



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# Tutorato UML 2

Ingegneria del Software

*Nico Pellegrinelli*

[nico.pellegrinelli@unibg.it](mailto:nico.pellegrinelli@unibg.it)

# Esercizio 1

## CHAT DISTRIBUITA



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# Testo dell'esercizio

Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi utenti lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza, o uscire dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- È comunque possibile inviare messaggi “privati” ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l’unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il moderatore della chat), con l’autorità di inviare avvisi ad altri utenti, di espellere utenti e di creare o eliminare stanze della chat. In ogni istante nella chat può esistere un solo moderatore.
- Il moderatore della chat, dopo aver inviato un messaggio di avviso all’utente interessato, può successivamente richiedere la sua espulsione dall’insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l’interazione dell’utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Use Case Diagram

Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi utenti lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza, o uscire dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- È comunque possibile inviare messaggi “privati” ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l’unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il moderatore della chat), con l’autorità di inviare avvisi ad altri utenti, di espellere utenti e di creare o eliminare stanze della chat. In ogni istante nella chat può esistere un solo moderatore.
- Il moderatore della chat, dopo aver inviato un messaggio di avviso all’utente interessato, può successivamente richiedere la sua espulsione dall’insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l’interazione dell’utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Use Case Diagram - Attori

Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi **utenti** lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza, o uscire dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- E comunque possibile inviare messaggi “privati” ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l’unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il **moderatore** della chat), con l’autorità di inviare avvisi ad altri utenti, di espellere utenti e di creare o eliminare stanze della chat. In ogni istante nella chat può esistere un solo moderatore.
- Il moderatore della chat, dopo aver inviato un messaggio di avviso all’utente interessato, può successivamente richiedere la sua espulsione dall’insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l’interazione dell’utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Use Case Diagram – Relazioni tra attori

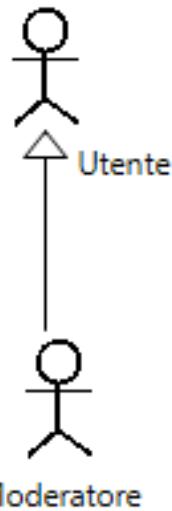
Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi **utenti** lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza, o uscire dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- E comunque possibile inviare messaggi “privati” ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l’unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di **utente** particolare (il **moderatore** della chat), con l’autorità di inviare avvisi ad altri utenti, di espellere utenti e di creare o eliminare stanze della chat. In ogni istante nella chat può esistere un solo moderatore.
- Il moderatore della chat, dopo aver inviato un messaggio di avviso all’utente interessato, può successivamente richiedere la sua espulsione dall’insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l’interazione dell’utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Use Case Diagram - Attori



# Use Case Diagram – Casi d'uso

Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi **utenti** lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può **entrare** per **scambiare messaggi** con gli altri utenti presenti nella stessa stanza, o **uscire** dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- E comunque possibile **inviare messaggi “privati”** ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l'unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il **moderatore** della chat), con l'autorità di **inviare avvisi** ad altri utenti, di **espellere utenti** e di **creare o eliminare stanze** della chat. In ogni istante nella chat può esistere un solo moderatore.
- Il moderatore della chat, dopo aver inviato un messaggio di avviso all'utente interessato, può successivamente richiedere la sua espulsione dall'insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l'interazione dell'utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Use Case Diagram – Relazioni tra casi d'uso

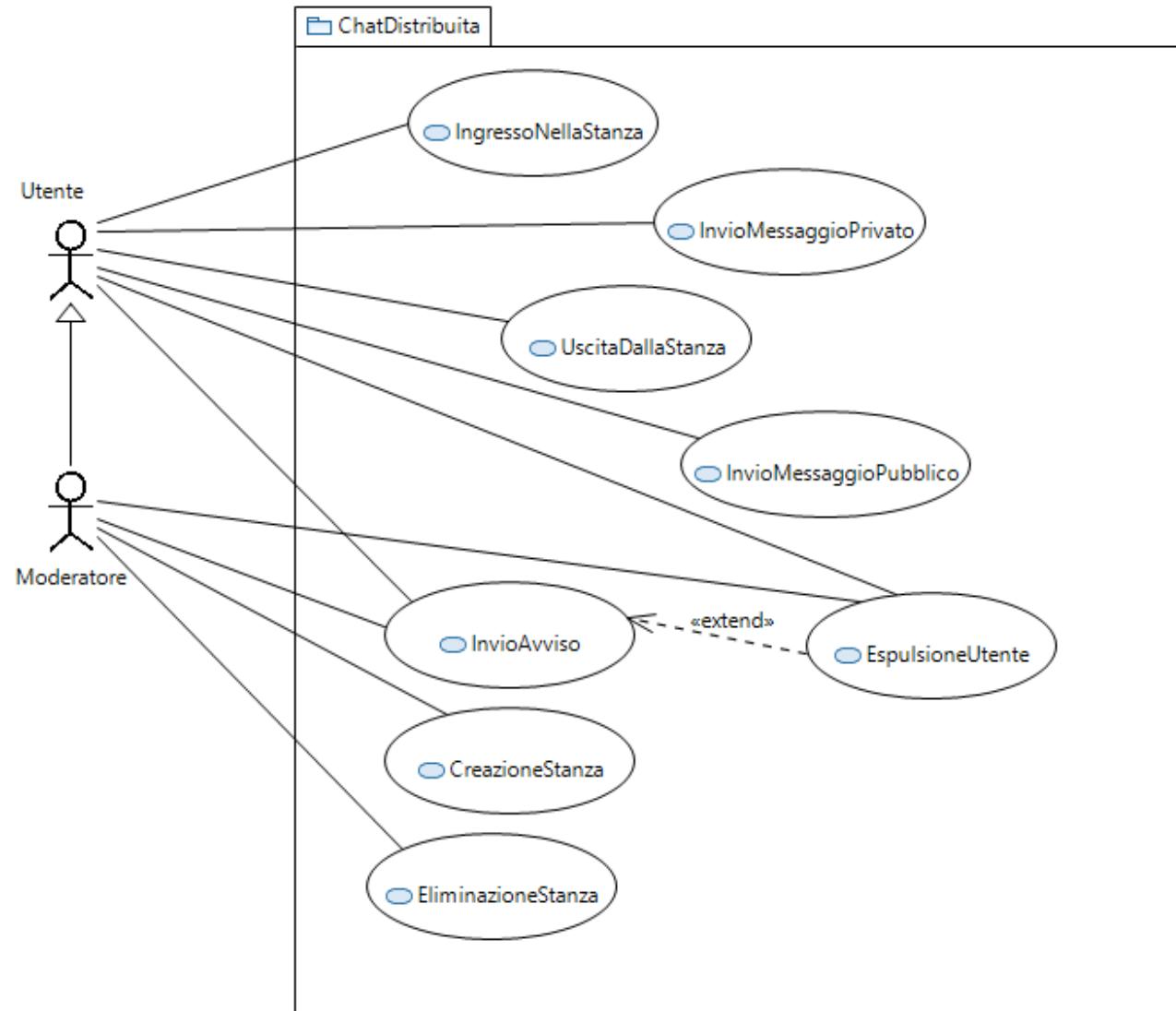
Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi **utenti** lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può **entrare** per **scambiare messaggi** con gli altri utenti presenti nella stessa stanza, o **uscire** dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- E comunque possibile **inviare messaggi “privati”** ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l'unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il **moderatore** della chat), con l'autorità di **inviare avvisi** ad altri utenti, di **espellere utenti** e di **creare o eliminare stanze** della chat. In ogni istante nella chat può esistere un solo moderatore.
- Il moderatore della chat, **dopo aver inviato un messaggio** di avviso all'utente interessato, **può successivamente richiedere la sua espulsione** dall'insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l'interazione dell'utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Use Case Diagram - Soluzione



# Use Case Diagram – Descrizione casi d'uso

Identificare per il caso d'suo CREAZIONE DI UNA STANZA:

- Nome
- Breve descrizione
- Pre-condizioni
- Post-condizioni
- Situazioni di errore
- Stato del sistema nel caso in cui ci sia un errore
- Attori
- Trigger
- Processo standard
- Processo alternativo



# Use Case Diagram – Descrizione casi d'uso

Caso d'suo CREAZIONE DI UNA STANZA :

- **Nome**

Creazione di una stanza virtuale

- **Breve descrizione**

Il moderatore crea una nuova stanza virtuale

- **Pre-condizioni**

Il moderatore ha accesso alla chat

- **Post-condizioni**

La stanza virtuale è stata creata

- **Situazioni di errore**

La stanza virtuale è già esistente

- **Stato del sistema nel caso in cui ci sia un errore**

Nessuna nuova stanza virtuale è stata creata

- **Attori**

Moderatore



# Use Case Diagram – Descrizione casi d'uso

- **Trigger**

Il moderatore richiede attraverso l'interfaccia testuale di creare una nuova stanza virtuale

- **Processo standard**

1. Il moderatore fa login nella chat
2. Il moderatore seleziona l'opzione "Crea nuova stanza virtuale"
3. Il moderatore inserisce il nome della stanza virtuale
4. Il sistema crea la nuova stanza virtuale
5. Il sistema conferma al moderatore la creazione della nuova stanza virtuale

- **Processo alternativo**

4. Il sistema notifica al moderatore che una stanza con quel nome già esiste



# Class Diagram

Vogliamo progettare una semplice chat, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi utenti lo scambio di messaggi in rete:

- Esistono diverse stanze virtuali (ciascuna legata a una tematica specifica) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza, o uscire dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- È comunque possibile inviare messaggi “privati” ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l’unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il moderatore della chat), con l’autorità di inviare avvisi ad altri utenti, di espellere utenti e di creare o eliminare stanze della chat. In ogni istante nella chat può esistere un solo moderatore.
- In moderatore della chat, dopo aver inviato un messaggio di avviso all’utente interessato, può successivamente richiedere la sua espulsione dall’insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l’interazione dell’utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Class Diagram – Individuazione delle classi

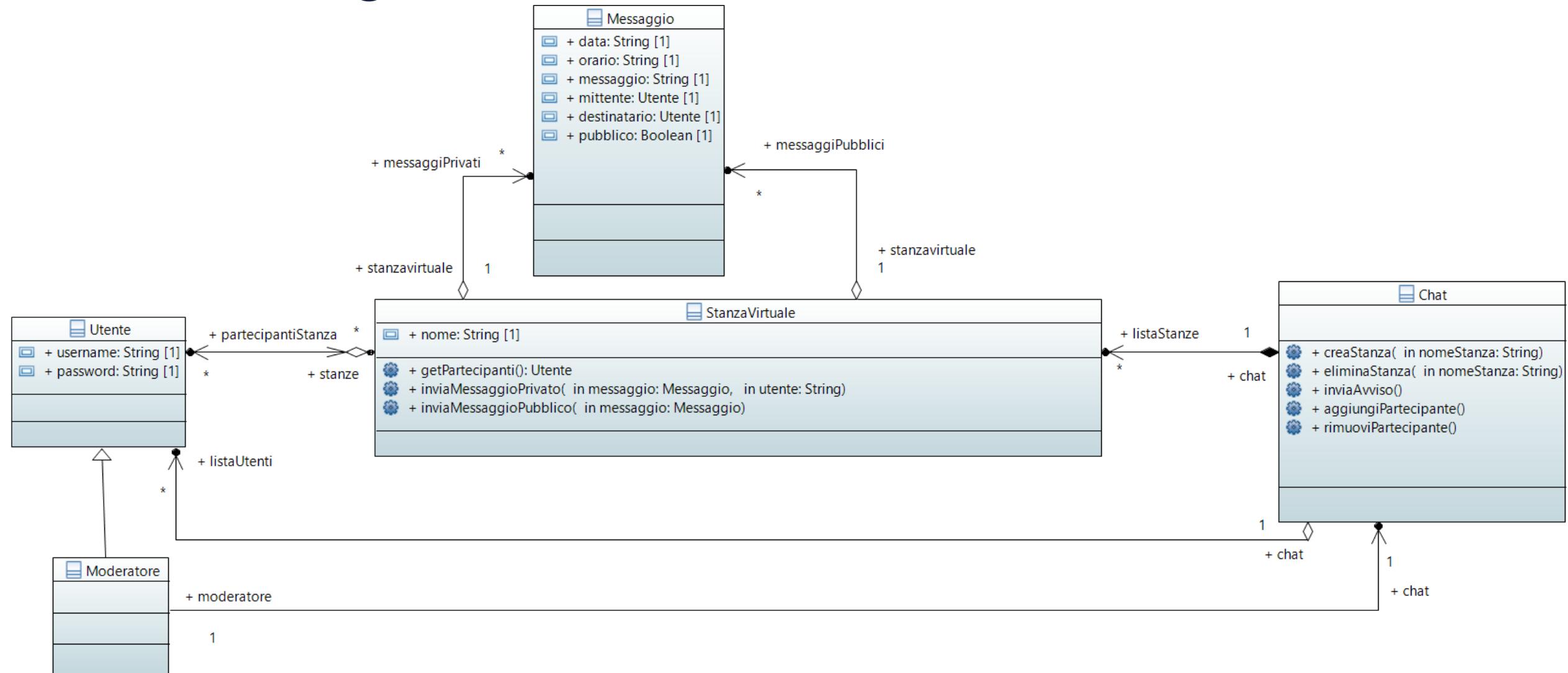
Vogliamo progettare una semplice **chat**, caratterizzata dai requisiti informali descritti nel seguito, che consente a diversi **utenti** lo scambio di **messaggi** in rete:

- Esistono diverse **stanze virtuali** (ciascuna legata a una tematica specifica) nelle quali un utente può entrare per scambiare messaggi con gli altri utenti presenti nella stessa stanza, o uscire dalla stanza.
- I messaggi inviati da un utente sono trasmessi a tutti gli altri partecipanti presenti nella stanza.
- E comunque possibile inviare messaggi “privati” ad uno specifico partecipante presente nella stessa stanza. In questo caso il destinatario selezionato sarà l’unico a ricevere il messaggio.
- Il sistema prevede la presenza di una tipologia di utente particolare (il **moderatore** della chat), con l’autorità di inviare avvisi ad altri utenti, di espellere utenti e di creare o eliminare stanze della chat. In ogni istante nella chat può esistere un solo moderatore.
- In moderatore della chat, dopo aver inviato un messaggio di avviso all’utente interessato, può successivamente richiedere la sua espulsione dall’insieme dei partecipanti alla chat.
- Ogni successivo tentativo di invio di un messaggio da parte di un utente espulso deve essere rifiutato dal sistema.

Vogliamo che l’interazione dell’utente con il sistema avvenga attraverso una semplice interfaccia testuale utilizzata per inviare messaggi agli altri utenti e per visualizzare i messaggi ricevuti, oltre agli avvisi provenienti dal moderatore della chat.



# Class Diagram – Soluzione



# Esercizio 2

## Istruzione universitaria



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# State Machine Diagram

Si richiede di rappresentare uno state machine diagram per il seguente problema:

Inizialmente, il programma di studio è inattivo. Quando inizia un nuovo semestre e le tasse sono state pagate (e quindi lo studente si iscrive al corso di studi), il programma di studio diventa attivo.

Le tasse universitarie devono essere pagate entro l'inizio di ogni semestre. Se questo non viene fatto, il programma di studio passa ad essere nuovamente inattivo.

Durante il corso di un programma di studi, lo studente passa attraverso la laurea triennale, la magistrale ed il dottorato. In tutti questi casi, non è possibile iscriversi al corso successivo se non si è completato il precedente.

In ogni momento, lo studente può lasciare l'università.

Se lo studente non paga le tasse universitarie per uno specifico semestre (ad esempio perché vuole prendere una pausa di un semestre) deve comunque essere possibile per lo studente tornare allo stato raggiunto prima della pausa nel momento in cui torna a pagare le tasse.



# State Machine Diagram – Stati e Transizioni

Si richiede di rappresentare uno state machine diagram per il seguente problema:

Inizialmente, il programma di studio è **inattivo**. Quando **inizia un nuovo semestre** e le tasse sono state pagate (e quindi lo studente si iscrive al corso di studi), il programma di studio diventa **attivo**.

Le tasse universitarie devono essere pagate entro l'inizio di ogni semestre. Se questo non viene fatto, il programma di studio **passa ad essere nuovamente inattivo**.

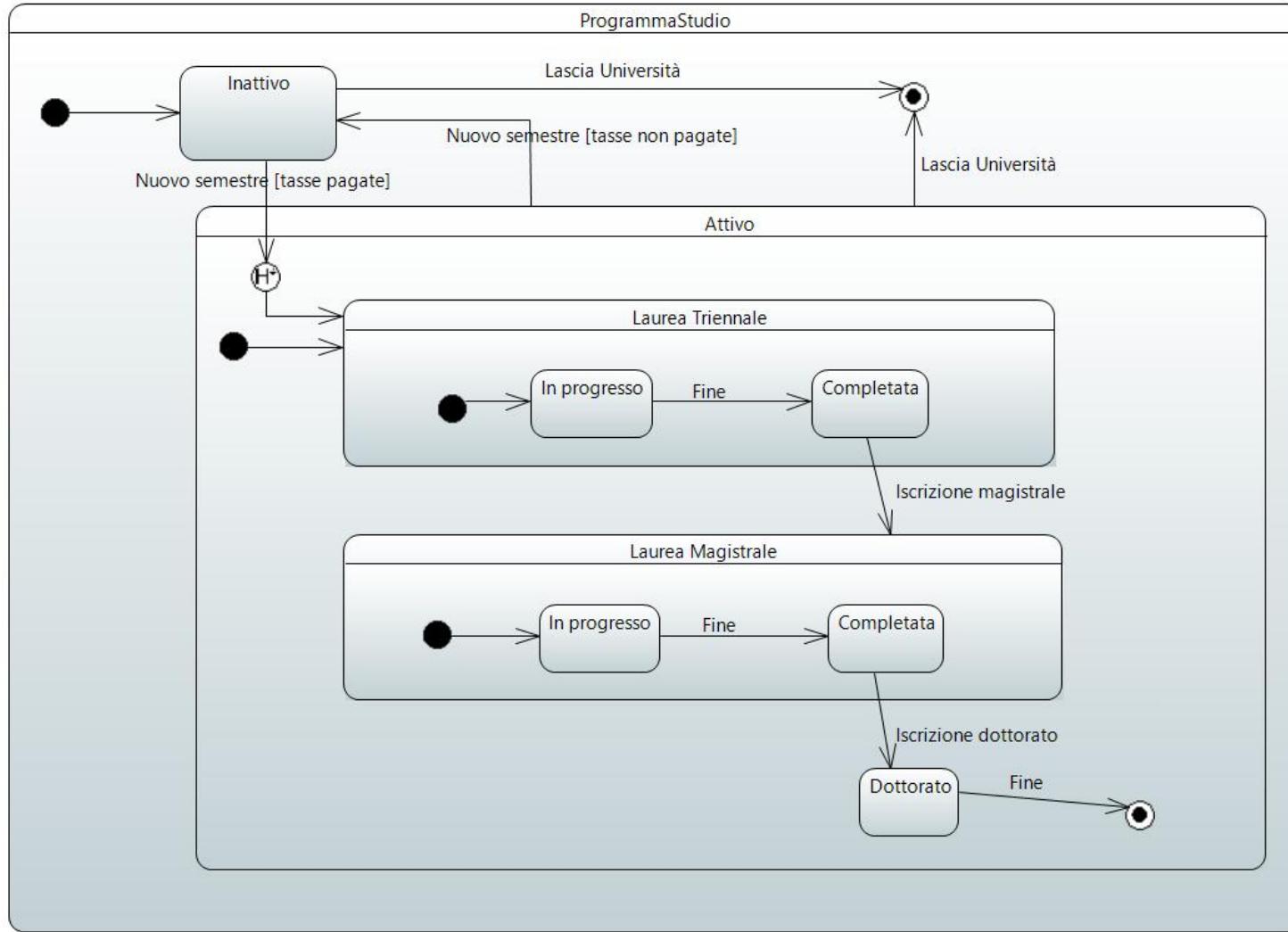
Durante il corso di un programma di studi, lo studente passa attraverso la **laurea triennale**, la **magistrale** ed il **dottorato**. *In tutti questi casi, non è possibile **iscriversi** al corso successivo se non si è completato il precedente.*

In ogni momento, lo studente può **lasciare l'università**.

Se lo studente non paga le tasse universitarie per uno specifico semestre (ad esempio perché vuole prendere una pausa di un semestre) deve comunque essere possibile per lo studente **tornare allo stadio raggiunto prima della pausa** nel momento in cui torna a pagare le tasse.



# State Machine Diagram – Soluzione



# Esercizio 3

## Spedizione Posta



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# Sequence e Communication Diagram

Descrivere con un sequence diagram le operazioni che caratterizzano l'invio di una lettera dal momento della spedizione fino a quello della ricezione da parte del destinatario.

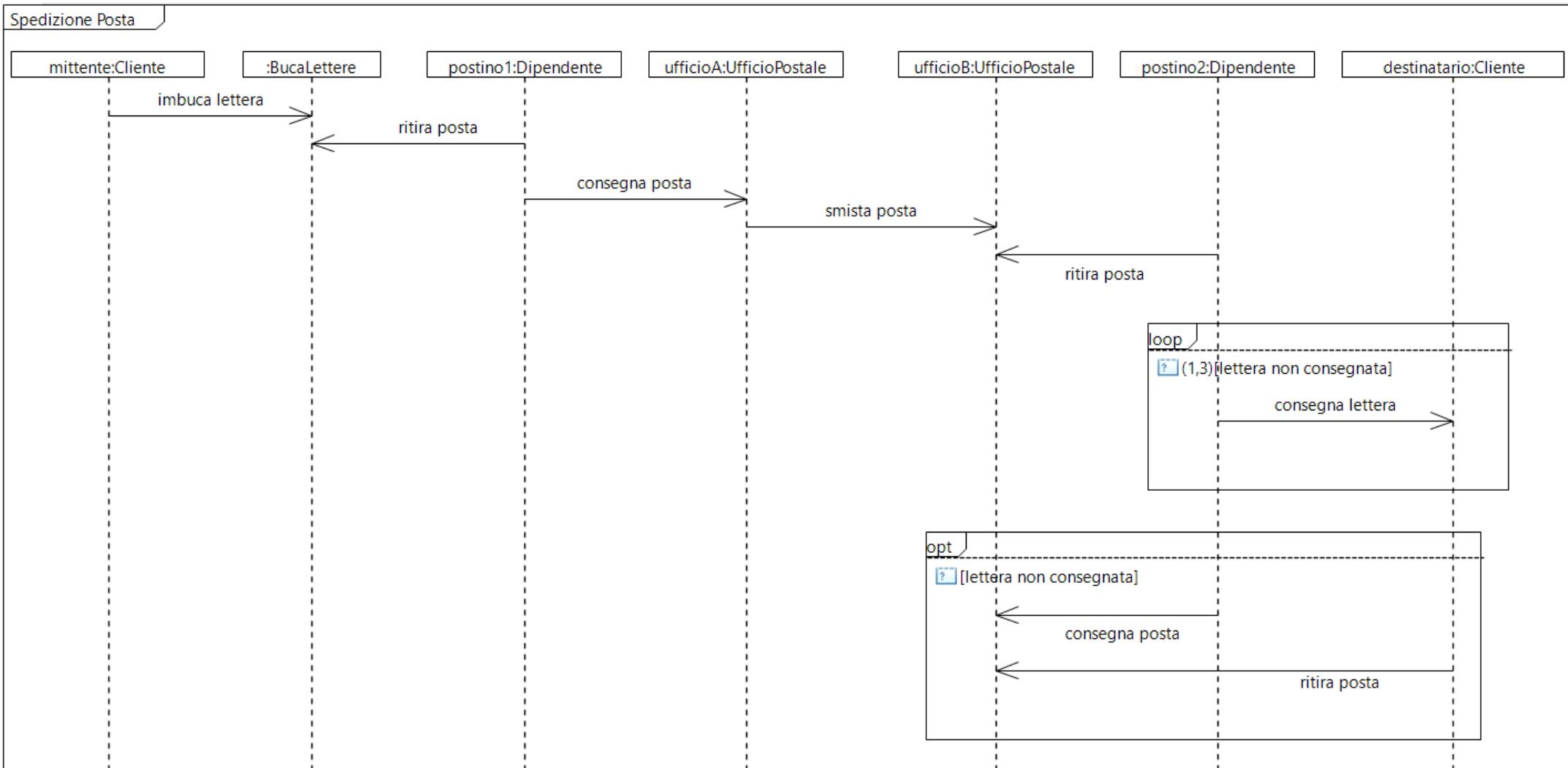
In particolare :

- il mittente imbuca la lettera nella buca per lettere
- un postino la ritira e la consegna all'ufficio postale
- l'ufficio postale la smista verso l'ufficio postale di destinazione
- un secondo postino la ritira e prova a consegnarla al destinatario un massimo di tre volte
- dopo tre volte la porta all'ufficio postale e il destinatario deve andare direttamente a ritirarla

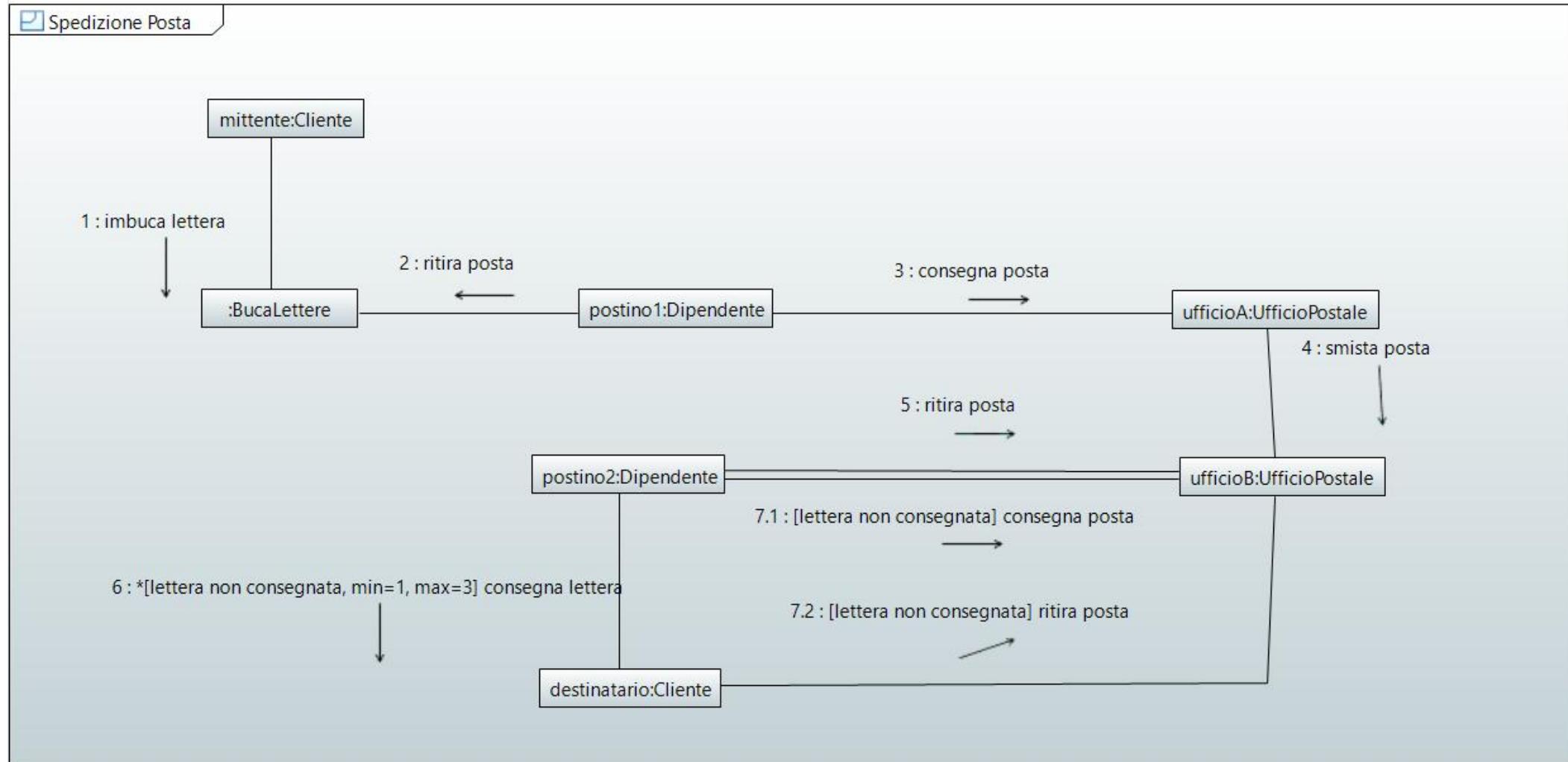
Tradurre il sequence diagram in un communication diagram .



# Sequence Diagram - Soluzione



# Communication Diagram - Soluzione



## Esercizio 4

# Organizzazione colloquio



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione

# Activity Diagram

Attraverso un activity diagram si modelli il processo di organizzazione di un colloquio lavorativo da parte di una agenzia interinale.

L'agenzia dopo aver esaminato il CV del candidato e aver valutato se sia idoneo o meno invia un messaggio di richiesta di colloquio. A questo punto le possibilità sono due: la persona risponde accettando o declinando l'invito oppure passa una settimana e l'agenzia non ottiene risposta. A fronte dell'accettazione viene inviato un messaggio con i dettagli del luogo ed ora del colloquio. In caso di esito negativo (rifiuto o non risposta), l'agenzia manda comunque un messaggio di notifica.



# Activity Diagram - Individuazione delle attività

Attraverso un activity diagram si modelli il processo di organizzazione di un colloquio lavorativo da parte di una agenzia interinale.

L'agenzia dopo **aver esaminato il CV** del candidato e aver valutato se sia idoneo o meno **invia un messaggio di richiesta di colloquio**. A questo punto le possibilità sono due: **la persona risponde accettando o declinando** l'invito oppure **passa una settimana e l'agenzia non ottiene risposta**. A fronte dell'accettazione **viene inviato un messaggio** con i dettagli del luogo ed ora del colloquio. In caso di esito negativo (rifiuto o non risposta), l'agenzia **manda comunque un messaggio di notifica**.



# Activity Diagram

