

# Esercitazione

Vincenzo Gervasi, Laura Semini  
Ingegneria del Software  
Dipartimento di Informatica  
Università di Pisa

# Riassunto lezione precedente

## Outline della lezione

- Lezione precedente:
  - Introduzione a UML
  - Diagramma dei casi d'uso
- Questa lezione

# Chiavi magnetiche

# Chiavi magnetiche

- Per motivi di sicurezza, un'organizzazione ha deciso di realizzare un sistema secondo il quale a ogni dipendente è assegnata una chiave magnetica per accedere (aprire) determinate stanze. I diritti di accesso dipenderanno in generale dalla posizione e dalle responsabilità del dipendente. Quindi sono necessarie operazioni per modificare i diritti di accesso posseduti da una chiave se il suo proprietario cambia ruolo nell'organizzazione.

# I Requisiti

- Come si assegna una chiave?

Si verifica l'identità del dipendente, e si verifica il suo ruolo. Quindi si assegnano i diritti collegati al suo ruolo.

- Come si modificano i diritti?

Si verificano l'identità del dipendente e il suo nuovo ruolo. Quindi si aggiornano i diritti.

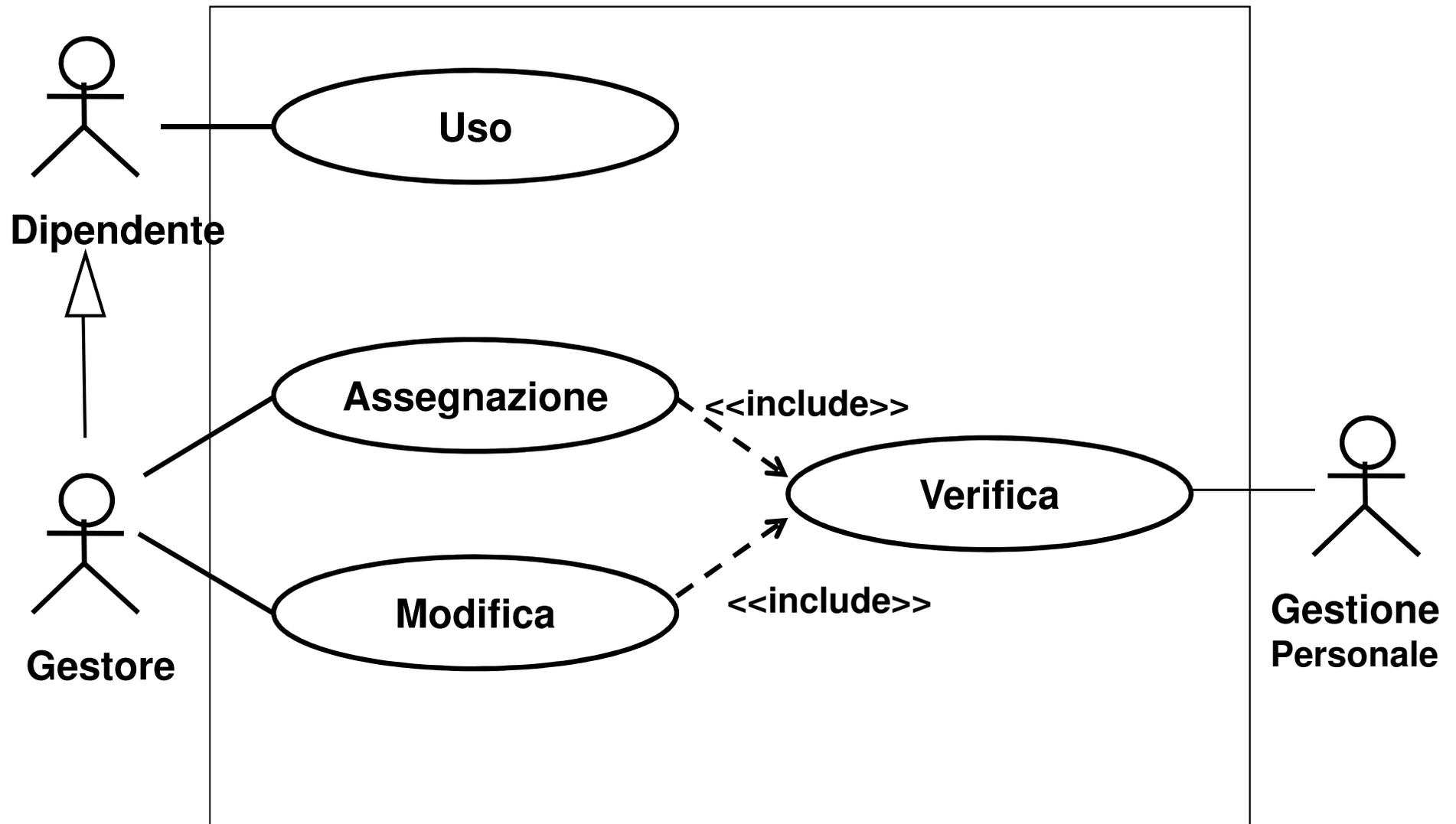
- Come funziona la chiave?

Quando viene inserita la chiave, si verifica se i diritti della chiave sono adeguati per la porta che si sta cercando di aprire. In caso positivo, si apre la porta, altrimenti la porta rimane chiusa, e viene mandato un allarme in centrale.

- Attenzione a definire i confini del sistema

Db dipendenti?

# Chiavi magnetiche: Casi d'uso



## Caso d'uso: assegnazione

<i>Breve descrizione:</i>	Assegnazione iniziale dei diritti, al momento della consegna della chiave.
<i>Attori primari:</i>	Gestore.
<i>Attori secondari:</i>	Nessuno.
<i>Precondizioni:</i>	Chiave non assegnata.
<i>Sequenza degli eventi principale:</i>	<ol style="list-style-type: none"><li>1. Esegue Verifica</li><li>2. assegna diritti</li></ol>
<i>Postcondizioni:</i>	Chiave associata all'impiegato, con i diritti collegati al suo status corrente.
<i>Sequenze alternative degli eventi:</i>	Verifica fallisce.

## Caso d'uso: modifica

<i>Breve descrizione:</i>	Variazione dei diritti del detentore della chiave.
<i>Attori primari:</i>	Gestore.
<i>Attori secondari:</i>	Nessuno.
<i>Precondizioni:</i>	Chiave assegnata.
<i>Sequenza degli eventi principale:</i>	<ol style="list-style-type: none"><li>1. Esegue Verifica</li><li>2. aggiorna diritti</li></ol>
<i>Postcondizioni:</i>	Diritti coerenti con il nuovo status del detentore.
<i>Sequenze alternative degli eventi:</i>	Verifica fallisce.

## Caso d'uso: verifica

<i>Breve descrizione:</i>	Effettua le verifiche previste.
<i>Attori primari:</i>	Nessuno.
<i>Attori secondari:</i>	Gestione Personale.
<i>Precondizioni:</i>	Nessuna
<i>Sequenza degli eventi principale:</i>	1. verifica dell'identità 2. verifica dei diritti
<i>Postcondizioni:</i>	verifica dell'identità positiva verifica dei diritti positiva
<i>Sequenze alternative degli eventi:</i>	Verifiche fallite.

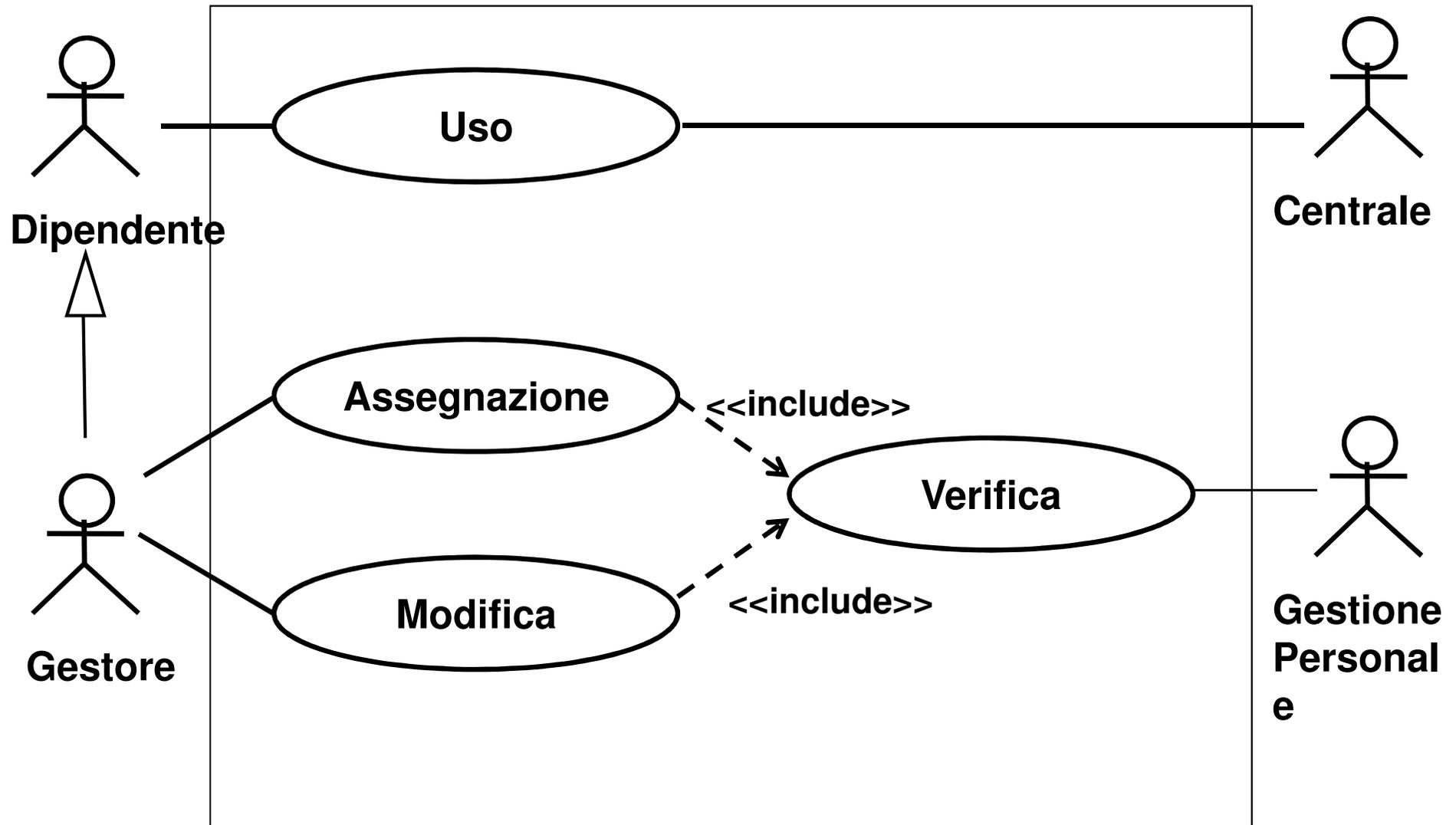
## Caso d'uso: uso (della chiave)

<i>Breve descrizione:</i>	Uso della chiave per entrare in una stanza.
<i>Attori primari:</i>	Dipendente.
<i>Attori secondari:</i>	Nessuno.
<i>Precondizioni:</i>	Nessuna
<i>Sequenza degli eventi principale:</i>	<ol style="list-style-type: none"><li>1. Verifica dei diritti della chiave (assegnata);</li><li>2. Apertura della porta.</li></ol>
<i>Postcondizioni:</i>	Porta aperta.
<i>Sequenze alternative degli eventi:</i>	Diritti mancanti

## Sequenza alternativa degli eventi

Sequenza alternativa degli eventi: diritti mancanti	
<i>Breve descrizione:</i>	Blocco di una chiave e allarme.
<i>Attori primari:</i>	Nessuno.
<i>Attori secondari:</i>	<u>Centrale.</u>
<i>Precondizioni:</i>	Chiave inserita, verifica fallita.
<i>Sequenza degli eventi principale:</i>	<ol style="list-style-type: none"><li>1. Blocca la carta</li><li>2. allarme alla centrale</li></ol>
<i>Postcondizioni:</i>	Porta chiusa, carta bloccata.
<i>Sequenze alternative degli eventi:</i>	Nessuno.

# Chiavi magnetiche: Casi d'uso

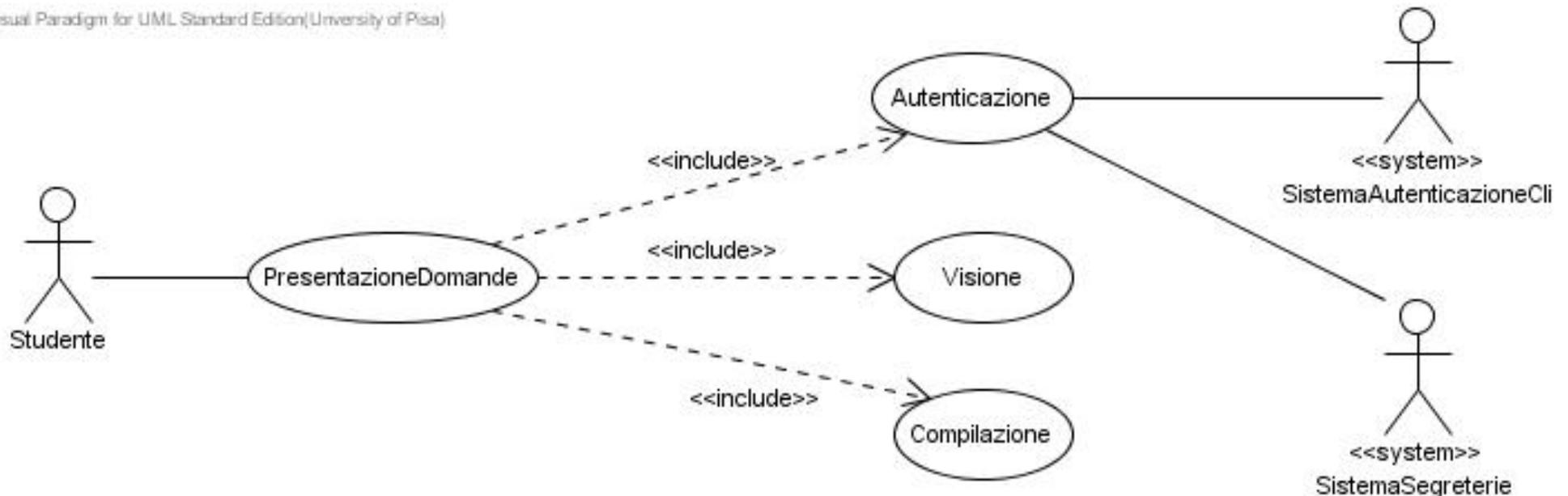


**Tirocini**

# Tirocini (Modello statico)

Il caso d'uso *PresentazioneDomande* prevede tre fasi: *Autenticazione* (comprende l'immissione degli esami non registrati), *Visione* (della lista progetti) e *Compilazione* (della domanda).

Visual Paradigm for UML Standard Edition(University of Pisa)



# Caso d'uso: Compilazione

Breve descrizione: *Permette di compilare la domanda di tirocinio.*

Attore principale: *Studente*

Attore secondario: *Nessuno*

PreCondizioni: *Studente autenticato, esami completi, lista progetti visualizzata.*

PostCondizioni: *Domanda compilata.*

Sequenza principale degli eventi:

1. *se (lo studente vuole)*

1. *lo Studente indica la matricola di un collega*

2. *per (ogni progetto)*

1. *lo Studente esprime un giudizio tra: inaccettabile, bassa, media, alta*

2. *se (giudizio alto)*

1. *il sistema assegna il voto 5*

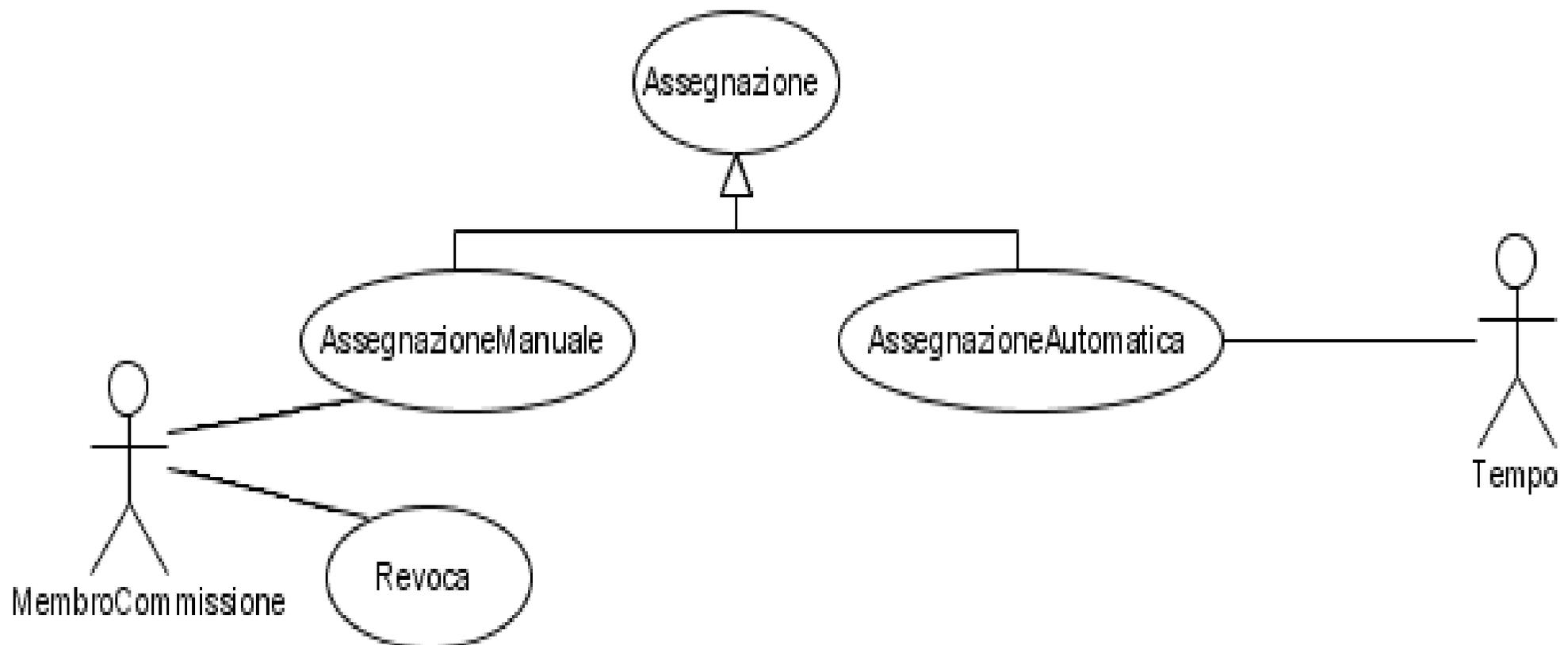
3. *se (lo studente vuole)*

1. *per (ogni progetto con preferenza alta)*

1. *se (lo studente vuole)*

1. *lo Studente esprime un voto tra 1 e 10*

# assegnazione dei tirocini, sia manuale sia automatica, e la revoca.



# Revoca

Breve descrizione: *Viene disfatta un'assegnazione tra uno studente e un tirocinio*

Attore principale: *MembroCommissione*

Attori secondari: *Nessuno*

PreCondizioni: *Tirocinio assegnato allo studente*

PostCondizioni: *Tirocinio non assegnato, studente non assegnato.*

Sequenza principale degli eventi:

1. *Il MembroCommissione richiama il tirocinio*
2. *Il sistema lo visualizza*
3. *Il MembroCommissione annulla l'assegnazione*
4. *Il sistema chiede conferma*
5. *Il MembroCommissione conferma*
6. *Il sistema modifica il tirocinio*

Sequenza alternativa degli eventi:

1. *Il MembroCommissione non conferma*
2. *Il MembroCommissione termina la sessione dopo la visione*

# Grande distribuzione

# Grande Distribuzione

## Modifica dei prezzi di vendita

Il ClientDelPuntoVendita viene usato dal responsabile del punto vendita per modificare i prezzi di vendita (a negozio chiuso) ...

La modifica di un prezzo si riflette automaticamente nella modifica del prezzo esposto su un piccolo display posto sugli scaffali dove sono esposti i prodotti.

**INOLTRE:**

La politica della catena è che le modifiche dei prezzi possono essere fatte una sola volta nelle 24 ore e non possono superare il 20 per cento.

# Grande distribuzione

- Per casa
  1. Dare la narrativa del caso d'uso ModificaDeiPrezzi, tenendo conto che la politica della catena è che le modifiche dei prezzi possono essere fatte una sola volta nelle 24 ore e non possono superare il 20 per cento.
  2. Dare un diagramma dei casi d'uso che comprenda : InvioOrdini, RicezioneFornitura, GenerazioneResocontoStato-Magazzino, e CalcoloTempiMedi-Consegna.
  3. Dare la narrativa del caso d'uso InvioOrdini, assumendo che un ordine venga fatto solo se non è possibile il trasferimento interno

# Modifica dei prezzi di vendita

Caso d'uso: ModificaDeiPrezzi

Breve descrizione: *Permette di modificare i prezzi di vendita di alcuni prodotti e la loro visualizzazione sul display allo scaffale.*

Attore principale: *ResponsabilePuntoVendita*

Attore secondario: *Nessuno*

PreCondizioni: *Punto vendita chiuso*

PostCondizioni: *Prezzi aggiornati*

Sequenza principale degli eventi:

1. *per (ogni prezzo da aggiornare)*
  1. *il Responsabile richiama il prodotto*
  2. *il Responsabile indica il nuovo prezzo*
  3. *il Sistema aggiorna la scheda del prodotto*
  4. *il Sistema aggiorna il display*

Sequenza alternativa degli eventi: *Variazione di prezzo superiore al 20% per almeno un articolo.*

Breve descrizione: *Gli articoli in questione vengono modificati in più o meno 20%.*

Sequenza alternativa degli eventi: *Prezzo di almeno un articolo già modificato nelle 24 ore.*

Breve descrizione: *Il prezzo degli articoli in questione non viene modificato e il Responsabile viene avvertito per ognuno.*

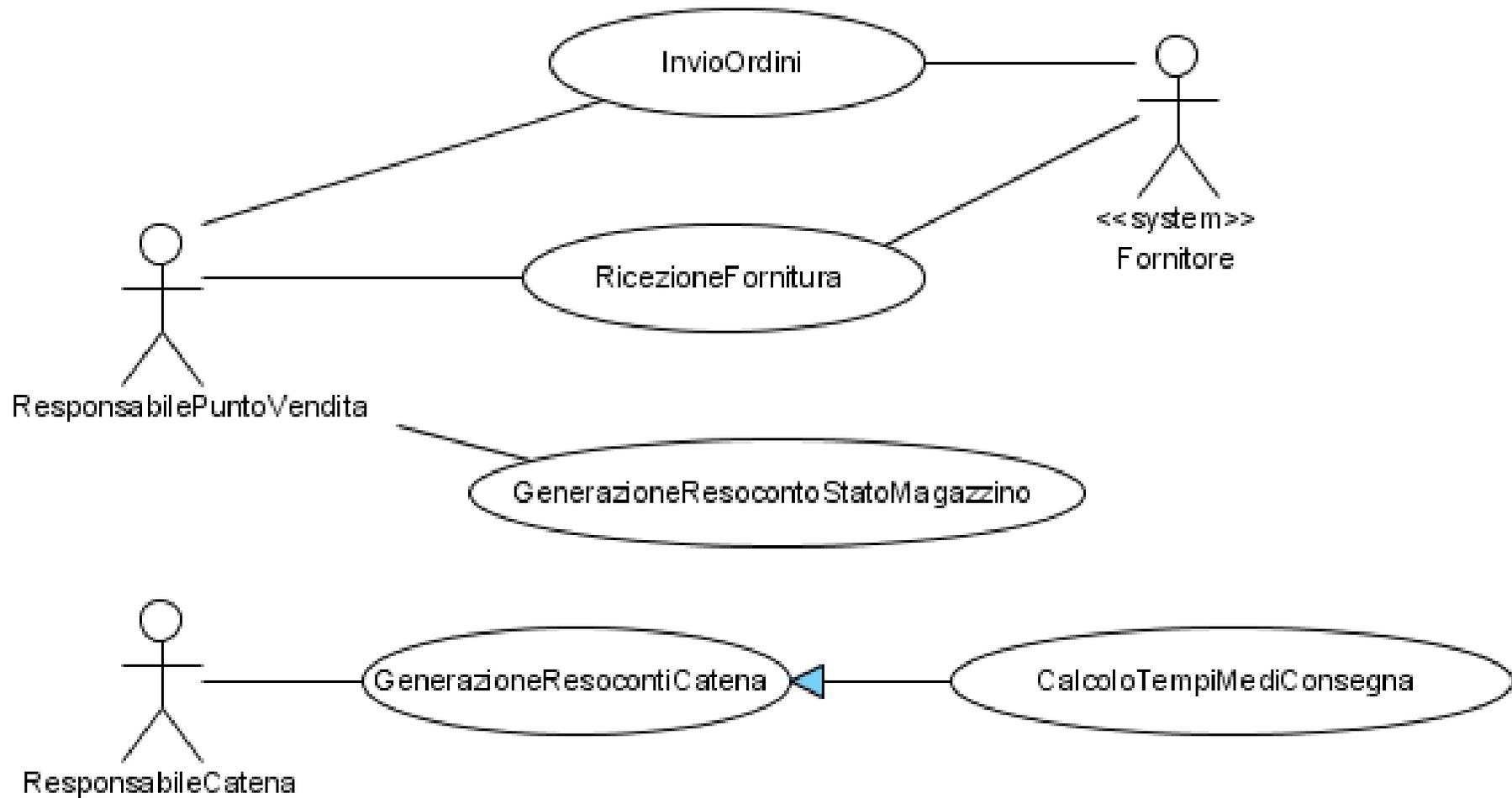
# Grande Distribuzione

InvioOrdini, RicezioneFornitura, GenerazioneResocontoStato-Magazzino, e CalcoloTempiMedi-Consegna.

- Il ServerDelPuntoVendita mantiene l'inventario del corrispondente punto vendita. Il ClientDelPuntoVendita viene usato dal responsabile del punto vendita .... per fare gli ordini. Quando dei prodotti arrivano al punto vendita, il responsabile controlla la corrispondenza rispetto all'ordine fatto prima di inserirli nell'inventario. Inoltre, il TradingSystem permette al responsabile del punto vendita di generare resoconti sullo stato del magazzino.
- Un ServerDellaCatena è collegato a tutti i punti vendita. Usando un ClientDellaCatena, il responsabile della catena può generare diversi tipi di resoconti. Ad esempio, TradingSystem permette di calcolare i tempi medi di consegna da un fornitore ai punti vendita.

# Grande Distribuzione

InvioOrdini, RicezioneFornitura, GenerazioneResocontoStato-Magazzino, e CalcoloTempiMedi-Consegna.



# Grande Distribuzione

**InvioOrdini, RicezioneFornitura, GenerazioneResocontoStato-Magazzino, e CalcoloTempiMedi-Consegna.**

Caso d'uso: InvioOrdini

Breve descrizione: *Permette di inviare un ordine di alcuni prodotti a dei fornitori*

Attore principale: *ResponsabilePuntoVendita*

Attore secondario: *Fornitore*

PreCondizioni: *Prodotti sotto scorta, impossibile il trasferimento interno*

PostCondizioni: *Prodotti ordinati*

Sequenza principale degli eventi:

1. *per (ogni fornitore)*
  1. *il ResponsabilPuntoVendita chiede al TradingSystem di creare un nuovo ordine per il fornitore*
  2. *per (ogni prodotto da ordinare a quel fornitore)*
    1. *il ResponsabilPuntoVendita chiede al TradingSystem di aggiungere il prodotto all'ordine*
  3. *il Responsabile chiede a TradingSystem l'invio dell'ordine*

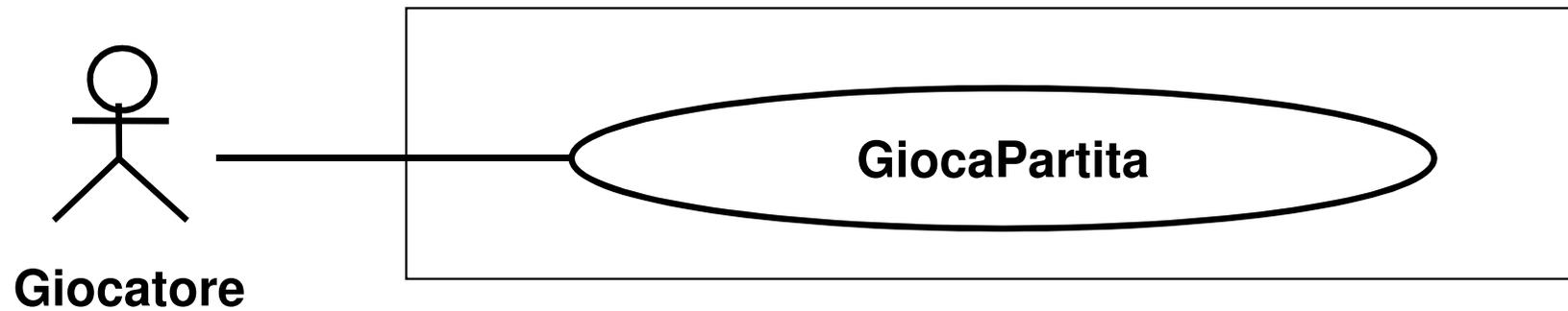
Sequenza alternativa degli eventi:

*Fornitore non più convenzionato con la catena  
Prodotto non più in commercio.*

# Cops and Robbers (?????)

Contrived....

# Cops and Robbers

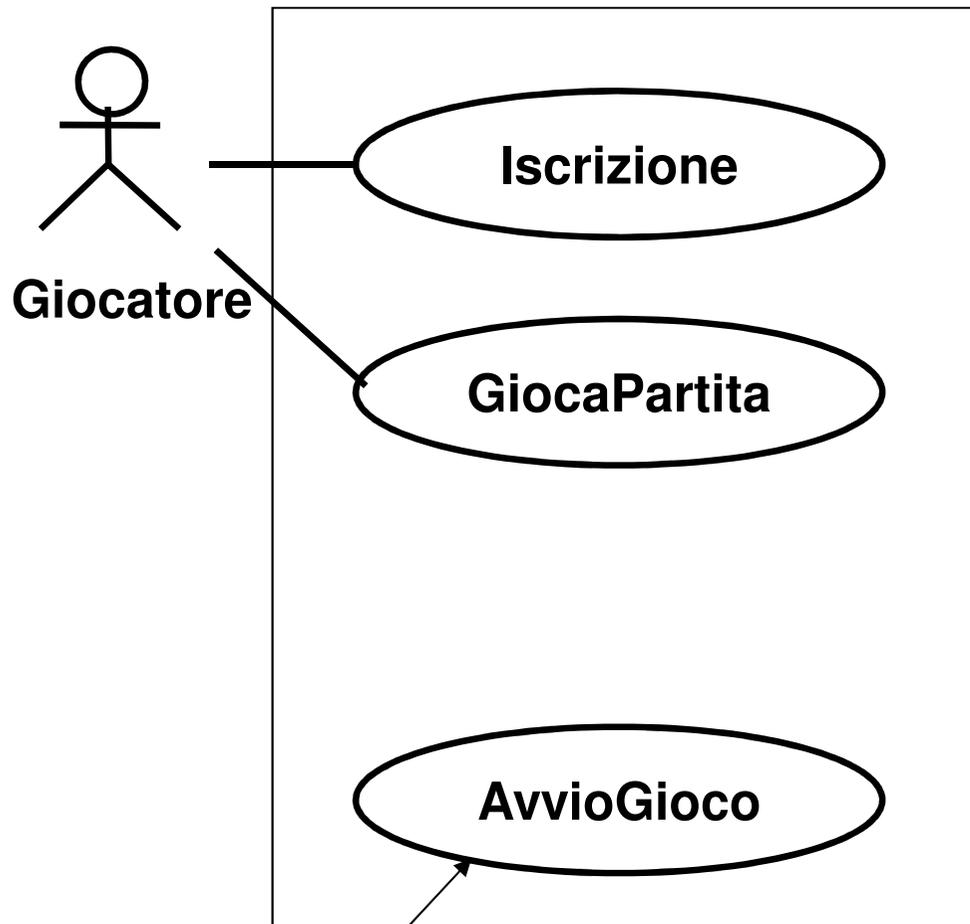


➤ Troppo poco.....

# CopsAndRobbers

- CopsAndRobbers:
  - deve rispettare le Regole del gioco descritte in Appendice A;
  - deve notificare a tutte le entità ogni modifica degli oggetti passivi del mondo virtuale;
  - deve far vedere a ogni entità solo le entità presenti nella stessa stanza;
  - deve lasciare il mondo virtuale in uno stato globale consistente, dopo ogni mossa;
  - deve trattare tutti i ladri e tutte le guardie in modo fair.
- Per l'amministrazione del gioco, CopsAndRobbersAdmin:
  - deve ricevere le richieste di connessione al gioco da parte dei giocatori;
  - quando ha ricevuto esattamente  $k$  richieste, deve
    - inviare ai giocatori la posizione degli oggetti all'interno della mappa, e
    - notificare a ogni giocatore che può iniziare il gioco;
  - deve inviare ai giocatori che ne fanno richiesta la lista dei giocatori che partecipano al gioco, con la loro energia;
  - deve notificare la vincita/perdita ai giocatori.
- Per permettere lo svolgimento del gioco, PeerCopsAndRobbers:
  - deve dar corso alle richieste del giocatore, e notificargliene l'esito ;
  - deve muovere le guardie assegnategli.

# Cops and Robbers errato



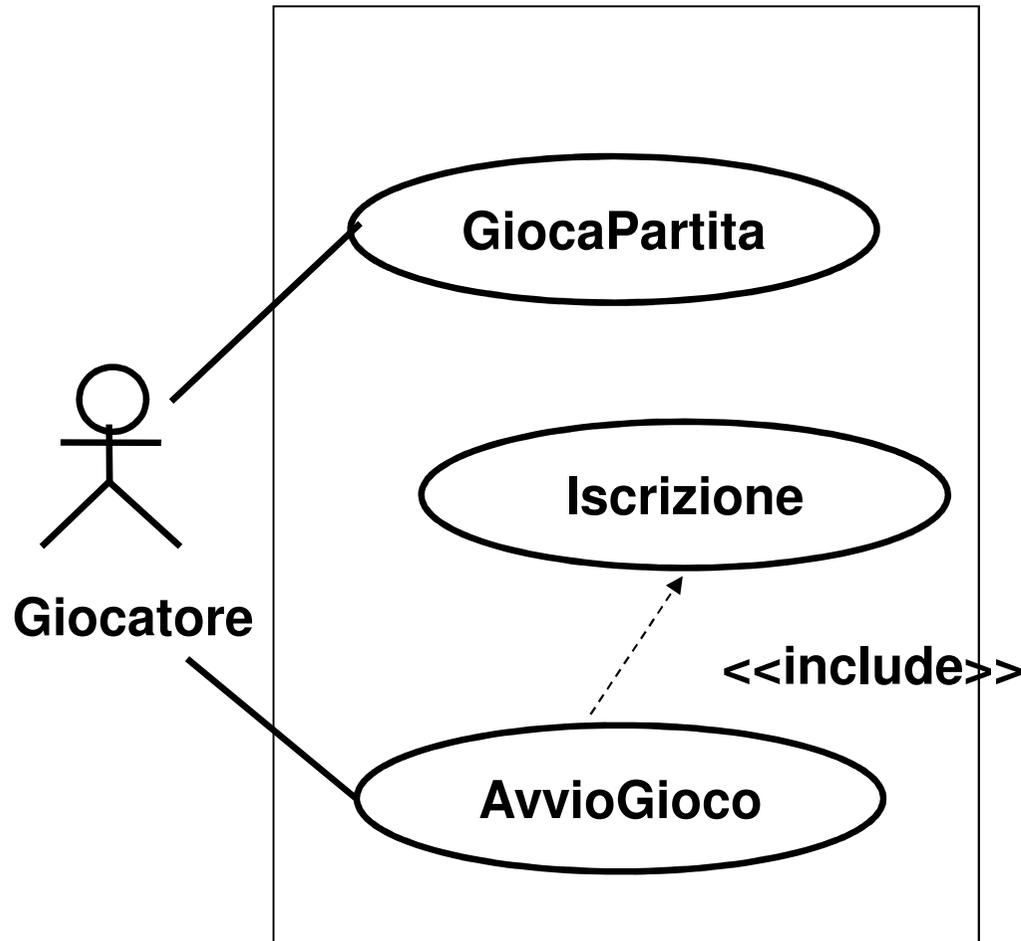
- **Iscrizione**
  - Postcondizione: giocatore iscritto
- **AvvioGioco**
  - Precondizione: k giocatori iscritti
  - Posiziona oggetti nella mappa
  - Postcondizione: gioco avviato
- **GiocaPartita**
  - Precondizione: gioco avviato
  - Postcondizione: vincono ladri o vincono guardie

**Chi è l'attore??**

with one or more actors when providing its service. Note that two use cases that specify the same subject do not communicate with each other because each of them individually describes a complete usage of the system. They may interact indirectly through shared actors.

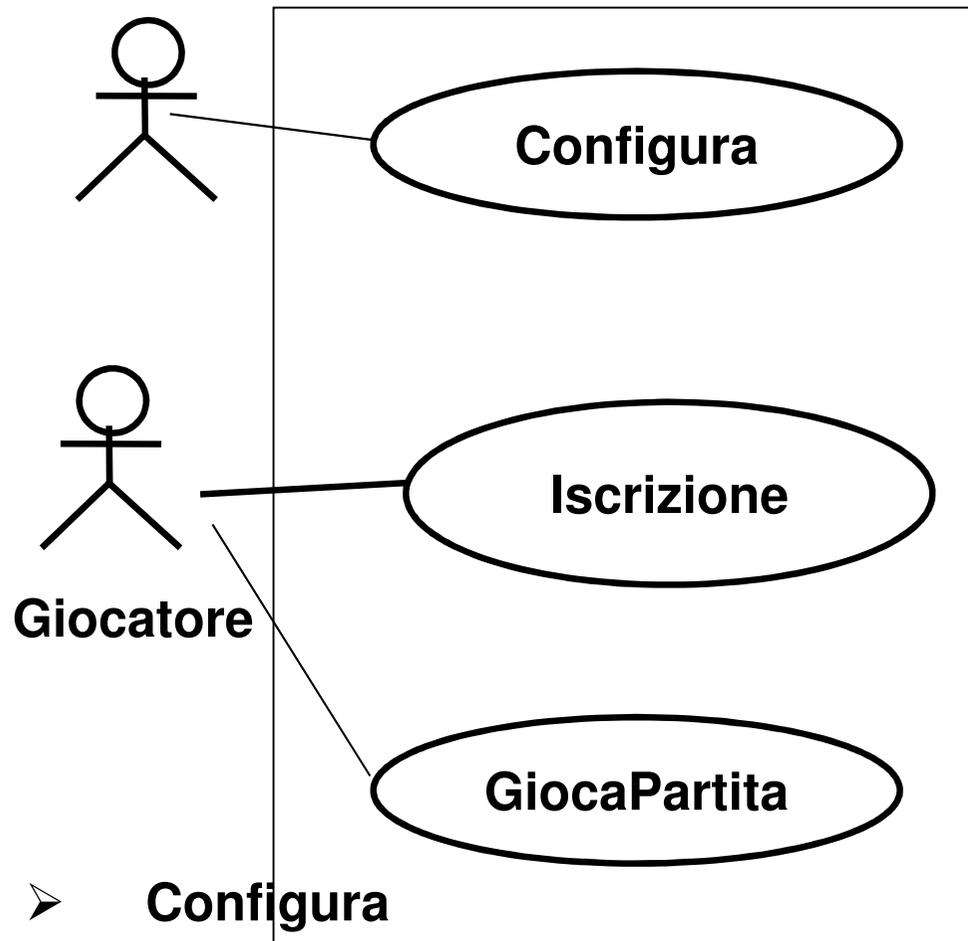
**Non passa il controllo, i dati si**

# Cops and Robbers



- **Iscrizione**
  - Postcondizione: giocatore iscritto
- **AvvioGioco**
  - Postcondizione: gioco avviato
  - Passi Principali:
    - Include Iscrizione
    - If k giocatori iscritti
    - Posiziona oggetti nella mappa
    - Notifica a ogni giocatore che può iniziare il gioco;
- **GiocaPartita**
  - Precondizione: gioco avviato
  - Postcondizione: vincono ladri o vincono guardie
  - Passi: ad ogni turno muove

# Cops and Robbers



## ➤ Configura

- **Passi Principali:**
  1. Decide k
  2. Avvia una sessione di iscrizioni
  3. while (gioco avviato)
    1. Avvia nuova sessione di iscrizioni

## ➤ Iscrizione

- **Postcondizione: gioco avviato o giocatore prenotato**
- **Passi Principali:**
  1. Il giocatore chiede di iscriversi
  2. Il giocatore viene iscritto
  3. If k giocatori prenotati
    1. posiziona oggetti nella mappa
    2. notifica a ogni giocatore prenotato l'inizio del gioco;

## ➤ GiocaPartita

- **Precondizione: gioco avviato**
- **Postcondizione: vincono ladri o vincono guardie**
- **Passi principali:**
  - while(vivo)
    - muovi
    - aspetta esito mossa
  - ricevi notifica causa terminazione: morte, cattura, fine gioco.

# Scomporre GiocaPartita???

- Possibili casi d'uso:
  - Muovi
  - Richiedi energia altrui
  
- A questo punto la fine del gioco è un passo (l'ultimo!) di Avvio Gioco che diventerebbe ArbitraGioco