

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

## Relazione di stage - Bozza

Nicola Moretto (matr. 578258)

27 novembre 2012

# Indice

<b>1. Progetto</b>	<b>8</b>
1.1. Genesi . . . . .	8
1.2. Reti sociali . . . . .	8
1.3. Architettura . . . . .	9
1.3.1. Contenuti informativi . . . . .	9
1.3.2. Sistema di classificazione . . . . .	13
1.4. Requisiti e vincoli . . . . .	13
1.4.1. Criterio di classificazione . . . . .	14
1.4.2. Interfaccia grafica . . . . .	14
<b>2. Stage</b>	<b>16</b>
2.1. Piano di stage . . . . .	16
2.1.1. Obiettivi e requisiti . . . . .	16
2.1.2. Pianificazione . . . . .	17
2.2. Norme di stage . . . . .	18
2.3. Criterio di classificazione . . . . .	21
2.3.1. Enciclopedia del sapere . . . . .	21
2.3.2. Ambiguità sintattica . . . . .	23
2.3.3. Ambiguità semantica . . . . .	26
2.3.4. Modello relazionale . . . . .	28
2.4. Interfaccia grafica . . . . .	30
2.4.1. Risultati di ricerca . . . . .	31
2.4.2. Filtri di ricerca . . . . .	32
2.4.3. Navigazione dei contenuti . . . . .	36
<b>3. Valutazioni finali</b>	<b>41</b>
3.1. Obiettivi . . . . .	41
3.1.1. Criterio di classificazione . . . . .	41
3.1.2. Interfaccia grafica . . . . .	41
3.2. Competenze professionali . . . . .	41
3.2.1. Criterio di classificazione . . . . .	42
3.2.2. Interfaccia grafica . . . . .	42
3.3. Stage e università . . . . .	43

## *Indice*

---

A. Glossario	44
B. Criterio di classificazione	47
C. Interfaccia grafica	51

## Elenco delle figure

1.1. Gerarchia degli utenti nelle piattaforme web tradizionali . . . . .	10
1.2. Struttura di un contenuto informativo . . . . .	12
2.1. Diagramma di Gantt . . . . .	19
2.2. Valore informativo di un contenuto . . . . .	21
2.3. Relazioni tra le entità del dominio . . . . .	22
2.4. Ambiguità sintattica di un'entità (+ modello relazionale) . . . . .	23
2.5. Etichette primarie e secondarie (+ modello relazionale) . . . . .	25
2.6. Pagina di disambiguazione [2] . . . . .	26
2.7. Accezioni di un'etichetta (+ modello relazionale) . . . . .	27
2.8. Accezioni chiave e sinonimiche (+ modello relazionale) . . . . .	27
2.9. Modello relazione del criterio di classificazione . . . . .	28
2.10. Modello ad oggetti delle accezioni . . . . .	29
2.11. Diagramma delle classi - Package <i>model.filter</i> . . . . .	35
2.12. Diagramma delle classi - Package <i>view.content</i> . . . . .	39

## Elenco delle tabelle

2.1. Pianificazione settimanale delle attività . . . . .	18
2.2. Configurazione dell'ambiente di lavoro . . . . .	19
2.3. Lista dei parametri per i filtri di ricerca . . . . .	33
2.4. Elenco dei parametri - suddivisi per tipo - per i filtri di ricerca . . . . .	34
2.5. Famiglie di prodotti . . . . .	37
2.6. Famiglie di prodotti . . . . .	37



# 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 Srl*. 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** La prima occorrenza di termini propri o di provenienza straniera divenuti di uso corrente nella lingua italiana sono evidenziati in *corsivo*, mentre le parole o espressioni che assumono particolare significato nel presente contesto sono riportate in MAIUSCOLETTO.

**Codice e formule** I nomi di tabelle, classi, package, ... impiegano uno stile di carattere sans serif, mentre i frammenti di codice o formule impiegano un carattere a **spaziatura fissa**.

# 1. 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 turistici.

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 considerarne 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 riguardi 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.

#### 1.3.1. 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 risultano immediatamente e intuitivamente comprensibili agli utenti, a prescindere dal loro livello di esperienza.



Figura 1.1.: Gerarchia degli utenti nelle piattaforme web tradizionali

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 pensiero rappresenta una forma di comunicazione adatta ad esprimere un contenuto di lunghezza breve e con contenuti superficiali (considerazioni, stati d'animo, freddure, ...).

### **Evento**

L'evento identifica e 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.



Figura 1.2.: Struttura di un contenuto informativo

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 aggiuntivi 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.

## 2. 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.

**Criteri 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 per 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, riuscendo a conciliare due esigenze distinte:

- dev'essere sufficientemente astratto e flessibile per adattarsi alla molteplicità di varianti tematiche in cui la piattaforma stessa può essere declinata;
- dev'essere ottimizzato per 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. Infine, alla luce di possibili evoluzioni nello sviluppo della piattaforma, si desidera 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 ridotto o elevato numero di contenuti, a prescindere dalla classe del dispositivo impiegato (*smartphone, tablet, notebook, ...*).



Il primo passo consiste nell'individuare le informazioni essenziali ad una rapida e precisa identificazione dei contenuti (titolo, autore, data, ...) e 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 essere comunque accessibili, ma solo su esplicita richiesta dell'utente. In questo ambito si inseriscono una serie di 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 perciò individuare opportuni criteri di raggruppamento, ordinamento e collocamento 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 intende aggiungere la possibilità per l'utente di filtrare i contenuti visualizzati in accordo a proprietà (argomento, autore, data di pubblicazione, tipo) o metadati associati (attinenza, 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 innanzi tutto in considerazione alcuni casi d'uso classici:

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 (contenuti affini, ...);
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.

## 2. Stage

---

ATTIVITÀ	ORE DI LAVORO
<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 2.1.: Pianificazione settimanale delle attività

### 2.2. Norme di stage

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

**Documentazione** La documentazione è stata 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.



**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 desiderabili);
- $y \in \{1, 2, \dots\}$  è il numero associato ad un requisito.

**Tracciamento dei casi d'uso** Il tracciamento delle dipendenze tra casi d'uso e requisiti software è stato 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.

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

$$\text{CONTA.SE}(A:Z; "X")$$

ove:

- $A:Z$  corrisponde all'intervallo di celle di una singola riga o colonna;
- $X$  rappresenta il pattern da cercare;
- $\text{CONTA.SE}$  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 0 (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 0 (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 2.2.: 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.

[...]

Le enciclopedie possono essere generali, e contenere voci su qualsiasi argomento (lettere, scienze, arti), oppure specialistiche e concentrarsi su un particolare campo della conoscenza (ad esempio un'enciclopedia di medicina o di filosofia). [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, 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'intera conoscenza 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).

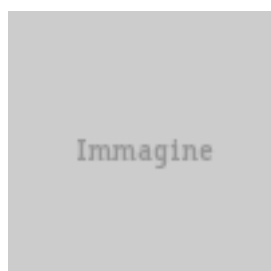


Figura 2.3.: Relazioni tra le entità del dominio

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 multi-a-molti tra le entità, che distingue entità referenti e riferite e permette così di interpretare il dominio  $D$  come una struttura a grafo orientato ove:

- ciascun nodo rappresenta un'entità;
- ciascun arco uscente identifica un'entità riferita;
- ciascun arco uscente 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  $E$  della piattaforma.

Il modello illustrato presenta tuttavia due notevoli inconvenienti, che devono essere risolti per soddisfare i requisiti e gli obiettivi previsti per il criterio di classificazione: 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.

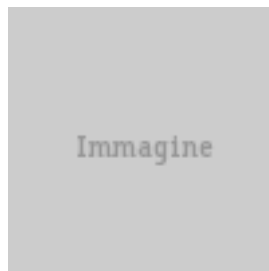


Figura 2.4.: Ambiguità sintattica di un'entità (+ modello relazionale)

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 quando 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 multi-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 associato a ciascuna entità è pari a:

$$\alpha = \frac{|E_0| + |E_1| + \dots}{|D|} = \frac{\sum |E_i|}{|D|} \quad (2.1)$$

ove

$$\sum |E_i| = |E| \quad (2.2)$$

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

---

<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 termini cercati (entità o etichette) siano presenti, uno per volta, e che la complessità computazionale sia equivalente in entrambi i casi.



**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 che pubblica, 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 siano 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.



Figura 2.5.: Etichette primarie e secondarie (+ modello relazionale)

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).

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.

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.

#### 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 2.6.: Pagina di disambiguazione [2]

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

$$f(a_{j,k}) : A_j \rightarrow D \quad (2.3)$$

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

$$a_{j,k} \neq a_{j,k'} \implies f(a_{j,k}) \neq f(a_{j,k'}) \quad (2.4)$$

L'equazione 2.1 conserva la propria validità, mentre la 2.2 dev'essere modificata per riflettere la differente molteplicità della suddetta relazione, ossia il fatto che ciascuna etichetta possa riferire una o più entità:

$$\sum |E_i| = \sum |B_i| = |A| = \sum |A_j| \geq |E| \quad ^3 \quad (2.5)$$

<sup>3</sup>La prima uguaglianza esprime una banale identità tra le etichette di un'entità e le accezioni corrispondenti; la seconda discende dal fatto che  $P' = B_1, B_2, \dots$  rappresenta una partizione di  $A$  rispetto alle entità, mentre la successiva uguaglianza si deduce analogamente dal fatto che  $P'' = A_1, A_2, \dots$  è anch'essa partizione di  $A$  (rispetto alle etichette); l'ultima disuguaglianza si deduce osservando che  $\forall j |A_j| \geq 1$ , ossia per ciascuna etichetta del dizionario esiste almeno un'accezione.



Figura 2.7.: Accezioni di un'etichetta (+ modello relazionale)

Si registrano inoltre rilevanti ripercussioni in merito alla distinzione tra etichette primarie e secondarie. A ciascuna entità continuano ad essere associate una e una sola etichetta primaria, oltre ad 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 ( $a_{j,1}, \dots, a_{j,|A_j|}$ ), 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 2.8.: Accezioni chiave e sinonimiche (+ modello relazionale)

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

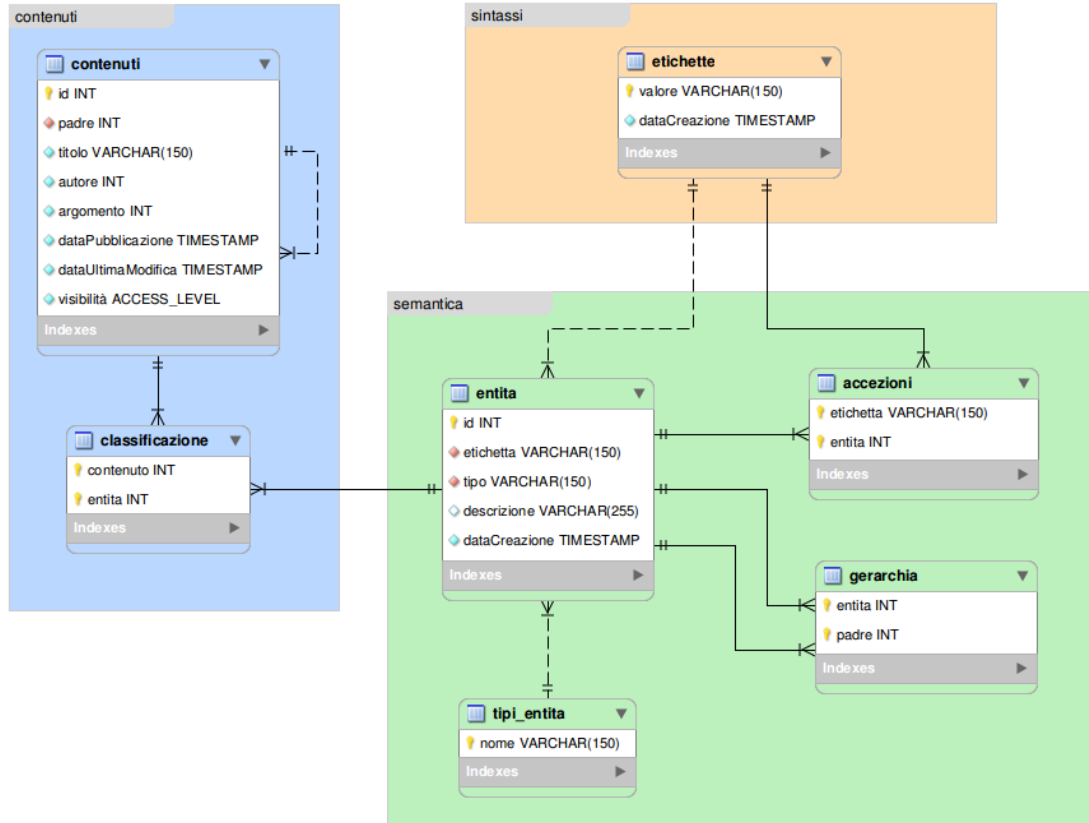


Figura 2.9.: Modello relazionale del criterio di classificazione

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 2.9, la sola tabella **contenuti** risulta importata dal modello relazionale della piattaforma per evidenziare alcune relazioni fondamentali. Le rimanenti sono organizzate in tre *layer* (**contenuti**, **semantica** e **sintassi**) 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 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).

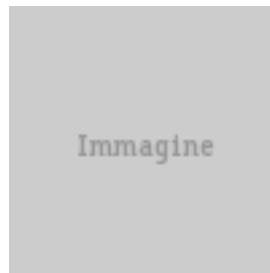


Figura 2.10.: Modello ad oggetti delle accezioni

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>4</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**, ossia **entita.etichetta**, che identifica la corrispondente etichetta primaria nella tabella **etichette**. Così facendo si riescono ad esprimere tutti i vincoli referenziali senza dover definire le sottoclassi.

## 2.4. Interfaccia grafica

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

1. impostare i parametri iniziali di ricerca, ossia le parole chiave e l'ambito;
2. modificare a posteriori 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 di ciascun contenuto, i metadati di classificazione e i legami reciproci;

---

<sup>4</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.

5. mostrare la discussione associata ad un contenuto informativo;
6. impostare i filtri personalizzati (solo per utenti autenticati).

Nella fase di progettazione devono essere tenuti in debita considerazione alcuni requisiti di qualità desiderabili:

### **Intuitiva**

L'interfaccia grafica deve risultare agevole 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, ...*).

### **Multipiattaforma**

L'interfaccia grafica dev'essere fruibile dal maggior numero possibile di dispositivi (*computer, tablet, smartphone, ...*), avendo ben presenti le differenti modalità di interazione.

### **Ordinata**

L'interfaccia grafica dev'essere in grado di rappresentare in maniera ordinata ed efficace le informazioni, a prescindere dal numero di contenuti caricati: quelle essenziali devono essere immediatamente disponibili e facilmente identificabili, mentre quelle aggiuntive o accessorie devono essere comodamente accessibili.

Per semplificare le fasi di analisi e progettazione il modello del prodotto software è stato suddiviso in tre componenti:

### **Ricerca**

Insieme dei parametri e criteri di ricerca (*parole chiave, ambito di ricerca, ...*).

### **Filtri**

Insieme dei filtri di raffinamento della ricerca.

### **Contenuti**

Insieme dei contenuti informativi corrispondenti ai criteri di ricerca e dei relativi strumenti di navigazione.

#### **2.4.1. Risultati di ricerca**

L'insieme dei contenuti informativi corrispondenti ai criteri di ricerca dev'essere gestito in modo che sia possibile:

- aggiornare i risultati di ricerca in seguito ad una variazione dei parametri di ricerca;
- gestire il sottoinsieme dei contenuti da visualizzare, in accordo alla configurazione dei filtri di ricerca;

- aggiornare lo stato di visualizzazione dei contenuti in modo intelligente (restrizione vs ampliamento) a trasparente rispetto ai filtri;
- mantenere in memoria i risultati di ricerca durante la consultazione di una discussione.

- restituisce i valori di inizializzazione dei parametri per i filtri di ricerca

L'architettura MVC del sistema dev'essere in grado di:

1. aggiornare in tempo reale i risultati in tempo reale a fronte di alterazioni dei valori impostati sui filtri di ricerca;

### 2.4.2. Filtri di ricerca

**Requisiti** 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  $u$ , ciascuno dei quali - in un dato istante - dev'essere ammissibile ( $u \in U_A$ ) o bloccato ( $u \in U_B$ ). Deve quindi valere che  $U_A \cap U_B = \emptyset$  e  $U_A \cup U_B = U$ .

Inizialmente tutti i valori per un certo parametro sono ammessi, ossia  $U_a = U$  e  $U_b = \emptyset$ : l'azione dell'utente consiste nell'alterare 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 soddisfa tutti i parametri. A seconda che ciascun contenuto assuma un singolo valore (autore, data di pubblicazione, argomento, ...) o una lista (emozioni, etichette, intenzioni, ...), il parametro risulta soddisfatto se e solo se:

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

**Parametri** Al termine dell'analisi viene redatta una lista completa dei parametri che soddisfano il requisito iniziale, visibile nella tabella 2.3 (tra parentesi si riporta il nome della classe corrispondente, ove disponibile).<sup>5</sup>

---

<sup>5</sup>Il termine ETICHETTE riportato nella tabella designa in seguito il criterio di classificazione illustrato nella sezione 2.3.



PROPRIETÀ	CRITERI DI CLASSIFICAZIONE
Autore ( <code>model.User</code> )	Argomento ( <code>model.Topic</code> )
Data di pubblicazione	Emozioni ( <code>model.Emotion</code> )
Tipo ( <code>model.Type</code> )	Etichette ( <code>model.Entity</code> )
Giudizi ( <code>model.Rating</code> )	Interessi ( <code>model.Interest</code> )
	Intenzioni ( <code>model.Intention</code> )

Tabella 2.3.: Lista dei parametri per i filtri di ricerca

**Classificazione** Per identificare i requisiti e i casi d'uso correlati ai filtri di ricerca conviene classificare i parametri in relazione alla natura dell'insieme  $U$  (ordinato, finito, ...) e alle possibili modalità d'interazione dell'utente (selezione di un intervallo di valori, impostazione di una soglia, ...), che determinano e alterano i sottoinsiemi  $U_A$  e  $U_B$ .

L'analisi si conclude con l'individuazione di quattro classi di filtri di ricerca:

#### A lista di valori

Il filtro a lista di valori consente di autorizzare o bloccare selettivamente ciascun elemento dell'insieme  $U$ . I casi d'uso includono:

- la consultazione della lista dei valori ammessi;
- la consultazione della lista dei valori bloccati;
- l'autorizzazione di un valore;
- il blocco di un valore;
- l'azzeramento del filtro (autorizzare tutti i valori).

#### A soglia di valore

Nel filtro a soglia di valore l'insieme  $U$  è ordinato e rappresenta un intervallo caratterizzato da un minimo  $\min \in U$  ed un 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 soddisfacibili risulta così definito:  $U_A = \{u \in U : u \geq x\}$ .

#### Ad intervallo

Nel filtro ad intervallo l'insieme  $U$  è ordinato e si caratterizza per un minimo  $\min \in U$  ed un massimo  $\max \in U$ . L'utente è libero di 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 interruttore**

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 elencati e descritti in precedenza. I casi d'uso prevedono:

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

LISTA DI VALORI	SOGLIA DI VALORE	INTERVALLO	INTERRUTTORE
Autore	Giudizi	Data di pubblicazione	Interessi
Argomento			
Emozioni			
Etichette			
Intenzioni			
Tipo			

Tabella 2.4.: Elenco dei parametri - suddivisi per tipo - per i filtri di ricerca

**Progettazione** La fase di progettazione deve integrare nell'architettura MVC della piattaforma le classi e i parametri dei filtri di ricerca:

**Model**

Ciascun tipo di filtro di ricerca è rappresentato dalla classe corrispondente nel package `model.filter`: `FList` (a lista di valori), `FRange` (ad intervallo), `FSwitch` (ad interruttore) ed `FValue` (a soglia di valore). Ciascuna di esse implementa un'interfaccia standard (`Filter`) per ridurre l'accoppiamento con le componenti del `CONTROLLER` e facilitare l'estensibilità del sistema (aggiunta di nuovi tipi di filtri).

A parità di classe i parametri differiscono - dal punto di vista di un filtro - 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 2.3 rappresenta - a livello di tipo - il corrispondente parametro per il



Figura 2.11.: Diagramma delle classi - Package *model.filter*

filtro di ricerca (il 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 rendere i primi indipendenti dai secondi facilitando l'aggiunta, l'aggiornamento e la rimozione dei parametri stessi, si definisce una classe astratta (`model.Metadata`), che tutte le suddette classi devono estendere. Essa include un metodo astratto (`Metadata.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, ossia ricavata dai risultati di ricerca (autore, data di pubblicazione, ...): nel primo caso la lista di valori viene memorizzata nella classe come campo dati statico.

I filtri vengono quindi creati a partire da una lista di valori, istanze di una sottoclasse concreta di `model.Metadata`: per favorire il disaccoppiamento si introduce una collezione di classi (`model.criteria`), che incapsulano tali valori e li rendono accessibili mediante un'interfaccia standard adeguata alla specifica classe del filtro: `CList` (a lista di valori), `CRange` (ad intervallo) e `CValue` (a soglia di valore).

Questo ulteriore livello di indirezione permette di conseguire un altro obiettivo, ossia poter inizializzare un filtro ad interruttore con qualsiasi tipo di criterio, nascondendone le specifiche alle classi che ne gestiscono essenzialmente lo stato di attivazione (`FSwitch` e `FVSwitch`).

Occorre garantire la corretta combinazione di istanze delle tre classi (v. tabella 2.4): la soluzione al problema consiste nell'applicazione del design pattern *Abstract Factory*, a garanzia di una corretta e consistente creazione dei filtri.

## View

Ciascuna classe di filtro richiede un componente grafico differente, che metta a disposizione dell'utente gli strumenti e le opzioni adeguate per consentire un'interazione completa rispetto a quanto previsto dai rispettivi casi d'uso: `FVList` (a lista

di valori), `FVRange` (ad intervallo), `FVSwitch` (ad interruttore) e `FVValue` (a soglia di valore). Le classi elencate implementano la medesima interfaccia (`ViewFilter`) e devono essere inizializzate con un oggetto di tipo `Filter`.

Occorre assicurare che il tipo dinamico di questo oggetto sia coerente e consistente rispetto al tipo di componente grafico che stiamo costruendo (v. tabella 2.5): per tale ragione il costruttore di ciascuna classe presa in esame richiede un argomento avente tipo statico corrispondente alla sottoclasse concreta di `Filter`, che rappresenta il filtro corrispondente.

In aggiunta si estende il *design pattern* *Abstract Factory* in modo da gestire in modo trasparente l'intero processo di creazione di un filtro a *runtime*: il parametro, il filtro e il componente grafico.

### Controller

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

Per indicare quali parametri specifici parametri siano compatibili con un certo generico, si definisce un'interfaccia da essi implementata e utilizzata come tipo base nei metodi di quest'ultimo componente.

Resta inoltre da considerare che ciascun parametro specifico recupera le informazioni di inizializzazione (i valori di  $U$ ) in modo diverso: una lista statica di classe o dinamica, basata su un'interrogazione ai risultati di ricerca: tale meccanismo dev'essere reso trasparente nell'interfaccia in questione.

**Abstract Factory** Il design pattern *Abstract Factory* consente di nascondere i dettagli relativi alle classi istanziate durante il processo di creazione di un filtro, riepilogate nella tabella ??.

### 2.4.3. Navigazione dei contenuti

La componente di navigazione e consultazione dei risultati rappresenta un elemento cardine del sistema software poiché ha il compito di presentare all'utente i risultati di ricerca e le relative informazioni in modo chiaro e ordinato, a prescindere dal numero di contenuti e dalle specifiche del dispositivo utilizzato (dimensione, risoluzione dello schermo, modalità d'interazione, ...).

Seguendo un approccio *bottom-up* ci si focalizza innanzi tutto sulla rappresentazione di un contenuto informativo (proprietà, metadati, relazioni) e in seguito sulla progettazione dell'interfaccia, che li presenti all'utente in modo da facilitarne la fruizione.

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

	Metadata	Criterion	Filter	FilterController	FilterView
<i>Argomento</i>					
<i>Attinenza</i>					
<i>Autore</i>					
<i>Data di pub.</i>					
<i>Emozioni</i>					
<i>Giudizi</i>					
<i>Intenzioni</i>					
<i>Interessi</i>					
<i>Tipo</i>					

Tabella 2.5.: Famiglie di prodotti

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.

Ciò consente di rendere più efficiente lo sfruttamento dello spazio disponibile e di comunicare all'utente le sole informazioni rilevanti e utili per stabilire la pertinenza del contenuto: ove il contenuto soddisfi le aspettative dell'utente, suscitandone l'interesse, quest'ultimo può accedere e consultare le informazioni aggiuntive.

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

Tabella 2.6.: Famiglie di prodotti

Una volta distinte le informazioni in essenziali o aggiuntive occorre scegliere - per ciascuna di esse - la modalità di rappresentazione (grafica o testuale) più adatta per esprimerla e comunicarla all'utente, in modo da risultare intuitivamente comprensibile. Per soddisfare tali requisiti di qualità si fissano alcune linee guida:

- mantenere un bilanciamento equilibrato tra informazioni testuali e grafiche, garantendo un adeguato livello di accessibilità (cite::WCAG);
- utilizzare in modo limitato e non esclusivo il colore;
- ricorrere a forme grafiche per esprimere informazioni quantitative o definite su un insieme limitato di valori, così da facilitare rispettivamente l'associazione 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 direttamente 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) e semantica 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 alla primeva esigenza di renderli immediatamente riconoscibili e comprensibili da qualsiasi categoria di utenti. La scelta della forma associate a ciascun tipo si avvale di analisi sociologiche svolte da altri membri del team di progetto.

### Titolo

Il titolo è riportato in forma testuale.

**Classi di contenuti** Dal momento che la rappresentazione visiva di ciascuna classe di contenuti differisce essenzialmente per la forma elementare scelta, che a sua volta influenza il posizionamento ed il *rendering* delle informazioni, si ritiene opportuno definire una gerarchia di classi ove:

- la classe base (**ContentView**) rappresenta il componente grafico per un generico contenuto (incluse le relative informazioni associate, tra cui le reciproche relazioni);
- le sottoclassi (**CVAnswer**, **CVEvent**, ...) identificano le possibili classi di un contenuto, a ciascuna delle quali è associata la specifica forma elementare e un insieme di regole per la visualizzazione delle informazioni.

Questa scelta progettuale intende facilitare l'aggiornamento delle specifiche di *rendering* di una classe di contenuti (variazione della forma o dell'organizzazione delle informazioni) e ne facilita l'integrazione di sostitutive o aggiuntive.



Figura 2.12.: Diagramma delle classi - Package *view.content*

**Cronologia** I risultati di una ricerca devono essere presentati secondo una struttura chiara ed ordinata, che ne agevoli la consultazione da parte dell'utente e che sia adeguatamente fruibile a prescindere dal numero di risultati o dal dispositivo utilizzato (risoluzione, dimensione dello schermo, modalità d'interazione, ...).

**UC.3 - Consultazione dei risultati di ricerca** La rilevanza di un contenuto informativo rispetto ai criteri di ricerca è infine determinata dal grado di corrispondenza rispetto alle entità cercate: maggiore è il numero di riscontri, maggiore sarà approssimativamente la rilevanza attribuita nel contesto della ricerca.

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 principali:

**Corrispondenza completa:**  $E_s \subseteq E_c$

Al contenuto risultano assegnate tutte le entità cercate (massima attinenza).

**Corrispondenza parziale:**  $E_s \cap E_c \neq \emptyset$

Al contenuto risulta assegnata parte delle entità cercate (media attinenza).

**Nessuna corrispondenza:**  $E_s \cap E_c = \emptyset$

Al contenuto non risulta assegnata alcuna entità cercata (attinenza nulla).



## 3. Valutazioni finali

### 3.1. Obiettivi

Nel corso dello stage sono state svolte due attività correlate, ma radicalmente diverse sul fronte del 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 nella piattaforma;
2. l'analisi e la progettazione di un'interfaccia grafica per la consultazione dei risultati di una ricerca sui contenuti informativi.

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

#### 3.1.1. Criterio di classificazione

#### 3.1.2. Interfaccia grafica

### 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 S(L)S sin dai primi albori ed ha rappresentato un'opportunità di crescita professionale molto rilevante.

Sin dalle attività precedenti e in particolare nel corso dello stage ho potuto apprezzare gli enormi benefici - in termini di competenze professionali acquisite - derivanti dall'operare all'interno di un progetto caratterizzato 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 diverse.

L'analisi, la progettazione e lo sviluppo della piattaforma S(L)S 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).

#### 3.2.1. Criterio di classificazione

La progettazione del criterio di classificazione ha richiesto nella fase iniziale dello stage 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 coniugare - nella soluzione più semplice e adatta alla piattaforma *Social (Life) Shuttle* - gli aspetti essenziali riuscendo al contempo a superarne i limiti evidenziati.

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 riguardano 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.

#### 3.2.2. Interfaccia grafica

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. Queste, seppur preliminari, hanno rappresentato un contributo cruciale nel processo di identificazione e comprensione delle variegate 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 effettuare l'analisi dei requisiti, che ha permesso di delineare i casi d'uso degli utenti e individuare alcuni requisiti fondamentali: in questa fase si è rivelata decisiva per riuscire a svolgere le attività previste

in modo da rispettare il piano di lavoro un'organizzazione del lavoro 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 abbastanza 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 e di straniamento riscontrato nell'ambito del progetto S(L)S è rappresentato dal taglio fortemente multidisciplinare, a cui tutti sono chiamati a concorrere ed attingere e che costringe a rivedere un modello di pensiero ed una *forma mentis* principalmente preparate e abituate a focalizzarsi sull'aspetto principalmente tecnico e tecnologico, avente maggior pertinenza ed interesse rispetto alle figure professionali degli informatici.

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

- il percorso continua a livello professionale

## A. 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.

### G

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

Editor di testo ufficiale dell'ambiente desktop GNOME.

### L

**LaTeXila** - <http://projects.gnome.org/latexila/>

Editor 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.

## B. 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  $d_i$  ( $E_i \subset E$ ).

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

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

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

## 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  $d_i$  ( $b_{i,k} \in B_i$ ).

## Codice SQL

### Etichette

**Ricerca di un'etichetta** Restituisce le stringhe 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
```

## C. Interfaccia grafica

### Legenda

$U$

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. ( $u \in U$ )

### Casi d'uso

### Requisiti

### Architettura

Model

View

Controller

# 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].