DaCENA

Documento di Design

Nome gruppo: SOKE
Numero assegnato: 24

Componenti del gruppo:
761678 Angiolillo Andrea
761702 Belingheri Omar
761763 Khayam Adam

INTRODUZIONE

Scopo del documento

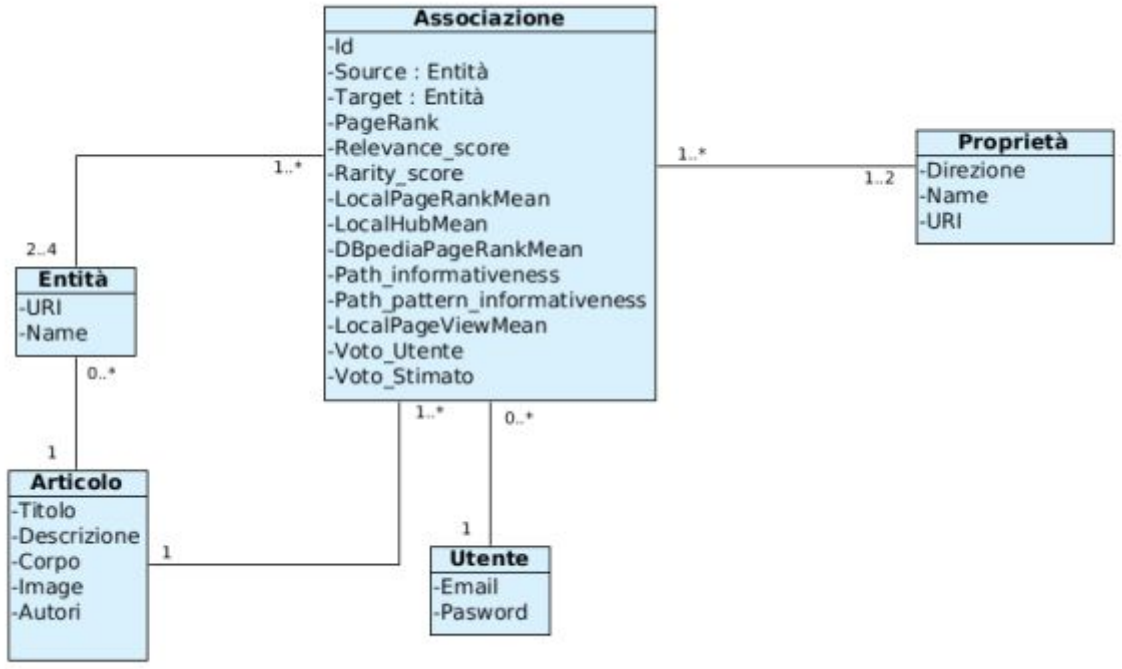
Lo scopo di questo documento è quello di illustrare le scelte architetturali che sono state adottate nell'implementazione del progetto DaCena. DaCena è un software che consente agli utenti di visualizzare le informazioni contenute negli articoli di giornale che leggono mediante dei Knowledge Graphs: in questo modo essi vengono facilitati nella ricerca di informazioni aggiuntive correlate a quelle contenute nell'articolo. Inoltre, gli utenti possono migliorare la qualità del grafo visualizzato mediante un sistema di votazione delle associazioni in esso contenute: un algoritmo di online learning aggiorna conseguentemente il grafo, tenendo conto sia dell'importanza "oggettiva" delle entità che delle opinioni dell'utente. Per tutti i dettagli sui casi d'uso, i requisiti, e il funzionamento dell'applicazione, si veda il documento relativo.
Nelle prossime sezioni presenteremo, in ordine:

1. architettura del problema
2. architettura logica
3. architettura concreta
4. architettura di deployment
5. scelte implementative

Architettura del Problema

Modello dei dati

Per poter capire i diagrammi di attività mostrati nei paragrafi successivi, è innanzitutto necessario avere chiaro quali dati vengono utilizzati nel nostro sistema (e quindi, nelle attività). Di seguito il diagramma del modello dei dati.



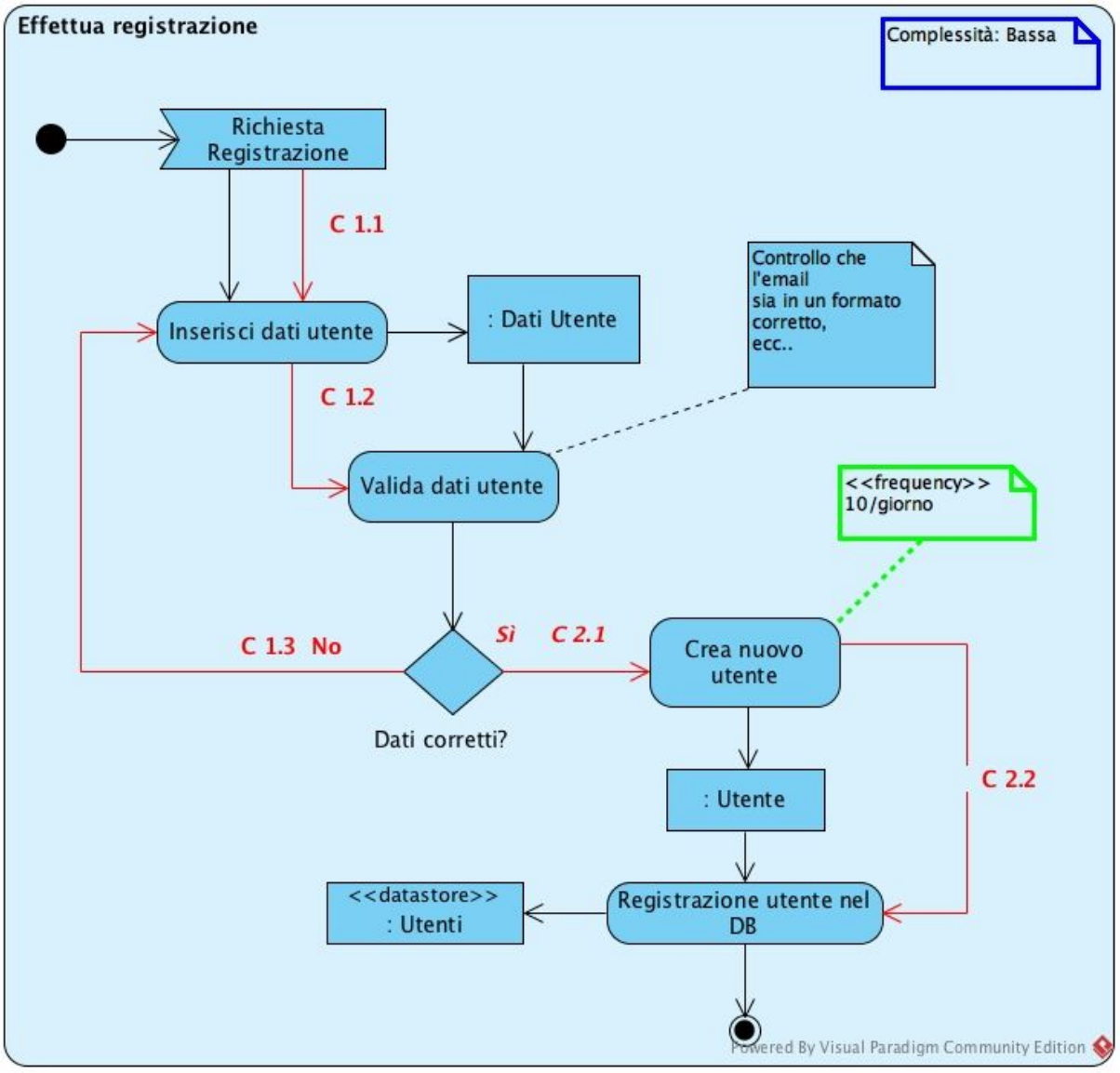
Diagrammi delle attività

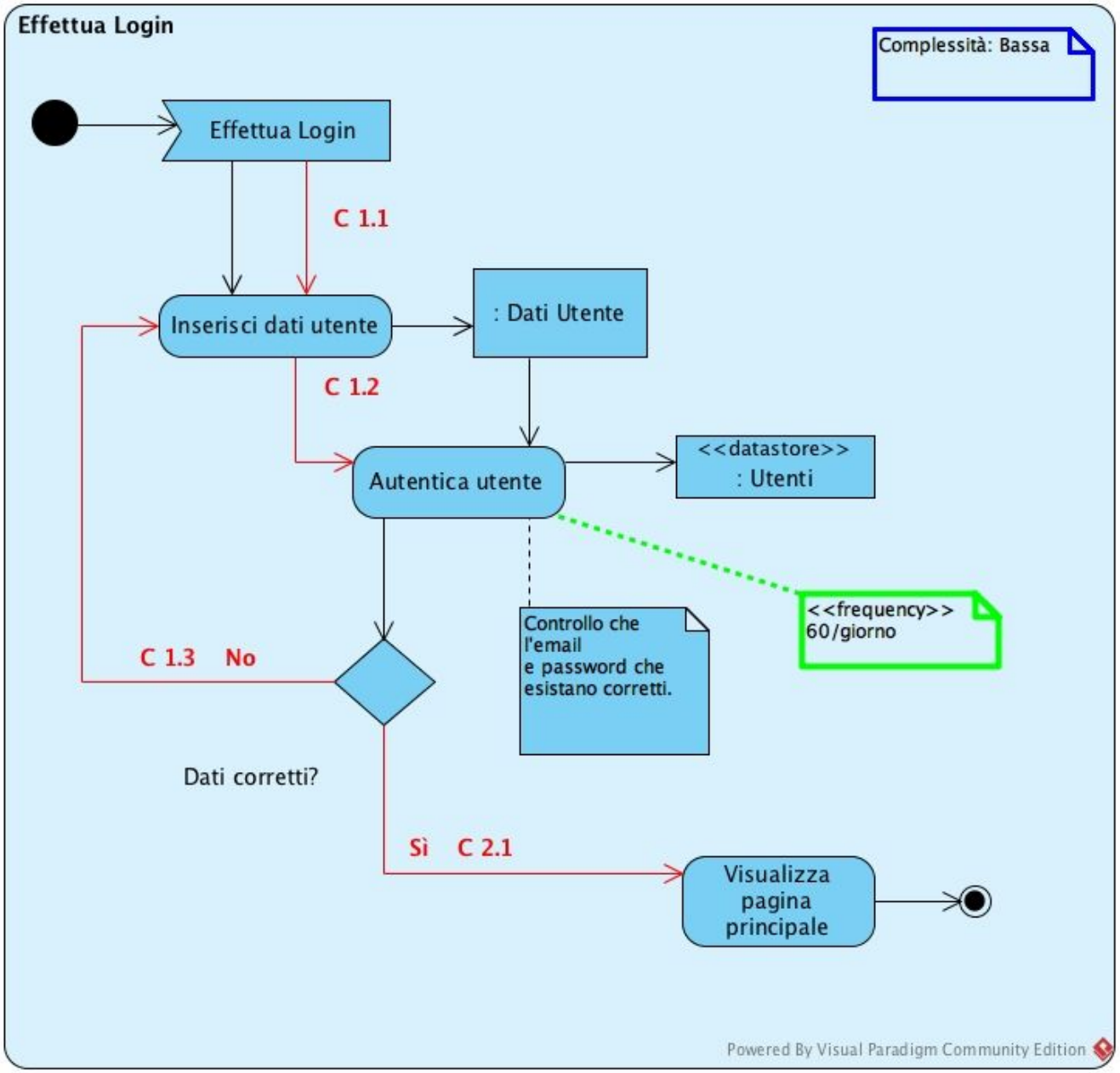
Le attività sono state individuate a partire dai requisiti funzionali (specificati nell'apposito documento), e sono le seguenti:

- Effettua registrazione
- Effettua login
- Modifica informazioni utente
- Aggiungi articolo
- Elimina articolo
- Estrai associazioni
- Effettua clustering
- Visualizza grafo
- Genera grafo

Registrazione e Login

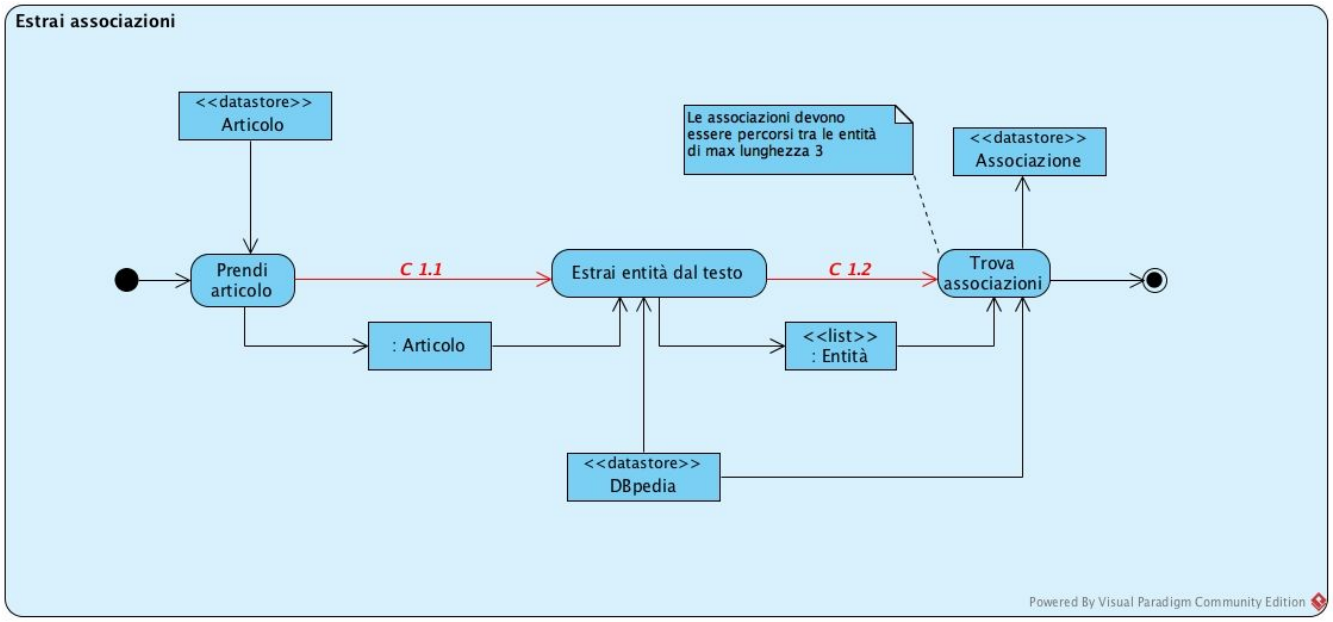
Le prime attività che presentiamo sono quelle che consentono all'utente di registrarsi e loggarsi sulla piattaforma. Per la registrazione, è sufficiente che l'utente fornisca come dati un'email valida e una password.




Effettua Login
Complessità: Bassa
Effettua Login
C 1.1
Inserisci dati utente
: Dati Utente
C 1.2
Autentica utente
<<datastore>>
: Utenti
<<frequency>>
60/giorno
Controllo che
l'email
e password che
esistano corretti.
C 1.3 No
Dati corretti?
Sì C 2.1
Visualizza
pagina
principale
Powered By Visual Paradigm Community Edition

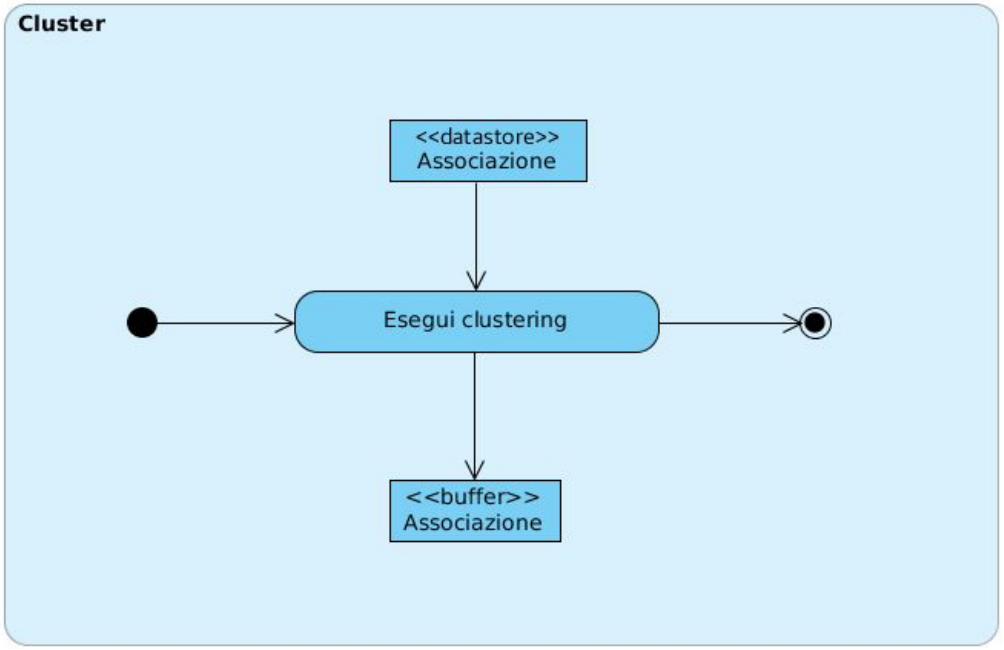
Estrai Associazioni

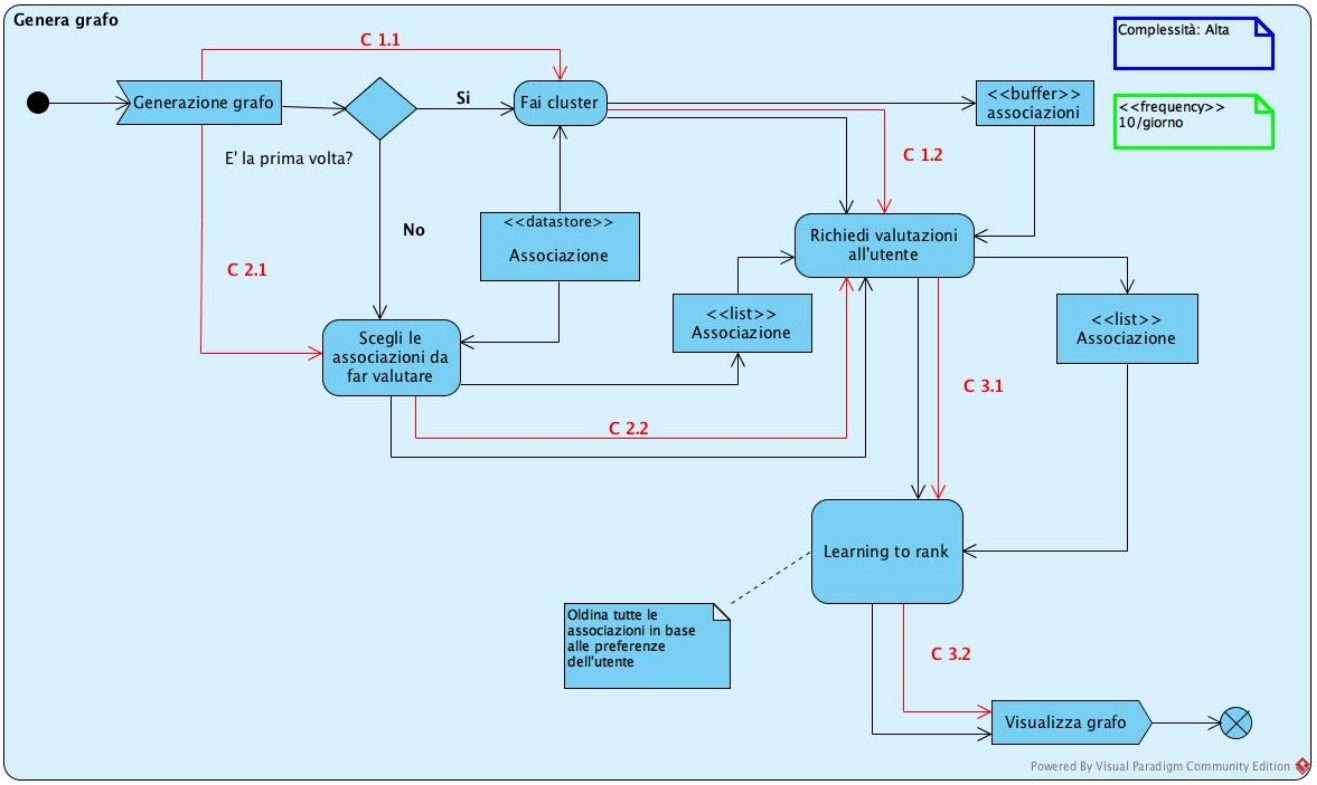
Questa attività si occupa di estrarre, dato un articolo, le entità più importanti presenti in esso e le relazioni che le collegano. Per fare ciò si sfrutta DBpedia, un database contenente informazioni strutturate prese da Wikipedia.

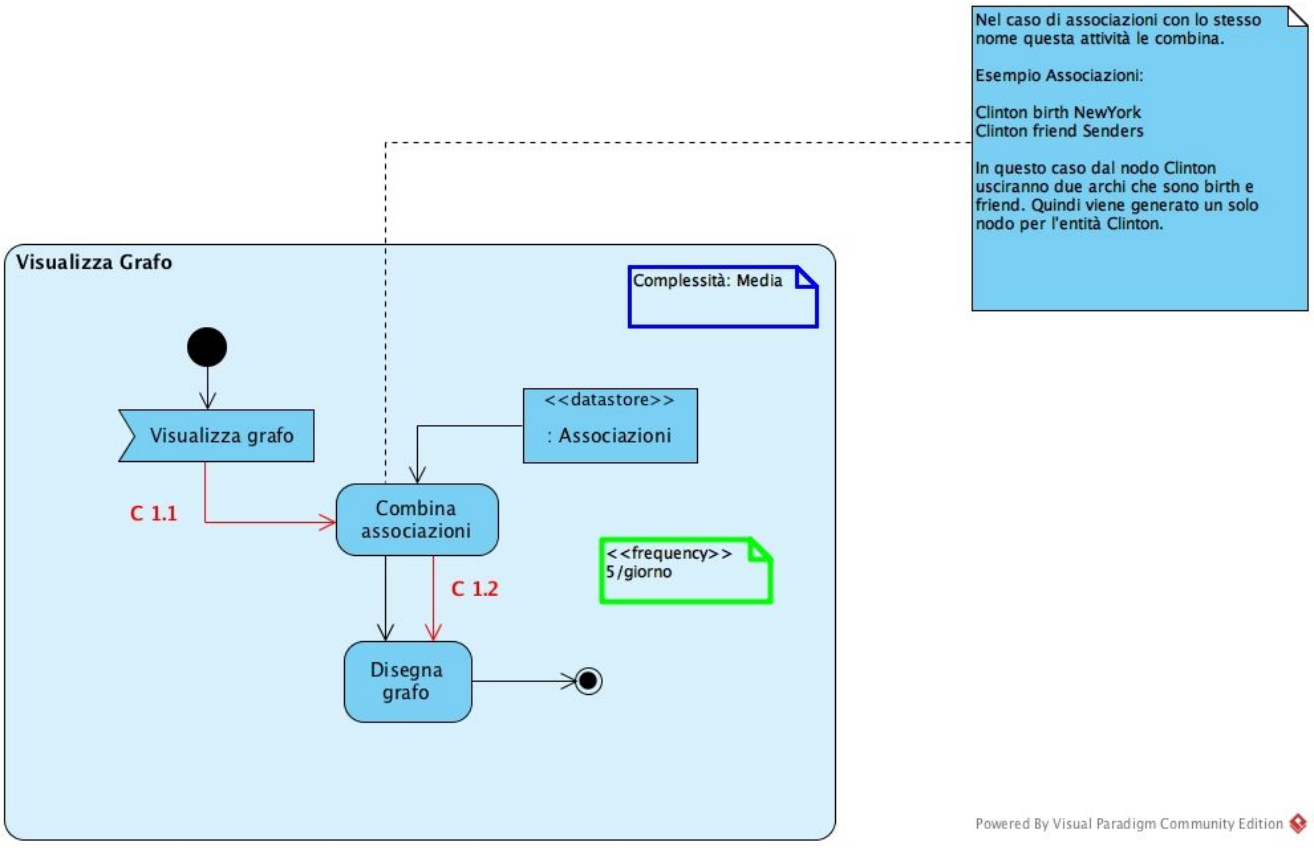


Clustering, generazione e visualizzazione grafo

Per capire come il grafo viene generato, bisogna avere chiaro il funzionamento dell'algoritmo di learning sottostante. Inizialmente viene fatto clustering sulle associazioni contenute nell'articolo, in modo da scegliere quali mostrare all'utente: quest'ultimo le valuterà in base a quanto le ritiene importanti, e un algoritmo di apprendimento cercherà di catturare le preferenze dell'utente, aggiornando il grafo in base ai feedback appena ottenuti. Se l'utente non sarà soddisfatto del grafo ottenuto, potrà decidere di valutare nuove associazioni in futuro.

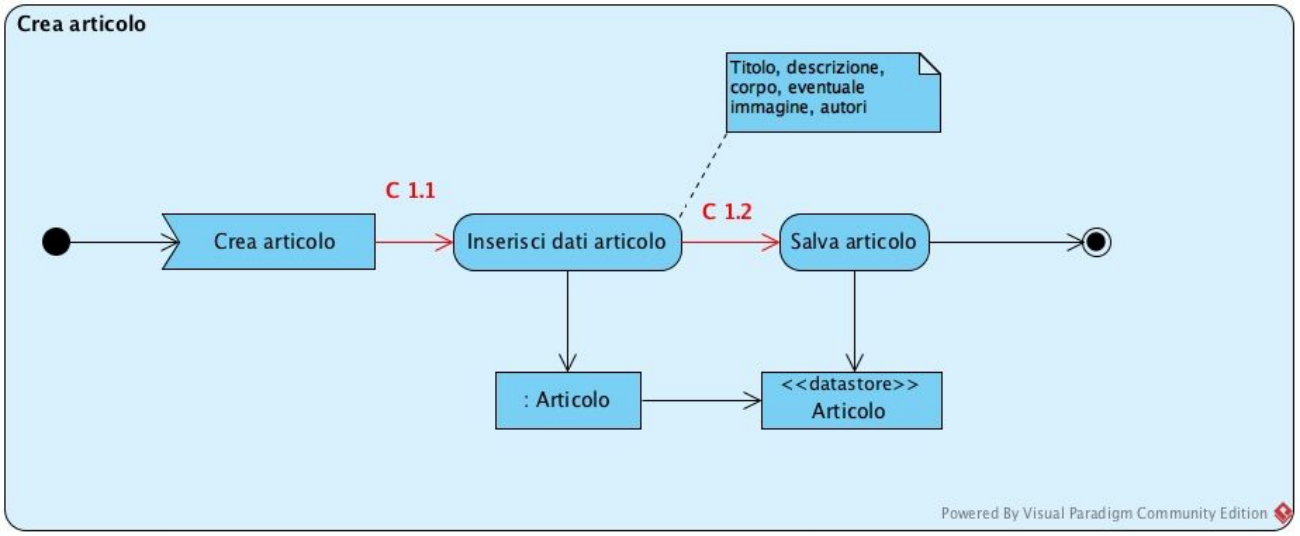

Cluster
<<datastore>>
Associazione
Esegui clustering
<<buffer>>
Associazione

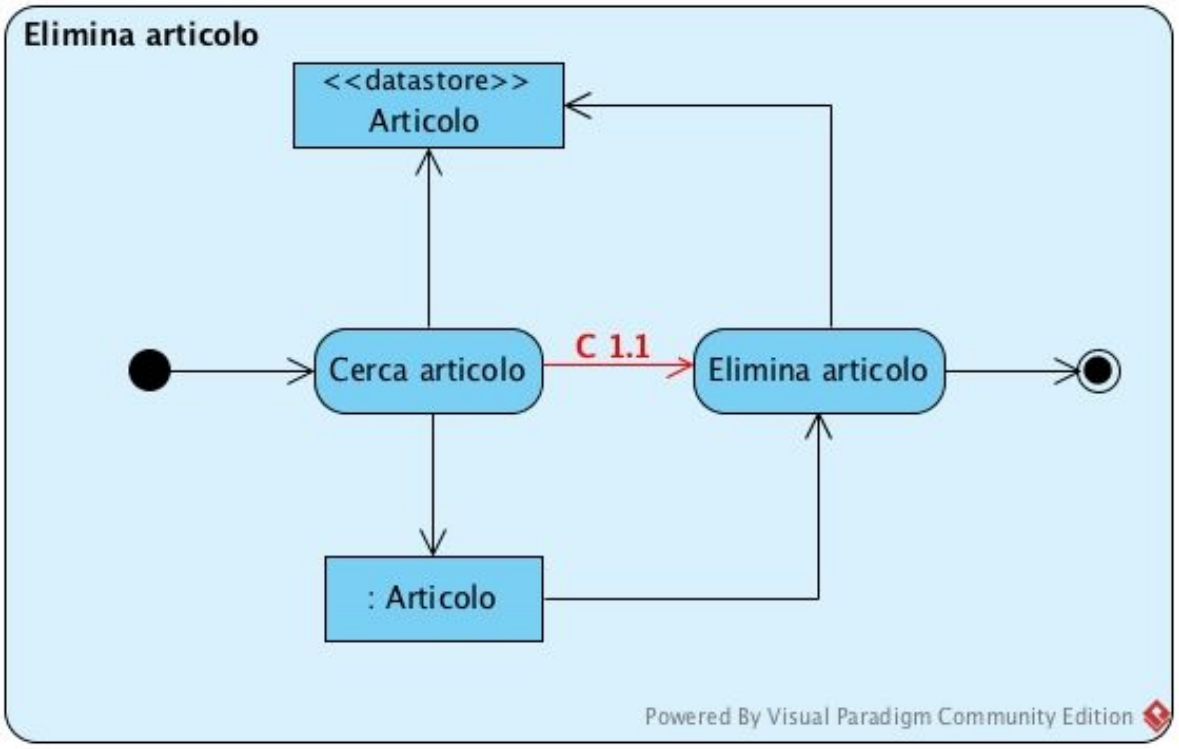

Genera grafo
Complessità: Alta
C 1.1
Generazione grafo
Si
Fai cluster
<<buffer>>
associazioni
<<frequency>>
10/giorno
E' la prima volta?
C 1.2
No
<<datastore>>
Associazione
Richiedi valutazioni
all'utente
C 2.1
<<list>>
Associazione
<<list>>
Associazione
Scegli le
associazioni da
far valutare
C 3.1
C 2.2
Learning to rank
Oldina tutte le
associazioni in base
alle preferenze
dell'utente
C 3.2
Visualizza grafo
Powered By Visual Paradigm Community Edition



Crea ed elimina articolo

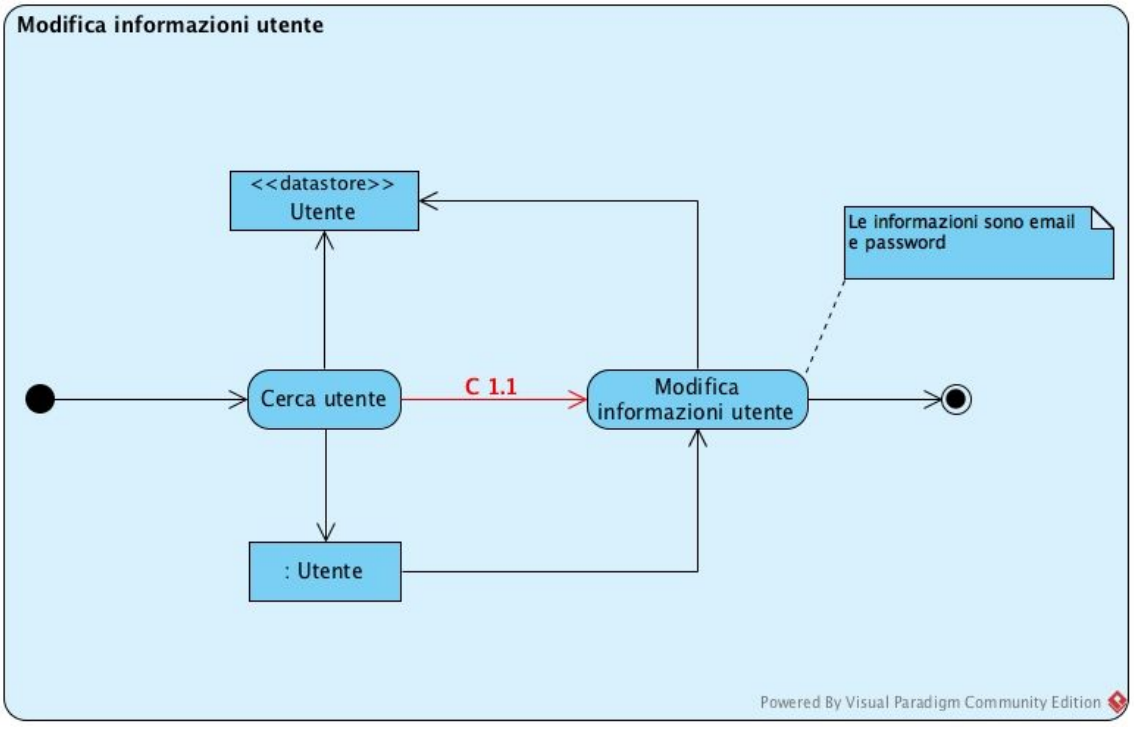
L'amministratore può aggiungere e rimuovere articoli dal sistema DaCENA.





Modifica Informazioni Utente

L'amministratore può modificare le informazioni dell'utente, come la sua email e la sua password.



Architettura Logica

In questo capitolo verranno mostrati i principali componenti software del nostro sistema, e si spiegheranno le scelte effettuate nella loro definizione.
Innanzitutto proponiamo due soluzioni, che verranno poi confrontate per scegliere quale sia la migliore:

- Partizionamento a 4 componenti:
- Partizionamento a 6 componenti.

Componente	Attività	Numero istanze
Gestore Client	Effettua registrazione	5000
	Effettua login	
Gestore Amministratore	Modifica informazioni utente	5
	Aggiungi articolo	
	Elimina articolo	
	Estrai associazioni	
Gestore Learner	Genera grafo	5000
	Visualizza grafo	
Cluster	Effettua clustering	5

Componente	Attività	Numero istanze
Gestore Client	Effettua registrazione	5000
	Effettua login	
Gestore Amministratore	Modifica informazioni utente	5

	Aggiungi articolo	
	Elimina articolo	
Gestore Learner	Genera grafo	5
Visualizzatore	Visualizza grafo	5000
Cluster	Effettua clustering	5
Annotator	Estrai associazioni	1

Footprint

Il footprint è stato calcolato sulle seguenti dimensioni:

- Frequenza: la frequenza con cui l'attività viene eseguita;
- Complessità: la complessità dell'attività;
- Ritardo: il tempo trascorso tra l'inizio e la fine dell'attività;
- Extra-Flow: il flusso di informazioni verso gli attori esterni del sistema;
- Intra-Flow: il flusso di dati verso le altre componenti;
- Sharing: la condivisione delle informazioni con le altre componenti.

Per le prime tre dimensioni, i valori del footprint sono stati calcolati numericamente. Per quanto riguarda le rimanenti, si è cercato di stimare dei valori partendo dalle informazioni rappresentate nei diagrammi di attività.
Verrà illustrato ora il procedimento seguito per la stima numerica del footprint per le dimensioni della frequenza, complessità e del ritardo.

La prima fase è stata quella di assegnare dei valori alle tre dimensioni considerate per ogni attività.

Attività	Frequenza	Complessità	Ritardi
Effettua registrazione	Ore	Bassa	Secondi
Effettua login	Ore	Bassa	Secondi
Modifica informazioni utente	Giorni	Basso	Secondi
Aggiungi articolo	Giorni	Bassa	Secondi
Elimina articolo	Giorni	Bassa	Secondi
Estrai associazioni	Giorni	Media	Minuti

Effettua clustering	Giorni	Media	Millisecondi
Visualizza grafo	Giorni	Media	Millisecondi
Genera grafo	Giorni	Alta	Millisecondi

Ogni dimensione può assumere i seguenti valori:

- Frequenza = {Secondi, Minuti, Ore, Giorni}
- Complessità = {Bassa, Media, Alta}
- Ritardo = {Millisecondi, Secondi, Minuti }

Una volta mostrati i valori assunti dalle attività per le tre dimensioni, abbiamo calcolato il footprint di ogni singolo componente C calcolando lo spread nel seguente modo:

$$\text{Spread}_{D,C} = \frac{\textit{numero di valori assunti da D nella componente C}}{\textit{numero totale di valori assumibili da D}}$$

Calcoliamo dunque, per ogni soluzione di partizionamento ipotizzata, il footprint delle dimensioni complessità, frequenza, ritardo.

Quattro componenti - footprint

Footprint Complessità			
Gestore Client	Gestore Amministratore	Gestore Learner	Cluster
Bassa	Bassa	Alta	Media
Bassa	Bassa	Media	
	Bassa		
	Media		
Spread			
⅓ = 0.33	⅔ = 0.66	⅔ = 0.66	⅓ = 0.33
Media			
0.50			

Footprint Frequenza			
Gestore Client	Gestore Amministratore	Gestore Learner	Cluster
Ore	Giorni	Secondi	Minuti
Ore	Giorni	Secondi	
	Giorni		
	Giorni		
Spread			
¼ = 0.25	¼ = 0.25	¼ = 0.25	¼ = 0.25
Media			
0.25			

Footprint Ritardi			
Gestore Client	Gestore Amministratore	Gestore Learner	Cluster
Secondi	Secondi	Millisecondi	Millisecondi
Secondi	Secondi	Millisecondi	
	Secondi		
	Minuti		
Spread			
⅓ = 0.33	⅔ = 0.5	⅓ = 0.33	⅓ = 0.33
Media			
0.37			

Sei componenti - footprint

Footprint Complessità					
Gestore Client	Gestore Amministratore	Gestore Learner	Visualizzatore	Cluster	Annotator
Bassa	Bassa	Alta	Media	Media	Media
Bassa	Bassa				
	Bassa				
Spread					
⅓ = 0.33	⅓ = 0.33	⅓ = 0.33	⅓ = 0.33	⅓ = 0.33	⅓ = 0.33
Media					
0.33					

Footprint Frequenza					
Gestore Client	Gestore Amministratore	Gestore Learner	Visualizzatore	Cluster	Annotator
Ore	Giorni	Secondi	Secondi	Minuti	Giorni
Ore	Giorni				
	Giorni				
Spread					
¼ = 0.25	¼ = 0.25	¼ = 0.25	¼ = 0.25	¼ = 0.25	¼ = 0.25
Media					
0.25					

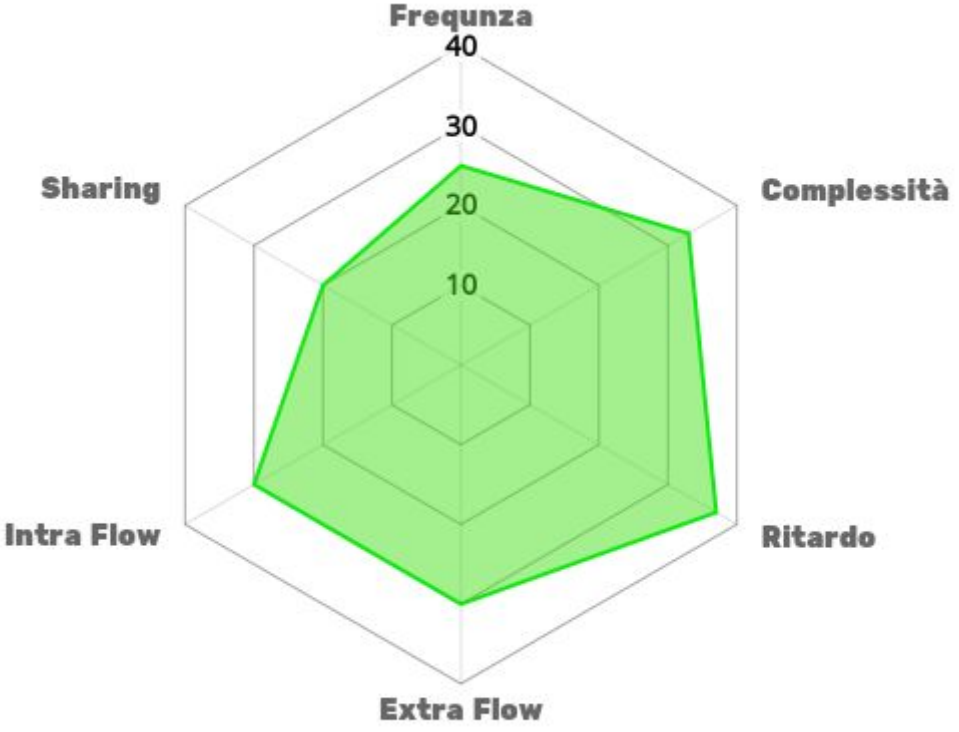
Footprint Ritardi					
Gestore Client	Gestore Amministratore	Gestore Learner	Visualizzatore	Cluster	Annotator
Secondi	Secondi	Millisecondi	Millisecondi	Millisecondi	Minuti
Secondi	Secondi				
	Secondi				
Spread					
⅓ = 0.33	⅓ = 0.33	⅓ = 0.33	⅓ = 0.33	⅓ = 0.33	⅓ = 0.33
Media					
0.33					

Una volta calcolato il footprint per ogni singola componente rispetto ad una data dimensione, il footprint totale è ricavato facendo la media dei footprint di ogni singola componente. Nella tabella seguente vengono mostrati i valori totali delle dimensioni, compresi nell'intervallo (0, 100], del footprint totale delle due soluzioni considerate. Si noti come lo sharing sia diminuito notevolmente dalla prima alla seconda soluzione: separando il visualizzatore e il gestore learner in due componenti diverse, si riesce a istanziare quest'ultimo un numero molto inferiore di volte, portando così ad un enorme miglioramento.

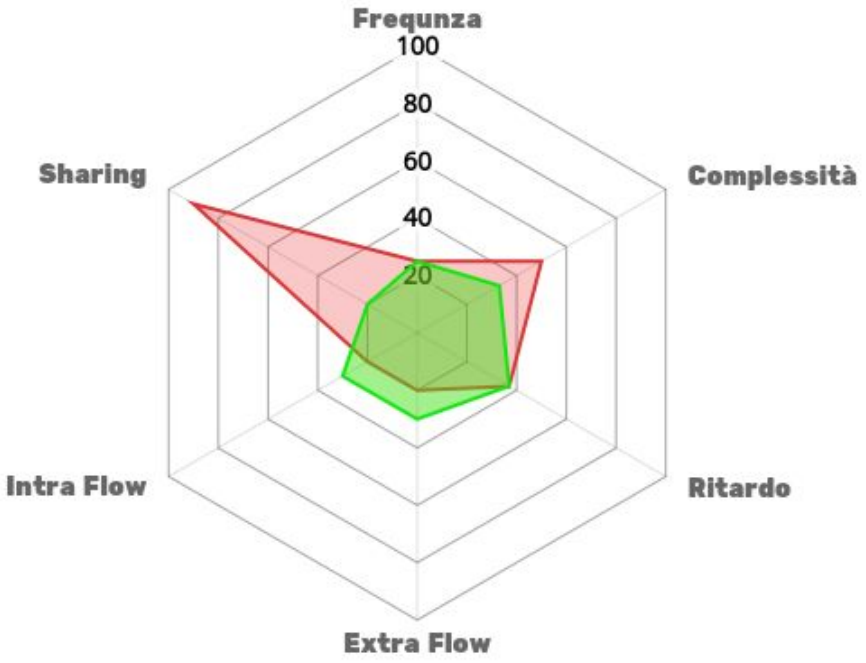
	4 Componenti	6 Componenti
Frequenza	25	25
Complessità	50	33
Ritardo	37	37
Extra Flow	20	30
Intra Flow	20	30
Sharing	90	20

Footprint 4 componenti

Footprint 6 componenti



Footprint a confronto



Quest'ultimo grafo mostra un confronto tra la footprint della soluzione a quattro componenti (rosso) e quella a 6 componenti (verde). Si è scelto di adottare la seconda soluzione per ridurre notevolmente lo sharing e la complessità.

Gestore Client

Questo componente fornisce delle funzionalità base all'utente, come accedere alla piattaforma attraverso l'operazione di login oppure registrare un account all'interno del sistema.

Gestore Amministratore

Componente per la gestione dei dati sugli utenti e gli articoli. Consente all'amministratore di aggiungere ed eliminare articoli, e di modificare i dati sugli utenti (email e password).

Gestore Learner

Componente che si occupa di eseguire l'algoritmo di online learning to rank, che serve a catturare le preferenze dell'utente in base ai suoi interessi quando visualizza i grafi e ne vota le associazioni.

Cluster

Componente che effettua clustering sulle associazioni estratte dagli articoli.

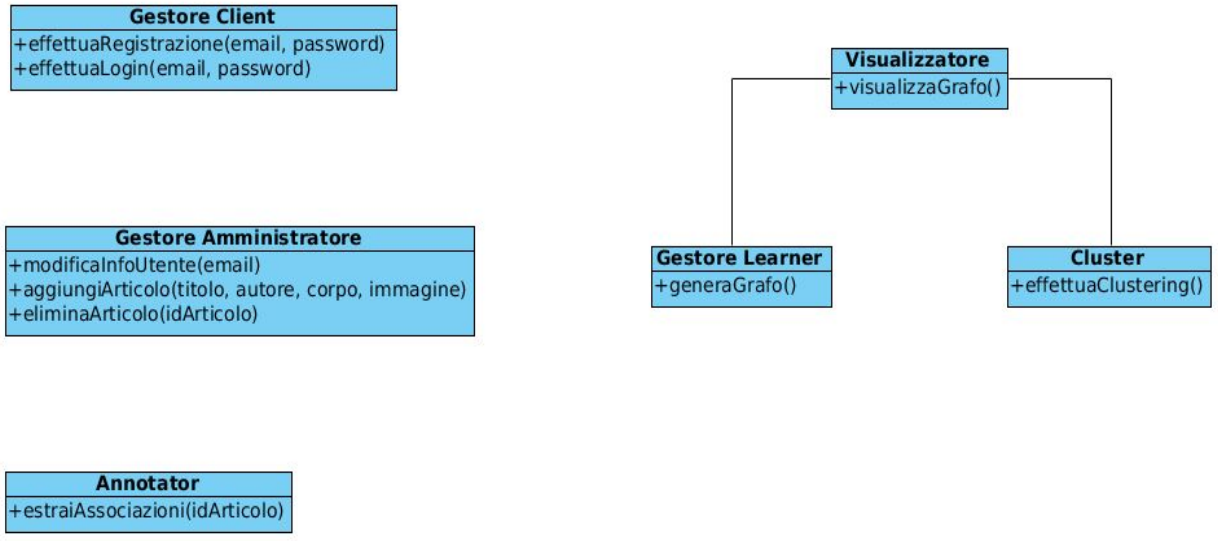
Annotator

Componente che, dato un articolo, si occupa di estrarne le entità principali e le relazioni che le collegano.

Architettura Concreta

Diagramma delle componenti concrete

Presentiamo ora le componenti concrete del sistema DaCENA. Le attività elencate in questo documento sono state raggruppate nelle sei componenti concrete descritte precedentemente. Il diagramma sottostante mostra le componenti e i loro metodi principali, che vengono invocati dall'applicazione.

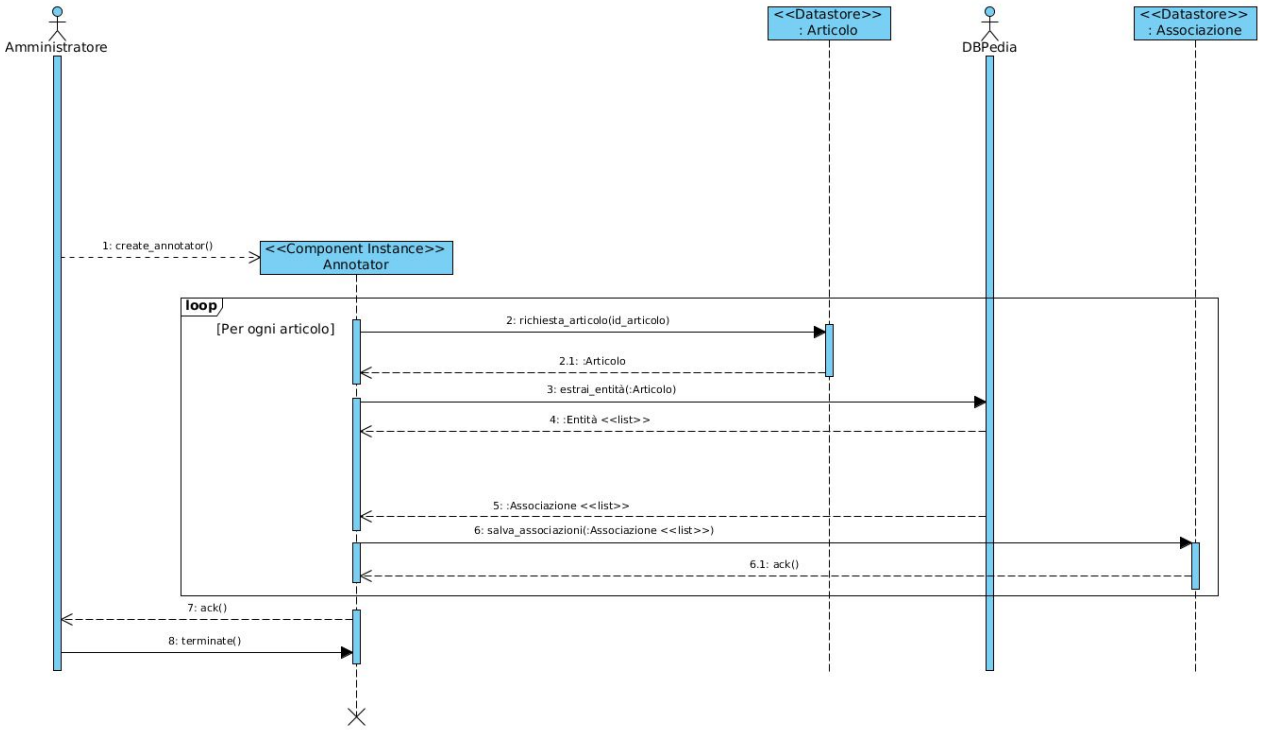


Diagrammi di sequenza

Mostriamo di seguito alcuni diagrammi di sequenza, focalizzandoci sulle attività più complesse e che coinvolgono più attori e componenti.

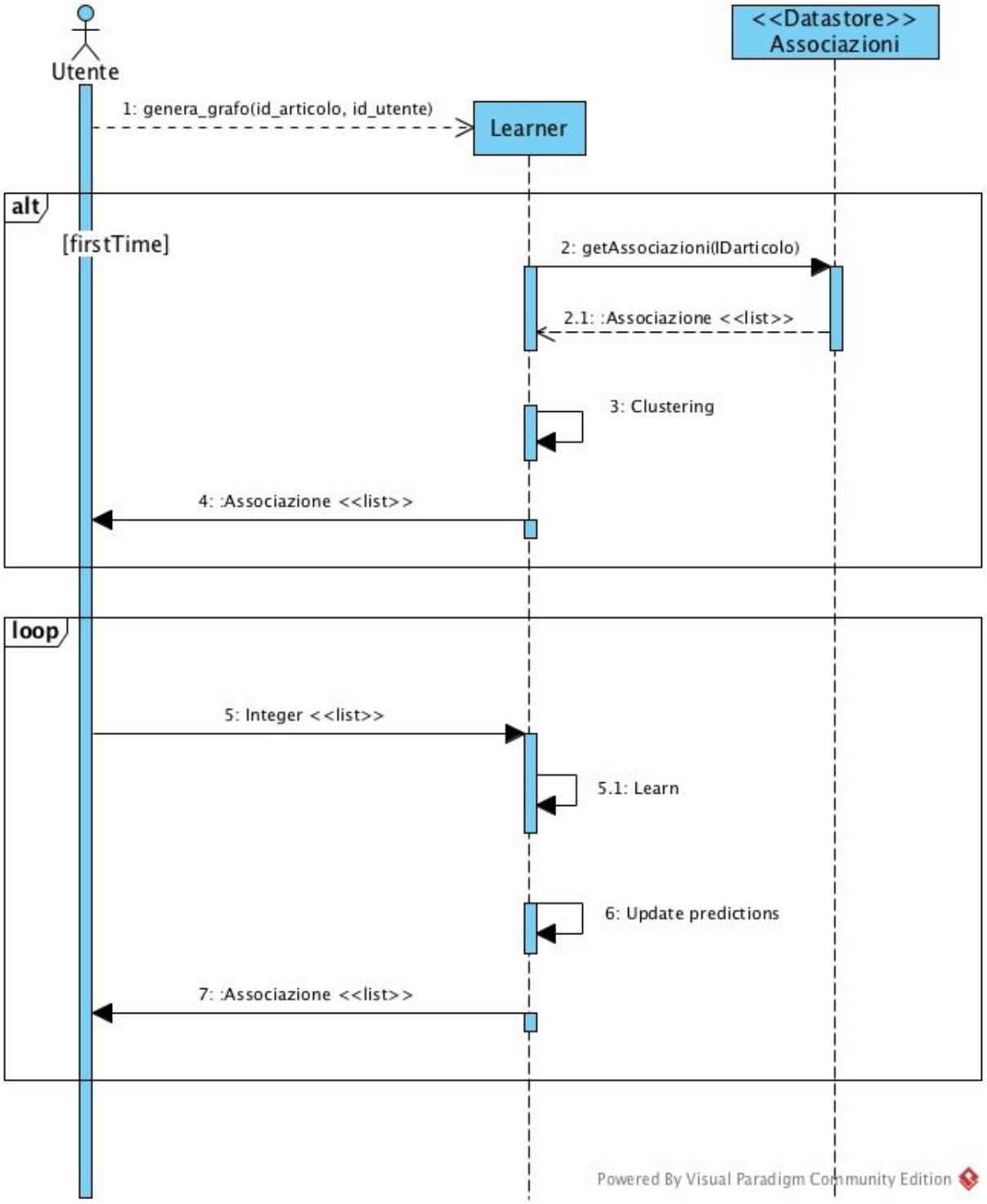
Estrai associazione

In questo diagramma di sequenza viene mostrata la sequenza di operazioni che vengono svolte per l'estrazione delle associazioni.



Learning

In questo diagramma vengono mostrate le operazioni svolte per la fase di online learning



Architettura di Deployment

Di seguito verranno mostrati due diagrammi di deployment, che rappresentano due modi alternativi di organizzare i componenti.

Diagramma di deployment A

In questa prima soluzione si è cercato di alleggerire il client, che viene utilizzato semplicemente come interfaccia che carica gli articoli e che consente all'utente di comunicare con il server.

Pro:

- Il client si occupa solo di visualizzare l'articolo

Contro:

- Ogni volta che l'utente vuole visualizzare un articolo viene effettuata una richiesta al server

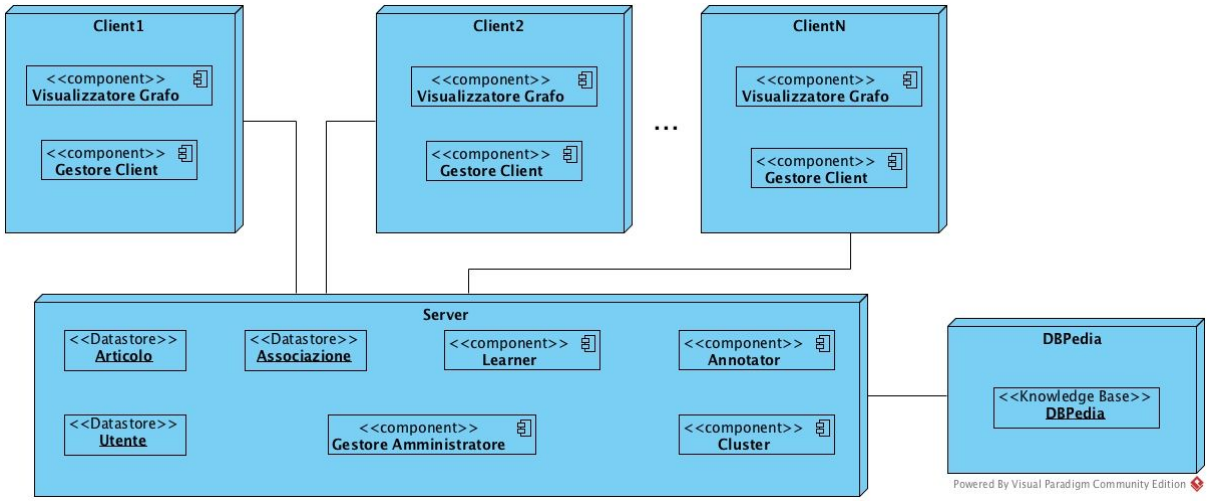


Diagramma di deployment B

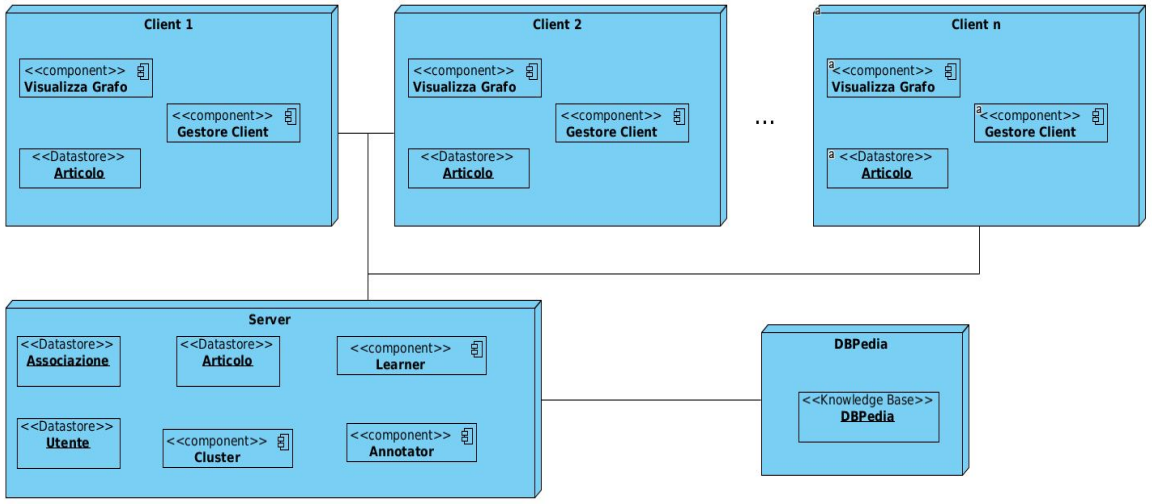
In questa soluzione si è scelto di fornire al client un datastore dove vengono salvati solo gli articoli visualizzati almeno una volta dall'utente.

Pro:

- Vengono effettuate meno richieste al server: quando l'utente decide di visualizzare per la prima volta un articolo, viene generata una richiesta al server per reperirlo. Successivamente alla prima visualizzazione, l'applicazione utilizza il datastore interno per visualizzare l'articolo.

Contro:

- Viene inserito un datastore nel client



Il secondo diagramma di deployment è stato scelto come soluzione finale in quanto si è voluto diminuire il più possibile il numero di richieste al server, anche a costo di appesantire leggermente il client (che comunque non deve fare calcoli computazionalmente pesanti).

Scelte Implementative

In questa sezione verranno presentate le tecnologie che sono state utilizzate per implementare il software.

Python

Si è scelto di sviluppare il backend in python, vista la grande disponibilità di librerie matematiche necessarie per il clustering e l'online learning to rank.
Alcune delle principali librerie utilizzate sono le seguenti:

- Pandas e Numpy: per la manipolazione delle strutture dati;
- Scikit-learn: per i metodi di clustering e per gli algoritmi di apprendimento automatico. Ne sono stati provati diversi, e alla fine si è scelto *Multinomial Naive Bayes* in quanto è risultato essere il più efficiente.

L'algoritmo scelto per effettuare il clustering è stato Dirichlet, necessario in situazioni in cui non si conosca il numero K di cluster da generare (come nel nostro caso).

Electron

Il framework Electron permette di definire applicazioni web cross-platform utilizzando Javascript, HTML e CSS.
La scelta è ricaduta su questo framework in modo da rendere il nostro software disponibile su tutti i sistemi operativi, requisito che ci eravamo imposti.