

## Software Architecture Documentation

# Sistema Controllo Allarmi

Gruppo PSSS\_1846

- Paudice Genny - M63/000448
- Riccio Carlo - M63/000445
- Silvestro Rolando - M63/000482

<b>1</b>	<b>Requisiti informali del sistema software.....</b>	<b>5</b>
<b>2</b>	<b>Analisi del sistema .....</b>	<b>6</b>
2.1	<b>Diagrammi dei casi d'uso .....</b>	<b>6</b>
2.1.1	Descrizione del caso d'uso [Generazione Allarme].....	7
2.1.2	Descrizione del caso d'uso [Gestione e Segnalazione Allarme] .....	8
2.1.3	Descrizione del caso d'uso [Visualizza Elenco di Richieste di Intervento] .....	8
2.1.4	Descrizione del caso d'uso [Visualizza Statistiche Allarmi] .....	9
2.1.5	Descrizione del caso d'uso [Visualizza Elenco Allarmi] .....	9
2.2	<b>Diagrammi di contesto.....</b>	<b>10</b>
2.2.1	Diagramma di contesto [Software Context Diagram] .....	10
2.3	<b>Modello del dominio - System Domain Model .....</b>	<b>12</b>
2.4	<b>Diagrammi dinamici .....</b>	<b>13</b>
2.4.1	Activity - Generazione Allarme .....	13
2.4.2	Communication - Generazione Allarme .....	14
2.4.3	Activity - Gestione e Segnalazione Allarme.....	15
2.4.4	Communication - Gestione e Segnalazione Allarme .....	16
2.4.5	Activity - Visualizza Elenco di Richieste di Intervento.....	17
2.4.6	Communication - Visualizza Elenco di Richieste di Intervento .....	18
2.4.7	Activity - Visualizza Statistiche Allarmi.....	19
2.4.8	Communication - Visualizza Statistiche Allarmi.....	20
2.4.9	Activity - Visualizza Elenco Allarmi.....	21
2.4.10	Communication - Visualizza Elenco Allarmi .....	22
<b>3</b>	<b>Diagrammi architetturali .....</b>	<b>23</b>
3.1	<b>Vista architetturale [Client-Server] .....</b>	<b>23</b>
3.1.1	Descrizione .....	23
3.1.2	Modello.....	23
3.1.3	Elementi non standard .....	23
3.1.3.1	Catalogo delle tipologie di elementi .....	23
3.1.4	Razionale .....	24
3.1.5	Relazioni con altri modelli/elementi di modello .....	24
3.2	<b>Vista architetturale Client Amministratore .....</b>	<b>25</b>
3.2.1	Descrizione .....	25
3.2.2	Modello.....	25
3.2.3	Elementi non standard .....	25
3.2.3.1	Catalogo delle tipologie di elementi .....	26
3.2.4	Razionale .....	27
3.2.5	Relazioni con altri modelli/elementi di modello .....	27
3.3	<b>Vista architetturale Client Gestore .....</b>	<b>27</b>
3.3.1	Descrizione .....	27
3.3.2	Modello.....	28

3.3.3	Elementi non standard.....	28
3.3.3.1	Catalogo delle tipologie di elementi.....	28
3.3.4	Razionale .....	29
3.3.5	Relazioni con altri modelli/elementi di modello .....	29
<b>3.4</b>	<b>Vista architettuale Client Centralina .....</b>	<b>29</b>
3.4.1	Descrizione .....	29
3.4.2	Modello .....	30
3.4.3	Elementi non standard.....	30
3.4.3.1	Catalogo delle tipologie di elementi.....	30
3.4.4	Razionale .....	31
3.4.5	Relazioni con altri modelli/elementi di modello .....	31
<b>3.5</b>	<b>Vista architettuale Server COR .....</b>	<b>31</b>
3.5.1	Descrizione .....	31
3.5.2	Modello .....	32
3.5.3	Elementi non standard.....	33
3.5.3.1	Catalogo delle tipologie di elementi.....	33
3.5.4	Razionale .....	34
3.5.5	Relazioni con altri modelli/elementi di modello .....	34
<b>3.6</b>	<b>Vista architettuale della connessione RMI .....</b>	<b>34</b>
3.6.1	Descrizione .....	34
3.6.2	Modello .....	35
3.6.3	Razionale .....	35
3.6.4	Relazioni con altri modelli/elementi di modello .....	36
<b>3.7</b>	<b>Vista di deploy .....</b>	<b>36</b>
3.7.1	Descrizione .....	36
3.7.2	Modello .....	37
3.7.3	Elementi non standard.....	37
3.7.3.1	Catalogo delle tipologie di elementi.....	37
<b>4</b>	<b>Beyond .....</b>	<b>38</b>
<b>4.1</b>	<b>Vista architettuale di dettaglio - Client Centralina.....</b>	<b>38</b>
4.1.1	Descrizione .....	38
4.1.2	Modello .....	39
4.1.3	Relazioni con altri modelli/elementi di modello .....	39
<b>4.2</b>	<b>Vista architettuale di dettaglio - ServerCOR.....</b>	<b>380</b>
4.2.1	Descrizione .....	380
4.2.2	Modello .....	380
4.2.3	Relazioni con altri modelli/elementi di modello .....	390
<b>4.3</b>	<b>Sequence diagram di dettaglio [Generazione Allarme - lato client] .....</b>	<b>381</b>
4.3.1	Descrizione .....	381
4.3.2	Modello .....	41
<b>4.4</b>	<b>Sequence diagram di dettaglio Gestione e Segnalazione Allarme - lato server..</b>	<b>41</b>
4.4.1	Descrizione .....	41
4.4.2	Modello .....	41

<b>5</b>	<b>Testing.....</b>	<b>43</b>
	5.1 Generazione Allarme e Segnalazione Allarme .....	43
	5.2 Statistiche Amministratore.....	45
	5.3 Elenco Richieste Intervento Gestore .....	47

# 1 Requisiti informali del sistema software

Si vuole realizzare un sistema software per il controllo automatico di allarmi domestici provenienti da abitazioni registrate.

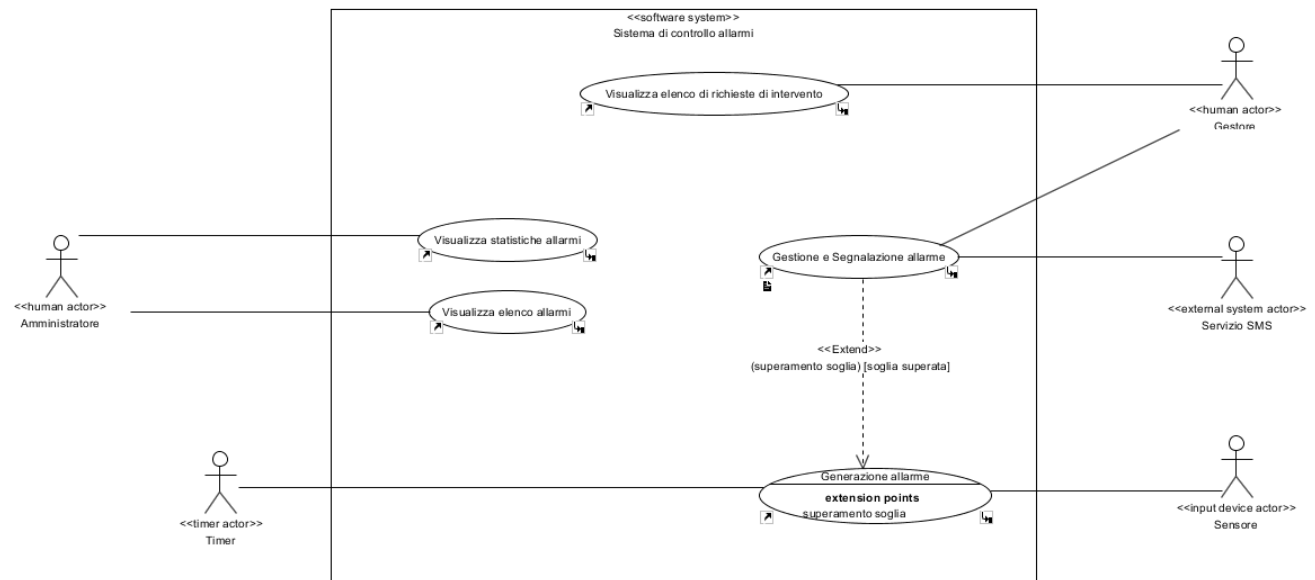
1. **Generazione di una segnalazione di allarme.** La centralina controlla a intervalli di tempo regolari il valore rilevato da ciascun sensore e ne verifica l'eventuale superamento della soglia di allarme. In tal caso, essa deve generare un allarme e svolgere una serie di azioni. In particolare, deve inviare un comando di attivazione ai segnalatori acustici (sirene di allarme) posti presso l'abitazione, un messaggio di allarme alla centrale operativa remota, ed un SMS al proprietario dell'abitazione con indicazione di data, ora e tipo di allarme verificatosi. Il messaggio di allarme inviato alla centrale operativa include le informazioni di data, ora, identificativo univoco dell'abitazione di origine e l'identificativo del sensore che ha generato l'allarme.
2. **Gestione allarme da parte della centrale** Alla ricezione di una segnalazione allarme, la centrale verifica che questa arrivi da una delle abitazioni da essa gestita e sia di una tipologia che essa sia in grado di gestire. In caso affermativo, un sms al proprietario dell'abitazione deve essere inviato indicando la tipologia di allarme e la data e ora in cui si è verificato. Inoltre una richiesta di intervento, associata alla segnalazione di allarme deve essere registrata e un'apposita notifica inviata al gestore della tipologia di allarme.
3. **Visualizza elenco di richieste di intervento di un dato tipo.** Il gestore di un particolare tipo di allarme deve visualizzare la lista delle richieste di intervento relative a segnalazione di allarme aperte della tipologia da lui gestita. In particolare per ciascuna richiesta di intervento sarà riportata la data della sua creazione e l'abitazione cui fa riferimento.
4. **Visualizza statistiche su una tipologia di allarme in un dato periodo.** L'amministratore della centrale deve poter visualizzare le statistiche relative alle segnalazioni di allarmi di una data tipologia in un dato periodo. A tal fine, tramite un'apposita interfaccia grafica, l'amministratore seleziona la tipologia di allarme tra quelle gestite dalla centrale e inserisce il periodo per il quale vuole ottenere le statistiche (sotto forma di data di inizio e data di fine). Il sistema controlla che è stato inserito un periodo valido e mostra la lista delle abitazioni interessate da almeno una segnalazione di allarme della tipologia selezionata nel periodo indicato. Per ciascuna abitazione saranno riportati: il numero di segnalazioni di allarmi sollevate e il numero di segnalazioni correttamente risolte (stato "Chiusa").
5. **Visualizza elenco allarmi maggiormente sollevati/non risolti.** L'amministratore della centrale deve poter visualizzare l'elenco delle tipologie di allarmi ordinati in base al numero di segnalazioni di allarme avute o in base al numero di segnalazione che presentano richieste di intervento associate con esito "Non Risolutivo".

## 2 Analisi del sistema

### Stakeholder principali:

1. Proprietario: il possessore dell'abitazione che viene avvisato con un SMS quando si è aperta una segnalazione di allarme
2. Amministratore: colui che gestisce le abitazioni registrate, può visualizzare statistiche.
3. Gestore: colui che viene avvisato di un allarme e che deve intervenire nell'abitazione da cui proviene l'allarme

### 2.1 Diagrammi dei casi d'uso



Nome elemento	Tipo	Descrizione
Amministratore	Attore Primario	Utente umano che interagisce con il sistema per visualizzare statistiche
Timer	Attore Primario	Timer esterno che controlla periodicamente i sensori dell'abitazione

Gestore	Attore Primario e secondario	Utente umano che viene avvisato in caso di allarme e che interagisce con il sistema per visualizzare richieste di intervento da gestire
Sensore	Attore Secondario	Dispositivo di input che viene interrogato dal sistema
Servizio SMS	Attore Secondario	Sistema esterno per l'invio di un SMS al proprietario in caso di allarme
Generazione allarme	Caso d'uso	Descrive la generazione di un messaggio di allarme e l'attivazione dei segnalatori acustici
Gestione e Segnalazione Allarme	Caso d'uso	Descrive la registrazione di una segnalazione di allarme e di una richiesta di intervento
Visualizza elenco di richieste di intervento	Caso d'uso	Descrive la richiesta di visualizzare le richieste di intervento da parte d un gestore
Visualizza Statistiche allarmi	Caso d'uso	Descrive la procedura con cui l'amministratore visualizza le statistiche degli allarmi
Visualizza elenco allarmi	Caso d'uso	Descrive la procedura con cui l'amministratore visualizza l'elenco degli allarmi

*Tabella 1: Elementi del diagramma dei casi d'uso*

Nelle sottosezioni successive è riportata una descrizione dettagliata di ogni caso d'uso.

### **2.1.1 Descrizione del caso d'uso [Generazione Allarme]**

**Sommario:** Descrive la generazione di un messaggio di allarme e l'attivazione dei segnalatori acustici

**Attori:** Timer e sensori

**Precondizione:** Centralina e sensori attivi

**Sequenza principale:**

1. Il sistema controlla la soglia del sensore
2. Se la soglia è superata
  - 2.1 <<superamento soglia>> (estende il secondo caso d'uso)
  - 2.2 Il sistema attiva i segnalatori acustici
  - 2.3 Il sistema raccoglie le informazioni di allarme (data,ora ecc)
  - 2.4 Il sistema invia l'allarme alla centrale operativa.

### **2.1.2 Descrizione del caso d'uso [Gestione e Segnalazione Allarme]**

**Sommario:** Controllo dell'allarme generato e segnalazione al gestore

**Attori:** Gestore, Servizio SMS

**Precondizione:** Messaggio di allarme ricevuto dalla centrale operativa remota

**Sequenza principale:**

1. Il sistema verifica i dati inviati dall'allarme
2. Se i dati sono corretti
  - 2.1 Il sistema apre e registra una segnalazione di allarme con relativa richiesta di intervento.
  - 2.2 Il sistema raccoglie le info per l'invio del SMS.
  - 2.3 Il sistema richiede l'invio di un SMS (al Servizio SMS) al proprietario con data, ora e tipo allarme.
  - 2.4 Il sistema invia notifica del tipo allarme al gestore corrispondente.

### **2.1.3 Descrizione del caso d'uso [Visualizza Elenco di Richieste di Intervento]**

**Sommario:** Richiesta di visualizzazione elenco di richieste di intervento

**Attori:** Gestore

**Precondizione:** Esiste un elenco di richieste di intervento

**Sequenza principale:**

1. Il gestore richiede di visualizzare un elenco di richieste di intervento
2. Il sistema raccoglie data creazione e abitazione di riferimento



3. Il sistema mostra a video l'elenco delle richieste di intervento relative alle segnalazioni di allarme

#### **2.1.4 Descrizione del caso d'uso [Visualizza Statistiche Allarmi]**

**Sommario:** Richiesta di visualizzazione statistiche allarmi

**Attori:** Amministratore

**Precondizione:** Esistono statistiche

**Sequenza principale:**

1. L'amministratore seleziona la tipologia di allarme
2. L'amministratore inserisce una data d'inizio e una data di fine
3. Se la data di inizio è minore della data di fine
  - 3.1 Il sistema raccoglie il numero di segnalazione di allarme sollevate e il numero di segnalazioni correttamente risolte
  - 3.2 Il sistema mostra a video le statistiche

##### **2.1.4.1 Scenario alternativo**

Se la data di inizio è maggiore di quella di fine il sistema mostra a video un messaggio di errore

#### **2.1.5 Descrizione del caso d'uso [Visualizza Elenco Allarmi]**

**Sommario:** Richiesta di visualizzazione elenco allarmi

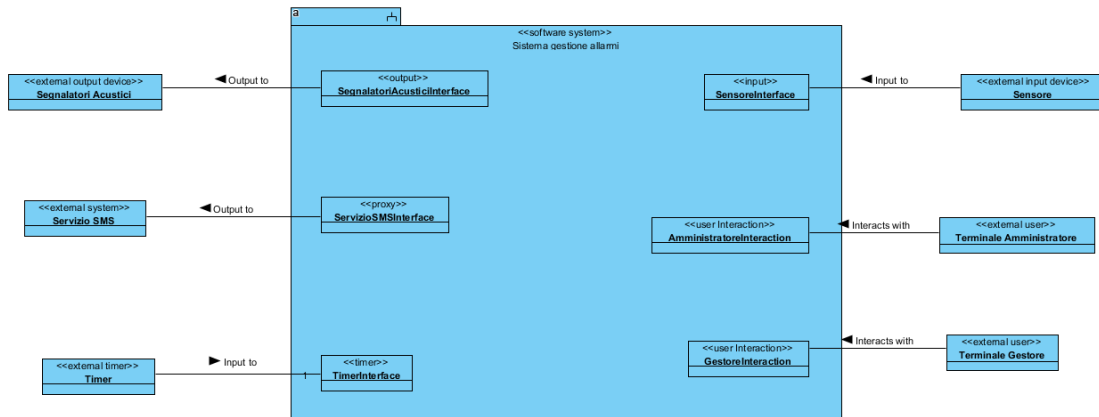
**Attori:** Amministratore

**Precondizione:** Esistono segnalazioni di allarme

**Sequenza principale:**

1. L'amministratore seleziona l'ordinamento desiderato (sollevati o non risolti)
2. Il sistema raccoglie le informazioni per la creazione della richiesta
3. Il Sistema mostra l'elenco delle tipologie d'allarme ordinato in base alla richiesta

## 2.2 Diagrammi di contesto



### 2.2.1 Diagramma di contesto [Software Context Diagram]

A seguito dell'analisi effettuata sui casi d'uso, il diagramma di contesto, di cui sopra, evidenzia gli oggetti necessari per il soddisfacimento di ogni requisito, delimitando inoltre i confini tra il sistema interno di gestione degli allarmi e gli oggetti esterni che permettono l'interazione con gli utenti dello stesso.

Riportiamo una breve descrizione dell'individuazione delle boundaries classes e per ognuna la corrispondente interazione con le classi esterne

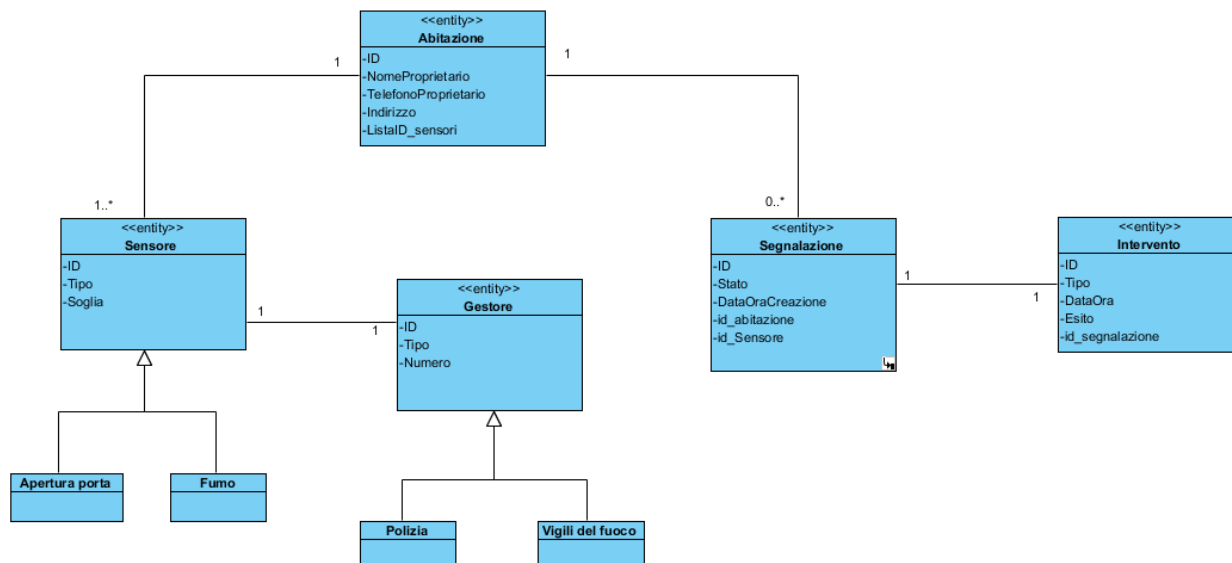
- *AmministratoreInteraction*: interagisce con l'amministratore grazie al terminale esterno *Terminale Amministratore*
- *GestoreInteraction*: interagisce con il gestore grazie al terminale esterno *Terminale Gestore*
- *SensoreInterface*: permette il prelievo del valore rilevato sul dispositivo *Sensore*
- *SegnalatoriAcusticiInterface*: comunicano con il dispositivo di output *Segnalatori Acustici*
- *ServizioSMSInterface*: comunica con il *Sistema SMS esterno* per effettuare l'invio di un SMS
- *TimerInterface*: comunica con un oggetto di controllo Timer Esterno *Timer*

Nome elemento	Tipo	Descrizione
Terminale Amministratore	External user	Postazione che permette all'amministratore di visualiz-

		zare statistiche e allarmi
Terminale Gestore	External user	Postazione che permette al gestore di visualizzare richieste di intervento aperte
Sensore	External input device	Dispositivo che fornisce i dati rilevati al sistema
Segnalatori Acustici	External output device	Dispositivo che in caso di superamento soglia si attivano (sirene acustiche)
Servizio SMS	External system	Servizio esterno per la messaggistica
Timer	External Timer	Timer esterno che fornisce la nozione di tempo reale al sistema
AmministratoreInteraction	User Interaction	Si interfaccia con Terminale Amministratore fornendo statistiche ed elenco allarmi
GestoreInteraction	User Interaction	Si interfaccia con Terminale Gestore fornendo elenco di richieste di intervento
SensoreInterface	Input	Si interfaccia con un dispositivo di input Sensore per prelevare il valore rilevato
SegnalatoriAcusticiInterface	Output	Si interfaccia con un dispositivo di output per attivare i Segnalatori acustici
<i>ServizioSMSInterface</i>	Proxy	Comunica con il servizio esterno di invio SMS, richiedendo l'invio di un SMS
TimerInterface	Timer	Comunica con il timer esterno

Tabella 2: Elementi del diagramma di contesto

## 2.3 Modello del dominio



Nome elemento	Descrizione
Sensore	Memorizza tipo e soglia di ogni sensore
Abitazione	Memorizza il nome e il telefono del proprietario, l'indirizzo della casa e la lista degli id dei sensori che possiede
Segnalazione	Memorizza data e ora della creazione della segnalazione lo stato e l'abitazione e il sensore a cui fa riferimento
Intervento	Memorizza data e ora dell'intervento, il tipo di allarme, l'esito e la segnalazione a cui fa riferimento
Gestore	Memorizza tipo di allarme gestito e numero di telefono

## **2.4 Diagrammi dinamici**

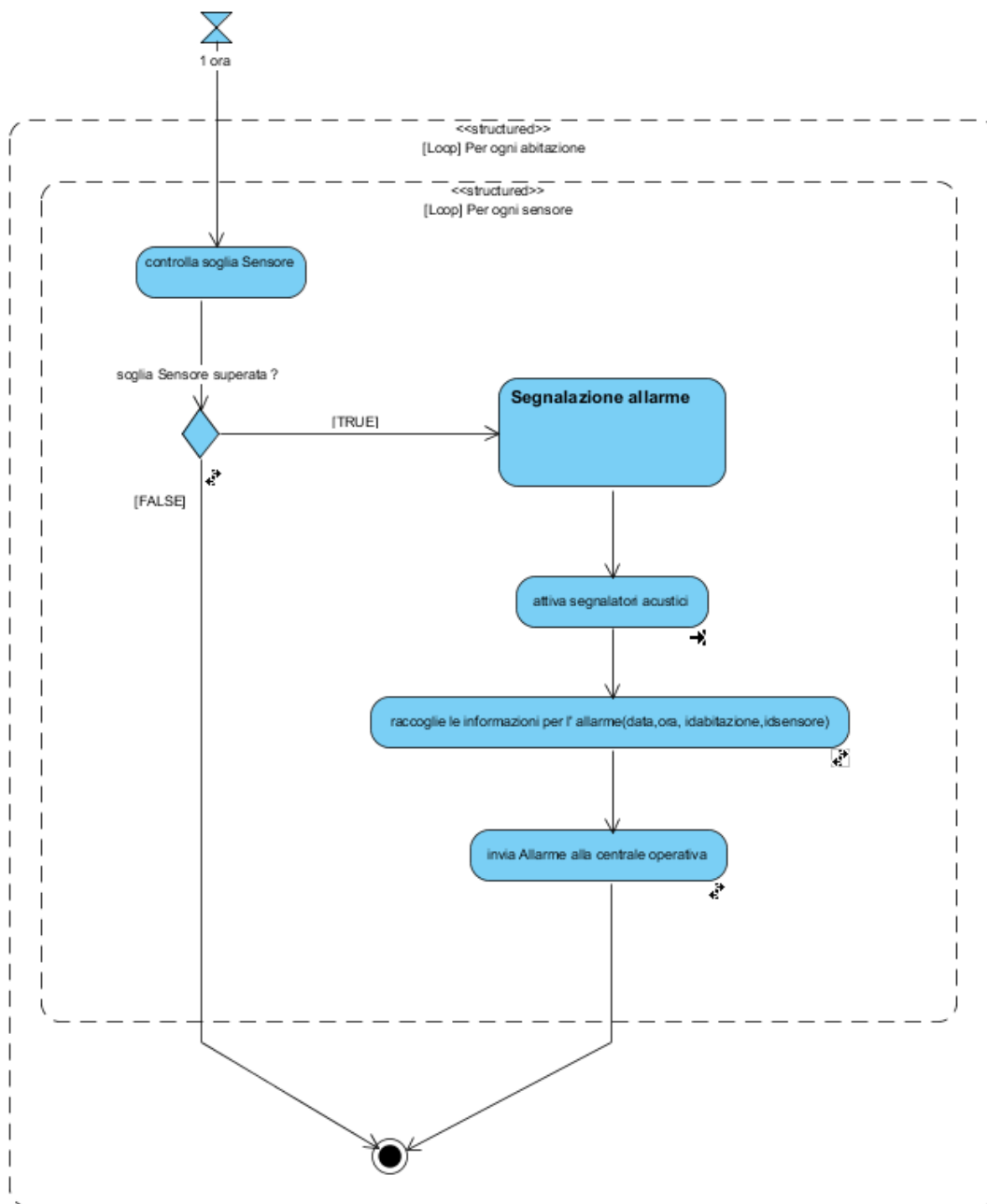
### **2.4.1 Activity Diagram - Generazione Allarme**

#### **2.4.1.1 Descrizione**

Periodicamente la centralina controlla il valore di soglia di tutti i sensori presenti nell'abitazione e se la soglia viene superata attiva i segnalatori acustici e invia un messaggio di allarme alla centrale operativa remota.

Di particolare importanza si segnala l'attività in grassetto “**Segnalazione Allarme**” a cui è collegato concettualmente il diagramma del successivo caso d'uso che, per esigenze di progetto in Visual Paradigm si è preferito collegarlo come sub-diagram del caso uso esteso.

### 2.4.1.2 Modello



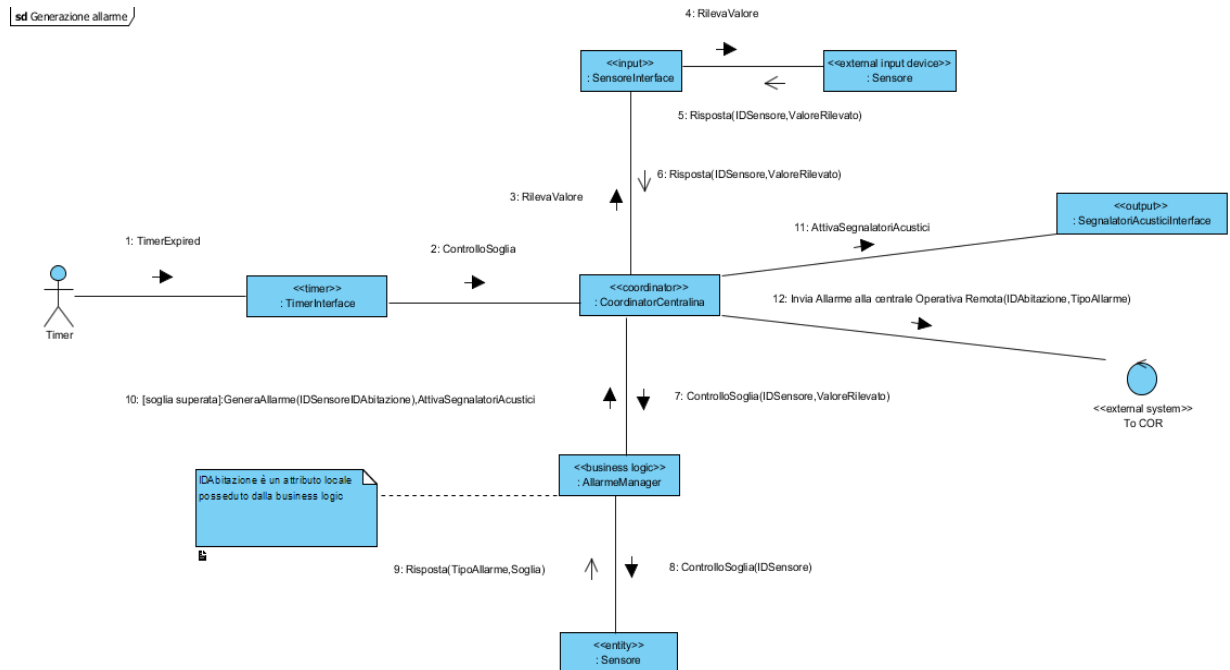
## 2.4.2 Communication Diagram – Generazione Allarme

### 2.4.2.1 Descrizione

Il timer fa partire la comunicazione che viene gestita da un oggetto di controllo coordinator che rileva il valore dal sensore fisico attraverso la sua interfaccia e lo invia alla business logic AllarmeManager che controlla il valore rilevato non prima di aver prelevato la relativa soglia

dall'entity sensore. Il diagramma è parziale perché in caso di superamento soglia viene esteso un altro caso d'uso

### 2.4.2.2 Modello

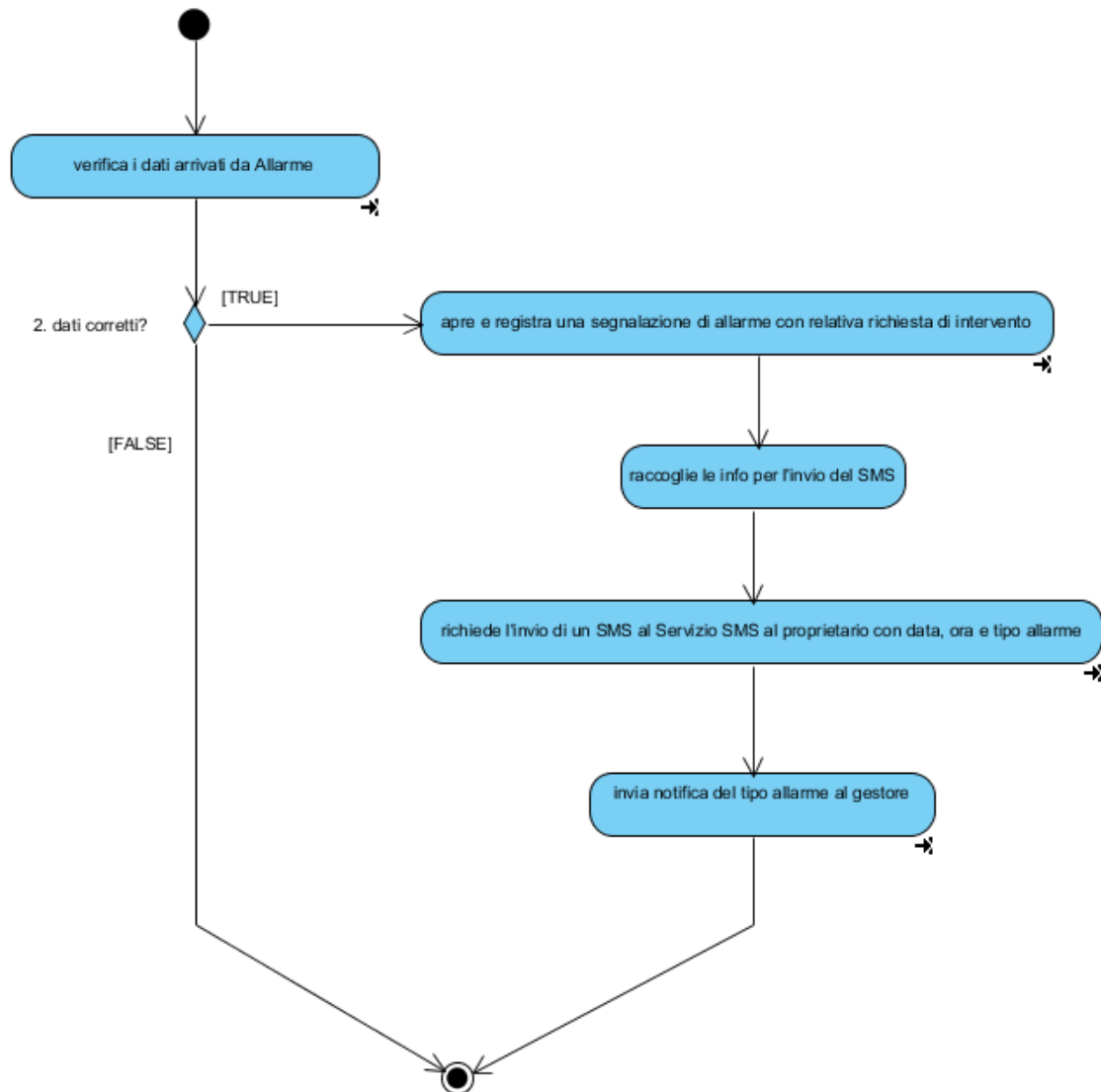


## 2.4.3 Activity Diagram - Gestione e Segnalazione Allarme

### 2.4.3.1 Descrizione

Al superamento della soglia il messaggio arriva alla centrale operativa che verifica se i dati ricevuti sono corretti e in caso affermativo apre e registra una segnalazione di intervento con relativa richiesta di intervento, invia un SMS al proprietario dell'abitazione da cui proviene l'allarme e notifica il gestore appropriato

### 2.4.3.2 Modello



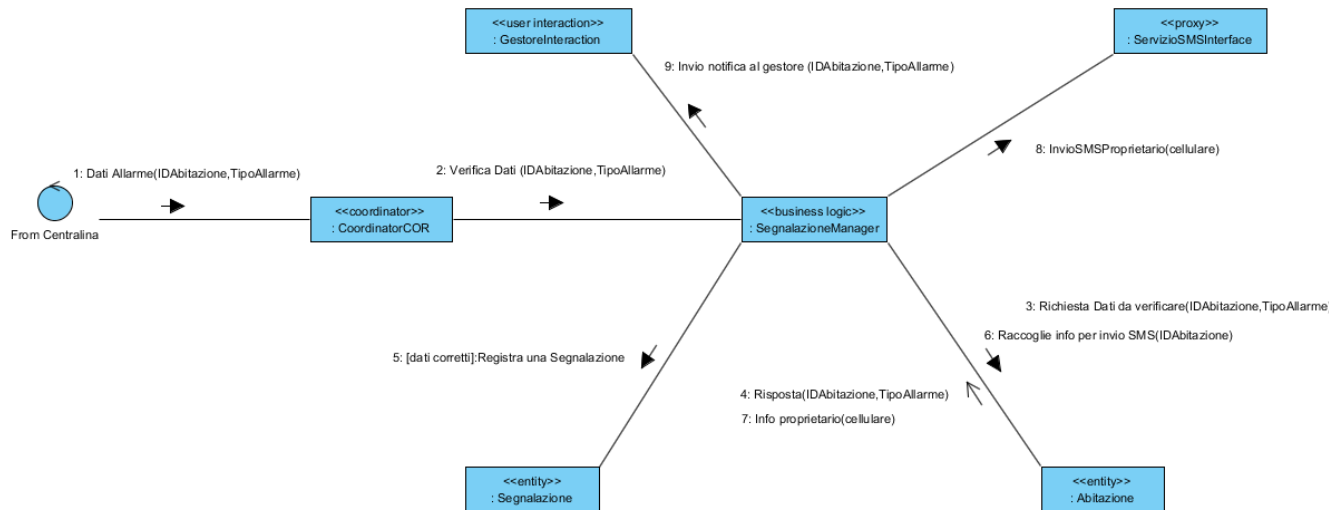
## 2.4.4 Communication Diagram - Gestione e Segnalazione Allarme

### 2.4.4.1 Descrizione

Il caso d'uso comincia come estensione del precedente, i dati inviati dalla centralina alla centrale operativa vengono raccolti da un coordinator che li affida ad una business logic SegnalazioneManager che si occupa di aprire e registrare una segnalazione oltre ad avvisare il proprietario con un SMS

### 2.4.4.2 Modello



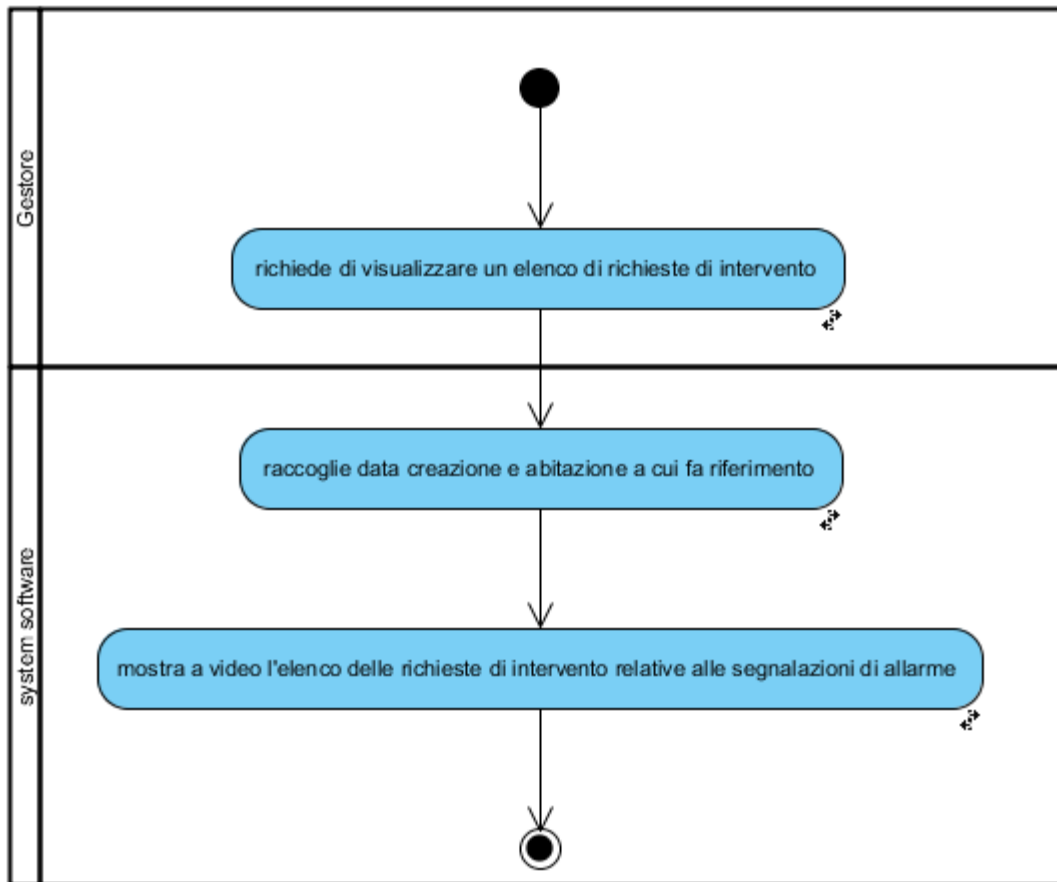


## 2.4.5 Activity - Visualizza Elenco di Richieste di Intervento

### 2.4.5.1 Descrizione

In questo caso d'uso abbiamo inserito le swimline che mostrano le diverse attività tra gestore e sistema: il gestore richiede di visualizzare un elenco di richieste di intervento, il sistema elabora la richiesta e mostra a video la richiesta

### 2.4.5.2 Modello



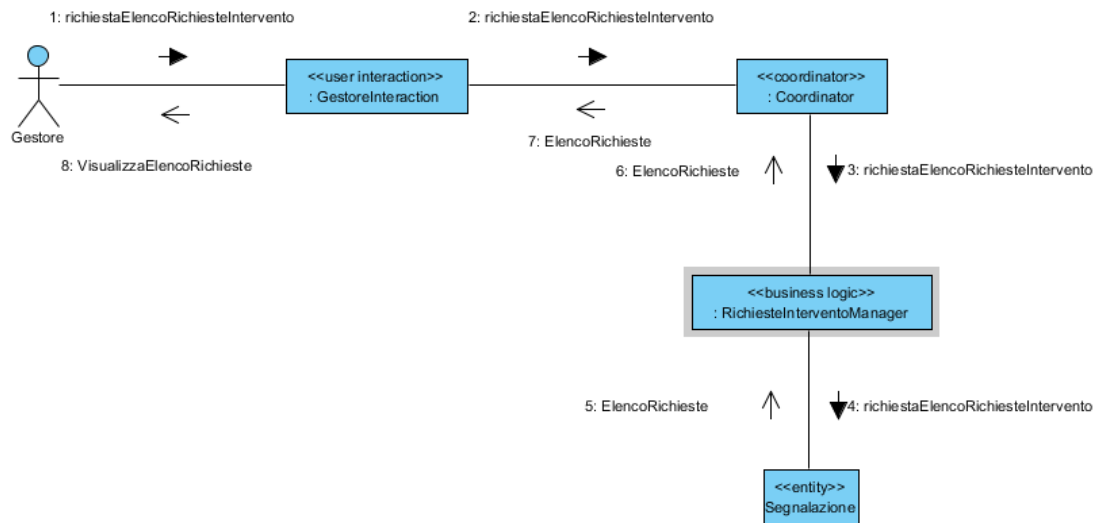
## 2.4.6 Communication - Visualizza Elenco di Richieste di Intervento

### 2.4.6.1 Descrizione

Il gestore fa partire la comunicazione con la sua richiesta ed un apposita business logic si occupa di soddisfarla

## 2.4.6.2 Modello

sd Visualizza elenco di richieste di intervento-Communication Diagram

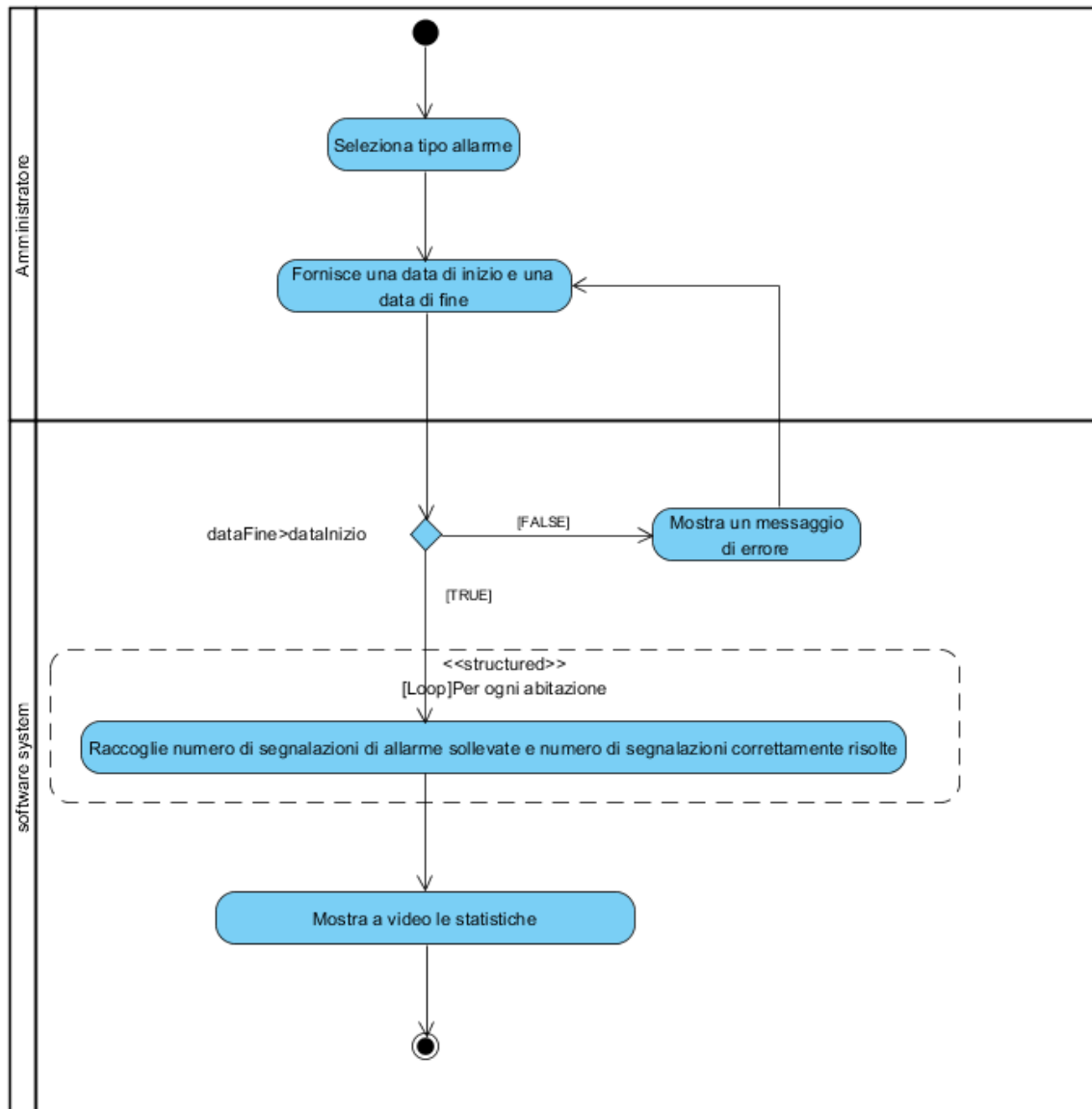


## 2.4.7 Activity - Visualizza Statistiche Allarmi

### 2.4.7.1 Descrizione

L'amministratore seleziona il tipo di allarme e fornisce una data di inizio e fine, se la data inserita è coerente il sistema raccoglie e mostra a video le statistiche altrimenti mostra un messaggio di errore. Anche in questo caso le swimline permettono di separare i compiti tra utente umano e sistema

### 2.4.7.2 Modello

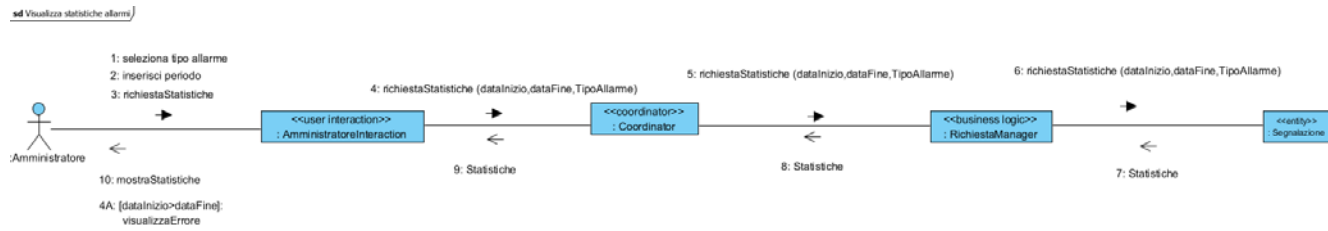


## 2.4.8 Communication - Visualizza Statistiche Allarmi

### 2.4.8.1 Descrizione

L'amministratore far partire la comunicazione che viene prima gestita da un coordinator che si occupa del controllo dei dati di input e poi in seguito viene elaborata e soddisfatta dal sistema tramite una business logic "RichiestaManager"

## 2.4.8.2 Modello

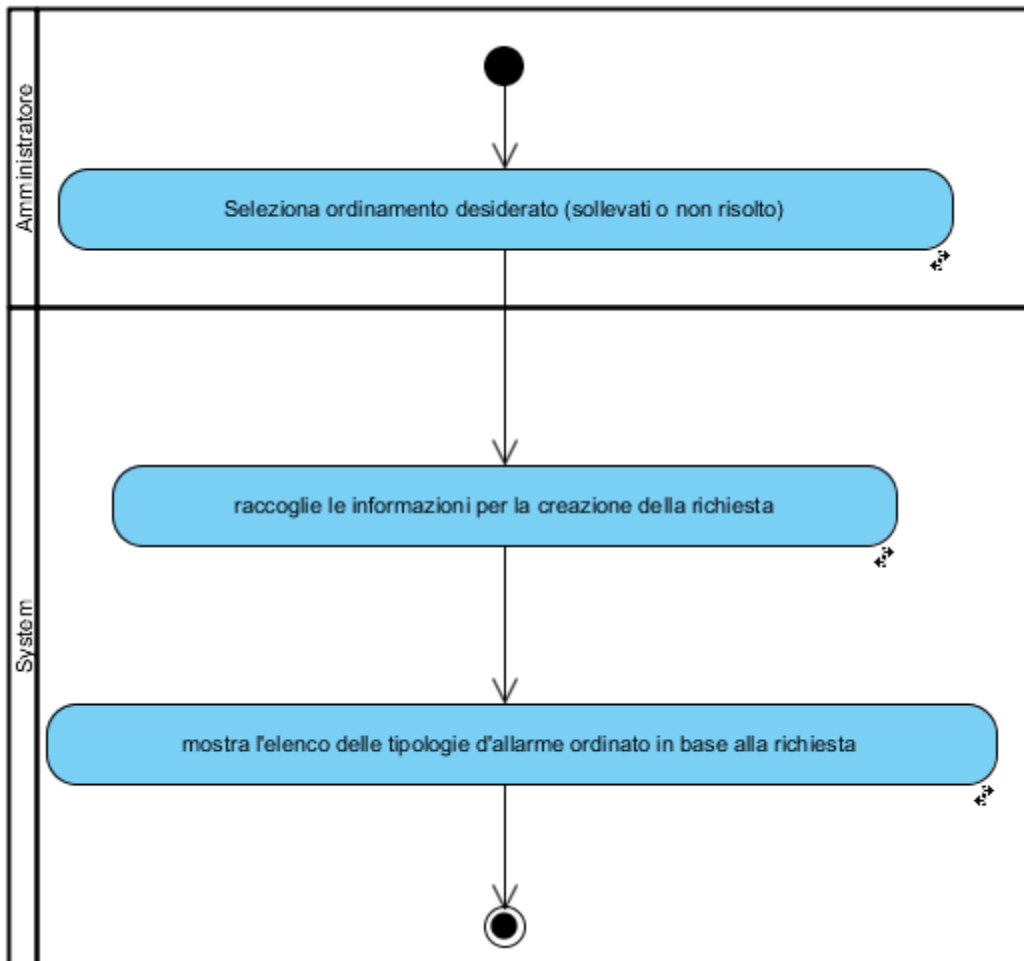


## 2.4.9 Activity - Visualizza Elenco Allarmi

### 2.4.9.1 Descrizione

L'amministratore seleziona l'ordinamento degli allarmi desiderato, il sistema elabora e mostra l'elenco ordinato in base alla richiesta.

### 2.4.9.2 Modello

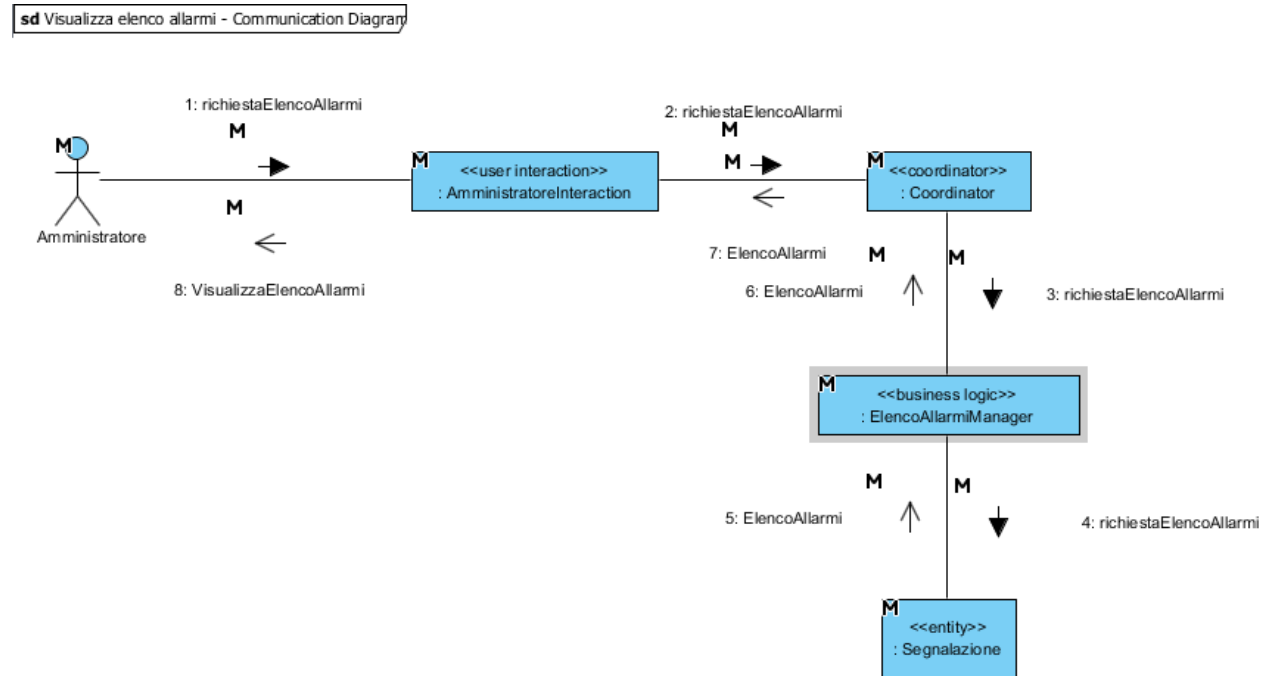


## 2.4.10 Communication - Visualizza Elenco Allarmi

### 2.4.10.1 Descrizione

L'amministratore richiede gli allarmi, un coordinator gestisce la richiesta che viene elaborata da una business logic che raccoglie gli allarmi e li mostra al richiedente

### 2.4.10.2 Modello



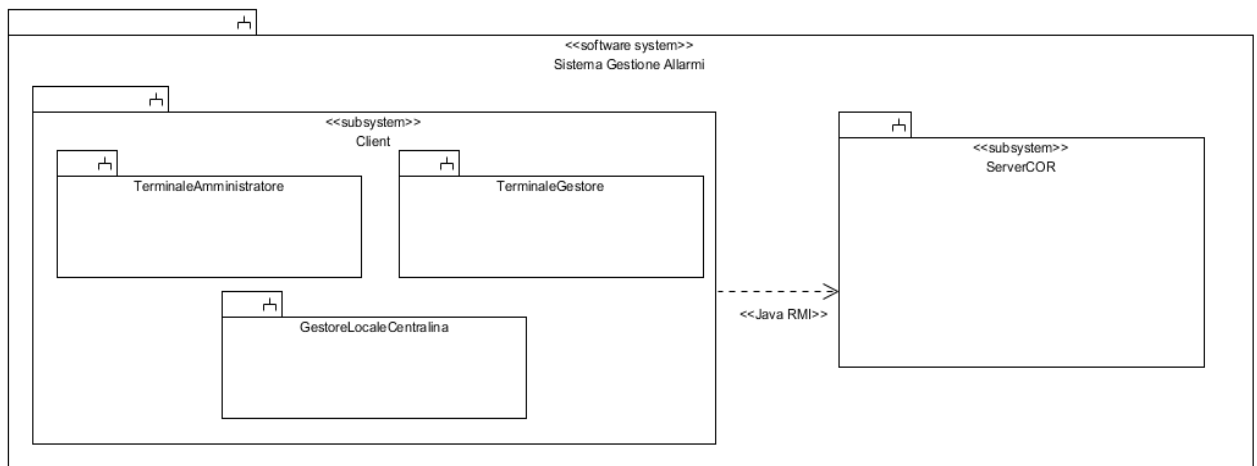
## 3 Diagrammi architetturali

### 3.1 Vista architetturale [Client-Server]

#### 3.1.1 Descrizione

La soluzione scelta per realizzare il sistema è una soluzione Client-Server. La modellazione delle componenti architetturali è presentata con una vista di decomposizione in subsystem, il cui dettaglio è chiarito meglio nei diagrammi architetturali che seguiranno. Alla vista di decomposizione viene applicato inoltre uno stile d'uso che definisce l'unica relazione di uso stereotipata <<Java RMI>>. Tale relazione chiarisce il meccanismo di comunicazione tra la parte client e server del sistema.

#### 3.1.2 Modello



#### 3.1.3 Elementi non standard

##### 3.1.3.1 Catalogo delle tipologie di elementi

Nome	Stereotipo testuale	Acronimo	Descrizione
TerminaleAmministratore	<<Subsystem>>	TA	Terminale dell'amministratore
TerminaleGestore	<<Subsystem>>	TG	Terminale del gestore
GestoreLocaleCentralina	<<Subsystem>>	GLC	Centralina locale presente in un'abitazione

ServerCentraleOperativa Remota	<<Subsystem>>	COR	Parte server del Sistema software
Sistema Gestione Allarmi	<<SoftwareSystem>>	SGA	Intero Sistema software

*Tabella 4 - Catalogo delle tipologie di elementi*

### 3.1.3.2 Catalogo delle tipologie di relazioni

Nome	Stereotipo testuale	Acronimo	Descrizione
JavaRMI	<<Java RMI>>	RMI	Paradigma per l'invocazione dei servizi del server

*Tabella 5 - Catalogo delle tipologie di relazioni*

### 3.1.4 Rationale

La scelta di utilizzare un architettura client-server è stata effettuata considerando la facilità di scalabilità con cui possono essere replicate le diverse componenti architetturali del sistema. Dato che le funzionalità sono pensate per essere offerte in remoto, allora si è scelto come paradigma di comunicazione quello offerto da Java RMI.

### 3.1.5 Relazioni con altri modelli/elementi di modello

Link di tracciabilità	Descrizione
Vista Architetturale del Client Centralina	Vista a moduli del client centralina
Vista Architetturale del Client Amministratore	Vista a moduli del client amministratore
Vista Architetturale del Client Gestore	Vista a moduli del client gestione
Vista Architetturale del Server	Vista a moduli del server che comunica con i client del sistema



Vista Architettuale della connessione RMI	Vista che definisce la connessione remota tra i subsystem del sistema
-------------------------------------------	-----------------------------------------------------------------------

Tabella 6: Relazioni con altri modelli o elementi di modello

## 3.2 Vista architettuale Client Amministratore

### 3.2.1 Descrizione

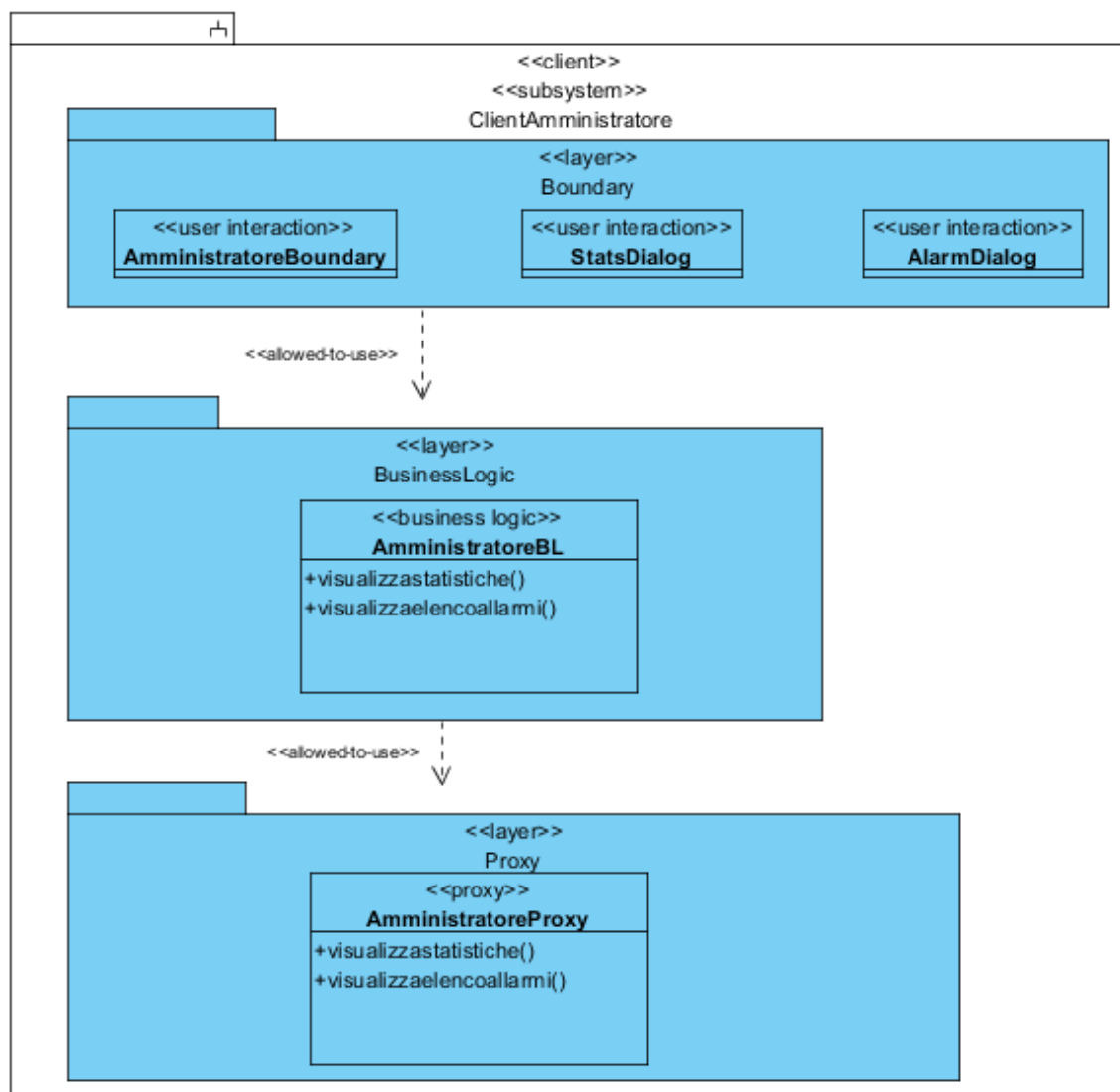
La modellazione della postazione di visualizzazione è definita come subsystem, dato che esso è una parte del sistema. Tale subsystem è modellato mediante una vista layered a tre layers dove:

1. *Boundary Layer*: implementa il boundary tra l'utente e il sistema:
  - *Amministratore Boundary* corrisponde all'interfaccia con il quale viene fatta la richiesta.
  - *StatsDialog* corrisponde alla schermata sulla quale vengono visualizzate le statistiche
  - *AlarmDialog* corrisponde alla schermata sulla quale vengono visualizzati gli allarmi
2. *Business Logic Layer* implementa la logica applicativa del Client Amministratore
3. *Proxy Layer*: implementa l'interfacciamento del client verso il server.

Il primo layer rappresenta il boundary tra l'utente e le funzionalità del sistema.

Gli stili che sono stati usati in questa vista sono **decomposition style** in package/classi, **layered style** che fissa dipendenze di uso tra i diversi layer e moduli del subsystem presentato.

### 3.2.2 Modello



### 3.2.3 Elementi non standard

#### 3.2.3.1 Catalogo delle tipologie di elementi

Nome	Stereotipo testuale	Acronimo	Descrizione
ClientAmministratore	<<Subsystem>>	CA	Terminale dell'amministratore
Boundary layer	<<boundary >>	CABL	boundary tra l'utente e il sistema
Business logic layer	<<business logic>>	CABLL	Logica di business del client

Proxy layer	<<proxy>>	CAPL	Interfacciamento verso il lato server del sistema
-------------	-----------	------	---------------------------------------------------

### 3.2.4 Rationale

La scelta di modellare il client dell'amministratore a livelli è pensata in funzione dell'espandibilità che può essere fatta sul sistema e in funzione della manutenibilità del sistema stesso, in termini di future aggiunta di altri moduli ai layer esistenti.

### 3.2.5 Relazioni con altri modelli/elementi di modello

Link di tracciabilità	Descrizione
Vista Architettuale del Server	Vista del server che comunica con il client amministratore attraverso un interfaccia grafica dove richiede statistiche

## 3.3 Vista architetturale Client Gestore

### 3.3.1 Descrizione

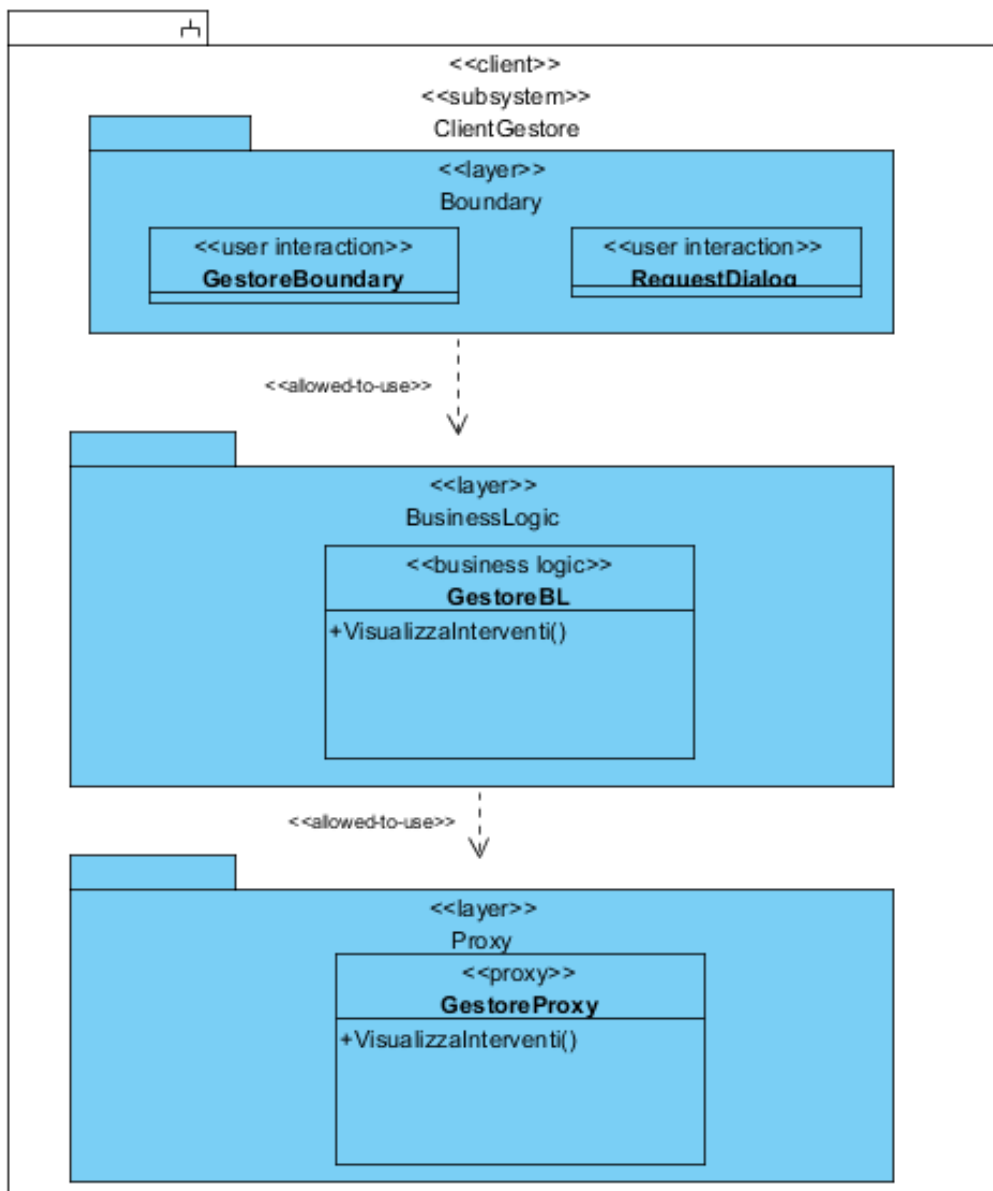
La modellazione del client gestore è definita come subsystem, dato che esso è una parte del sistema. Tale subsystem è modellato mediante una vista layered a tre layers dove:

1. *Boundary Layer*: implementa il boundary tra l'utente e il sistema:
  - *GestoreBoundary* corrisponde all'interfaccia con il quale viene fatta la richiesta.
  - *RequestDialog* corrisponde alla schermata sulla quale vengono visualizzate le statistiche
2. *Business Logic Layer* implementa la logica applicativa del client gestore
3. *Proxy Layer*: implementa l'interfacciamento del client verso il server.

Il primo layer rappresenta il boundary tra l'utente e le funzionalità del sistema.

Gli stili che sono stati usati in questa vista sono **decomposition style** in package/classi, **layered style** che fissa dipendenze di uso tra i diversi layer e moduli del subsystem presentato.

### 3.3.2 Modello



### 3.3.3 Elementi non standard

#### 3.3.3.1 Catalogo delle tipologie di elementi

Nome	Stereotipo testuale	Acronimo	Descrizione
------	---------------------	----------	-------------

ClientGestore	<<Subsystem>>	CG	Terminale del gestore
Boundary layer	<<boundary >>	CGBL	boundary tra l'utente e il sistema
Business logic layer	<<business logic>>	CGBLL	Logica di business del client gestore
Proxy layer	<<proxy>>	CGPL	Interfacciamento verso il lato server del sistema

### 3.3.4 Rationale

La scelta di modellare il client del gestore a livelli è pensata in funzione dell'espandibilità che può essere fatta sul sistema e in funzione della manutenibilità del sistema stesso, in termini di future aggiunte di altri moduli ai layer esistenti.

### 3.3.5 Relazioni con altri modelli/elementi di modello

Link di tracciabilità	Descrizione
Vista Architetturale del Server	Vista del server che comunica con il client gestore attraverso un interfaccia grafica dove richiede l'elenco delle richieste di intervento

## 3.4 Vista architetturale Client Centralina

### 3.4.1 Descrizione

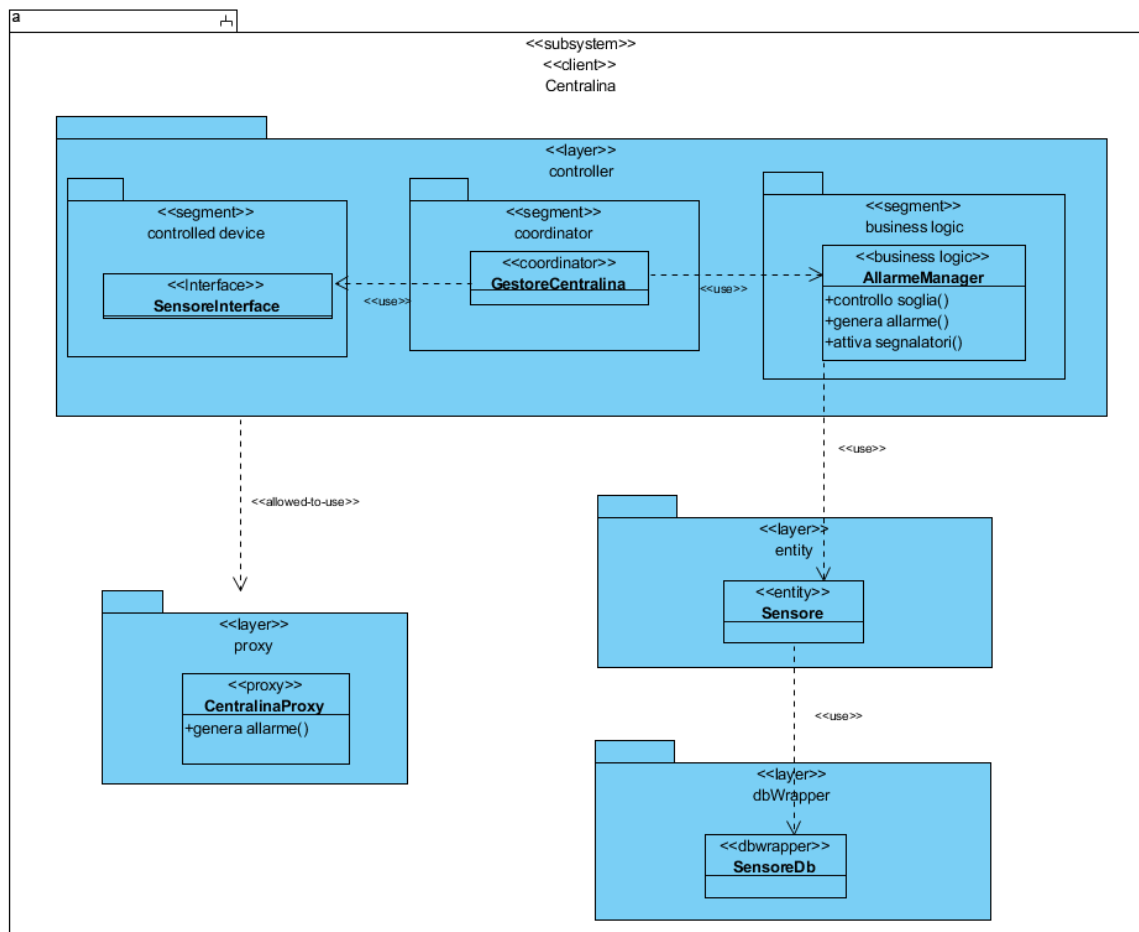
La modellazione del client centralina è definita come subsystem, dato che esso è una parte del sistema. Tale subsystem è modellato mediante una vista a livelli dove:

1. *Controller Layer* diviso in tre segment:
  - *Coordinator* implementa il controllo
  - *Business logic* implementa la logica applicativa della centralina
  - *Controlled device* implementa l'interfaccia del sensore di input
2. *Proxy Layer*: implementa l'interfacciamento del client verso il server.

3. *Entity layer*: implementa l'interfacciamento con la classe entità sensore
4. *DbWrapper layer*: implementa l'interfacciamento verso la parte di persistenza del sistema, ossia il database.

Gli stili che sono stati usati in questa vista sono **decomposition style** in package/classi, **layered style** e **uses style** che fissa dipendenze di uso tra i diversi layer e moduli del subsystem presentato.

### 3.4.2 Modello



### 3.4.3 Elementi non standard

#### 3.4.3.1 Catalogo delle tipologie di elementi

Nome	Stereotipo testuale	Acronimo	Descrizione
Client Centralina	<<Subsystem>>	CC	Centralina presente in

			una abitazione
Controller layer	<<controller>>	CCCL	Livello che contiene interfaccia controllo e business logic
Controlled device segment	<<segment>>	CDS	Interfaccia del dispositivo di input sensore
Coordinator segment	<<segment>>	CS	Controllo della centralina
Business logic segment	<<segment>>	BLS	Logica applicative della centralina
Entity layer	<<entity>>	CCEL	Interfacciamento con classe entità
DbWrapper layer	<<dbWrapper>>	CCDBL	Interfacciamento verso il database

### 3.4.4 Rationale

Considerata la natura particolare di questo client che, oltre a interfacciarsi con dispositivi esterni di input (sensori) e output (Segnalatori acustici), comunica con il server inviando un messaggio d'allarme si è pensato a questo particolare diagramma in cui sono presenti, oltre ad un layer proxy per la comunicazione con il server, anche i layer necessari per l'accesso al database.

### 3.4.5 Relazioni con altri modelli/elementi di modello

Vista Architettuale del Server	Vista del server che comunica con il client centralina attraverso un protocollo di comunicazione basato su JavaRMI
--------------------------------	--------------------------------------------------------------------------------------------------------------------

## 3.5 Vista architetturale Server COR

### 3.5.1 Descrizione

La modellazione del sottosistema server è definita mediante una vista a layer. In particolare si sono considerati quattro differenti layers:

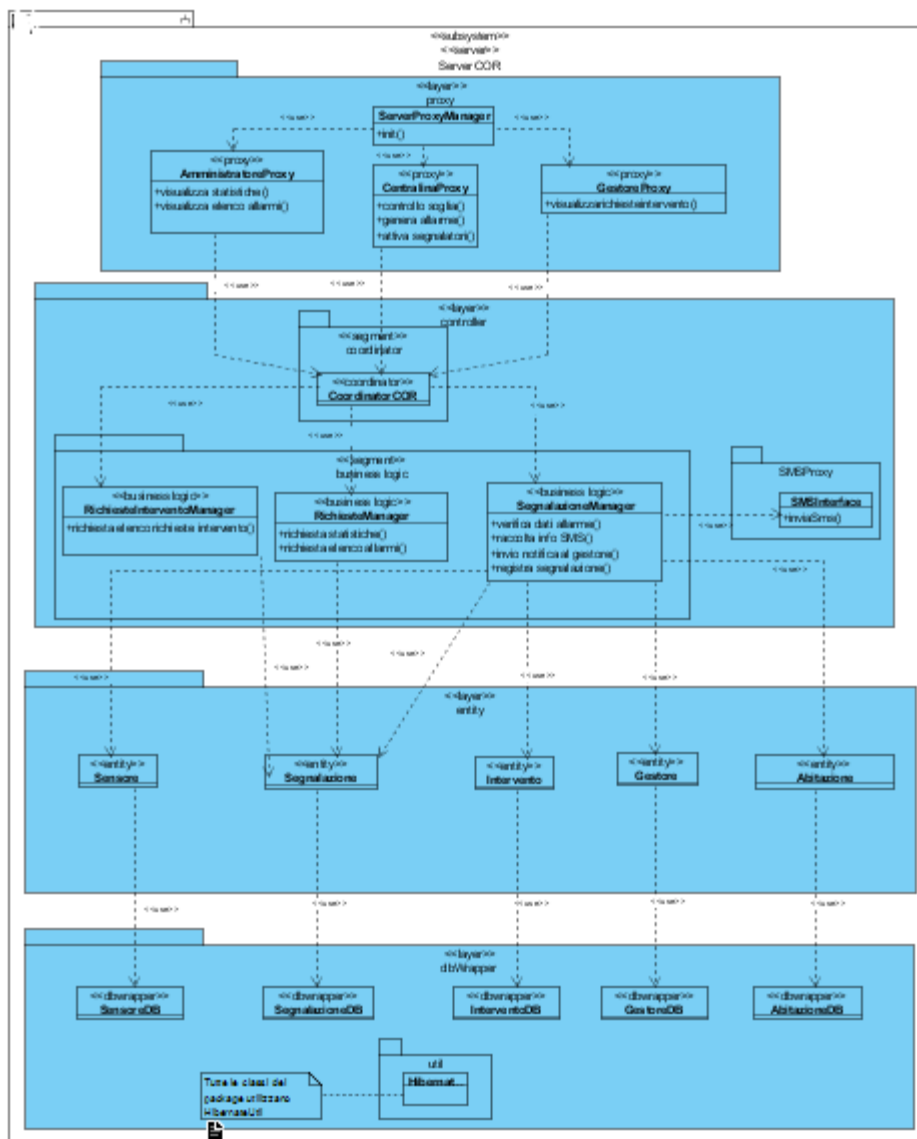
1. *Proxy\_Layer*: implementa l'interfacciamento verso i diversi client tramite Java RMI

2. *Controller Layer*: diviso in due segment:
  - *Coordinator*: implementa il controllo
  - *Business logic*: implementa la logica di business del server;
3. *Entity layer*: implementa l'interfacciamento con le classi entità
4. *DbWrapper layer*: implementa l'interfacciamento verso la parte di persistenza del sistema, ossia il database.

Gli stili che sono stati usati in questa vista sono **decomposition style** in package/classi, **layered** e **uses style** che fissa dipendenze di uso tra i diversi layer e moduli del subsystem.

### 3.5.2 Modello





### 3.5.3 Elementi non standard

#### 3.5.3.1 Catalogo delle tipologie di elementi

Nome	Stereotipo testuale	Acronimo	Descrizione
Server Centrale Operativa Remota	<<Subsystem>>	COR	Sistema Server
Boundary layer	<<boundary>>	CORBL	Boundary tra server e client
Controller layer	<<controller>>	CORCL	Controllo e logica applicativa del server

Entity layer	<<entity>>	COREL	Interfacciamento con classi entità
DbWrapper layer	<<dbWrapper>>	CORDB	Interfacciamento verso il database

### 3.5.4 Rationale

La scelta di definire una struttura a layer per il server è dettata dalla necessità di raggruppare moduli del sistema server in layer di comune responsabilità e per supportare il requisito di modificabilità. Ogni classe contenuta all'interno dei diversi package sono relazionati con altre classi con relazioni di uso, in quanto la correttezza di una funzionalità offerta è dipendente dalle classi del livello inferiore.

### 3.5.5 Relazioni con altri modelli/elementi di modello

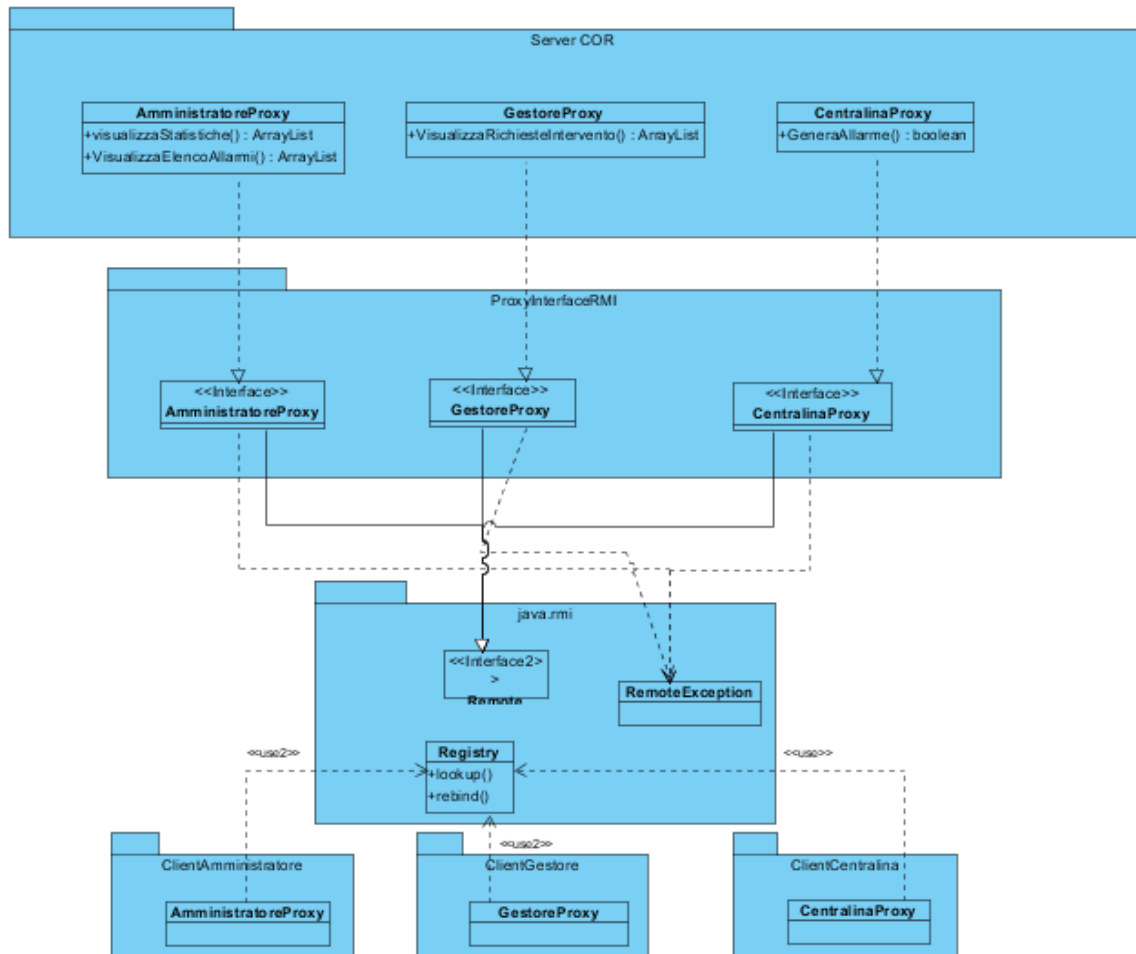
Link di tracciabilità	Descrizione
Vista Architettuale della centralina	Il server gestisce messaggi di allarme inviati dalla centralina e in base alla verifica dei dati decide se creare o meno una segnalazione ed un intervento
Vista Architettuale del client amministratore	Il server elabora richieste da parte dell'amministratore e mostra statistiche
Vista Architettuale del client Gestore	Il server elabora richieste da parte del gestore e mostra elenco di richieste di intervento

## 3.6 Vista architettuale della connessione RMI

### 3.6.1 Descrizione

Di seguito è mostrato il modello dell'intero sistema. Gli stili che sono stati usati in questa vista sono **decomposition style** in package/classi, **uses style** e **generalization style**. Il package ProxyRMIInterface identifica le classi che rappresentano l'interfaccia esposta ai client dal server che le fornisce.

### 3.6.2 Modello



### 3.6.3 Rationale

I dettagli dei diversi subsystem che sono contenuti nel modello di design presentato sono discussi nei corrispettivi paragrafi. La scelta di Java RMI come meccanismo di invocazione remota è dovuta alla necessità di deployare il sistema su nodi differenti, e quindi permettere che oggetti invocati su di un nodo (client) possano invocare funzionalità offerte da un altro nodo (server centralizzato).

### 3.6.4 Relazioni con altri modelli/elementi di modello

Link di tracciabilità	Descrizione
Vista Architeturale del Client	Vista del client centralina

Centralina	
Vista Architetturale del Client Amministratore	Vista del client amministratore
Vista Architetturale del Client Gestore	Vista del client gestore
Vista Architetturale del Server	Vista del server che comunica con i client del sistema

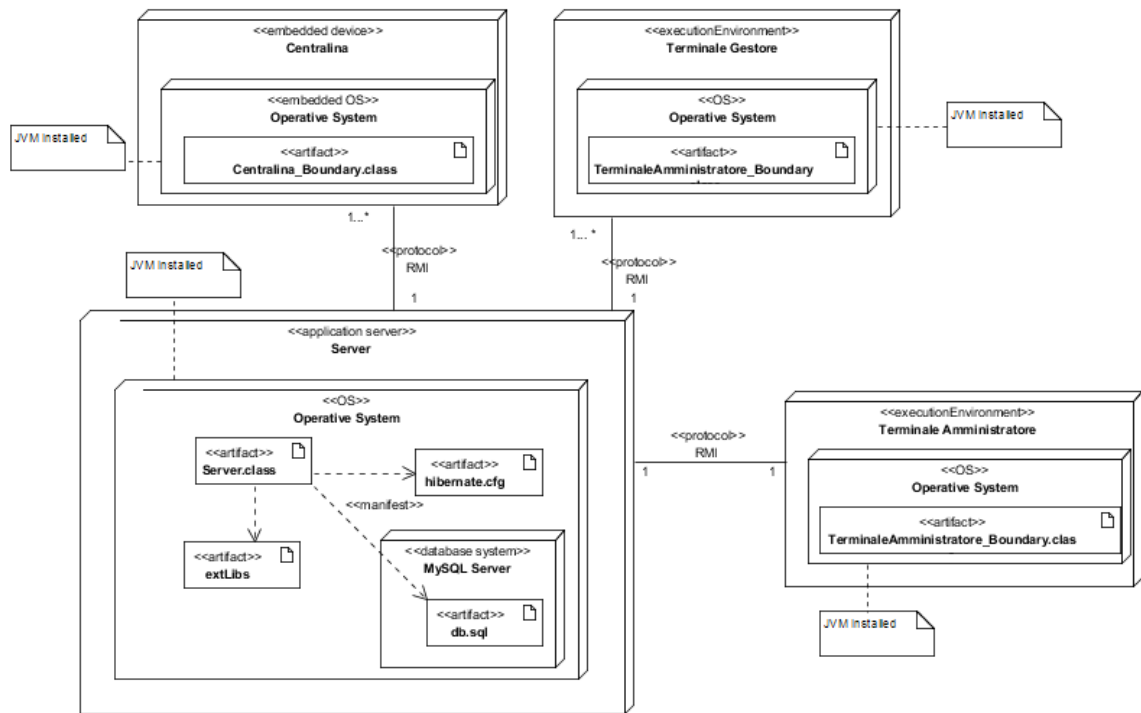
## 3.7 Vista di deploy

### 3.7.1 Descrizione

La parte client centralina del sistema che riguarda l'interazione con il server è stata deployata su nodi stereotipati <<embedded device>>, con una relazione molti a uno verso il server. Le interazioni con amministratore (unica) e gestore invece sono state deployate su nodi di tipo execution environment.

Il server invece è stato considerato come <<application server>>. La comunicazione tra le diverse componenti del sistema è stata realizzata mediante il protocollo che permette l'invocazione remota dei servizi del server, ossia RMI.

## 3.7.2 Modello



## 3.7.3 Elementi non standard

### 3.7.3.1 Catalogo delle tipologie di elementi

Nome	Stereotipo testuale	Acronimo	Descrizione
Operative System	<<embedded OS>>	OS	Sistema Operativo per sistemi che supportano JAVA
Centralina	<<embedded device>>	C	Centralina locale
TerminaleGestore	<< executionenvironment >>	TG	Terminale per il gestore
TerminaleAmministratore	<< executionenvironment >>	TA	Terminale per l'amministratore
RMI	<<protocol>>	RMI	Protocollo di comunicazione tra client e server
MySql Server	<<database system>>	MS	DBMS per mantenere la persistenza dei dati

## 4 Beyond

Come viste di Beyond si è scelto di utilizzare la tecnica del decomposition refinement al fine di descrivere in modo dettagliato le architetture del client centralina e del serveCOR. Inoltre si è scelto di documentare il comportamento di una interazione tra centralina locale e centrale operativa attraverso sequence diagram di dettaglio rispettando la coerenza sia delle module view presentate in precedenza sia del codice di progetto corrispondente.

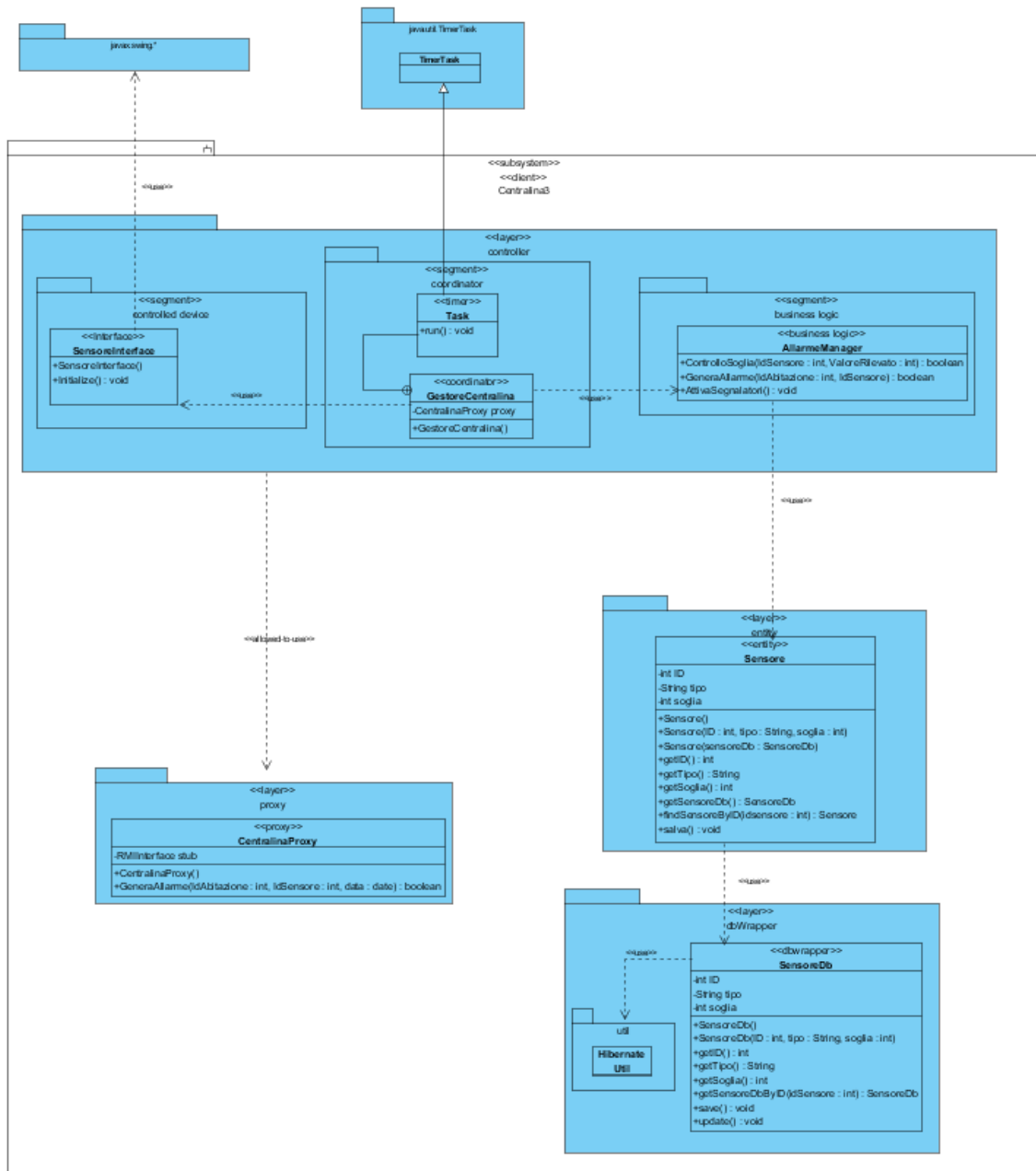
### 4.1 Vista architetturale di dettaglio – Client Centralina

#### 4.1.1 Descrizione

La descrizione dei componenti del subsystem è stata già fatta nel paragrafo 3.4.1.

I 2 segment del controller layer coordinator e business logic sono stati implementati in Java con una sola classe GestoreCentralina.Java inclusa nel package centralina.control mentre il segment controlled device rappresenta l'interfaccia del sensore simulata attraverso una interfaccia grafica.

### 4.1.2 Modello



### 4.1.3 Relazioni con altri modelli/elementi di modello

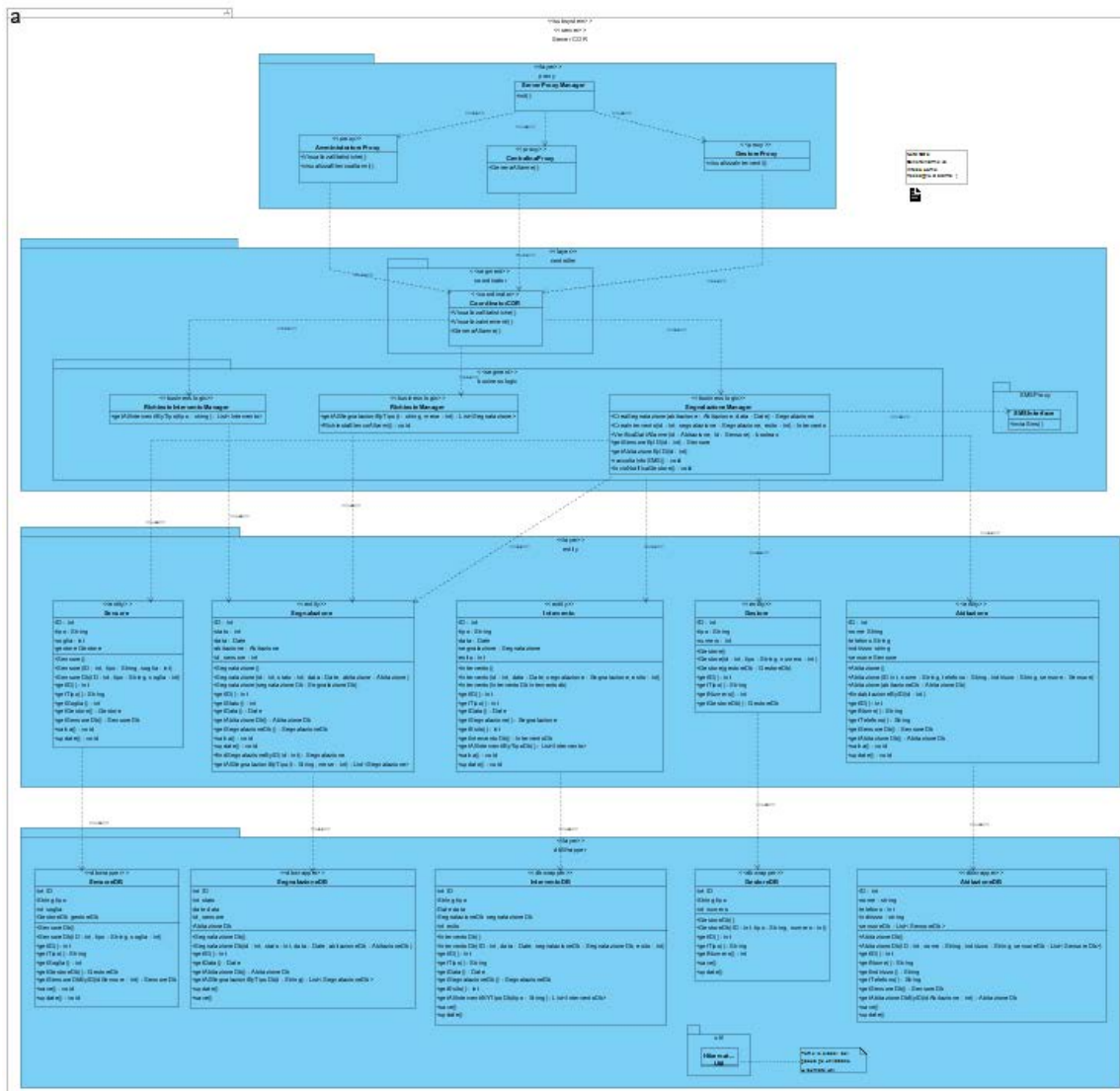
Cfr. paragrafo 3.4.5

## 4.2 Vista architetturale di dettaglio – ServerCOR

### 4.2.1 Descrizione

La descrizione del subsystem è stata già fatta nel paragrafo 3.5.1

### 4.2.2 Modello



### 4.2.3 Relazioni con altri modelli/elementi di modello

Cfr. paragrafo 3.5.5

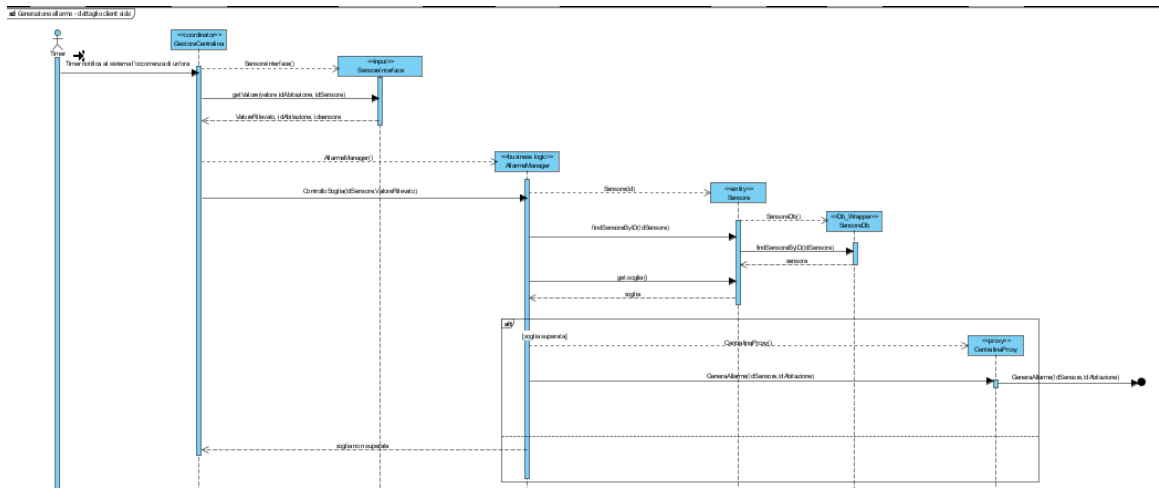


### 4.3 Activity diagram di dettaglio – Generazione Allarme lato client

### 4.3.1 Descrizione

Un timer periodicamente rileva il valore di un sensore da una interfaccia, la business logic della centralina si occupa di confrontare il valore rilevato con la soglia del relativo sensore e nel caso in cui la soglia venga superata si invia un allarme alla centrale operativa e si attivano i segnalatori acustici. Il diagramma seguente è stato inserito come subdiagram del caso d'uso del paragrafo 2.1.1

### 4.3.2 Modello



#### 4.4 Activity diagram di dettaglio - Gestione e Segnalazione Allarme - lato server

#### 4.4.1 Descrizione

Il server alla ricezione di un messaggio di allarme, verifica di poterlo gestire controllando i dati relativi quindi apre e registra nel database una segnalazione, apre e registra nel database un intervento avvisando il relativo gestore e invia un SMS per avvisare il proprietario dell'abitazione. Il seguente diagramma è stato inserito come subdiagram del relativo caso d'uso del paragrafo 2.1.2

### 4.4.2 Modello



## 5 Testing

### 5.1 Generazione e Gestione Segnalazione

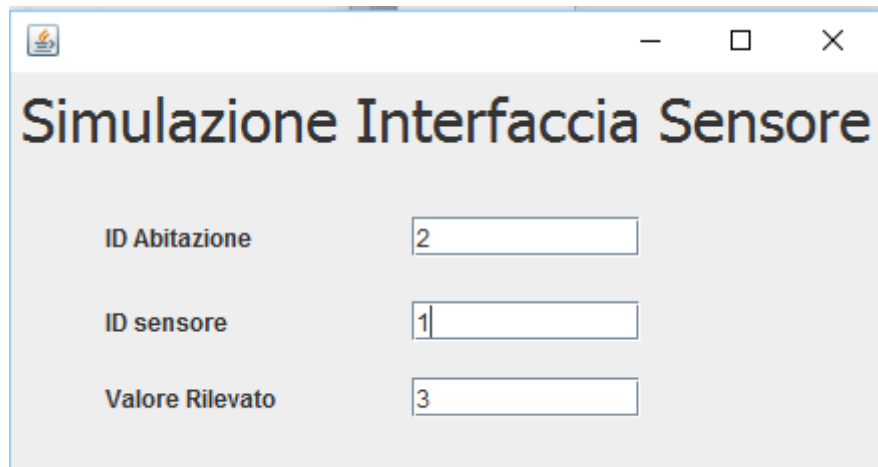
Per testare il funzionamento del codice si è provveduto ad inserire 3 entry nella tabella della centralina sensoredb (lato client)

ID	tipo	soglia
1	apertura porta	5
2	fumo	5
3	acqua	5

Allo stesso modo si è riempita la tabella abitazionedb lato server

ID	nome	telefono	indirizzo	id_sensore
2	Riccio	1234	via strada	1
1	Paudice	4321	via stradone	2

Dopo aver attivato il server si apre l'interfaccia che simula il sensore e dopo aver inserito un abitazione e un sensore si aspetta che il timer generi un numero random per simulare il valore istantaneo del sensore



Simulazione Interfaccia Sensore

ID Abitazione: 2

ID sensore: 1

Valore Rilevato: 3

se la soglia non viene superata si ottiene nella console lato client il messaggio relativo

```
Problems @ Javadoc Declaration Console
SensoreInterface [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (15 lug 2016, 10:22:23)
lug 15, 2016 10:22:48 AM org.hibernate.tool.hbm2ddl.TableMetadata <init>
INFO: HHH000126: Indexes: []
lug 15, 2016 10:22:48 AM org.hibernate.tool.hbm2ddl.SchemaUpdate execute
INFO: HHH000232: Schema update complete
Soglia bassa...nessun problema
Non Invio Allarme Alla Centrale
```

Se la soglia viene superata si ottiene nella console lato client il messaggio di invio alla centrale

Simulazione Interfaccia Sensore

ID Abitazione	<input type="text" value="1"/>
ID sensore	<input type="text" value="2"/>
Valore Rilevato	<input type="text" value="7"/>

```
Problems @ Javadoc Declaration Console
<terminated> SensoreInterface (1) [Java Application] C:\Program Files\Java\jre1.8.0_91\bin\javaw.exe (16 lug 2016, 16:24:38)
Invio Segnalazione alla Centrale Operativa
Invio Allarme Alla Centrale Riuscito
```

Mentre dal lato server viene mostrato un messaggio della creazione della segnalazione e dell'intervento

Problems @ Javadoc Declaration Console

New\_configuration (1) [Java Application] C:\Program Files\Java\jre1.8.0\_91\bin\javaw.exe (16 lug 2016, 17:24:04)

```

richiesta di Segnalazione... identificazione centralina
richiesta da parte della centralina: Paudice
Identificazione Sensore...
Allarme sensore di tipo: fumo
  VERIFICA DATI.....
Creiamo Segnalazione...
SEGNALAZIONE CREATA:
ID: 56
Nome Abitazione: Paudice
Data Segnalazione: Sat Jul 16 17:24:21 CEST 2016
Registro intervento...
Intervento Registrato
  
```

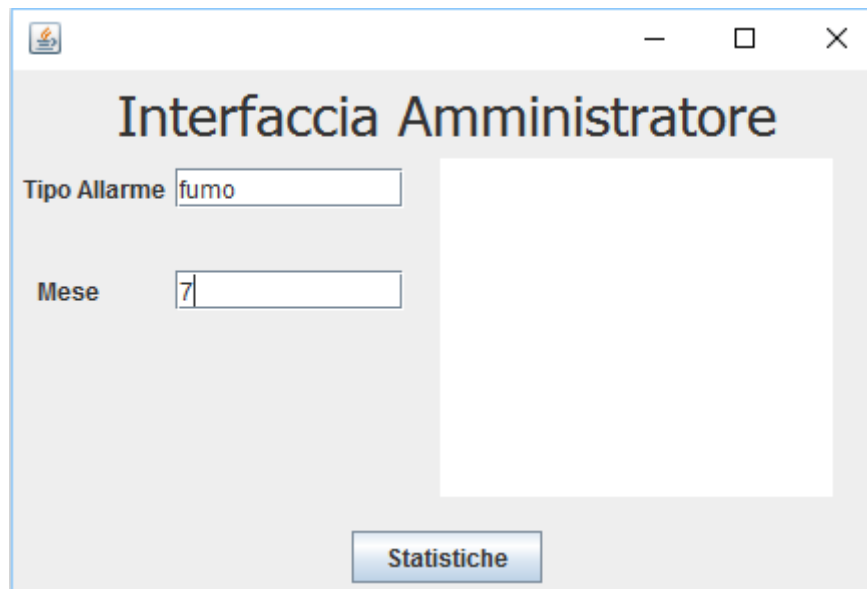
Contestualmente nel database vengono inserite le entry rispettivamente in segnalazione e interventodb

ID	stato	data ▾ 1	id_abitazione	id_sensore
8	0	2016-07-16 17:25:00	2	1
56	0	2016-07-16 17:24:21	1	2
99	0	2016-07-16 17:20:33	1	2
99	0	2016-07-16 17:20:33	1	2
81	0	2016-07-16 17:11:39	1	2
79	0	2016-07-16 17:09:24	1	2

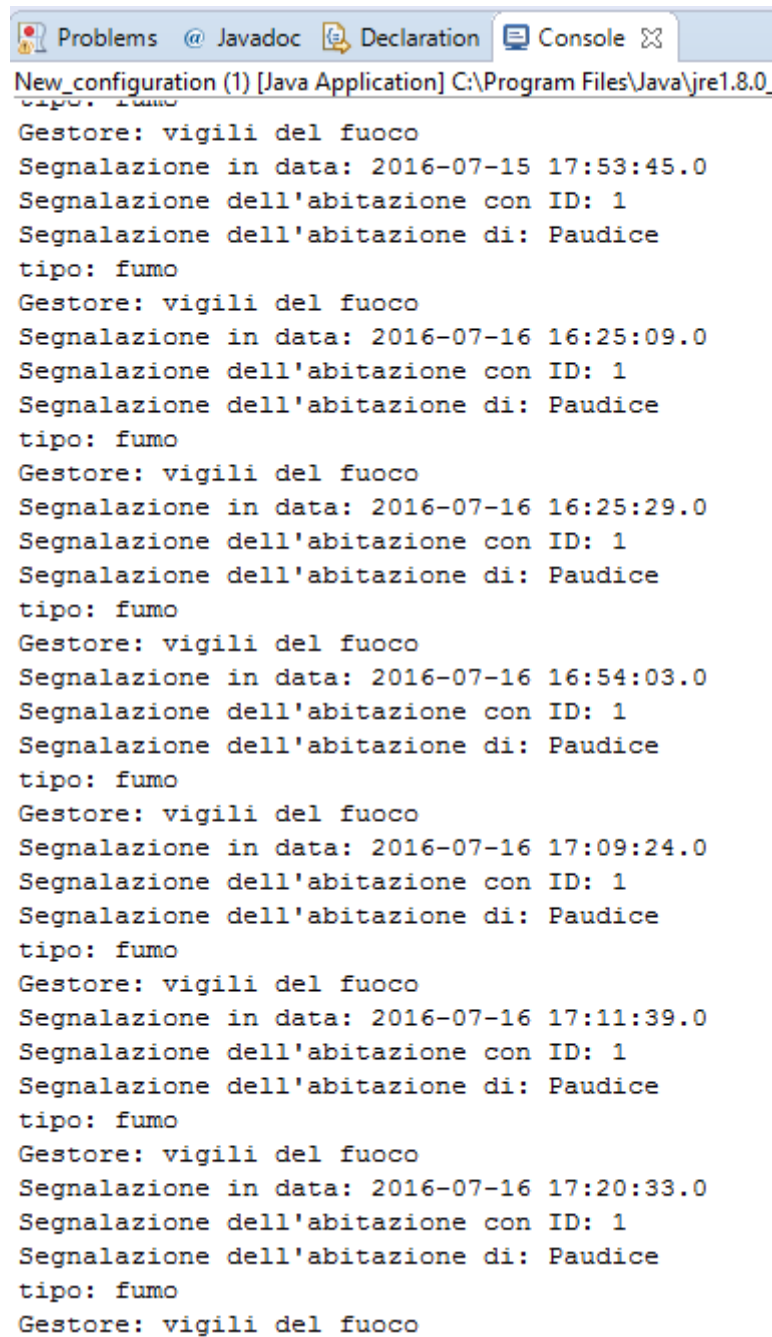
ID	tipo	data ▾ 1	esito	id_segnalazione
52	apertura porta	2016-07-16 17:25:00	0	8
144	fumo	2016-07-16 17:24:23	0	56
175	fumo	2016-07-16 17:20:35	0	99
116	fumo	2016-07-16 17:11:41	0	81
169	fumo	2016-07-16 17:09:25	0	79
74	apertura porta	2016-07-16 16:59:17	0	20

## 5.2 Statistiche Amministratore

Attraverso l'interfaccia messa a disposizione dalla classe di boundary del client amministratore inseriamo tipo e mese per il quale si desidera l'elenco e clicchiamo statistiche



Come risultato in console lato swerver si avrà ciò che è stato richiesto



The screenshot shows a Java IDE window with a console tab active. The console displays a series of log messages for a fire alarm system. Each entry includes the manager (Gestore), the alarm date and time (Segnalazione in data), the apartment ID (Segnalazione dell'abitazione con ID), the apartment address (Segnalazione dell'abitazione di), and the alarm type (tipo).

```
Problems @ Javadoc Declaration Console
New_configuration (1) [Java Application] C:\Program Files\Java\jre1.8.0_
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-15 17:53:45.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-16 16:25:09.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-16 16:25:29.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-16 16:54:03.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-16 17:09:24.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-16 17:11:39.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
Segnalazione in data: 2016-07-16 17:20:33.0
Segnalazione dell'abitazione con ID: 1
Segnalazione dell'abitazione di: Paudice
tipo: fumo
Gestore: vigili del fuoco
```

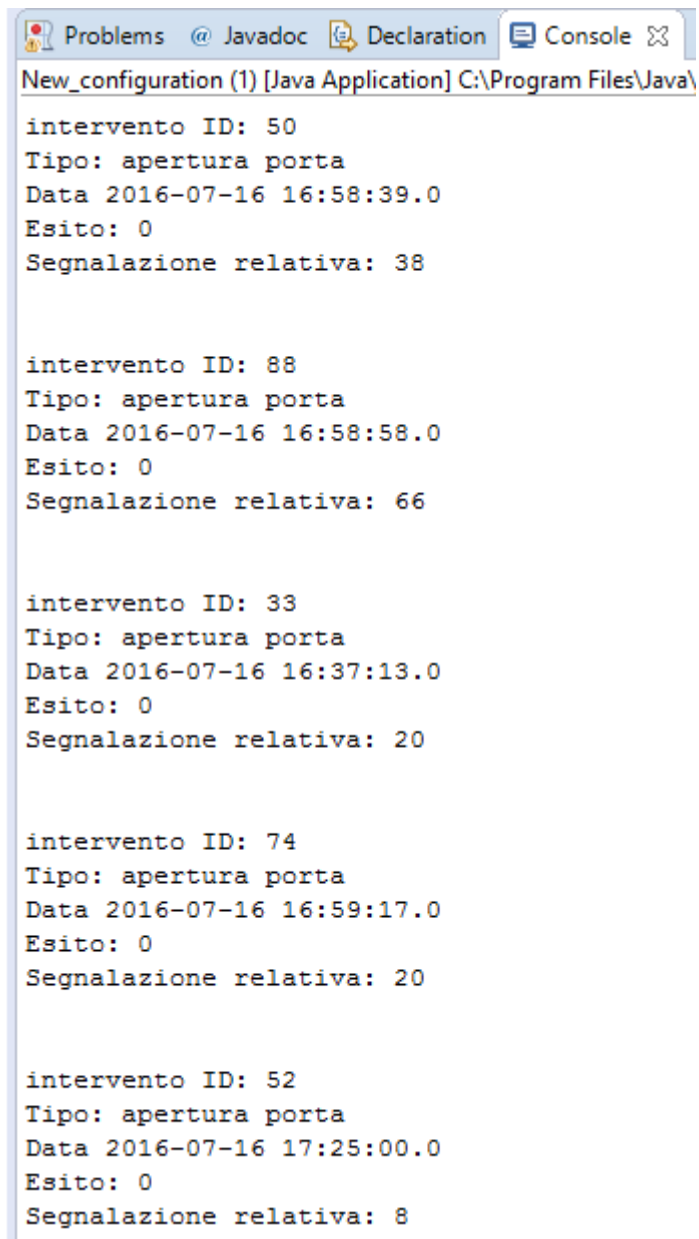
### 5.3 Elenco Richieste Intervento Gestore

Attraverso l'interfaccia gestore inserisco il tipo allarme



E il server mi restituisce l'elenco degli interventi del tipo scelto





The screenshot shows an IDE window with a tab labeled "Console". The title bar of the console window reads "New\_configuration (1) [Java Application] C:\Program Files\Java\". The console contains five blocks of log output, each representing an intervention. Each block includes the intervention ID, type, date and time, result, and a relative signal.

```
Problems @ Javadoc Declaration Console
New_configuration (1) [Java Application] C:\Program Files\Java\

intervento ID: 50
Tipo: apertura porta
Data 2016-07-16 16:58:39.0
Esito: 0
Segnalazione relativa: 38

intervento ID: 88
Tipo: apertura porta
Data 2016-07-16 16:58:58.0
Esito: 0
Segnalazione relativa: 66

intervento ID: 33
Tipo: apertura porta
Data 2016-07-16 16:37:13.0
Esito: 0
Segnalazione relativa: 20

intervento ID: 74
Tipo: apertura porta
Data 2016-07-16 16:59:17.0
Esito: 0
Segnalazione relativa: 20

intervento ID: 52
Tipo: apertura porta
Data 2016-07-16 17:25:00.0
Esito: 0
Segnalazione relativa: 8
```