

# **Sistema per il Monitoraggio e la Teleriabilitazione dei Pazienti**

**Progetto di Architettura del Software  
Università degli Studi di Milano-Bicocca**

**Matteo Breganni 869549  
Samuele Toniazza 918624**



# Assunzioni

- Ad ogni paziente viene consegnato un dispositivo wearable che viene indossato al polso. Il dispositivo wearable è composto dall'insieme dei sensori, una porta a cui è possibile collegare sensori plug-in, e uno schermo che permette all'utente di ricevere notifiche dal sistema per situazioni particolari (situazione di emergenza o di malfunzionamento rilevata, reminders per medicine e attività...).
- Oltre ai sensori integrati nel dispositivo wearable, sono presenti sensori “plug-in” opzionali, che vengono forniti al paziente nel caso in cui il suo piano terapeutico lo richieda. Questi dispositivi plug-in vengono collegati al dispositivo wearable temporaneamente per fare la rilevazione tramite un cavo.
- Per ogni sensore plug-in definiamo una frequenza massima di utilizzo prevista. Il piano terapeutico può quindi decidere la frequenza di utilizzo in base alla situazione medica del paziente, ma non può andare oltre questa frequenza massima.

# Assunzioni

- Il dispositivo wearable ha un livello di potenza computazionale sufficientemente alto, di modo da poter eseguire computazioni locali per esempio per elaborare i dati dei sensori.
- Il dispositivo wearable possiede una connettività wireless di che lo mantiene costantemente collegato alla rete (internet e telefonica) in modo autonomo, senza necessitare per esempio di un'applicazione sullo smartphone. Il dispositivo può anche essere accoppiato a reti wifi per mantenere la connessione più stabile per esempio nella residenza dell'utente.
- Il dispositivo wearable possiede anche un sistema GPS che può essere interrogato per ricavare la posizione del paziente quando necessario (situazioni di emergenza).

# Assunzioni

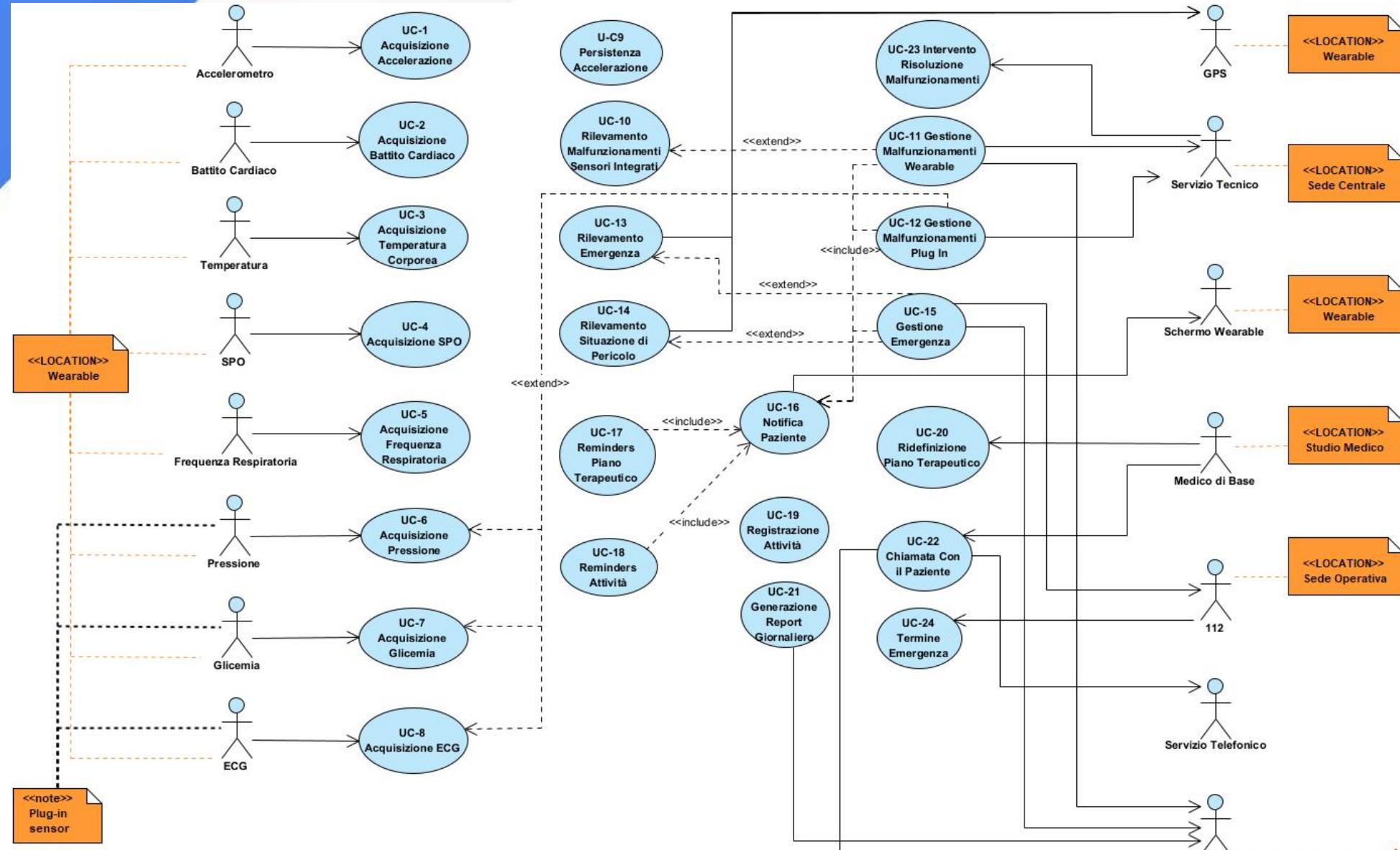
- Assumiamo i ruoli del medico curante e dei medici di turno nel seguente modo: il medico curante è quello che definisce il piano terapeutico e si mette in contatto con il paziente occasionalmente per discutere l'andamento del piano terapeutico e (se necessario) modificarlo. I medici di turno sono invece quelli che si interfacciano con il paziente quando viene rilevata una situazione potenzialmente pericolosa o di emergenza dai sensori del dispositivo, e sono quindi disponibili in qualsiasi momento.
- Assumiamo l'esistenza di un sistema esterno del 112 tramite il quale è possibile, in tempo reale, inviare i dati relativi alla situazione di pericolo o di emergenza rilevata. Il sistema esterno assegnerà la situazione automaticamente ad un medico di turno che analizza i dati ricevuti e prende delle decisioni (falso allarme, diagnosi, invio ambulanza più vicina...). I dati relativi alle decisioni prese dal 112 verranno rimandati al sistema, per la persistenza.

# Assunzioni

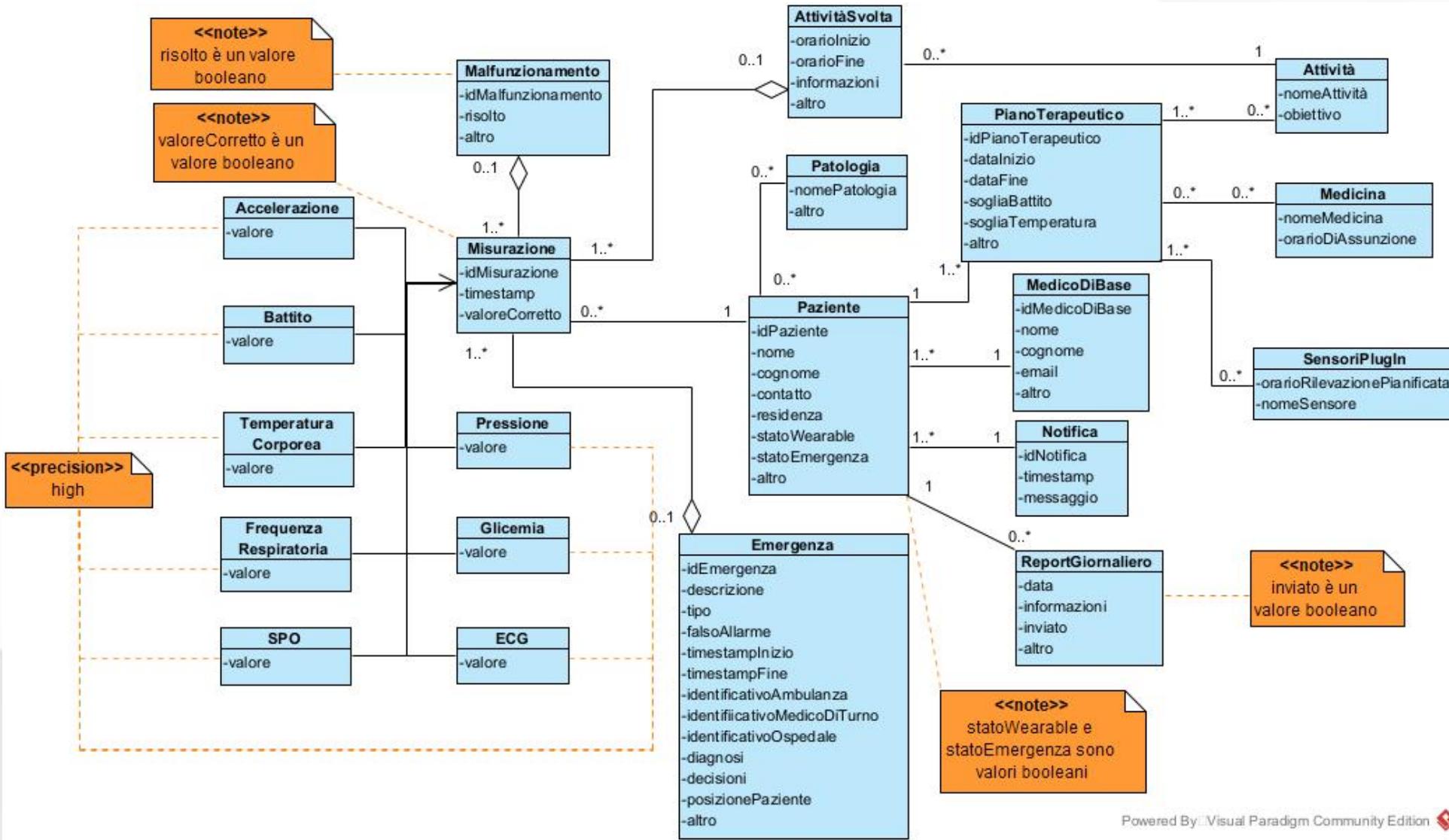
- Assumiamo che il paziente sia in grado di indossare il dispositivo wearable 24 ore su 24, come se avesse una batteria infinita.
- Assumiamo l'esistenza di un servizio tecnico che si occupa di risolvere i malfunzionamenti dei sensori del dispositivo wearable o di quelli plug-in. Alcuni malfunzionamenti possono essere risolti collegandosi in «remoto» tramite sistemi esterni del servizio tecnico.
  - Assumiamo che il servizio telefonico e il servizio email siano servizi esterni sempre disponibili in grado di supportare le comunicazioni.
  - Assumiamo che il sistema gestirà 200 pazienti, e quindi assumiamo che il sistema esterno 112 e quello di servizio tecnico possano supportare questa quantità di pazienti.

# Architettura del Problema

# Diagramma dei Casi d'Uso



# Modello di Dominio



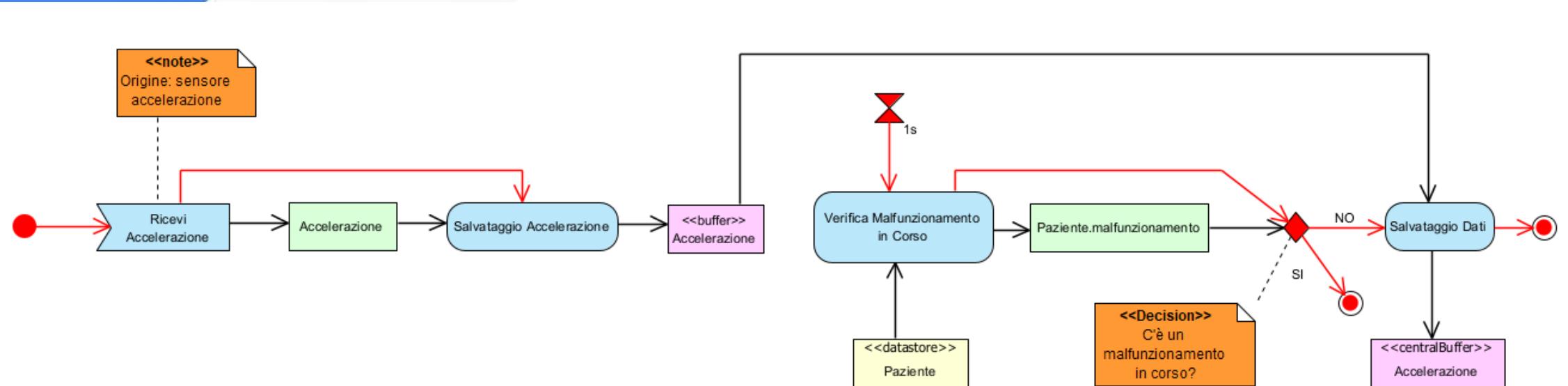
In alcune tabelle è riportato il campo “altro”, questo indica la possibile presenza di altri campi secondari, non importanti per la definizione dell’architettura, ma dipendenti dalle necessità specifiche.

# Note sui Diagrammi delle Attività

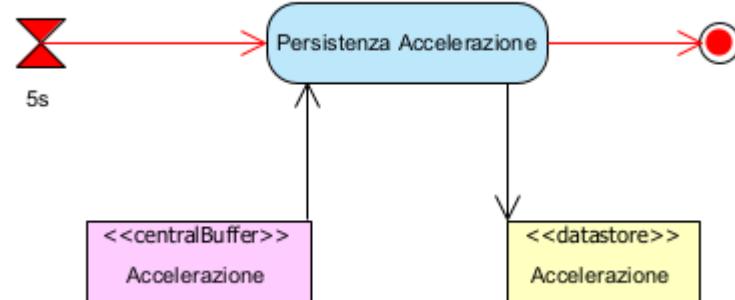
- Le frequenze riportate in ciascun diagramma delle attività si riferiscono al singolo paziente. Il numero totale di pazienti previsto è 200.
- I diagrammi delle attività (AD6-AD7-AD8) relativi ai sensori “plug-in” contengono le frequenze massime (per considerare il caso peggiore), queste possono essere definite diversamente tramite il piano terapeutico.
- Per ciascuna attività è stata stimata la complessità e il tempo massimo di esecuzione.
- I casi d'uso seguenti non sono stati definiti in quanto molto semplici (complexity bassa, impostano semplicemente il flag nel dato corrispondente):
  - UC-23 Intervento risoluzione malfunzionamenti (Frequency: ogni giorno; Delay: 5 secondi)
  - UC-24 Termine emergenza (Frequency: ogni 2 settimane; Delay: 5 secondi)
- I delay includono tutte le attività, interne o esterne.

# AD1 – Acquisizione Accelerazione

Complexity: bassa - Frequency: 100/secondo - Delay: 1s (prima parte), 1s (seconda parte)



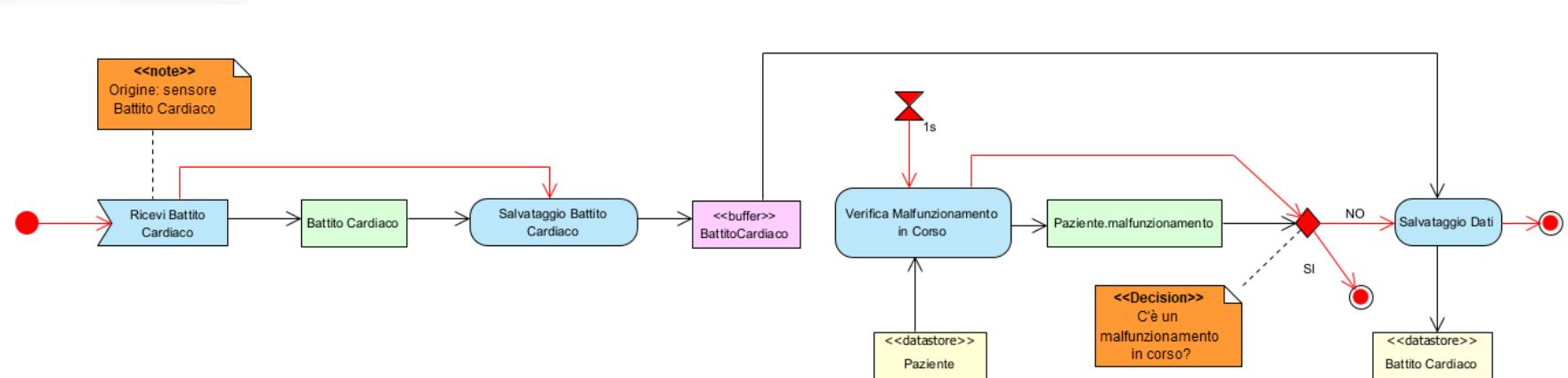
# AD9 - Persistenza Accelerazione



Complexity: bassa  
Frequency: ogni 5 secondi  
Delay: 1s

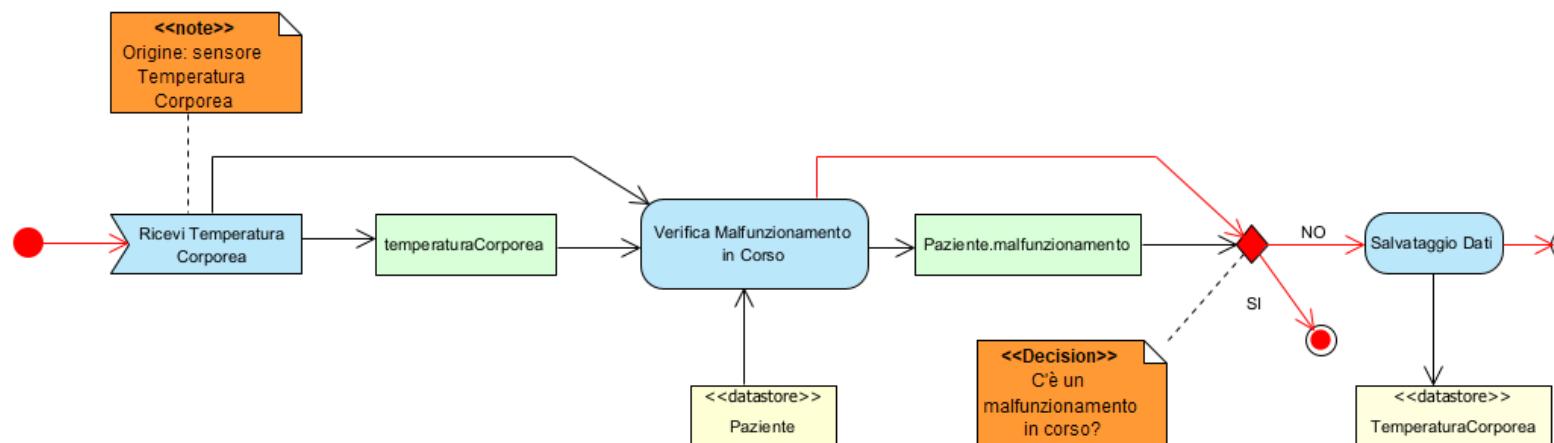
# AD2 - Acquisizione Battito Cardiaco

Complexity: bassa - Frequency: 2/secondo - Delay: 1 secondo (prima parte), 1 secondi (seconda parte)



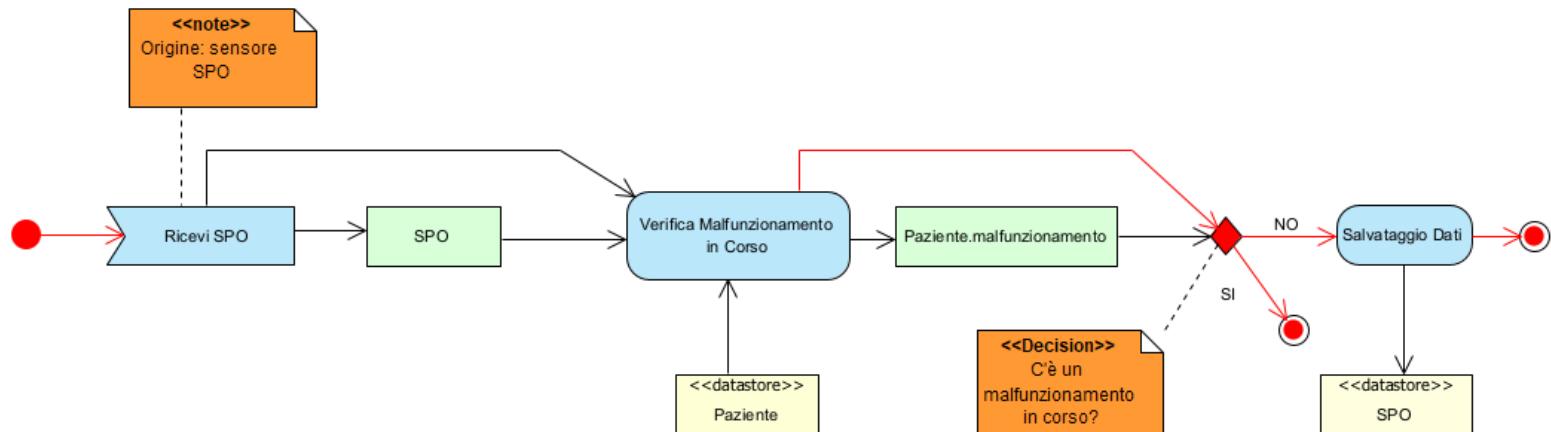
# AD3 - Acquisizione Temperatura Corporea

Complexity: bassa - Frequency: ogni 5 minuti - Delay: 2 secondi



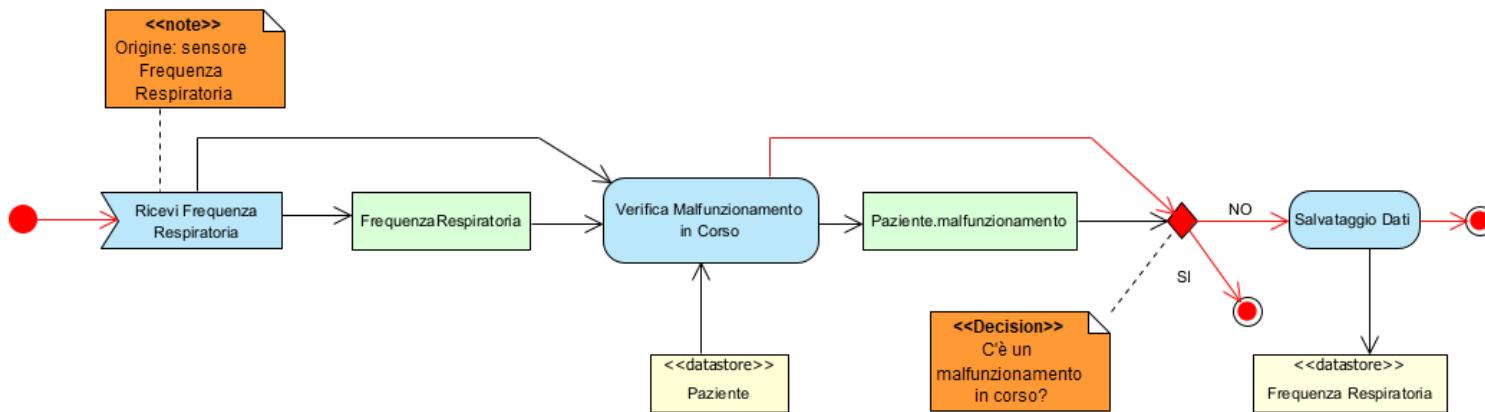
# AD4 - Acquisizione SPO

Complexity: bassa - Frequency: ogni minuto - Delay: 2 secondi



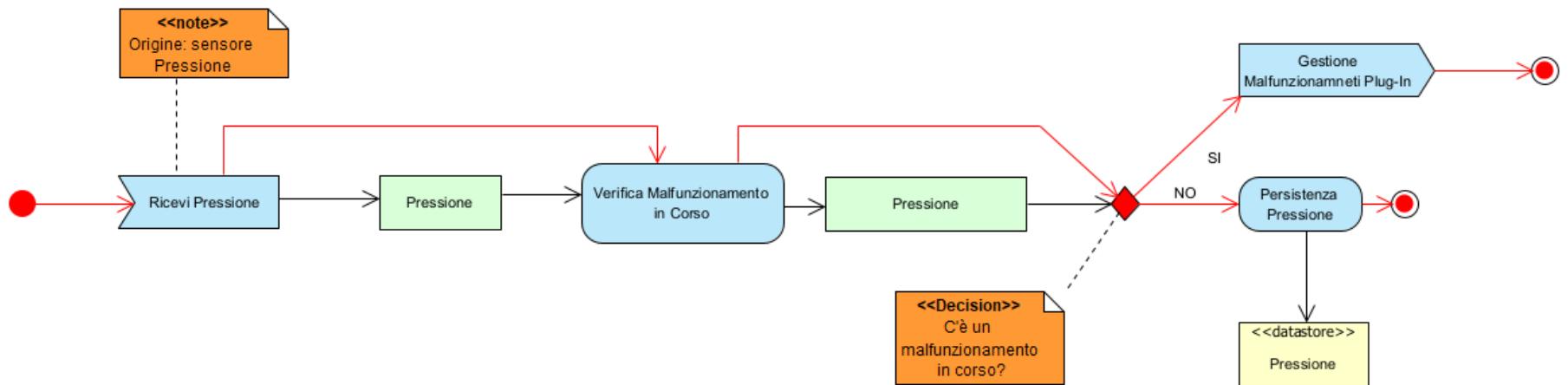
# AD5 - Acquisizione Frequenza Respiratoria

Complexity: bassa - Frequency: ogni minuto - Delay: 2 secondi



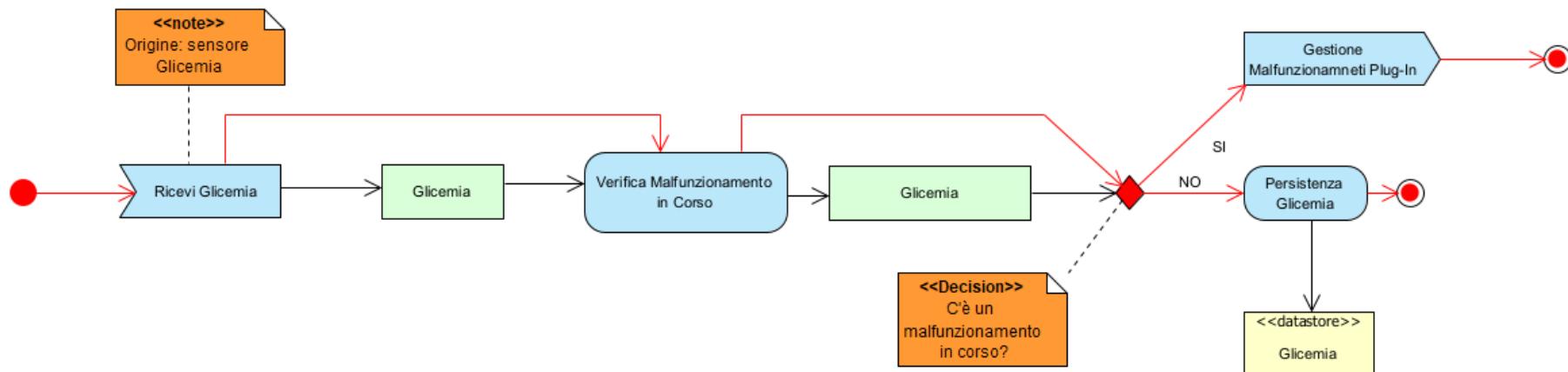
# AD6 - Acquisizione Pressione

Complexity: bassa - Frequency: ogni 15 minuti - Delay: 2 secondi



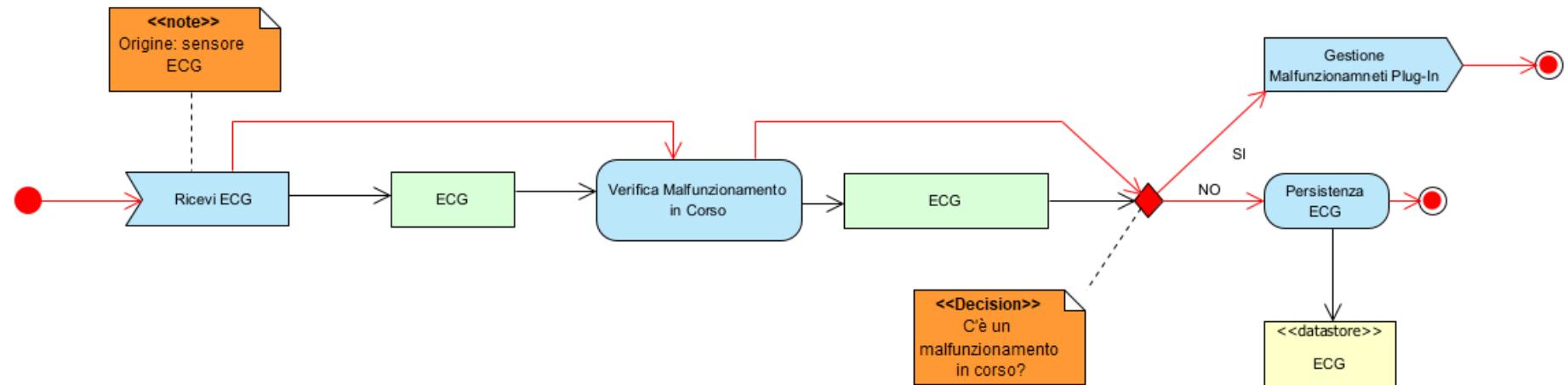
# AD7 - Acquisizione Glicemia

Complexity: bassa - Frequency: ogni 60 minuti - Delay: 2 secondi



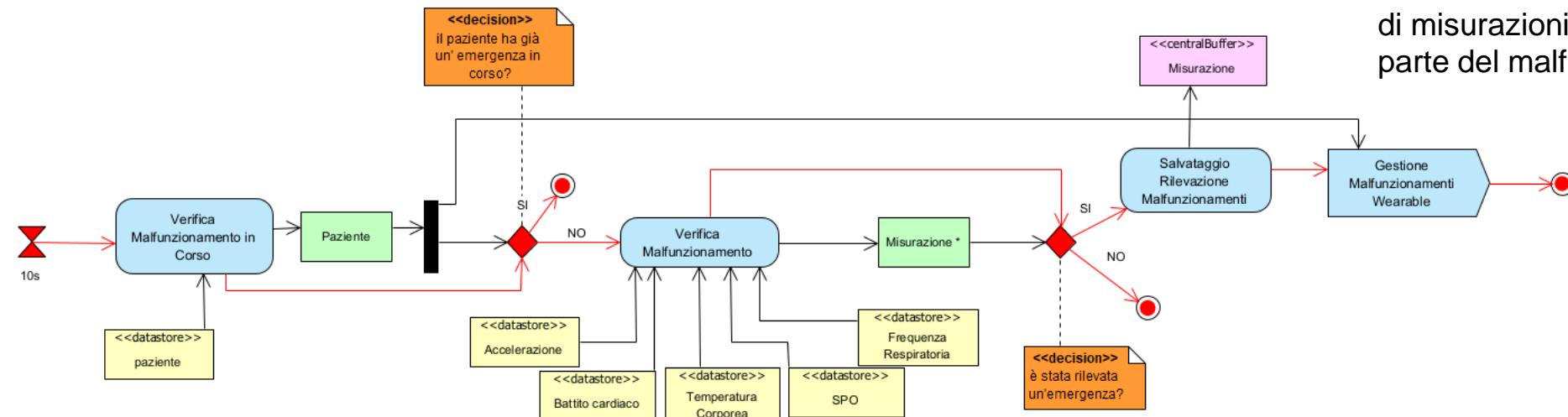
# AD8 - Acquisizione Elettrocardiogramma

Complexity: bassa - Frequency: ogni 30 minuti - Delay: 2 secondi



# AD10 - Rilevamento Malfunzionamento Sensori Integrati

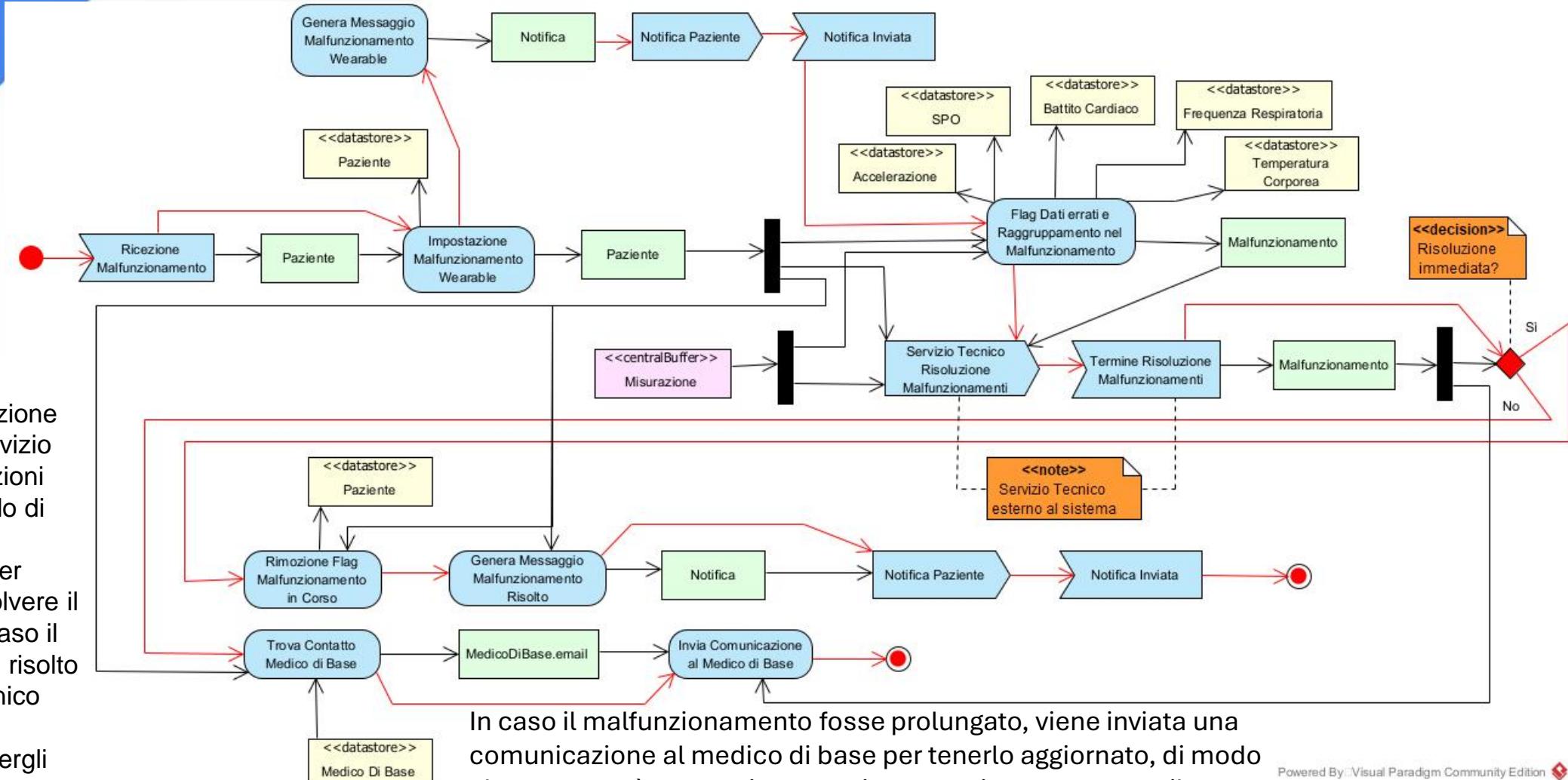
Complexity: alta - Frequency: ogni 10 secondi - Delay: 5 secondi



«Verifica Malfunzionamento» ritorna una collezione di misurazioni, ovvero quelle misurazioni che fanno parte del malfunzionamento.

# AD11 - Gestione Malfunzionamento Wearable

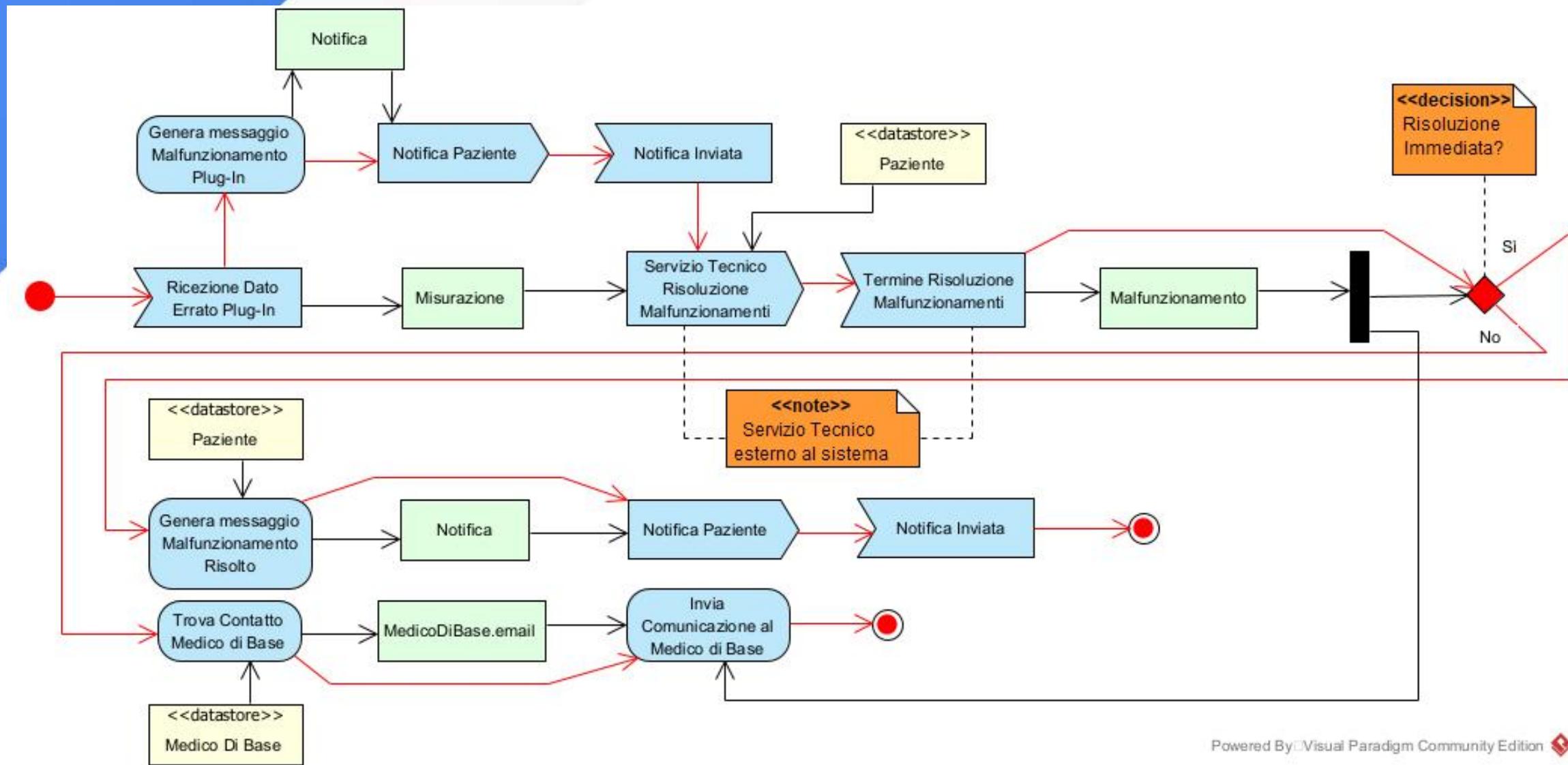
Complexity: media - Frequency: ogni giorno - Delay: 30 minuti



In caso il malfunzionamento fosse prolungato, viene inviata una comunicazione al medico di base per tenerlo aggiornato, di modo che esso potrà mettersi successivamente in contatto con il paziente e per esempio accordarsi su come procedere nel frattempo che il malfunzionamento viene risolto dal servizio tecnico (questo accade anche in AD12).

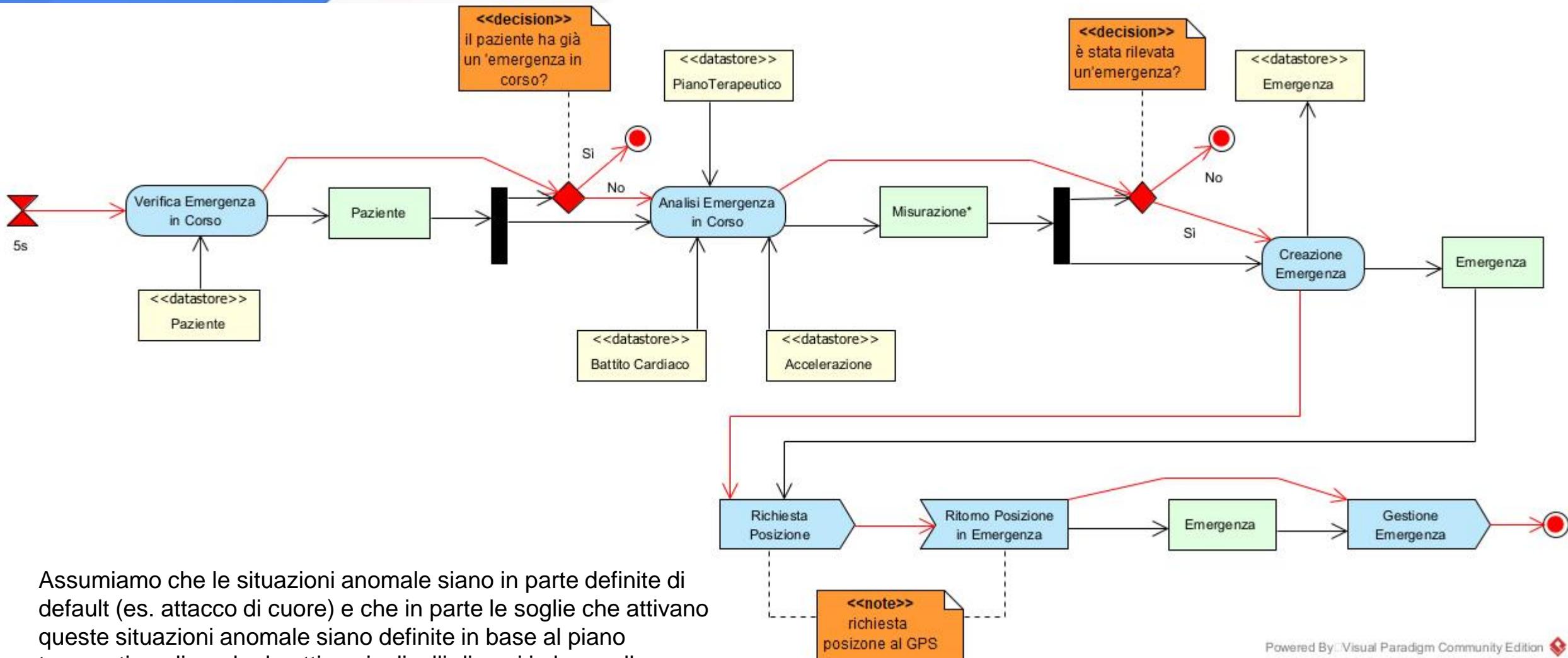
**AD12 - Gestione Malfunzionamento Plug-In**

Complexity: media - Frequency: ogni giorno - Delay: 30 minuti



# AD13 - Rilevamento Emergenza

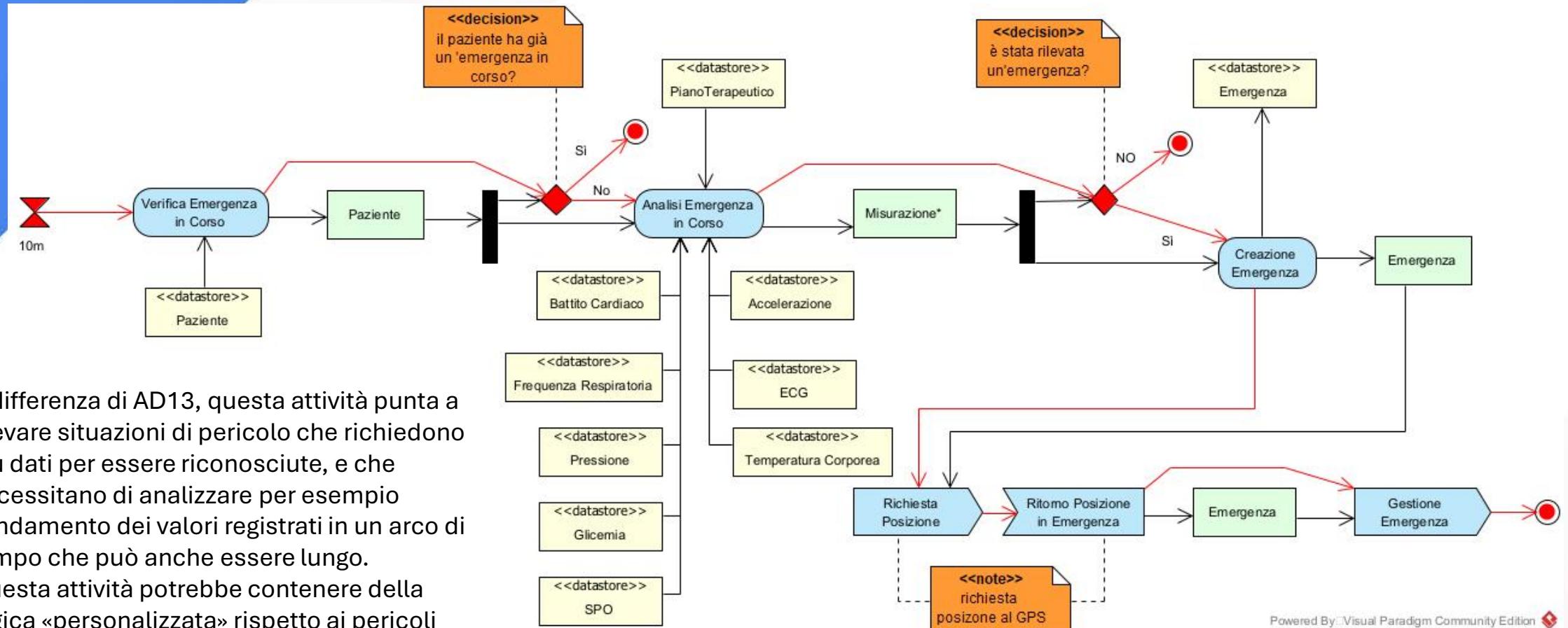
Complexity: alta - Frequency: ogni 5 secondi - Delay: 5 secondi



Assumiamo che le situazioni anomale siano in parte definite di default (es. attacco di cuore) e che in parte le soglie che attivano queste situazioni anomale siano definite in base al piano terapeutico, di modo da attivarsi a livelli diversi in base alla situazione del paziente.

# AD14 - Rilevamento Situazione di Pericolo

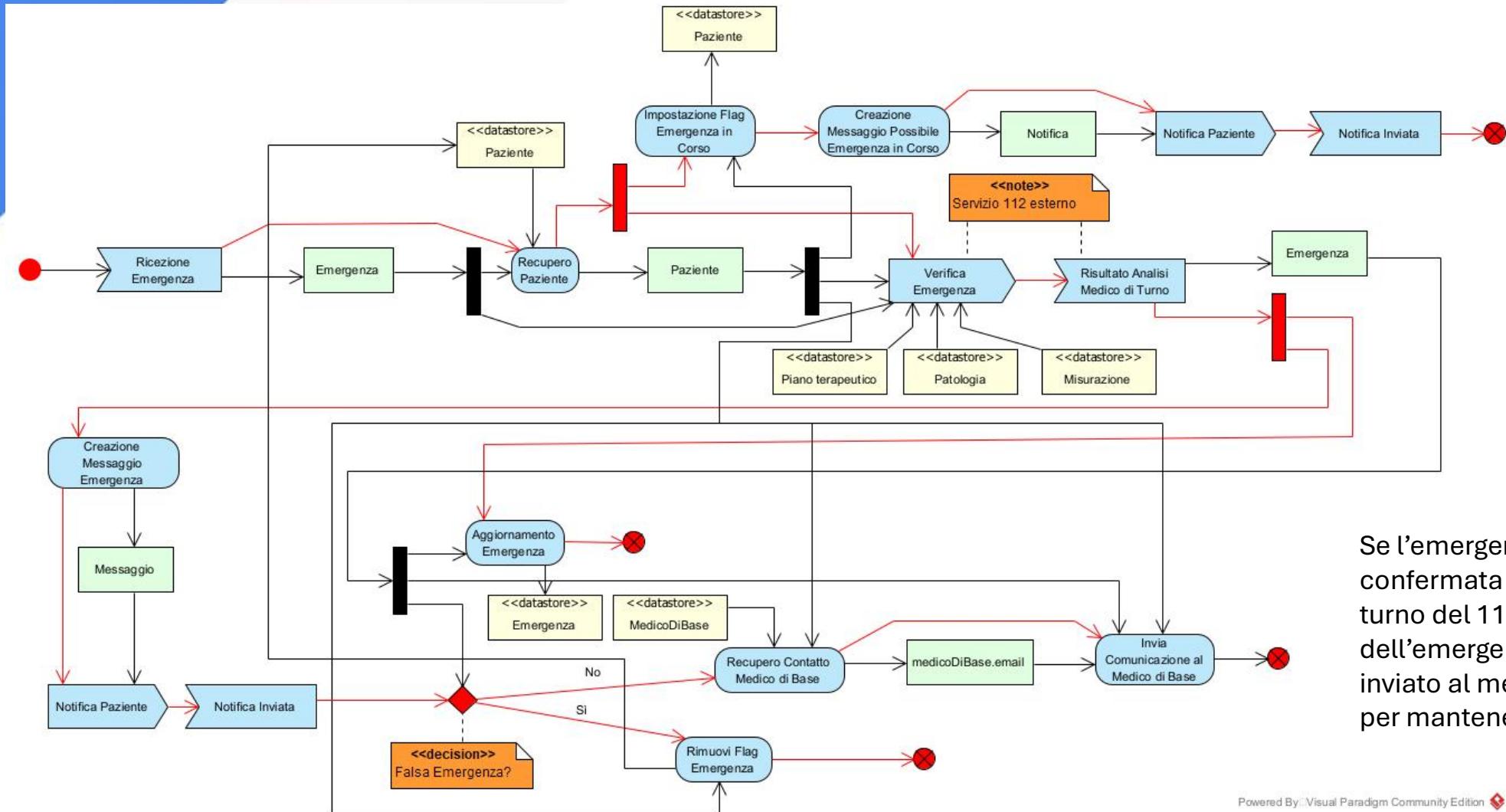
Complexity: alta - Frequency: ogni 10 minuti - Delay: 2 minuti



A differenza di AD13, questa attività punta a rilevare situazioni di pericolo che richiedono più dati per essere riconosciute, e che necessitano di analizzare per esempio l'andamento dei valori registrati in un arco di tempo che può anche essere lungo. Questa attività potrebbe contenere della logica «personalizzata» rispetto ai pericoli specifici di ciascun paziente, per esempio anche tramite le soglie custom del piano terapeutico.

# AD15 - Gestione Emergenza

Complexity: alta - Frequency: ogni 2 settimane - Delay: 2 minuti



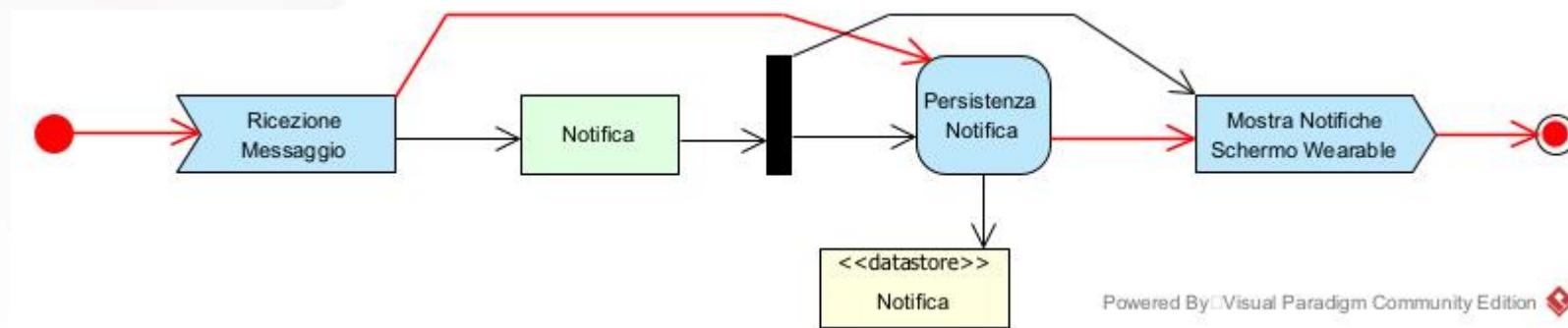
Se l'emergenza è confermata dal medico di turno del 112, il report dell'emergenza viene anche inviato al medico di base, per mantenerlo aggiornato.

Powered By Visual Paradigm Community Edition

L'attività di verifica emergenza è esterna, gestita dal medico di turno assegnato dal sistema esterno del 112. Il medico di turno analizza i dati ricevuti dal sistema e verifica se l'emergenza sia reale oppure se si sia trattato di un falso allarme. Sarà quindi il medico di turno stesso che potrà mettersi in contatto telefonico con il paziente e che tramite il sistema esterno provvederà ad inviare un'ambulanza se ritenuto necessario. L'attività ritorna quindi il dato Emergenza aggiornato.

# AD16 - Notifica Paziente

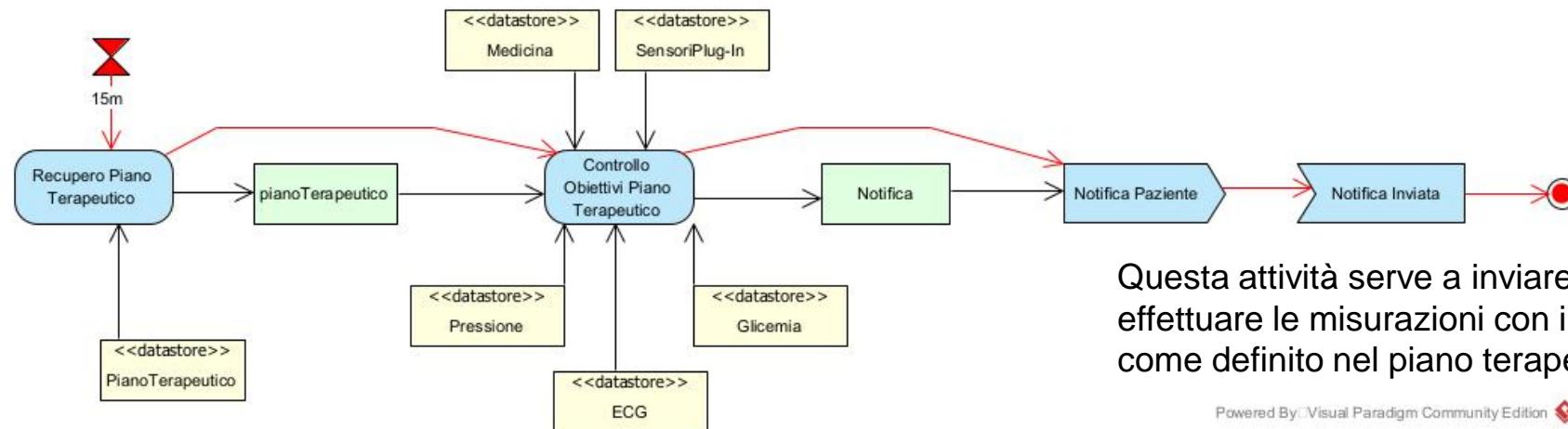
Complexity: bassa - Frequency: ogni 10 minuti - Delay: 5 secondi



Powered By Visual Paradigm Community Edition

# AD17 - Reminders Piano Terapeutico

Complexity: media - Frequency: ogni 15 minuti - Delay: 2 minuti

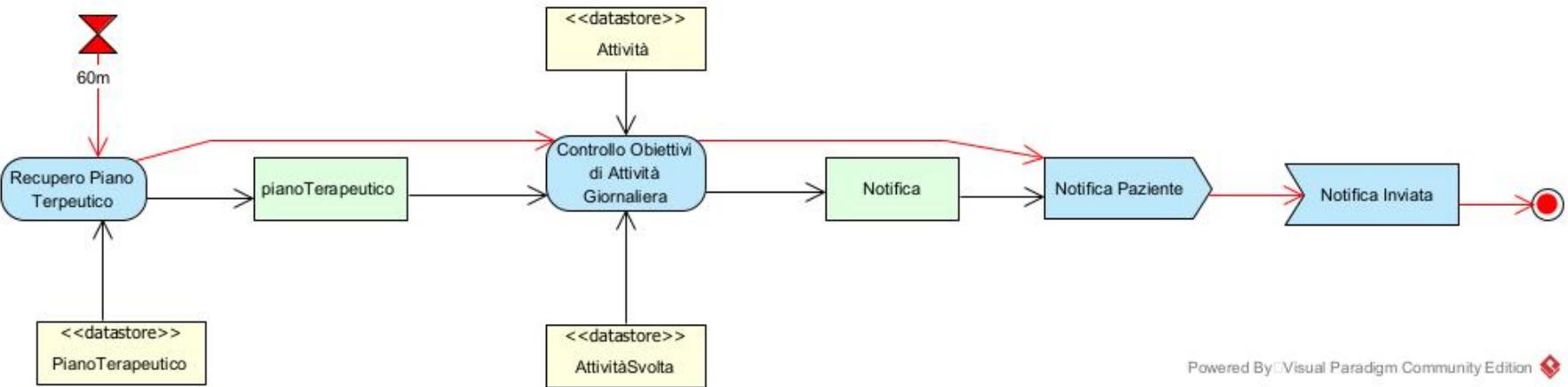


Questa attività serve a inviare promemoria al paziente per ricordargli di effettuare le misurazioni con i sensori plug-in e di assumere le medicine, come definito nel piano terapeutico.

Powered By Visual Paradigm Community Edition

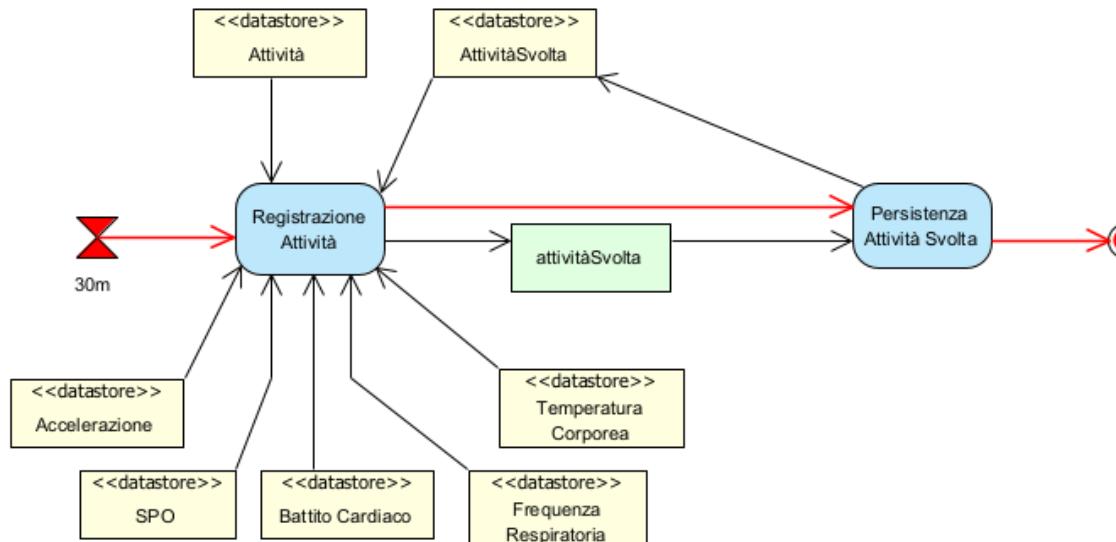
# AD18 - Reminders Attività

Complexity: media - Frequency: ogni 60 minuti - Delay: 2 minuti



Powered By Visual Paradigm Community Edition

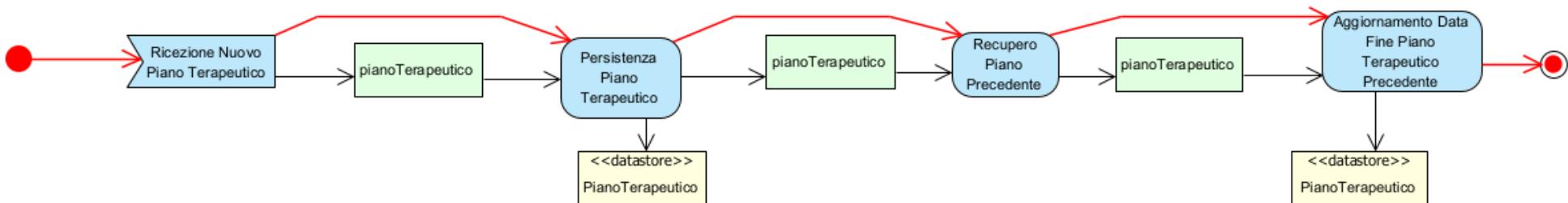
# AD19 - Registrazione Attività



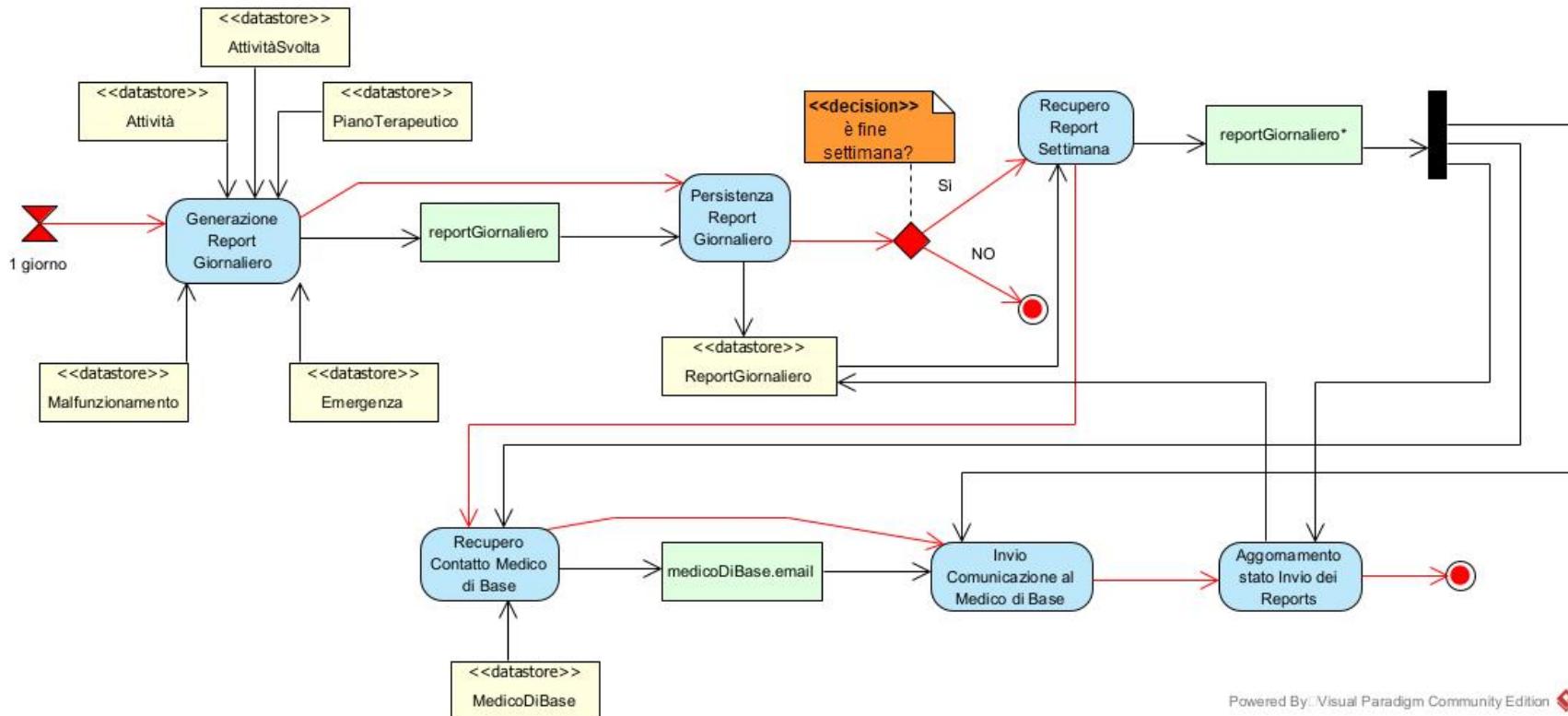
Complexity: alta  
Frequency: ogni 30 minuti  
Delay: 5 minuti

# AD20 - Ridefinizione Piano Terapeutico

Complexity: bassa - Frequency: ogni settimana - Delay: 30 secondi



# AD21 - Generazione Report Giornaliero

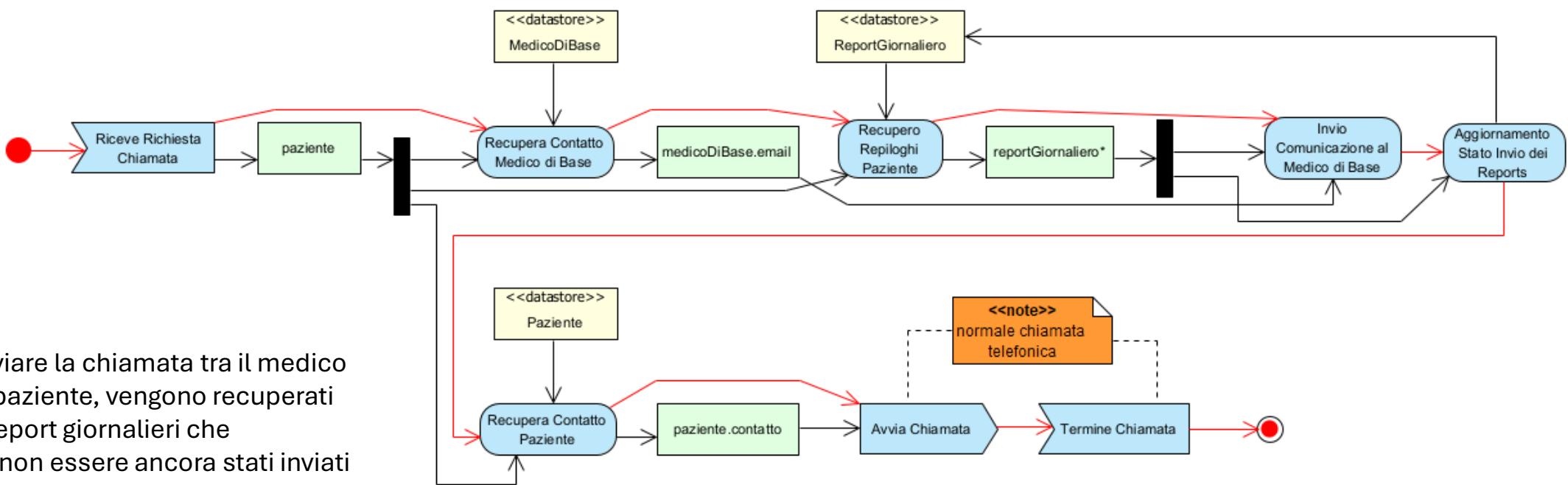


Quando un piano terapeutico viene ridefinito in AD20, viene comunque salvato quello precedente, impostando la data di termine. Questo può essere utile per analisi future, per esempio per confrontare emergenze passate con i dati corretti del piano terapeutico.

AD21 genera i report giornalmente e li invia al medico di base settimanalmente.

# AD22 - Chiamata con il paziente

Complexity: bassa - Frequency: ogni 2 giorni - Delay: 30 minuti



Prima di avviare la chiamata tra il medico di base e il paziente, vengono recuperati ed inviati i report giornalieri che potrebbero non essere ancora stati inviati al medico di base (se non siamo ancora arrivati a fine settimana), dato che il medico di base sarà molto probabilmente interessato a questi durante la chiamata.

# Architettura Logica

# Complexity-driven: possibili valori e cardinalità

Data la vastità di possibili frequenze abbiamo pensato avesse senso unire quelle molto simili, di modo che poi gli score calcolati rispecchino effettivamente lo spread.

**6 diversi valori di frequenza:** F1 «più di 1 al secondo», F2 «tra uno al secondo e uno al minuto (escluso)», F3 «da uno al minuto a ogni 5 minuti», F4 «da ogni 10 minuti a ogni 60 minuti», F5 «ogni 1 o 2 giorni», F6 «ogni settimana o 2 settimane».

**7 diversi valori di delay:** «1 secondo», «2 secondi», «5 secondi», «30 secondi», «2 minuti», «5 minuti», «30 minuti».

22 diversi tipi di dato per l'**abstraction**.

3 diverse possibilità per la **complexity** (bassa, media, alta).

4 diverse **locations** degli attori.

# Complexity-driven: complessità bassa

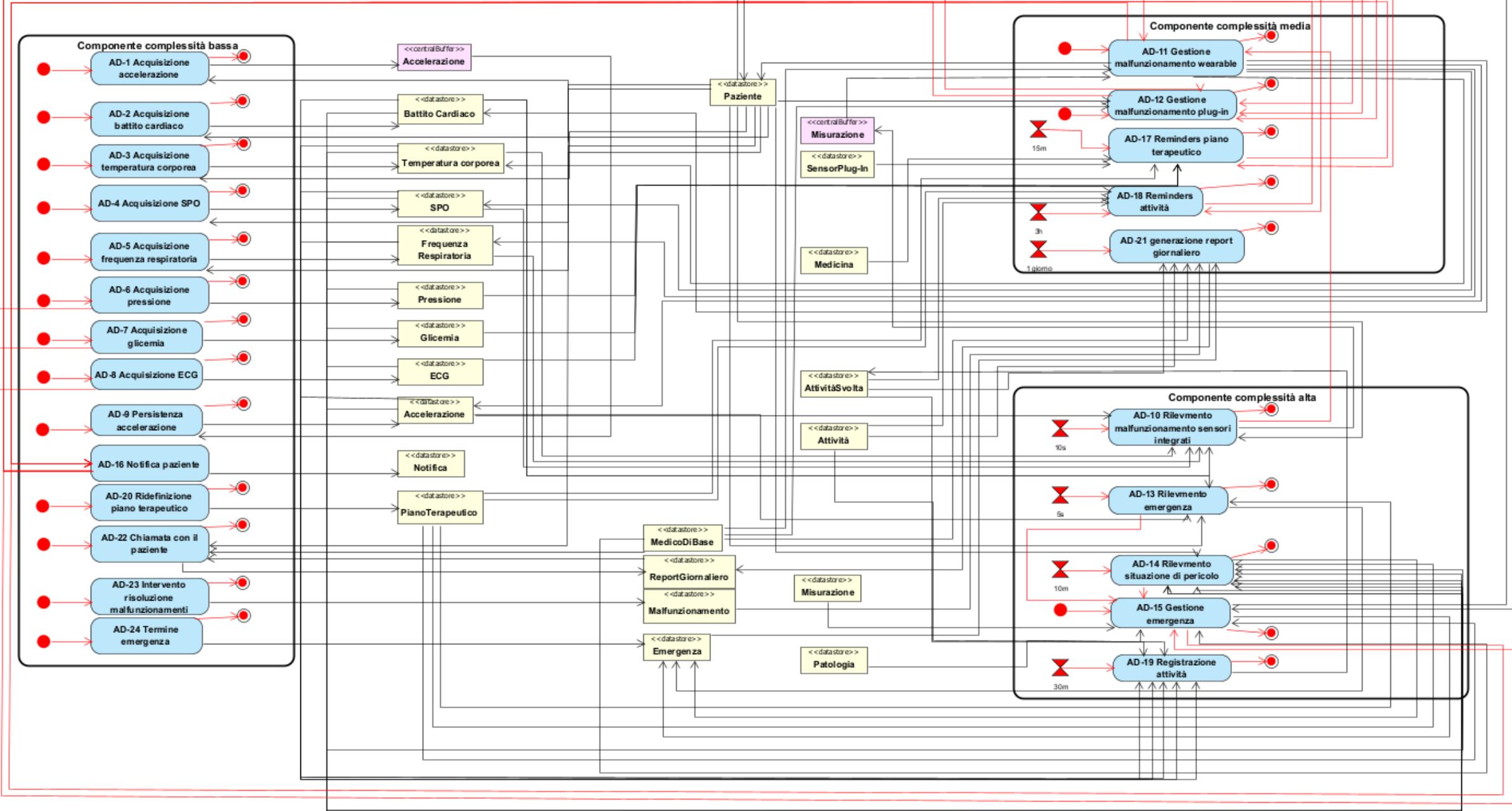
Diagramma attività		Complessità	Frequenza	Delay	Componente
AD1	Acquisizione accelerazione	Bassa	F1 - 100 volte al secondo	2 secondi	<b>Componente 1: complessità bassa</b>  <b>Molteplicità:</b> 1 per paziente (1x200 pazienti)
AD2	Acquisizione battito cardiaco	Bassa	F1 - 2 al secondo	2 secondi	
AD3	Acquisizione temperatura corporea	Bassa	F3 - Ogni 5 minuti	2 secondi	
AD4	Acquisizione SPO	Bassa	F3 - Ogni minuto	2 secondi	
AD5	Acquisizione frequenza respiratoria	Bassa	F3 - Ogni minuto	2 secondi	
AD6	Acquisizione pressione	Bassa	F4 - Ogni 15 minuti	2 secondi	
AD7	Acquisizione glicemia	Bassa	F4 - Ogni 60 minuti	2 secondi	
AD8	Acquisizione ECG	Bassa	F4 - Ogni 30 minuti	2 secondi	
AD9	Persistenza accelerazione	Bassa	F2 - Ogni 5 secondi	1 secondo	
AD16	Notifica paziente	Bassa	F4 - Ogni 10 minuti	5 secondi	
AD20	Ridefinizione piano terapeutico	Bassa	F6 - Ogni settimana	30 secondi	
AD22	Chiamata con il paziente	Bassa	F5 - Ogni 2 giorni	30 minuti	
AD23	Intervento risoluzione malfunzionamenti	Bassa	F5 - Ogni giorno	5 secondi	
AD24	Termine emergenza	Bassa	F6 - Ogni 2 settimane	5 secondi	

# Complexity-driven: complessità media

Diagramma attività		Complessità	Frequenza	Delay	Componente 2: complessità media  Molteplicità: 1 per paziente (1x200 pazienti)
AD11	Gestione malfunzionamento wearable	Media	F5 - Ogni giorno	30 minuti	
AD12	Gestione malfunzionamento plug-in	Media	F5 - Ogni giorno	30 minuti	
AD17	Reminders piano terapeutico	Media	F4 - Ogni 15 minuti	2 minuti	
AD18	Reminders attività	Media	F4 - Ogni 60 minuti	2 minuti	
AD21	Generazione report giornaliero	Media	F5 - Ogni giorno	5 minuti	

# Complexity-driven: complessità alta

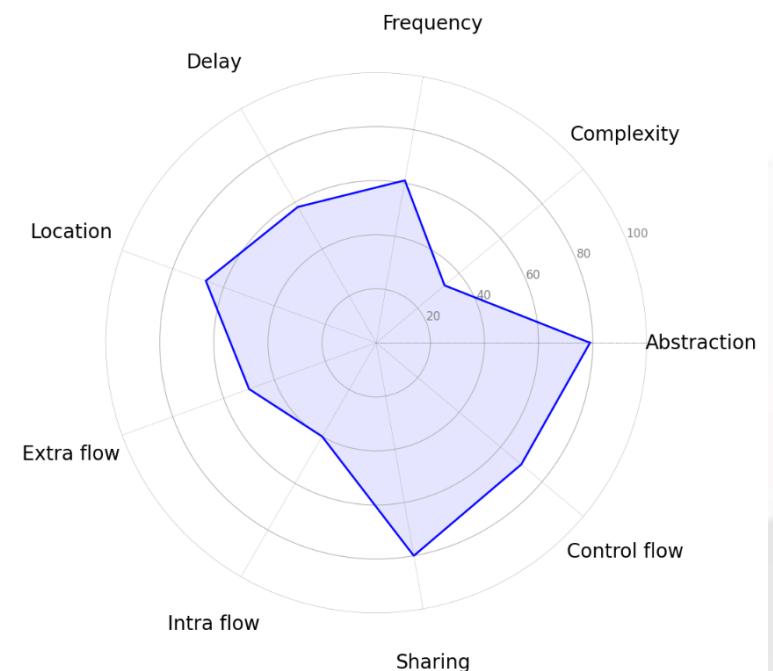
Diagramma attività		Complessità	Frequenza	Delay	Componente 3: complessità alta  Molteplicità: 1 per paziente (1x200 pazienti)
AD10	Rilevamento malfunzionamento sensori integrati	Alta	F2 - Ogni 10 secondi	5 secondi	
AD13	Rilevamento emergenza	Alta	F2 - Ogni 5 secondi	5 secondi	
AD14	Rilevamento situazione di pericolo	Alta	F4 - Ogni 10 minuti	2 minuti	
AD15	Gestione emergenza	Alta	F6 - Ogni 2 settimane	2 minuti	
AD19	Registrazione attività	Alta	F4 - Ogni 30 minuti	5 minuti	



# Complexity-driven: footprint

	Abstraction (data)	Complexity	Frequency	Delay	Location	Extra flow	Intra flow	Sharing	Control flow
<b>Componente 1</b> spread/value	16/22 = 73	1/3 = 33	6/6 = 100	6/7 = 86	4/4 = 100	Alto	Basso	Alto	Alto
<b>Componente 2</b> spread/value	18/22 = 82	1/3 = 33	2/6 = 33	3/7 = 30	2/4 = 50	Basso	Medio	Alto	Alto
<b>Componente 3</b> spread/value	18/22 = 82	1/3 = 33	3/6 = 50	4/7 = 57	2/4 = 50	Medio	Basso	Alto	Basso
<b>Totali complexity driven</b>	<b>79</b>	<b>33</b>	<b>61</b>	<b>58</b>	<b>67</b>	<b>50</b>	<b>40</b>	<b>80</b>	<b>70</b>

- Abstraction: ogni componente ha a che fare con molti diversi tipi di dato.
- Complexity: ogni componente contiene attività con complessità omogenee.
- Frequency: il primo componente contiene attività con frequenze molto diverse.
- Delay: il primo e terzo componente contengono attività con delay eterogenei.
- Location: nel primo componente le attività interagiscono con molti attori esterni.
- Extra flow: Il primo componente ha un'alta quantità di flussi dati con attori esterni, specialmente per colpa delle acquisizioni dei dati dai sensori.
- Sharing: le informazioni sono altamente condivise.
- Control flow: i primi due componenti hanno un control flow molto collegato.



# Functionality-driven: Acquisizione e persistenza dati

Diagramma attività		Complessità	Frequenza	Delay	Componente
AD1	Acquisizione accelerazione	Bassa	F1 - 100 volte al secondo	2 secondi	<b>Componente 1:</b> acquisizione e persistenza dati  <b>Molteplicità:</b> 1 per paziente (1x200 pazienti)
AD2	Acquisizione battito cardiaco	Bassa	F1 - 2 al secondo	2 secondi	
AD3	Acquisizione temperatura corporea	Bassa	F3 - Ogni 5 minuti	2 secondi	
AD4	Acquisizione SPO	Bassa	F3 - Ogni minuto	2 secondi	
AD5	Acquisizione frequenza respiratoria	Bassa	F3 - Ogni minuto	2 secondi	
AD6	Acquisizione pressione	Bassa	F4 - Ogni 15 minuti	2 secondi	
AD7	Acquisizione glicemia	Bassa	F4 - Ogni 60 minuti	2 secondi	
AD8	Acquisizione ECG	Bassa	F4 - Ogni 30 minuti	2 secondi	
AD9	Persistenza accelerazione	Bassa	F2 - Ogni 5 secondi	1 secondo	

## Functionality-driven: Gestione malfunzionamenti

Diagramma attività		Complessità	Frequenza	Delay	Componente
AD10	Rilevamento malfunzionamento sensori integrati	Alta	F2 - Ogni 10 secondi	5 secondi	<b>Componente 2: gestione malfunzionamenti</b>  <b>Molteplicità:</b> 1 per paziente (1x200 pazienti)
AD11	Gestione malfunzionamento wearable	Media	F5 - Ogni giorno	30 minuti	
AD12	Gestione malfunzionamento plug-in	Media	F5 - Ogni giorno	30 minuti	

## Functionality-driven: Gestione emergenze

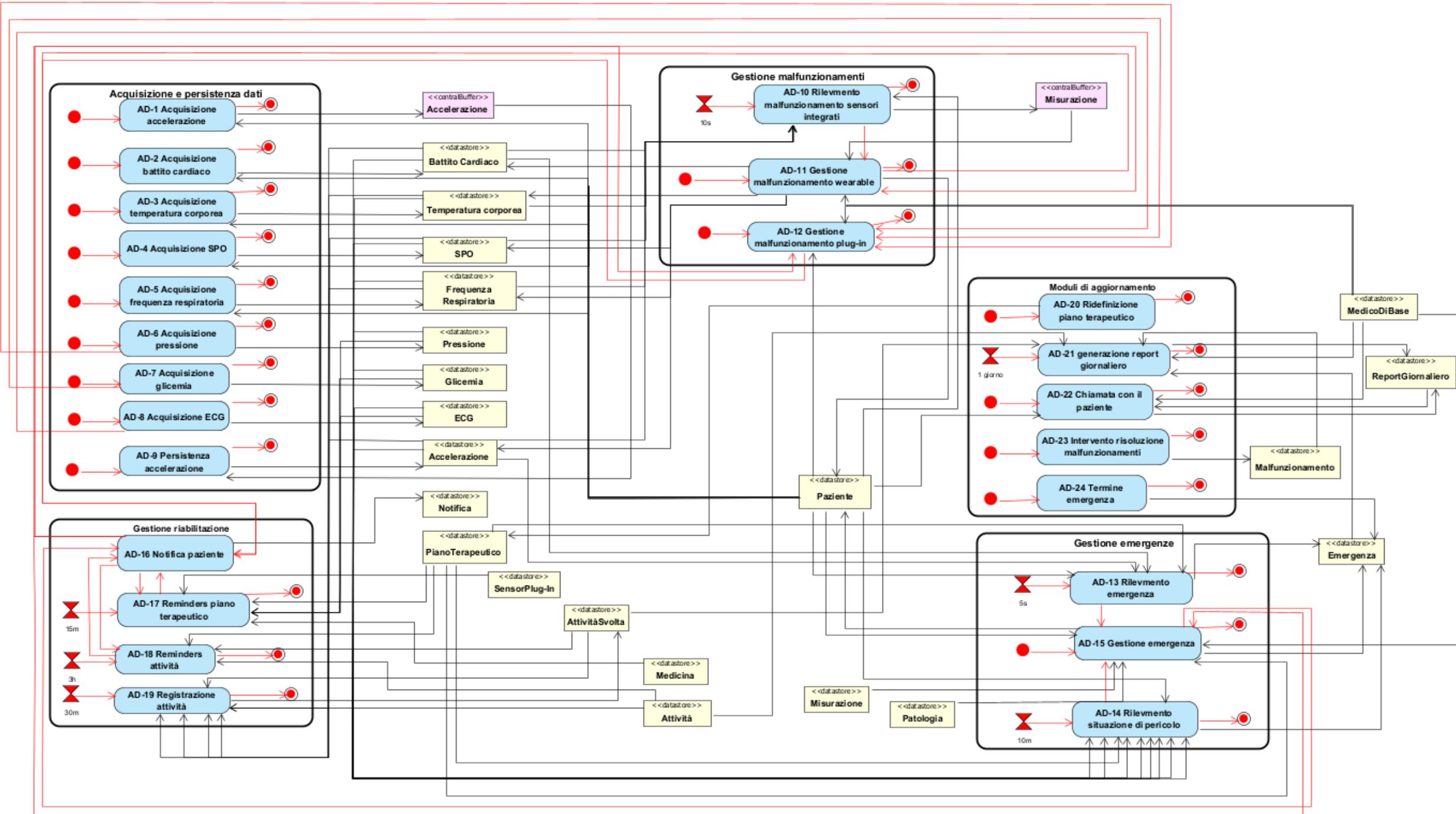
Diagramma attività		Complessità	Frequenza	Delay	Componente
AD13	Rilevamento emergenza	Alta	F2 - Ogni 5 secondi	5 secondi	<b>Componente 3: gestione emergenze</b>  <b>Molteplicità:</b> 1 per paziente (1x200 pazienti)
AD14	Rilevamento situazione di pericolo	Alta	F4 - Ogni 10 minuti	2 minuti	
AD15	Gestione emergenza	Alta	F6 - Ogni 2 settimane	2 minuti	

# Functionality-driven: Gestione riabilitazione

Diagramma attività		Complessità	Frequenza	Delay	Componente
AD16	Notifica paziente	Bassa	F4 - Ogni 10 minuti	5 secondi	<b>Componente 4: gestione riabilitazione</b>  <b>Molteplicità:</b> 1 per paziente (1x200 pazienti)
AD17	Reminders piano terapeutico	Media	F4 - Ogni 15 minuti	2 minuti	
AD18	Reminders attività	Media	F4 - Ogni 60 minuti	2 minuti	
AD19	Registrazione attività	Alta	F4 - Ogni 30 minuti	5 minuti	

# Functionality-driven: Moduli di aggiornamento

Diagramma attività		Complessità	Frequenza	Delay	Componente
AD20	Ridefinizione piano terapeutico	Bassa	F6 - Ogni settimana	30 secondi	<b>Componente 5: moduli di aggiornamento</b>  <b>Molteplicità:</b> 1 per paziente (1x200 pazienti)
AD21	Generazione report giornaliero	Media	F5 - Ogni giorno	5 minuti	
AD22	Chiamata con il paziente	Bassa	F5 - Ogni 2 giorni	30 minuti	
AD23	Intervento risoluzione malfunzionamenti	Bassa	F5 - Ogni giorno	5 secondi	
AD24	Termine emergenza	Bassa	F6 - Ogni 2 settimane	5 secondi	



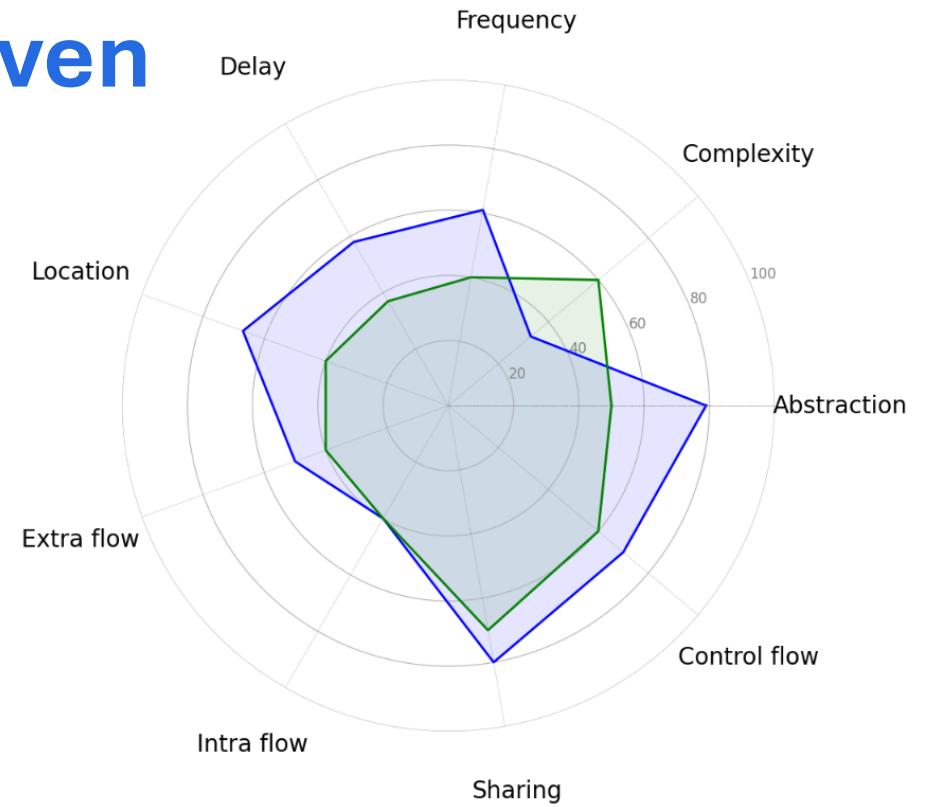
# Functionality-driven: footprint

	<b>Abstraction (data)</b>	<b>Complexity</b>	<b>Frequency</b>	<b>Delay</b>	<b>Location</b>	<b>Extra flow</b>	<b>Intra flow</b>	<b>Sharing</b>	<b>Control flow</b>
<b>Componente 1</b> spread/value	10/22 = 45	1/3 = 33	4/6 = 67	2/7 = 29	1/4 = 25	Alto	Basso	Alto	Alto
<b>Componente 2</b> spread/value	8/22 = 36	2/3 = 66	2/6 = 33	2/7 = 29	1/4 = 25	Basso	Medio	Medio	Alto
<b>Componente 3</b> spread/value	14/22 = 64	1/3 = 33	3/6 = 50	2/7 = 29	2/4 = 50	Medio-Basso	Medio	Alto	Basso
<b>Componente 4</b> spread/value	14/22 = 64	3/3 = 100	1/6 = 17	3/7 = 43	1/4 = 25	Basso	Medio	Medio	Medio
<b>Componente 5</b> spread/value	9/22 = 41	2/3 = 66	2/6 = 33	4/7 = 57	3/4 = 75	Medio	/	Medio	/
<b>Totali complexity driven</b>	<b>50</b>	<b>60</b>	<b>40</b>	<b>37</b>	<b>40</b>	<b>40</b>	<b>40</b>	<b>70</b>	<b>60</b>



# Comparazione tra complexity-driven e functionality-driven

	Complexity driven	Functionality driven
Abstraction	79	50
Complexity	33	60
Frequency	61	40
Delay	58	37
Location	67	40
Extra flow	50	40
Intra flow	40	40
Sharing	80	70
Control flow	70	60



- Abstraction: ogni componente continua ad avere un livello di abstraction abbastanza alto (principalmente data dall'alta quantità di sensori), la riduzione è comunque significativa.
- Complexity: il livello di complessità aumenta, ma comunque non è un problema perché rimane divisa, a parte il quarto componente che potrebbe essere diviso ulteriormente per ridurla.
- Frequency: le frequenze sono molto più omogenee rispetto a prima.
- Delay: anche i delay risultano più omogenei.
- Location: la divisione in funzionalità agisce bene nel ridurre lo score di location.
- Sharing: la condivisione di informazioni rimane comunque piuttosto alta.
- La soluzione functionality-driven risulta quindi ovviamente quella migliore.

# Architettura Concreta

# Scelta del tipo di componenti

**Componente acquisizione e persistenza dati:** per questo componente è stata scelta la soluzione **Stateful FarmManager**, data l'alta frequenza di generazione dei dati dai sensori. La scelta Stateful permette di assegnare ad ogni sensore un executor, di modo che questo non debba essere scelto dalla pool degli executor disponibili ogni volta, aiutando specialmente nel caso di sensori ad alta frequenza come accelerazione e battito cardiaco.

**Componente gestione malfunzionamenti:** qui invece è stata scelta la soluzione **Stateless FarmManager**, dato che in questo caso non è necessario mantenere lo stato tra esecuzioni diverse. Altre soluzioni come quella sequenziale non potrebbero funzionare, dato che se il processo è occupato da «gestione malfunzionamento plug-in», i malfunzionamenti sui sensori integrati non verrebbero controllati. La scelta one-shot sarebbe allo stesso modo errata, data la più alta frequenza dell'attività di riconoscimento dei malfunzionamenti che introdurrebbe molto overhead per le inizializzazioni. L'opzione «concorrente» potrebbe invece essere una buona alternativa, anche se l'inizializzazione di ciascun thread potrebbe comunque causare dell'overhead aggiuntivo.

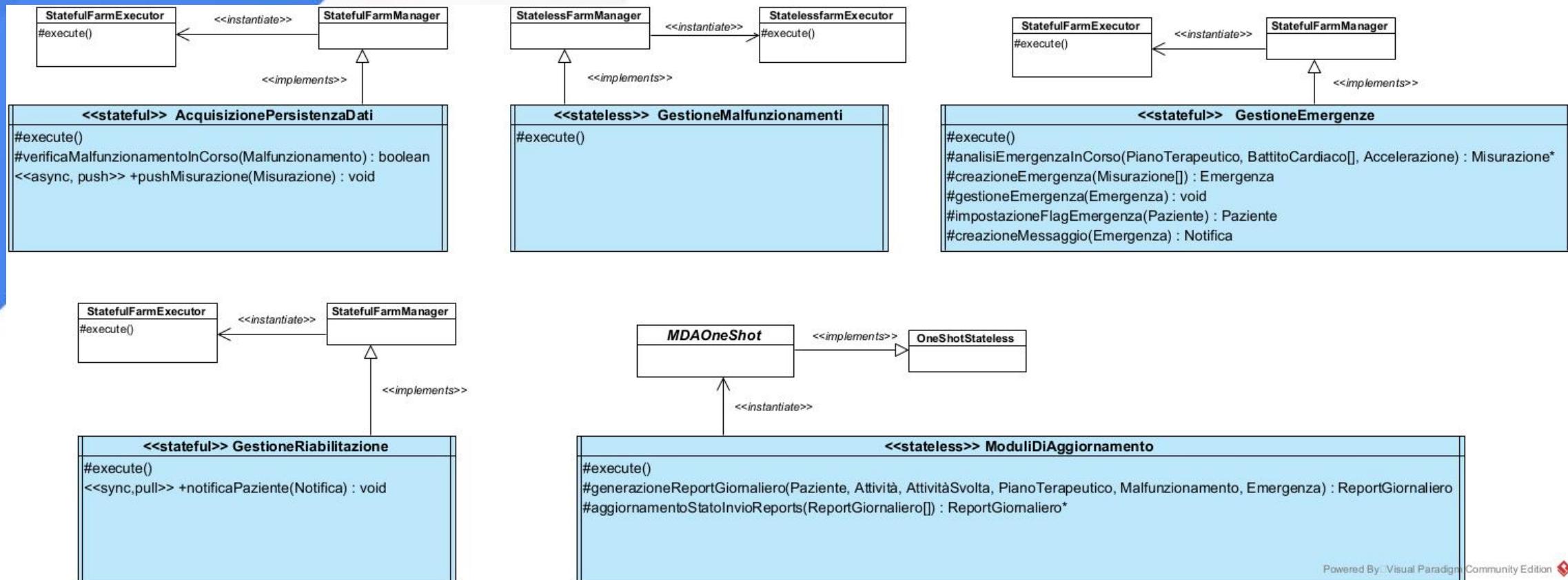
**Componente gestione emergenze:** per questo componente abbiamo scelto la soluzione **Stateful FarmManager**. I motivi sono simili al componente precedente. La scelta «stateful» invece ricade sul fatto che gli executor potrebbero giovare del mantenimento di alcuni dati tra le diverse esecuzioni, per quanto riguarda le attività di «rilevamento emergenza» e di «rilevamento situazione di pericolo» che potrebbero conservare alcuni dati per rendere le esecuzioni successive più veloci.

# Scelta del tipo di componenti

**Componente gestione riabilitazione:** per questo componente è stata scelta la soluzione **Stateful FarmManager**, per motivi simili al componente precedente, dato che anche qui il mantenimento dello stato nei diversi executors può aiutare a velocizzare le successive esecuzioni, dato che parte dei dati potrebbero essere uguali (per esempio se nel time step precedente stava registrando un'attività di corsa, questa potrebbe essere ancora in corso, ma per valutarla correttamente il componente avrebbe bisogno di recuperare anche alcuni dati già visti precedentemente).

**Componente moduli di aggiornamento:** la scelta è ricaduta sulla soluzione **Stateless OneShot**, dato che le attività di questo componente sono eseguite di rado e hanno una bassa urgenza, il che rende l'overhead dato dall'inizializzazione trascurabile.

# Diagramma architettura completa

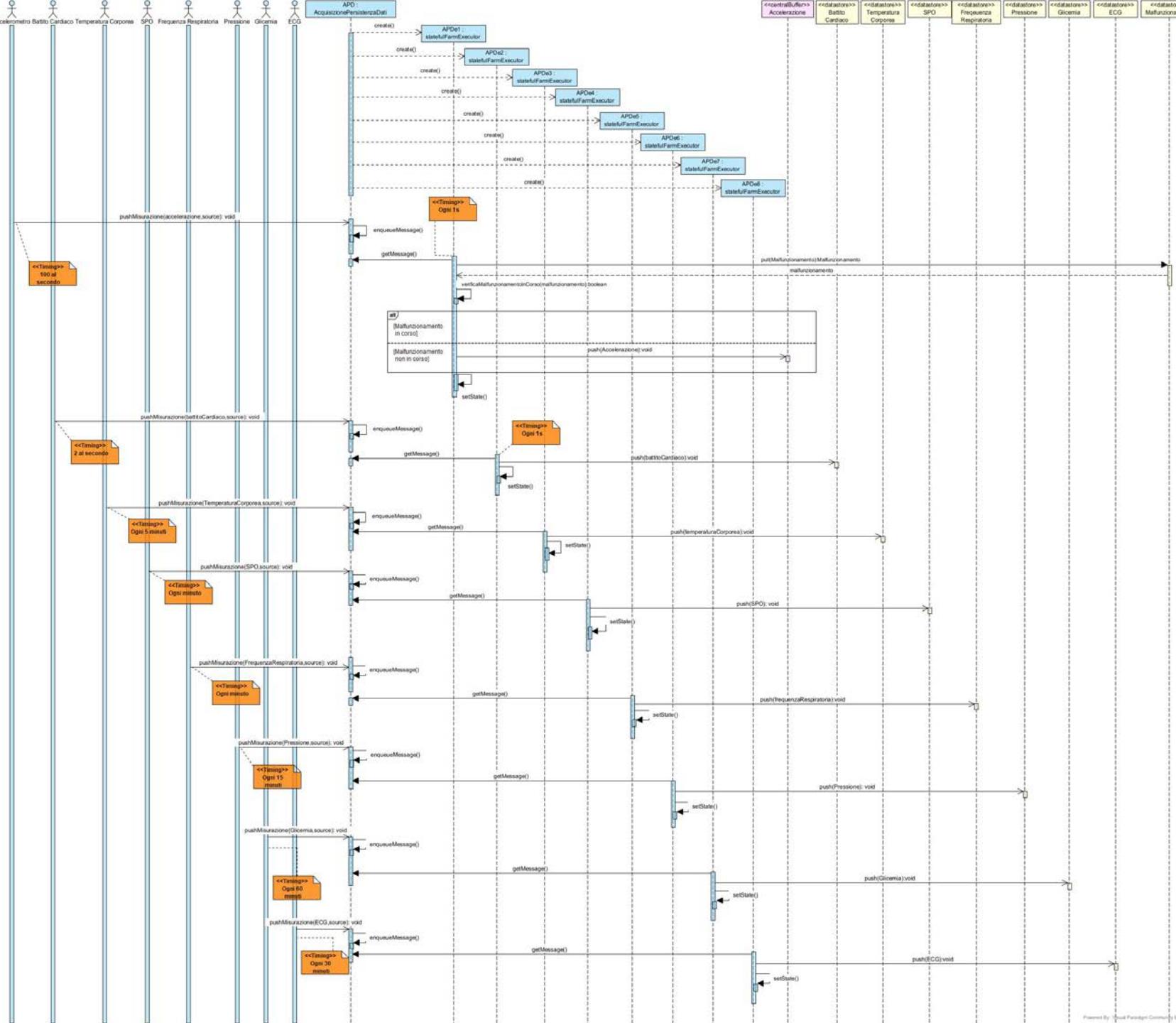


I metodi di ciascun componente dell'architettura completa rappresentano tutti i metodi necessari, utilizzati negli esempi implementati di seguito.

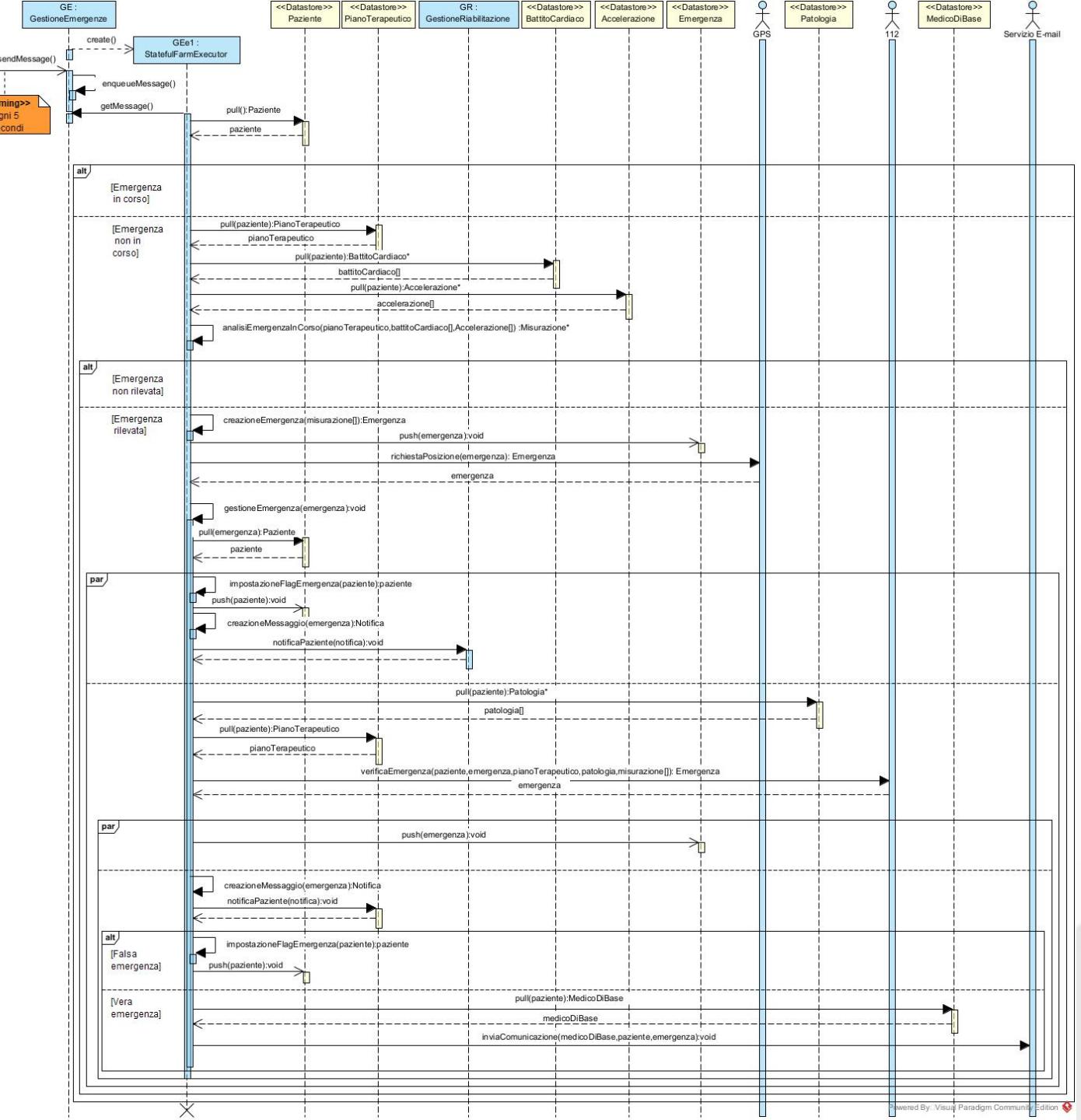
Non è stato sviluppato un esempio per il componente «GestioneMalfunzionamenti» in quanto questo sarebbe stato quasi identico nel funzionamento rispetto all'esempio sviluppato in «GestioneEmergenze».

# Componente acquisizione e persistenza dati

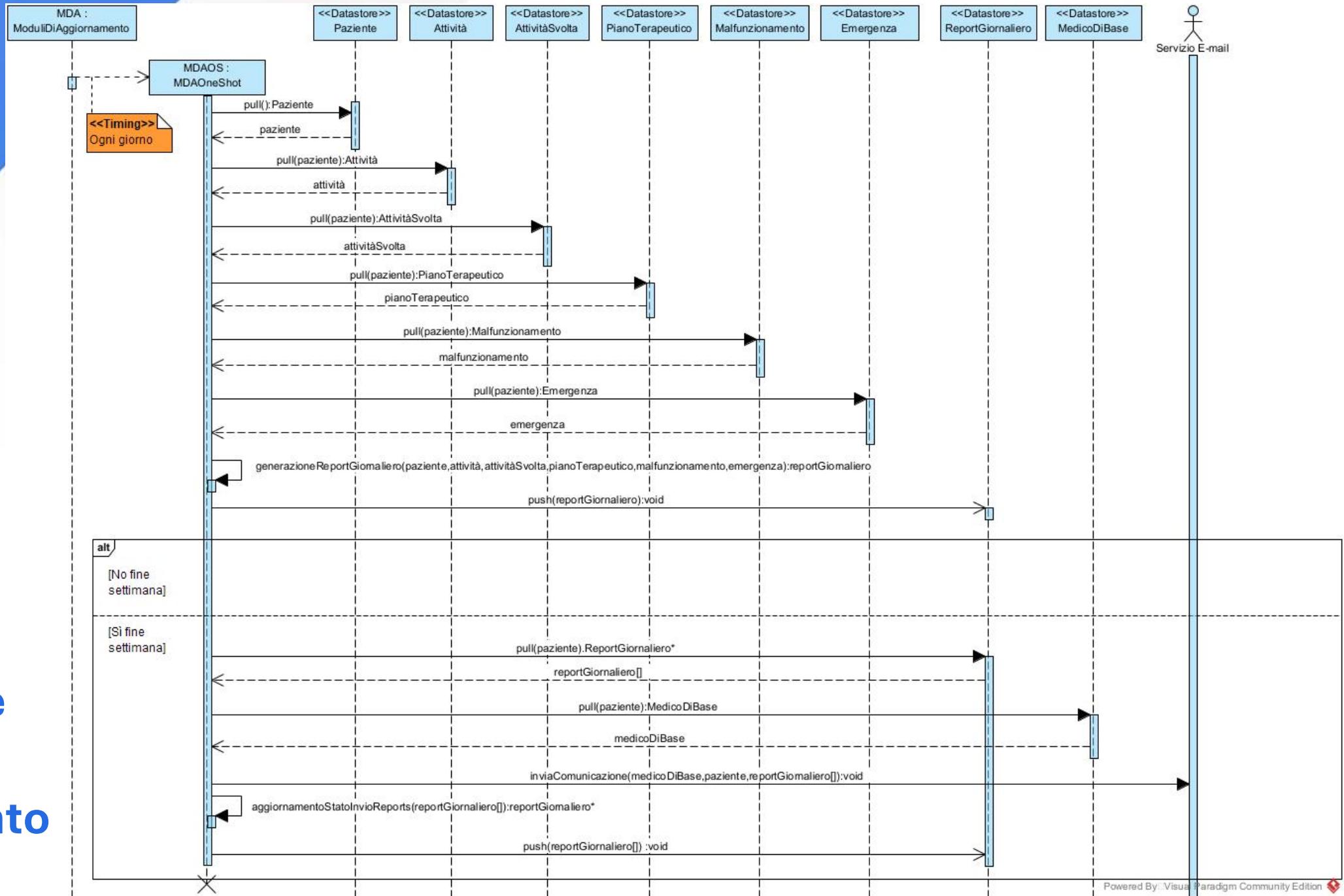
Il blocco «alt» centrale, insieme al poll dal datatore «Malfunzionamento», che verifica il malfunzionamento in corso andrebbe messo in tutte le sequenze, con la stessa logica. Lo abbiamo messo solamente nella prima per mantenere il diagramma leggibile.



# **Componente gestione emergenze e gestione riabilitazione**



# Componente moduli di aggiornamento



# Possibili miglioramenti ed espansioni

Nell'architettura logica, andare a dividere i componenti in modo diverso potrebbe avere dei vantaggi. Per esempio, per ridurre l'eterogeneità delle frequenze (specialmente in «Gestione Malfunzionamenti» e «Gestione Emergenze»), si potrebbe invece dividerli in «Rilevamenti» e «Gestioni», raggruppando le attività a frequenza più alta nei rilevamenti e quelle a frequenza più bassa nelle gestioni. Questo potrebbe avere effetti collaterali come l'aumento dell'eterogeneità delle complessità tra componenti, l'aumento dell'extra flow, dello sharing e del control flow.

In questo modo si potrebbe pensare, nell'architettura concreta, di avere «Rilevamenti» di tipo StatefulFarmManager, e «Gestioni» di tipo StatelessOneShot.

Una possibile espansione di questa architettura, sarebbe la gestione interna dei medici di turno e delle ambulanze (o comunque delle unità di emergenza), per esempio nel caso in cui il sistema comprendesse un personale medico adibito a monitorare e provvedere ai bisogni dei pazienti.

Si potrebbero anche espandere le funzionalità del dispositivo, prevedendo per esempio la possibilità di visionare ed interagire con recap giornalieri, attività svolte...