio un'etichetta, che la identifica sul piano sintattico e linguistico e la rende univocamente e consistentemente riconoscibile nella piattaforma, garantisce agli utenti la libertà di cercarla e riferirla con il termine o l'espressione preferito, contribuendo all'arricchimento del dizionario interno, senza tuttavia dover scendere a gravi compromessi in termini di efficienza del criterio di classificazione stesso.

L'ambiguità semantica presenta il sostanziale inconveniente di rendere la relazione tra entità ed etichette di tipo molti-a-molti: nella progettazione del diagramma relazionale si è tuttavia riusciti a conciliare tale molteplicità con i requisiti da essa imposti.

Le singole specifiche del criterio di classificazione sono globalmente riconducibili alla cruciale distinzione tra sintassi (le etichette) e semantica (le entità), che, pur presentando alcuni aspetti problematici, consente di definire in maniera logica e naturale la relazione tra entità e contenuti, stabilendo un legame forte e diretto con l'informazione stessa (ciò di cui si parla) piuttosto che con la forma linguistica con cui viene identificata.

La progettazione dell'interfaccia grafica si è conclusa anch'essa con il soddisfacimento dei requisiti fissati dal committente ed illustrati nella sezione 1.4.2, oltre a quelli emersi durante la fase di analisi dei requisiti.

Tra i problemi progettuali che hanno ricevuto maggiore attenzione, si annovera la gestione dei filtri di ricerca, che ha tenuto in debita

considerazione le esigenze di espandibilità (aggiunta, aggiornamento o rimozione di criteri di classificazione).

Il nodo della soluzione consiste nell'individuazione di alcune tipologie standard di filtri, caratterizzate da principi e meccanismi di funzionamento simili e sostanzialmente indipendenti dal tipo effettivo dei valori gestiti. Ciò che le differenzia è la natura dell'insieme  $U$  dei valori su cui sono definite (ordinato, finito, ...) e le modalità d'interazione dell'utente.

L'indipendenza tra il tipo di filtro e il parametro su cui opera incrementa sostanzialmente la riusabilità e la manutenibilità del codice, potendo definire una sola volta ciascun tipo di filtro e istanziarlo un numero arbitrario (a *runtime*) con i parametri opportuni, che forniscono l'insieme specifico dei valori su cui il filtro stesso agisce.

Il corretto funzionamento del meccanismo si basa sulla coerenza tra la natura dell'insieme  $U$  attesa dal filtro e l'insieme di valori forniti dal parametro. Per garantire la soddisfazione del vincolo si adotta un approccio basato sul *design pattern Abstract Factory*, che consente di garantire una corretta istanziazione delle componenti.

Ove possibile si ricorre ad alcuni ulteriori accorgimenti, come la definizione di costruttori con parametri aventi tipo statico corrispondente alla specifica classe con cui devono essere inizializzati.

La navigazione cronologica dei contenuti, la loro rappresentazione visiva mediante forme elementari e la scelta di forme espressive (grafiche o testuali) adeguate per comunicare le informazioni contribuiscono a soddisfare infine i requisiti di qualità connessi alla facilità d'uso da parte di qualsiasi categoria di utenti, alla fruibilità sulla maggior parte di dispositivi mobili e alla capacità di rappresentare le informazioni in maniera ordinata e chiaramente leggibile, a prescindere dal numero di risultati di ricerca.

## VALUTAZIONI FINALI

---

### 3.1 OBIETTIVI

Nel corso dello stage sono state svolte due attività correlate, ma differenti in merito al tipo di impegno e di competenze richieste:

1. lo sviluppo di un criterio di classificazione aggiuntivo per catalogare il sapere custodito nei contenuti pubblicati dagli utenti;
2. l'analisi e la progettazione di un'interfaccia grafica per la consultazione dei risultati di una ricerca sui contenuti informativi.

La progettazione del criterio di classificazione ha richiesto inizialmente un'attività di ricerca riguardante lo stato dell'arte dei sistemi di classificazione attualmente adottati nelle principali piattaforme di condivisione di contenuti:

- blog (*Drupal, Wordpress*);
- forum;
- social network (*Facebook, Twitter*).

I risultati di questa analisi hanno consentito di conoscere i principali meccanismi di catalogazione delle informazioni, di evidenziarne le principali analogie e differenze e di coglierne i punti di forza e di debolezza. A partire da queste informazioni ho inteso affrontare lo sviluppo di un criterio di classificazione, che riuscisse a coniugarne - nella soluzione più semplice e adatta alla piattaforma *Social (Life) Shuttle* - gli aspetti essenziali, riuscendo al contempo a superarne i limiti evidenziati.

I principali dubbi e criticità sono emersi all'atto di valutare l'interazione dell'utente in particolari scenari e casi d'uso: sebbene il criterio di classificazione risolva i problemi legati alle ambiguità sintattiche e semantiche, occorre considerare che la distinzione tra sintassi e semantica comporta la necessità di individuare esplicitamente le relazioni tra entità del dominio ed etichette del dizionario, al fine di garantire la consistenza e coerenza del criterio stesso, ogni qualvolta si definisce una nuova entità, si crea una nuova etichetta o si aggiunge un'accezione ad un'etichetta esistente.

In tali frangenti occorre individuare meccanismi che siano in grado di fornire aiuto e assistenza agli utenti nel processo di individuazio-



ne di tali legami o provvedere ad effettuare un controllo a posteriori finalizzato ad assicurare - con il miglior margine possibile - la consistenza del dizionario e del dominio, la correttezza o l'assenza dei suddetti legami tra entità ed etichette.

Un esempio emblematico riguarda l'individuazione dell'entità associata ad una nuova etichetta, utilizzata per la prima volta da un utente: in certi casi, ad esempio con i nomi propri, possono esistere molteplici varianti (nomi d'arte, soprannomi, ...) che differiscono sul piano sintattico, a volte marginalmente e a volte sensibilmente, e occorre individuare la correlazione esistente sul piano semantico.

Data la natura complessa del problema, queste criticità sono state raccolte e segnalate ai restanti membri del team e saranno oggetto di indagini da parte di persone qualificate. Nell'attesa si è ritenuto di dotare ogni declinazione della piattaforma di un dominio degli argomenti principali e di un dizionario delle espressioni più comuni, in modo da garantire una solida base rispetto alla quale catalogare e classificare le informazioni presenti nei contenuti pubblicati.

La fase di analisi dei requisiti ha invece approfondito le modalità d'interazione dell'utente generico con il sistema a partire dai requisiti essenziali illustrati nella sezione 1.4.2, riuscendo ad individuare requisiti impliciti e caratteristiche funzionali inizialmente non preventive.

La decisione, presa di comune accordo con il referente aziendale, di demandare la valutazione delle tecnologie di sviluppo al momento in cui le specifiche della piattaforma e le esigenze tecniche dell'interfaccia grafica fossero più stabili e definite, ha richiesto uno sforzo aggiuntivo di astrazione per quanto concerne la progettazione della soluzione, ma al contempo ha rappresentato uno spunto di riflessione per comprendere con maggior chiarezza il ruolo e l'identità della fase stessa nel ciclo di vita del software.

Nell'ambito dell'agnosticismo tecnologico, le attività si sono focalizzate sull'architettura e sulla presentazione dell'interfaccia grafica, fornendo ove possibile e utile delle linee guida o delle specifiche dettagliate riguardo approcci risolutivi (*design pattern*) o componenti di particolare interesse o rilevanza per il progetto, al fine di chiarire l'impianto progettuale.

Le soluzioni concepite hanno comunque tenuto conto delle esigenze di adattabilità a cambiamenti ed evoluzioni del sistema, incluse l'estensione o integrazione di funzionalità esistenti.

## 3.2 COMPETENZE PROFESSIONALI

Le attività di stage si inseriscono nell'ambito di un percorso di collaborazione con l'azienda *Sintesi Sas*, iniziato in occasione dell'iniziativa *Mimprendo* promossa da *Confindustria Padova*, che mi ha permesso di seguire il percorso evolutivo del progetto *Social (Life) Shuttle* sin dagli albori ed ha rappresentato un'opportunità di crescita professionale molto rilevante.

Sin dalle attività precedenti ma particolarmente nel corso dello stage ho potuto apprezzare gli enormi benefici - in termini di competenze professionali acquisite - derivanti dall'operare all'interno di un progetto segnato da un forte carattere multidisciplinare: il team di progetto include infatti figure provenienti da svariati settori professionali (psicologia, sociologia, marketing, economia, informatica, ingegneria, ...), che collaborano attivamente e a stretto contatto integrando le reciproche conoscenze e competenze.

Nel corso dell'attività di stage ho avuto occasione di confrontarmi con queste figure, da cui ho appreso informazioni che mi hanno consentito di svolgere le attività previste nel modo migliore e che mi hanno fatto comprendere come il successo di un progetto non possa in alcun modo prescindere dal contributo di competenze professionali diversificate.

L'analisi, la progettazione e lo sviluppo della piattaforma *Social (Life) Shuttle* non si limitano a considerare i soli aspetti tecnici o tecnologici nella concezione e nella valutazione delle soluzioni da adottare nell'intero ciclo di vita del software, ma cercano di comprendere ed interpretare le esigenze degli utenti e le dinamiche sociali. In questa piattaforma di socializzazione e di condivisione della conoscenza la tecnologia è diventata non il fine, bensì il mezzo attraverso il quale si cerca di concretizzare un certo tipo di esperienza (virtuale e non).

Durante la fase di integrazione del criterio di classificazione nel modello relazionale della piattaforma le competenze acquisite in ambito universitario mi hanno permesso di individuare soluzioni semplici ma efficaci, che sono state accolte e giudicate positivamente dal team di progetto. Le principali difficoltà incontrate hanno riguardato l'integrazione della propria soluzione nel modello relazionale esistente in maniera coerente e collaborativa, essendo diverse persone impegnate, seppur su fronti diversi, alla modifica e all'aggiornamento dello stesso.

Le attività previste per la seconda fase dello stage sono state caratterizzate solo in minima parte da attività di ricerca, integrata in molti aspetti da indagini di carattere sociologico svolte da altri membri del team di progetto.

Queste, seppur preliminari, hanno rappresentato un contributo cruciale nel processo di identificazione e comprensione delle esigenze di utenti con profili esperienziali diversificati e di valutazione delle scelte progettuali effettuate.

La maggior parte del tempo a disposizione è stato speso per svolgere un'accurata analisi dei requisiti, che ha permesso di delineare dettagliatamente i casi d'uso degli utenti e individuare alcuni requisiti fondamentali: in questa fase si è rivelata decisiva, per lo svolgimento delle attività secondo le scadenze previste dal piano di lavoro, un'organizzazione e una metodologia acquisiti e maturati durante la carriera universitaria.

La documentazione prodotta, in particolare, è stata giudicata assai favorevolmente per la completezza e l'organicità con cui sono stati presentati e illustrati in dettaglio i risultati del lavoro svolto nell'arco delle varie settimane.

### 3.3 STAGE E UNIVERSITÀ

Sebbene la formazione universitaria mi abbia consentito di superare piuttosto agevolmente le sfide e le difficoltà incontrate lungo il percorso di stage, grazie alle conoscenze e alle metodologie di lavoro acquisite, il principale fattore di novità riscontrato nell'ambito del progetto *Social (Life) Shuttle* è rappresentato dal taglio fortemente multidisciplinare, che obbliga a rivedere un modello di pensiero ed una *forma mentis* principalmente focalizzate sull'aspetto tecnico e tecnologico del prodotto.

Sin dalle prime comunicazioni con il referente aziendale, mi è risultato chiaro come l'attitudine a pensare in termini multidisciplinari fosse necessaria per comprendere lo spirito e gli obiettivi del progetto: maturato nel corso dei mesi, grazie al dialogo e alla collaborazione con gli altri membri del team di progetto, tale approccio ha rappresentato un elemento cruciale per integrarsi con successo nel progetto e comprenderne a fondo le reali esigenze ed il genuino spirito.

Durante l'attività di stage, si sono rivelati fondamentali per il conseguimento degli obiettivi fissati un approccio sistematico e organizzato alle fasi di analisi e progettazione dell'interfaccia grafica, frutto degli insegnamenti impartiti nel corso di *Ingegneria del Software* e dell'esperienza maturata durante l'attività di progetto collegata.

A riguardo, l'esperienza di stage mi ha consentito di rivalutare il ruolo decisivo svolto da attività in precedenza considerate di minor rilevanza, come la pianificazione accurata delle attività e la documentazione dei risultati ottenuti: in diversi frangenti la redazione stessa della documentazione mi ha consentito di rilevare incongruenze, criti-

cità o aspetti non sufficientemente approfonditi, permettendomi così di apportare tempestivamente correttivi e miglioramenti.

La varietà di tecnologie, linguaggi e prodotti con cui mi sono misurato durante l'intera esperienza nell'ambito del progetto *Social (Life) Shuttle* ha rafforzato via via la convinzione che le competenze metodologiche e la *forma mentis* rivestano un ruolo decisamente più rilevante rispetto alla conoscenza di linguaggi o tecnologie e meritino di essere affrontate sin dai primi corsi di laurea per stimolare un approccio organizzato e multidisciplinare al lavoro.

GLOSSARIO

---

## B

BITBUCKET - <https://bitbucket.org/>

Piattaforma web per la gestione delle attività di progetto con supporto a strumenti di controllo di versione distribuito.

## C

CAMELCASE

Convenzione per la scrittura di espressioni composte unendo le parole tra loro e mantenendo ciascuna iniziale in maiuscolo.

## D

DBMS (DATABASE MANAGEMENT SYSTEM)

Sistema software per la creazione, la manipolazione e l'interrogazione di basi di dati.

## G

GEDIT - <http://projects.gnome.org/gedit/>

Editor di testo ufficiale dell'ambiente desktop GNOME.

## L

LATEXILA - <http://projects.gnome.org/latexila/>

Editor  $\text{\LaTeX}$  integrato per l'ambiente desktop GNOME.

LIBREOFFICE CALC - <http://www.libreoffice.org/>

Applicazione per fogli di calcolo della suite di produttività *LibreOffice*.

M

MERCURIAL - <http://mercurial.selenic.com/>

Strumento multi piattaforma, gratuito ed open source per il controllo di versione distribuito.

MVC

Design pattern architetturale basato sulla scomposizione del sistema in tre componenti essenziali: *model* (logica di business), *view* (interfaccia grafica) e *controller* (logica di controllo).

MYSQL WORKBENCH - <http://www.mysql.it/products/workbench/>

Applicazione multi piattaforma, gratuita ed open source per la progettazione, lo sviluppo e l'amministrazione di database MySQL.

P

PDF (PORTABLE DOCUMENT FORMAT)

Formato di file per la rappresentazione di documenti in maniera indipendente dalla piattaforma hardware e software.

PENCIL - <http://pencil.evolus.vn/>

Applicazione multi piattaforma, gratuita ed open source per la realizzazione di prototipi di interfacce grafiche.

PROJECTLIBRE - <http://www.projectlibre.org/>

Applicazione multi piattaforma, gratuita ed open source per il *project management*, che consente di realizzare diagrammi di Gantt e di PERT, gestire le risorse allocate e le attività pianificate, ... .

R

RETE SOCIALE

Insieme di persone, aventi interessi in comune e inclini a collaborare e condividere idee o informazioni, e di relazioni di tipo esperienziale definite tra tali soggetti.

U

UBUNTU - <http://www.ubuntu.com/>

Distribuzione Linux gratuita derivata da Debian.

UML - <http://www.uml.org/>

Standard internazionale per un linguaggio di modellazione, che definisce un insieme di notazioni grafiche per la rappresentazione visiva di sistemi.

UMLET - <http://www.umlet.com/>

Applicazione multi piattaforma, gratuita ed open source per la realizzazione di diagrammi UML.

## CRITERIO DI CLASSIFICAZIONE

## LEGENDA

$i$   
 $1 \leq i \leq |D|, i \in \mathbb{N}.$

$j$   
 $1 \leq j \leq |E|, j \in \mathbb{N}.$

$k$   
 $1 \leq k \leq |A_j|, k \in \mathbb{N}.$

*Entità*

$D$   
 Dominio delle entità.

$d_i$   
 Entità del dominio ( $d_i \in D$ ).

*Etichette*

$E$   
 Dizionario delle etichette.

$E_i$   
 Insieme delle etichette relative all'entità  $d_i$  ( $E_i \subseteq E$ ).

$e_j$   
 Etichetta del dizionario ( $e_j \in E$ ).

$e_{i,j}$   
 Etichetta del dizionario relativa all'entità  $d_i$  ( $e_{i,j} \in E_i$ ).

$e_{i,0}$   
 Etichetta primaria dell'entità  $d_i$  ( $e_{i,0} \in E_i$ ).



*Accezioni* $A$ 

Insieme delle accezioni.

 $A_j$ Insieme delle accezioni di un'etichetta  $e_j$ . $B_i$ Insieme delle accezioni relative ad un'entità  $d_i$ . $a_{j,k}$ Accezione di un'etichetta  $e_j$  ( $a_{j,k} \in A_j$ ). $b_{i,k}$ Accezione relativa ad un'entità  $d_j$  ( $b_{i,k} \in B_i$ ).

## CODICE SQL

*Etichette*

**RICERCA DI UN'ETICHETTA** Restituisce le stringe corrispondenti alla sequenza di caratteri @termine inserita dall'utente, eventualmente adeguata allo standard di formato previsto.

```
SELECT *
  FROM etichette
 WHERE valore LIKE '@termine%'
```

**RICERCA DELLE ACCEZIONI DI UN'ETICHETTA** Restituisce la lista delle accezioni di un'etichetta @etichetta.

```
SELECT *
  FROM accezioni
 WHERE etichetta='@etichetta'
```

**NUMERO DI ASSEGNAZIONI DI UN'ENTITÀ** Restituisce il numero di contenuti cui sia stata assegnata l'entità @entita.

```
SELECT COUNT(*) AS num
  FROM classificazione
 WHERE entita='@entita'
```

**INSERIMENTO DI UN'ETICHETTA** Inserisce l'etichetta @stringa...

```
INSERT INTO etichette
  VALUES (@stringa,current_timestamp)
```

... e le associa almeno un'accezione, riferita all'entità @entita.

```
INSERT INTO accezioni(etichetta,entita)
VALUES (@etichetta,@entita)
```

### Entità

CALCOLO DI AFFINITÀ TRA DUE ENTITÀ L'affinità tra due entità è espressa dal numero di contenuti in cui compaiano entrambe.

```
SELECT COUNT(*) AS contenuti
FROM classificazione
WHERE entita='@entita1' OR entita='@entita2'
GROUP BY contenuto
HAVING COUNT(*)>=2
```

RICERCA DELLE ENTITÀ REFERENTI Restituisce la lista delle entità che riferiscono @entita.

```
SELECT padre
FROM gerarchia
WHERE figlio='@entita'
```

RICERCA DELLE ENTITÀ RIFERITE Restituisce la lista delle entità riferite da @entita.

```
SELECT figlio
FROM gerarchia
WHERE padre='@entita'
```

RICERCA DELLE ETICHETTE Restituisce la lista delle etichette con cui è riferibile l'entità @entita.

```
SELECT et.valore AS etichetta
FROM accezioni AS acc JOIN etichette AS et
ON (acc.etichetta=et.valore)
WHERE acc.entita='@entita'
```

### Contenuti

ASSEGNAZIONE DI UN'ETICHETTA Assegna l'entità @entita al contenuto @contenuto.

```
INSERT INTO classificazione(contenuto,entita)
VALUES (@contenuto,@entita)
```

**RICERCA DELLE ETICHETTE ASSEGNATE** Restituisce la lista delle entità assegnate ad un contenuto.

```
SELECT entita
FROM classificazione
WHERE contenuto='@contenuto'
```

**RICERCA DI CONTENUTI GENERICI** Restituisce i contenuti cui sia stata assegnata l'entità @entità.

```
SELECT contenuti
FROM classificazione
WHERE entita='@entita'
```

**RICERCA DI CONTENUTI SPECIFICI** Restituisce i contenuti cui siano state assegnate le entità @entita1, @entita2, ....

```
SELECT contenuto
FROM classificazione
WHERE entita='@entita1' [OR entita='@entita2' ...]
GROUP BY contenuto
HAVING COUNT(*)>=@num_etichette
```

INTERFACCIA GRAFICA

---

## LEGENDA

*Risultati di ricerca*

- $S$   
Insieme dei risultati di ricerca.
- $S_v$   
Insieme dei risultati di ricerca visualizzati ( $S_v \subseteq S$ ).
- $s$   
Risultato della ricerca ( $s \in S$ ).

*Filtri di ricerca*

- $U$   
Insieme dei possibili valori di una proprietà ( $U = U_A \cup U_B$ ).
- $U_A \subseteq U$   
Insieme dei valori autorizzati di una proprietà ( $U_A \cap U_B \neq \emptyset$ ).
- $U_B \subseteq U$   
Insieme dei valori bloccati di una proprietà ( $U_A \cap U_B \neq \emptyset$ ).
- $u$   
Valore di una proprietà ( $u \in U$ ).

*Cronologia*

- $r$   
Ampiezza di ciascuna unità di tempo (espressa in giorni, mesi o anni).
- $t$   
Numero delle unità di tempo visualizzate.
- $c$   
Numero massimo di contenuti visualizzati in ciascuna unità di tempo  $t_i$ .

$t_i$

Unità di tempo i-esima.

$c_i$

Numero di risultati di ricerca associati all'unità di tempo  $t_i$ .

# DIAGRAMMI DEI CASI D'USO

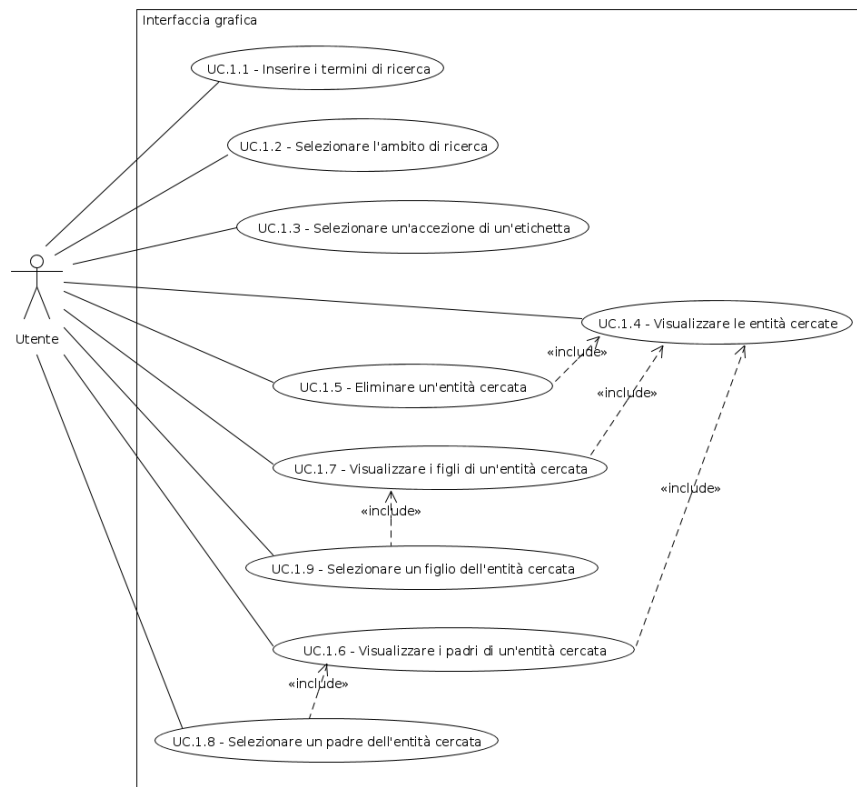


Figura 16: UC.1 - Ricerca di contenuti informativi

## INTERFACCIA GRAFICA

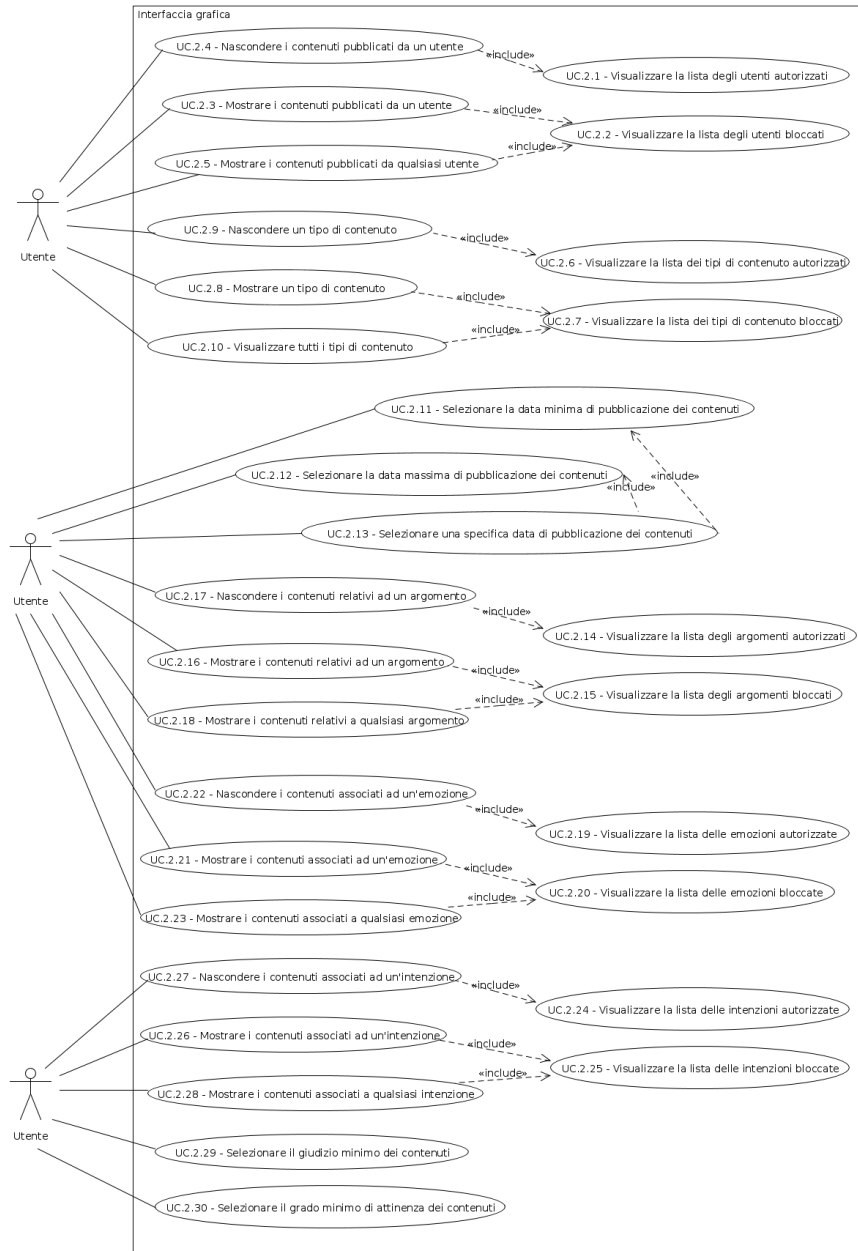


Figura 17: UC.2 - Raffinamento dei criteri di ricerca

## INTERFACCIA GRAFICA

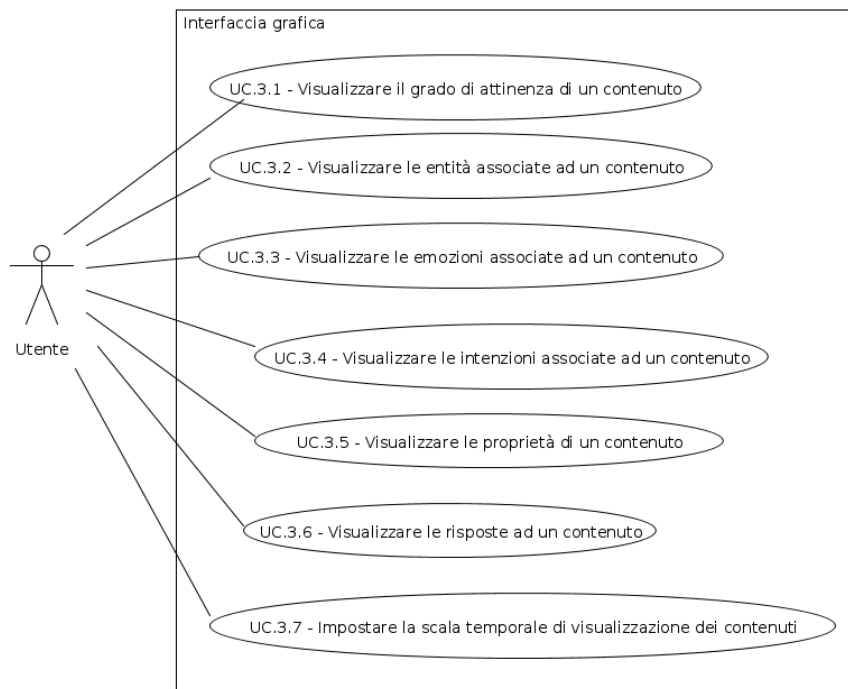


Figura 18: UC.3 - Consultazione dei risultati di ricerca

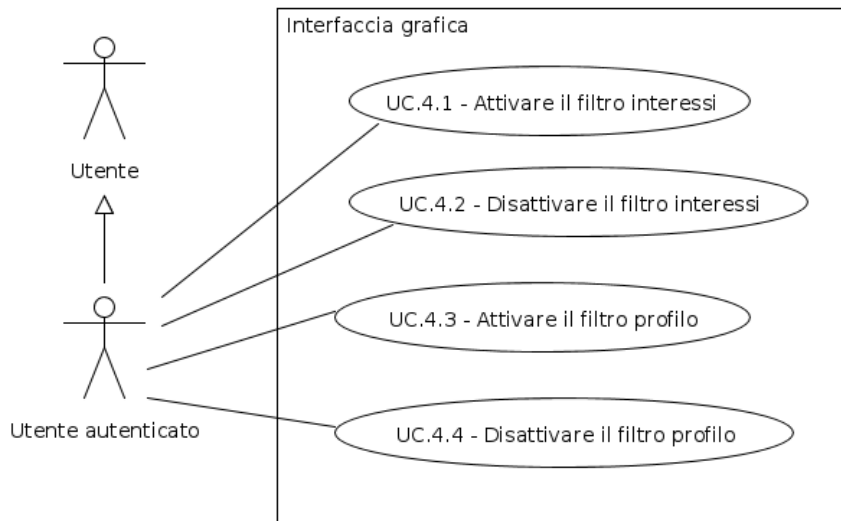


Figura 19: UC.4 - Gestione dei filtri utente

## BIBLIOGRAFIA

---

- [1] Wikipedia. Enciclopedia - wikipedia, l'enciclopedia libera, 2012.  
[Online; in data 23-novembre-2012].
- [2] Wikipedia. The newsroom - wikipedia, l'enciclopedia libera, 2012.  
[Online; in data 23-novembre-2012].