

Deliverable 3  
Friendly note

## **DOCUMENTO DI ARCHITETTURA**

### **INDICE**

DIAGRAMMA DELLE CLASSI

CODICE IN OCL

DIAGRAMMA DELLE CLASSI CON CODICE OCL

## Diagramma delle classi

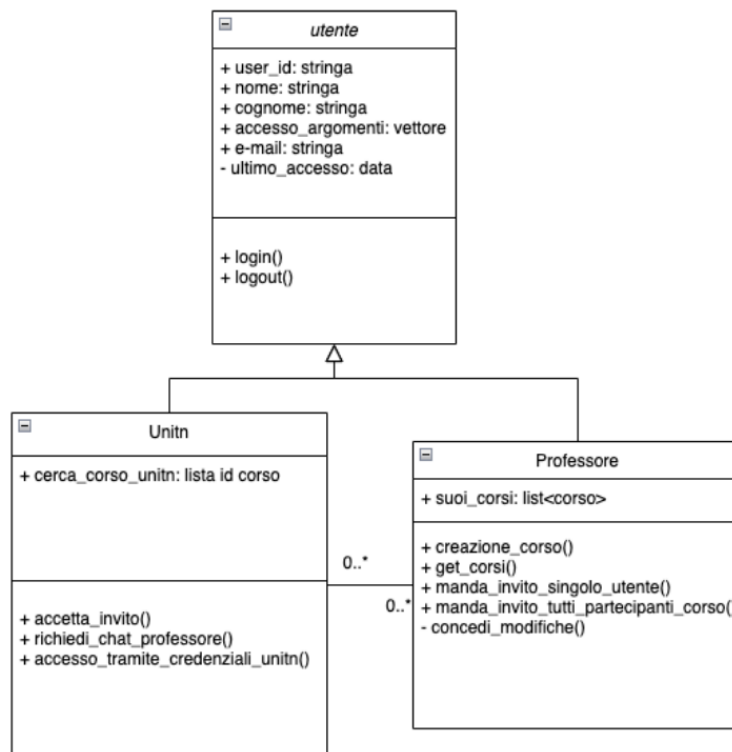
### Utente

Analizzando questo diagramma di contesto si possono individuare due attori collegati alla classe utente ovvero utente Unitn e utente Professore. L'utente Unitn, tramite le proprie credenziali, può accedere ai corsi universitari a cui appartiene inoltre può accettare inviti da professori dell'università di Trento.

L'utente Professore accedendo al sito tramite le proprie credenziali può vedere i propri corsi, creandone anche di nuovi relativi sempre alla propria materia di insegnamento. Inoltre può invitare altri utenti a partecipare al proprio corso.

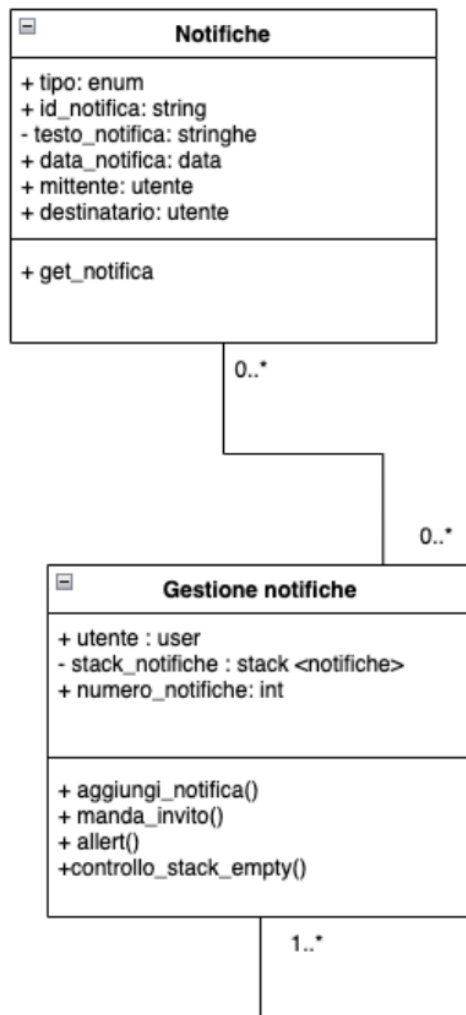
La classe utente presenta delle caratteristiche comuni sia per l'utente Unitn che per l'utente Professore come ad esempio e-mail , credenziali ed ultimo accesso.

Infine la classe utente è collegata alla classe Gestione Notifiche perché l'utente professore è in grado di autorizzare una modifica da parte dell'utente Unitn ad un file tramite sezione commento. Inoltre un utente professore può inoltrare l'invito di partecipare al proprio corso tramite notifiche e l'utente Unitn può accettare o meno tramite la visualizzazione della suddetta notifica.



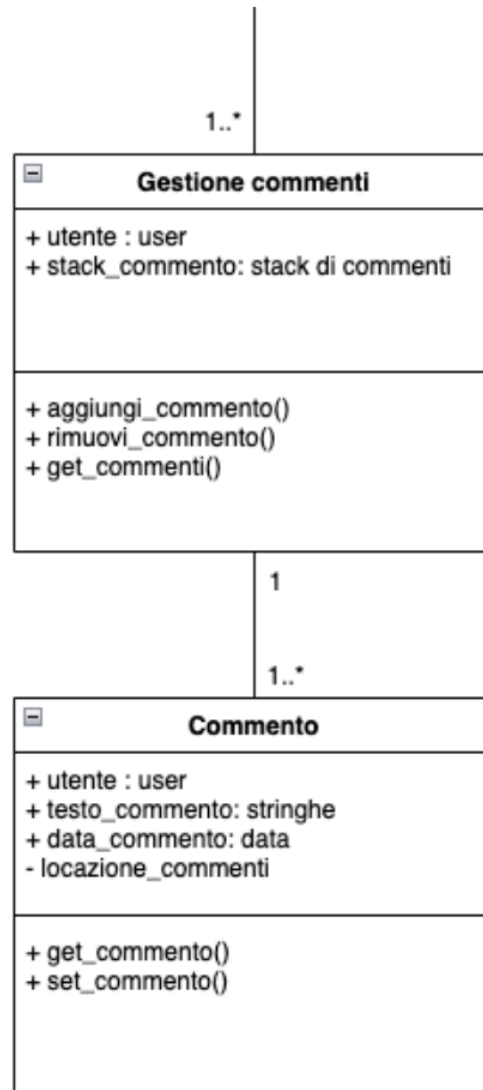
## Gestione Notifiche

Analizzando questo diagramma di contesto si può individuare l'attore notifica collegato alla classe Gestione Notifiche. Per quanto riguarda le notifiche sono costituite dal testo e dalla data della notifica, dal mittente e dal destinatario della notifica stessa. Le notifiche sono strettamente correlate alla gestione delle notifiche in quanto possiamo trovare caratteristiche aggiuntive riguardanti le notifiche stesse come ad esempio il numero e lo stack di notifiche. Quest'ultimo è costituito da tutte le notifiche apportate ad un singolo file che potranno essere visualizzate dalle più recenti alle più vecchie. La gestione delle notifiche è collegata alla gestione dei commenti poiché presentano delle caratteristiche comuni come ad esempio l'utente in quanto ha la possibilità di poter apportare delle modifiche al proprio commento potendovi accedere in qualsiasi momento



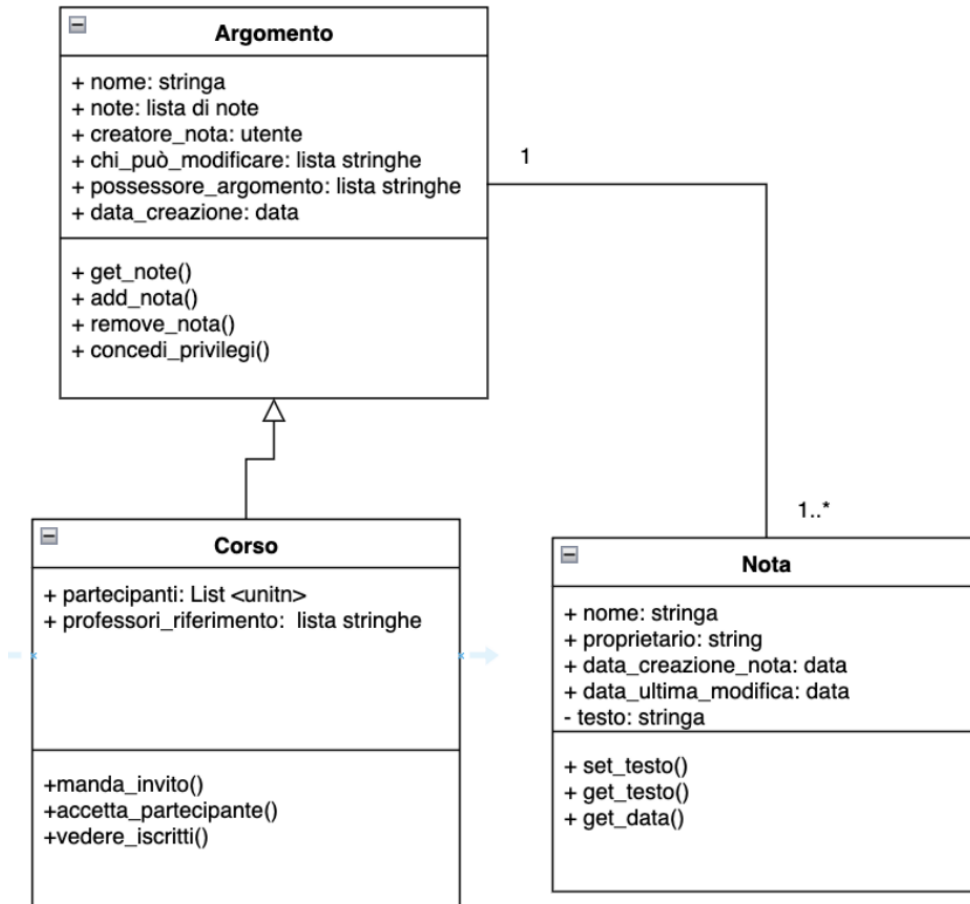
## Gestione commento

Analizzando questo diagramma di contesto si può individuare l'attore commento collegato alla classe commenti. Il commento è situato nell'apposita area commenti ed è costituito dal testo e dalla data del commento. I commenti sono strettamente correlate alla Gestione Commenti in quanto possiamo trovare delle caratteristiche che li riguardano l'utente e lo stack di commenti. Quest'ultimo costituito da tutti i commenti che si trovano all'interno della chat e possono essere visualizzati dai più recenti ai più vecchi. Come sopraindicato la gestione commenti è collegata alla gestione notifiche in quanto presentano delle analogie date dalla presenza dell'utente e dal numero di notifiche correlate al numero di commenti fatti in una chat



## Argomento

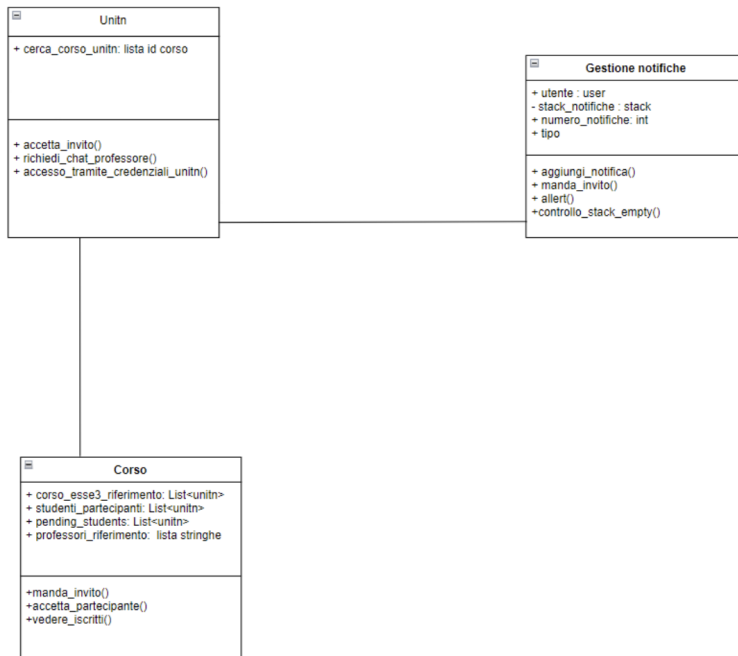
Analizzando questo diagramma di contesto si possono individuare due attori ovvero corso e nota. La sezione corso è costituita da tutti i partecipanti al corso che possono essere utenti Unitn e docente del corso ovvero utente Professore. Inoltre l'utente Professore può invitare nuovi partecipanti al corso tramite una funzione presente nella schermata del corso stesso. All'interno dell'argomento ci sarà uno spazio riservato alle note che saranno costituite da un nome, dalla data di creazione e dall'ultima modifica effettuata, inoltre sarà possibile per altri utenti vedere chi hai pubblicato quella nota.



**Codice in Object Constraint Language** In questo capitolo verrà descritta la logica prevista da alcune operazioni di alcune classi in modo formale. Ciò verrà fatto in Object Constraint Language (OCL) data la impossibilità di esprimere tali concetti con altri modi formali del contesto UML.

### accetta-invito

Un utente può accettare un invito quando l'invito appartiene al proprio Stack e la notifica è del tipo giusto



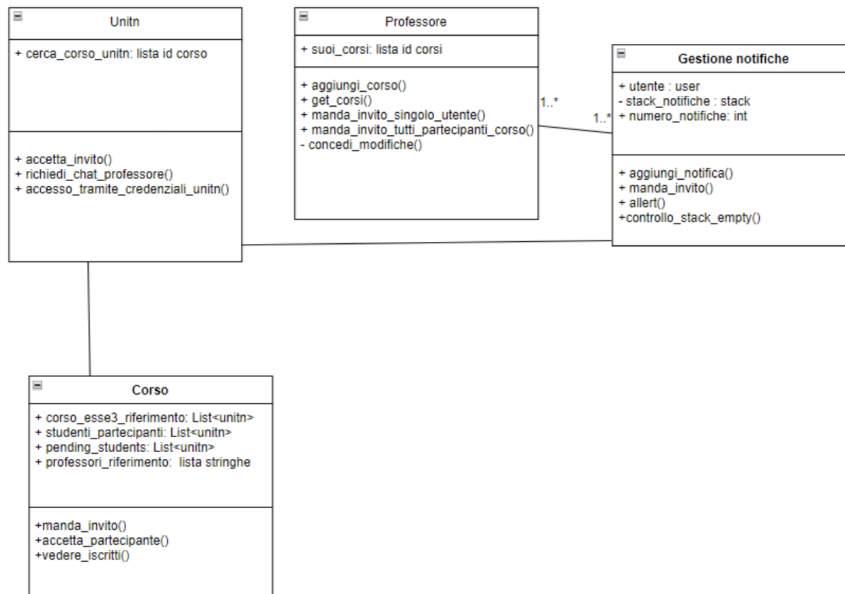
#### ContextUnitn::accetta\_invito

Pre :gestione\_notifiche.stack\_notifiche->includes<notifica> AND notifica.tipo=invito\_chat

Post: stack.pop(notifica)=true

## richiedi-chat-professore

Un utente può richiedere una chat col professore se e solo se l'utente stesso partecipa ad un corso in cui è presente anche il professore. Se ciò si verifica verrà aggiunta la notifica di richiesta allo stack di gestione delle notifiche del professore



ContextUnitn::richiedi\_chat\_professore()

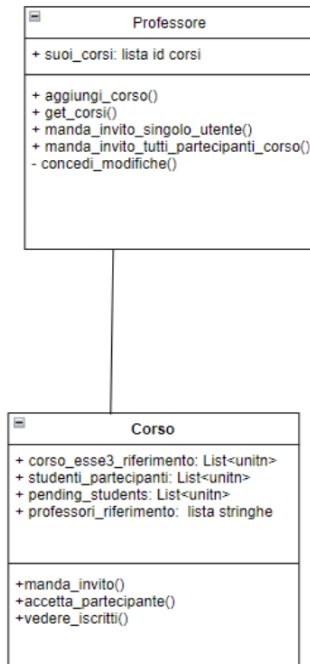
Precorso.partecipanti ->includes(unitn) and corso.professori\_riferimento ->includes<professore>

Post: gestione\_notifiche.stack\_notifiche=push(notifica)



## Creazione di un corso

Un professore può creare un corso se risulta insegnante dello stesso, ciò tramite i dati prelevati dalle API del sistema esse3. In quel caso verrà inviata una notifica di invito ad ogni partecipante al corso.



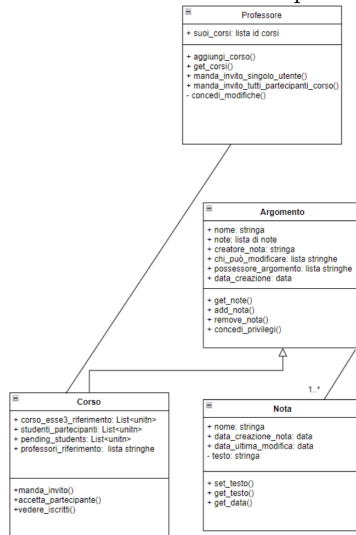
`Contextprofessore::creazione_corso()`

`Preprofessore.suoi_corsi ->includes<corso>`

`Post: corso.manda_invito(List <corso.corso_esse3_riferimento>)`

## Concessione di permessi di modifiche

Un professore può concedere i permessi di modifica ai suoi studenti se lui ne è il proprietario e se l'utente non ne è già in possesso. Se l'utente non è provvisto di suddette autorizzazioni verranno fornite aggiungendo l'utente alla lista di chi ha i permessi di modifica.



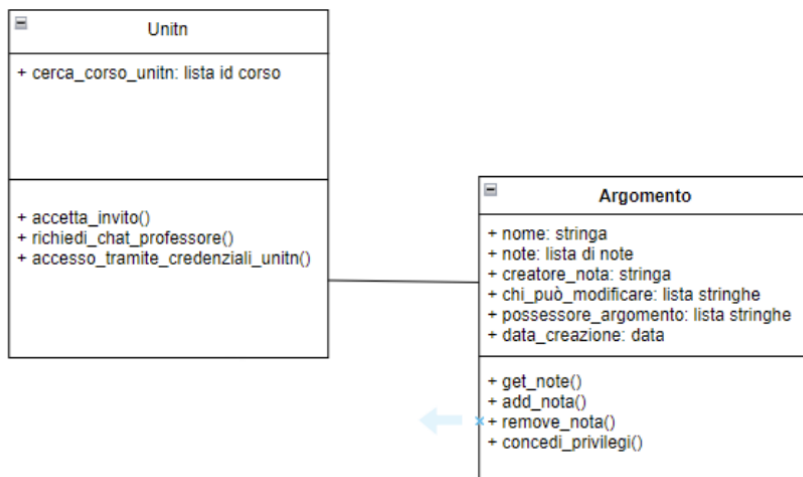
ContextProfessore::concedi\_modifiche()

Prenota.proprietario=professore.user\_id AND (argomento.chi\_può\_modificare ->includes<unitn> = false)

Post argomento.chi\_può\_modificare=push(unitn)

## Concessione privilegi

Un utente può concedere i permessi di modifica ad un argomento se ne è il proprietario e se l'utente che vuole apportare modifiche non ha già i privilegi verrà inserito nella lista degli utenti con permessi di modifica.



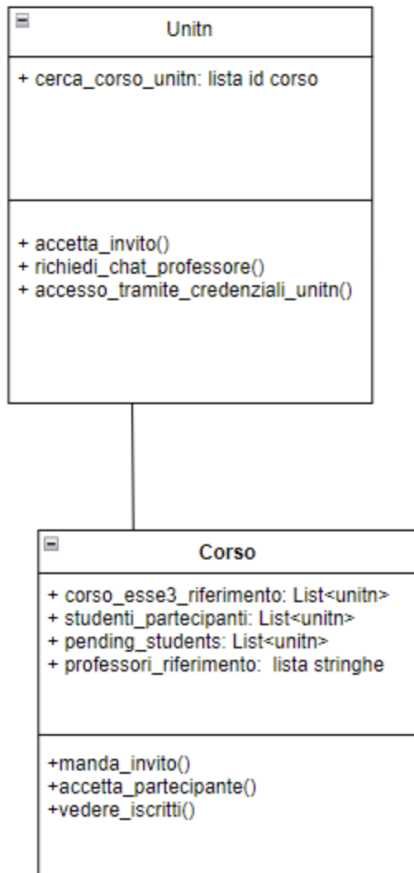
Contextargomento::concedi\_privilegi()

Pre: argomento.creatore\_nota=user.user\_id and (argomento.chi\_può\_modificare ->includes<user>=false)

Post: argomento.chi\_può\_modificare=push(user)

### accettazione di partecipanti

Un utente può essere accettato all'interno del corso se è associato come utente unitn e dopo essere stato accettato verrà aggiunto alla lista dei partecipanti al corso stesso



Contextcorse::accetta\_partecipante()

Pre: self instance of unitn

Post: corso.partecipanti=push(self)

## DIAGRAMMA DELLE CLASSI CON CODICE OCL

