

UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA
E SCIENZE INFORMATICHE

PERVASIVE COMPUTING

**Room Assistant Agents
supporting Voice-based
Hands-free Interactions for
Smart Hospitals**

Diego Pergolini
Luca Passeri
Giuseppe Pisano

14 Giugno 2019

Indice

1	Introduzione	3
2	Requisiti	3
2.1	Prima intervista	3
2.1.1	Caratteristiche della shock room	4
2.1.2	Caratteristiche della gastroenterologia	4
2.1.3	Requisiti nel contesto Shock Room	4
2.1.4	Requisiti nel contesto della gastroenterologia	5
2.2	Seconda intervista	5
2.2.1	Requisiti generali per il caso	5
2.2.2	Requisiti per la tracciabilità	6
2.2.3	Requisiti per la visualizzazione	6
2.3	Terza Intervista	6
2.3.1	Requisiti applicativi	6
2.3.2	Requisiti funzionali	7
2.3.3	Analisi dei requisiti	7
2.4	Requisiti per l'utente	7
3	Analisi del problema	9
4	Architettura	11
4.1	Introduzione	11
4.2	Architettura del sistema	11
4.3	Smart Hospital Microservice	14
4.4	Trauma Vocal Microservice	14
4.5	Gastro Vocal Microservice	14
4.6	Smart Screen Vocal Microservice	15
4.7	Generic Vocal Microservice	15
5	Interazione con Alexa	15
5.1	Informazioni Stanza	15
5.2	Iniziare o terminare caso su Trauma Tracker	16
5.3	Farmaci	16
5.3.1	Farmaci con unità di misura	16
5.3.2	Procedure farmacologiche	17

5.3.3	Infusioni Continue	17
5.4	Manovre	18
5.4.1	Manovre con parametri	18
5.4.2	Manovre tempo-dipendenti	18
5.4.3	Resto delle manovre	19
5.5	Diagnostica strumentale	19
5.5.1	Diagnostica Laboratorio	19
6	Implementazione	20
6.1	Microservizi	20
6.2	Skill Alexa	21
6.2.1	Skill Trauma Tracker	22
6.2.2	Skill Smart Hospital	23
6.2.3	Skill Gastroenterologia	23
6.2.4	Skill Smart Screen	23
7	Considerazioni Finali	23
7.1	Fault tolerance/resilienza	23
7.2	Scalabilità	23
7.3	Sicurezza	24
7.4	Portabilità/applicabilità generale	24
7.5	Manutenibilità ed estendibilità	24
8	Validazione del sistema	26
	Appendici	27
A	Integrazione	27
B	Demo	28
B.1	Scenario A	28
B.2	Scenario B	29

1 Introduzione

Il progetto si colloca nel contesto dell'ideazione di smart environment a supporto del lavoro del personale medico e infermieristico negli ospedali.

Nella fattispecie si vuole ideare un assistant agent che dia assistenza per le attività che vengono svolte in un certo ambiente ospedaliero (stanze, sale) abilitando una forma di interazione hands-free basata su riconoscimento vocale.

Come tecnologie abilitanti a disposizione potrà essere usato Amazon Echo plus, come esempio di assistente vocale che mette a disposizione opportune API e SDK per essere sfruttato nella progettazione dell'assistant agent.

Come caso di studio e scenario di riferimento si potrà considerare la gestione di un trauma ad opera del Trauma Team, presso l'Ospedale Bufalini, considerando come ambiente la Shock Room.

2 Requisiti

2.1 Prima intervista

In questo primo incontro il team si è confrontato con la dottoressa Costanza Martino, in modo da definire con chiarezza quali sono gli obbiettivi del progetto e perciò l'outcome atteso. All'incontro hanno partecipato anche il professor Ricci, il dottor Croatti e la dottoressa Montagna come referenti del, già esistente, sistema Trauma Tracker.

In primo luogo il committente ha definito come ambito di utilizzo due scenari principali:

- Shock Room: una stanza dedicata al trattamento dei pazienti particolarmente critici, e che quindi accedono al pronto soccorso con codici ad elevata priorità, tipicamente in codice "rosso".
- Gastroenterologia: stanza dedicata allo svolgimento di esami diagnostici in ambito gastroenterologico ed endoscopico.

La scelta di questi due scenari è stata adottata tenendo conto delle opposte condizioni ambientali, in modo da poter testare il comportamento del sistema in contesti anche molto differenti tra loro.

2.1.1 Caratteristiche della shock room

Essendo una sala dedicata al trattamento di casi particolarmente gravi, con interventi tempo-dipendenti e con la presenza dell'intero trauma team (fino a 10 persone), essa si presenta come un ambiente molto rumoroso e concitato. Proprio per i fattori appena presentati andrà valutato se l'Echo Plus a nostra disposizione può rappresentare uno strumento valido per cogliere l'input vocale in una stanza così rumorosa.

Ci è stato inoltre confermato che la connessione Wi-Fi è disponibile in shock room.

2.1.2 Caratteristiche della gastroenterologia

Questo ambiente è, differentemente dalla shock room, molto tranquillo e silenzioso, con, tipicamente, un solo soggetto coinvolto che esegue la visita sul paziente.

2.1.3 Requisiti nel contesto Shock Room

La priorità dello staff medico presente nella shock room durante un'emergenza è quella di tracciare con precisione gli interventi fatti senza dover perdere tempo e poter continuare a svolgere le proprie mansioni senza dover interagire manualmente con il sistema. Durante un'emergenza tutti i membri del team potrebbero essere coinvolti in qualche mansione, non permettendo l'utilizzo di un dispositivo fisico, come ad esempio, un tablet.

Alla luce delle esigenze espresse è quindi evidente la necessità di un'interazione hands-free quale quella basata su input vocale, che nello specifico del progetto, è rappresentata dal dispositivo Echo Plus con corrispondente assistente vocale denominato Alexa.

Già da questa intervista è emersa la possibilità che, in un futuro anche molto vicino, si possa pensare di utilizzare un microfono personale, considerato specialmente l'elevato livello di rumore presente nella shock room.

Nello specifico, le funzionalità dovrebbe esporre l'assistente (che per brevità d'ora in poi chiameremo Alexa) sono:

- **Supporto alla tracciabilità:** deve esser possibile effettuare il tracciamento delle azioni eseguite dal trauma team tramite comandi vocali direttamente impartiti dal trauma leader. Tali comandi dovranno rispecchiare quelli già presenti nel sistema Trauma Tracker sotto forma di

input di interfaccia grafica. Ogni volta che un comando viene impartito, Alexa dovrà confermare l'effettiva registrazione dell'azione.

- **Visualizzazione:** Alexa dovrà supportare dei comandi tramite i quali richiedere la visualizzazione di informazioni riguardo al caso clinico su dispositivi di output, ad esempio schermi.

2.1.4 Requisiti nel contesto della gastroenterologia

In questo caso, essendo il contesto non tempo-dipendente, si è ritenuto sufficiente pensare all'utilizzo di Alexa esclusivamente nell'ambito della tracciabilità delle azioni.

2.2 Seconda intervista

In questa seconda intervista con il committente abbiamo approfondito alcuni aspetti prioritari, l'incontro si è svolto con la presenza del dottor Emiliano Gamberini e la dottoressa Costanza Martino. Presenteremo quindi i requisiti raffinati risultanti dall'incontro. In questo caso si è approfondito lo scenario prioritario rappresentato cioè dalla Shock Room.

2.2.1 Requisiti generali per il caso

Il sistema dovrà essere predisposto in modo tale che il trauma leader possa iniziare a trattare un caso tramite un apposito comando, da quel momento in poi tutte le azioni riguarderanno quello specifico paziente. Un problema emerso riguarda l'identificazione del paziente, poiché, per motivi di criticità il team potrebbe incominciare ad operare prima che il paziente abbia ricevuto il codice identificativo del Pronto Soccorso.

Al fine di migliorare la precisione dei comandi impartiti è stata fatta l'assunzione che solo il trauma leader sia adibito alla comunicazione con Alexa, per esempio tramite microfono personale.

Si è inoltre assunto, per semplicità, che un solo team per volta interagisca con il sistema e che un team leader sia esclusivamente assegnato ad un paziente. In futuro il desiderata è che la comunicazione all'interno del team venga tracciata completamente, permettendo di osservare il tempo trascorso tra l'impartizione di ordine del leader e la sua esecuzione da parte della squadra. In questo sarebbe quindi possibile valutare l'efficienza di un team.

2.2.2 Requisiti per la tracciabilità

Per ora si è scelto di limitare i comandi impartibili ad Alexa a queste tre categorie tra quelle presenti in Trauma Tracker:

- Manovre
- Farmaci
- Diagnostiche

Ognuna di queste categorie presenta diverse possibili opzioni che adesso sono selezionabili tramite interfaccia grafica mentre il nostro sistema dovrà permetterne l'inserimento tramite voce. La struttura del comando sarà quindi formata dalla categoria seguita dall'opzione specifica.

Nel caso dei farmaci è previsto anche di specificare la quantità dello stesso. Considerando che per i comandi di tracciabilità non è previsto alcun tipo di output, si è ritenuto necessario pensare ad un meccanismo di conferma dell'effettivo inserimento. In particolare è stato ritenuto utile presentare su uno schermo i comandi appena inseriti, unitamente ad un feedback vocale fornito da Alexa. Nel caso in cui si inserisca erroneamente un informazione nel sistema, Alexa dovrà dare la possibilità di eliminarla, anche in un secondo momento.

2.2.3 Requisiti per la visualizzazione

In questo caso Alexa dovrà essere in grado di riconoscere specifici comandi adibiti alla visualizzazione di informazioni. Per ora ci si è limitati a fornire la possibilità di mostrare il Report Overview del caso presente in Trauma Tracker. In seguito il sistema dovrà essere in grado di gestire altri comandi di visualizzazione, come per esempio dati relativi alla cartella clinica del paziente o lo stato di altre stanze.

2.3 Terza Intervista

2.3.1 Requisiti applicativi

In questo terzo incontro il team si è incontrato con Alessandro Ricci, Angelo Croatti e Sara Montagna, in modo da confrontarsi sui nuovi elementi emersi dal secondo incontro ed accordarsi sulle possibili future strategie di implementazione per il nostro progetto.

In primo luogo è emerso che, al momento, non è contemplata l'ipotesi di integrare nel sistema microfoni attraverso i quali interagire con il sistema, il case study per ora includerà soltanto un Amazon Echo Plus presente nella stanza.

Essendo un progetto esplorativo e legato alla reportistica, non considereremo, per ora, aspetti di privacy e sicurezza. Il dottor Croatti ci ha, in questo incontro, fornito una lista dei comandi attualmente mappati sull'applicazione "Trauma Tracker", essi costituiranno il riferimento per lo sviluppo delle Skill caratteristiche del nostro progetto.

2.3.2 Requisiti funzionali

Il sistema che andremo a realizzare dovrà essere integrabile con l'applicativo Trauma Tracker, senza però andare a modificare le logiche interne dello stesso. Dovrà perciò fornire un'interfaccia indipendente dall'applicativo, che potrebbe, potenzialmente, essere utilizzata da varie porzioni di quello che, in futuro, sarà un intero ecosistema ospedaliero.

2.3.3 Analisi dei requisiti

Alla luce dei requisiti emersi, si è pensato di realizzare un modulo separato adibito al supporto al tracciamento di dati provenienti da vari sorgenti ospedaliere. Tale modulo dovrà essere interfacciabile, in modo tale che, sia supportato una modalità d'interazione totalmente hands-free, basata nello specifico, sulla piattaforma Alexa. Inoltre dovrà mettere a disposizione dei possibili interessati tutte le informazioni registrate.

2.4 Requisiti per l'utente

Si dà ora una vista dei requisiti per l'utente emersi durante le varie interviste in modo che possa essere possibile effettuare una validazione del prototipo sulla loro base:

- Dovrà essere possibile impartire comandi vocali per la registrazione delle azioni compiute in Trauma Room, offrendo un'interfaccia alternativa a quella fornita da TraumaTracker;
- l'interazione dovrà essere il più semplice e immediata possibile;

- appena impartito il comando, l'assistente vocale dovrà confermare la ricezione dello stesso;
- entro un tempo breve, dovrà essere notificata all'utente l'effettiva registrazione del comando, segnalando eventuali anomalie;
- dovrà essere possibile eliminare l'ultimo comando registrato, nel caso vengano inserite erroneamente delle azioni;
- il sistema creato dovrà essere facilmente integrabile con quello già in essere (Trauma Tracker);
- il sistema dovrà permettere l'integrazione di nuovi scenari applicativi, legati agli altri domini ospedalieri (Gastroenterologia);
- più dispositivi di input vocale, collocati in aree diverse dell'ospedale, dovranno permettere il tracciamento di azioni relative allo stesso caso;
- il sistema dovrà essere in grado di riconoscere con una buona accuratezza i comandi impartiti.

3 Analisi del problema

Sulla base dei requisiti emersi, si rende necessario condurre un'approfondita analisi del problema in modo da arrivare ad una chiara delineazione del dominio applicativo e quindi delle entità necessarie al funzionamento del sistema.

Il focus nella fase di raccolta dei requisiti è stato sul tracciamento delle azioni nel contesto del trauma. È però emersa la volontà di creare un sistema abbastanza generico da permettere il tracciamento delle azioni relative ai diversi domini ospedalieri nelle varie aree. Questo ha portato all'identificazione di alcuni elementi di prima classe:

- **funzionalità**, questo concetto rappresenta un sottoinsieme del dominio funzionalmente indipendente, come può essere il tracciamento di un caso in trauma room;
- **stanza**, la rappresentazione di una sala ospedaliera. Visto che ogni stanza di un ospedale ha delle sue peculiarità, ovviamente offrirà delle funzionalità diverse ai suoi utilizzatori;

Avendo ora identificato delle stanze con varie funzionalità, non resta che introdurre degli elementi che facciano da mediatori per l'utente, andando a registrare i comandi in base alle caratteristiche della stanza in cui vengono impartiti. Da questo derivano altre due entità principali del dominio:

- **interfaccia vocale**, dispositivo fisico abilitato alla raccolta di comandi vocali. Questo potrà essere, o posizionato staticamente in una stanza, oppure muoversi fra le varie aree, offrendo i suoi servizi in modo coerente con la posizione in cui si trova;
- **comando**, elemento cardine del sistema alla base delle interazioni fra le varie applicazioni. Ogni comando presenta una struttura tale da riflettere il contesto in cui è stato impartito. Quindi un comando dovrà contenere le informazioni relative alla funzionalità, all'interfaccia vocale e alla stanza da cui proviene. Unitamente a queste informazioni, ricorrenti in tutte le tipologie di comandi, saranno presenti altri dati specifici del dominio applicativo;

Alla luce di queste considerazioni viene mostrato un diagramma UML che riassume la struttura del dominio in oggetto:

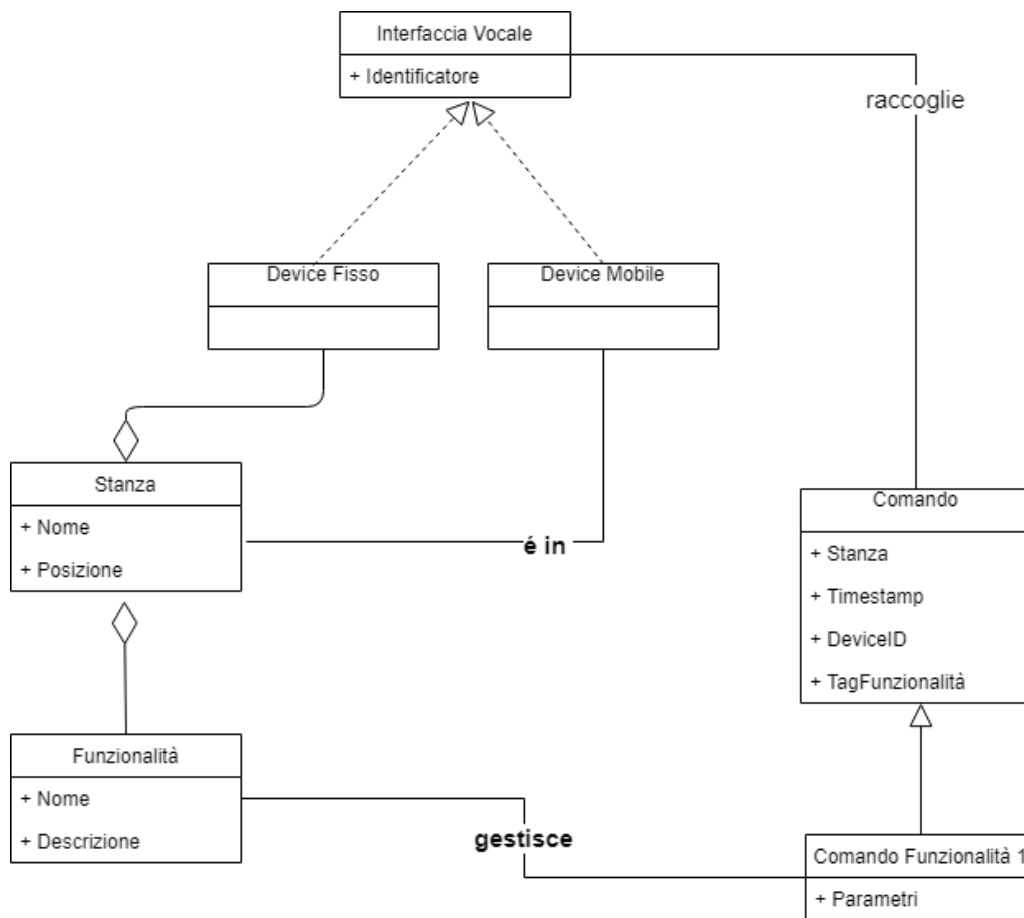


Figura 1: Diagramma UML del dominio

Gli elementi identificati sono risultati essere i componenti minimi necessari alla realizzazione di tutte le funzionalità richieste al sistema. Questa modellazione risulta essere inoltre estremamente flessibile, in quanto si può progettare il sistema prescindendo dai domini applicativi specifici.

4 Architettura

4.1 Introduzione

Questo progetto si trova all'interno di un contesto distribuito, in cui le informazioni provengono da varie fonti e devono essere rese disponibili a chiunque sia interessato.

E' chiaro che il focus sia sui dati, il che ci porta ad una progettazione del sistema secondo le best practices dell'Internet of Things.

Considerando la natura dei dati, i quali sono intrinsecamente strutturati e facilmente categorizzabili è stata pensata un'architettura che si adattasse il più naturalmente possibile ad essi. Si tratta infatti di comandi precisi che seguono una struttura predefinita, e ogni comando può essere classificato in base alla funzionalità a cui si riferisce. L'approccio a microservizi è risultato essere la scelta migliore, in quanto ogni servizio sarebbe stato adibito ad una funzionalità specifica: il fatto che tali funzionalità siano un numero limitato e quantificabile giustifica appieno la scelta di tale approccio.

Le funzionalità quindi vengono individuate implicitamente in base alle informazioni provenienti dall'assistente vocale.

Per ogni micro-servizio sono state pensate delle web API che rendessero le informazioni facilmente usufruibili a chiunque, seguendo gli standard del WoT. In particolare mettendo a disposizione delle WebSocket per realizzare una comunicazione reattiva in grado di distribuire le informazioni istantaneamente a tutti gli interessati ed inoltre facendo sì che ogni micro-servizio esponga una REST API che permetta di ottenere informazioni specificando appositi parametri.

4.2 Architettura del sistema

Alla luce di tali considerazioni, e del modello formale visto nella sezione precedente, è stata realizzata un'architettura generale del sistema.

Come evidenziato nella figura [2](#) si possono identificare 5 micro-servizi. Ognuno con peculiarità e funzionalità differenti. Essi sono:

- Smart Hospital Microservice
- Trauma Vocal Microservice
- Smart Screen Vocal Microservice

- Gastro Vocal Microservice
- Generic Vocal Microservice

Questi microservizi possono essere raggruppati in tre principali aree funzionali sulla base delle loro caratteristiche.

Lo Smart Hospital Microservice funge da punto d'ingresso, offrendo le informazioni anagrafiche sulle funzionalità offerte dal sistema ai vari device di riconoscimento vocale registrati in base alla stanza d'appartenenza.

Il Trauma Vocal Microservice, lo Smart Screen Vocal Microservice e il Gastro Vocal Microservice gestiscono ognuno una parte specifica del dominio applicativo considerato (funzionalità). Il primo è quello adibito alla gestione di tutte le interazioni relative al contesto delle Trauma Room, il secondo a quelle registrate nelle aree di Gastroenterologia, il terzo, più trasversale rispetto ai primi due, a tutte quelle aree attrezzate con gli strumenti di visualizzazione, in modo da permetterne il pilotaggio tramite input vocali.

Il Generic Vocal Microservice è un elemento di servizio con lo scopo di aggregare tutte le informazioni gestite e generate dai tre servizi descritti sopra. In questo modo è possibile monitorare facilmente il sistema nella sua globalità, astruendo l'organizzazione in microservizi adottata.

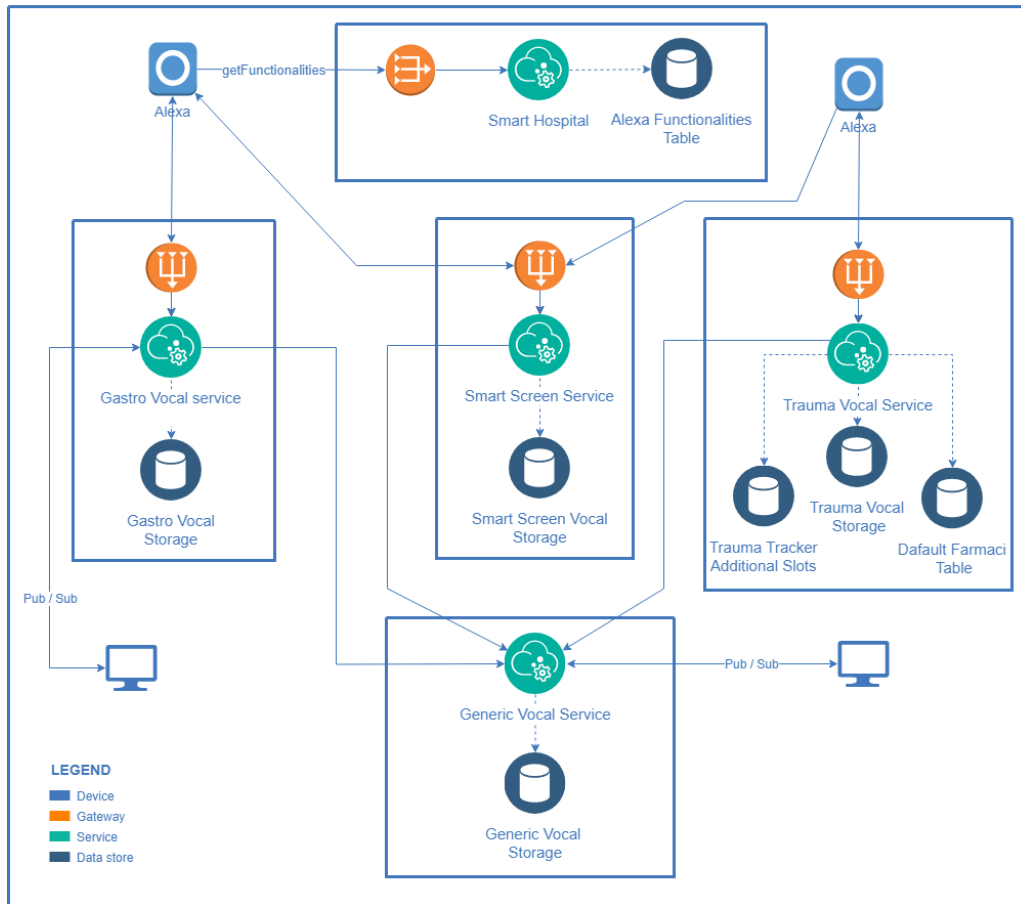


Figura 2: Architettura del sistema

Questa architettura, oltre a prestarsi molto bene alla realizzazione di un sistema completamente reattivo (resiliente, scalabile, reattivo, a scambio di messaggi), espone anche un'altra caratteristica molto importante: la flessibilità. Infatti, la divisione in servizi sulla base delle aree funzionali del dominio abilita l'aggiunta di funzionalità semplicemente tramite la realizzazione di un microservizio dedicato, senza andare a impattare sulla parte del sistema già in essere. Oltretutto l'adozione di un modello di comunicazione standard (REST) con i dispositivi di input vocale (Amazon Alexa nell'implementazione attuale) rende possibile sostituire facilmente questi elementi, abilitando teoricamente uno scenario in cui sono dei dispositivi eterogenei fra loro a essere incaricati dell'acquisizione dei dati.

Si andrà ora a descrivere più nel dettaglio le funzionalità offerte dai vari servizi.

4.3 Smart Hospital Microservice

Questo microservizio, collegato ad una tabella di storage denominata *AlexaFunctionalities*, fornisce la possibilità di ottenere, dato un certo Alexa in una certa stanza, quale funzionalità sono presenti in essa e come è possibile interagirci tramite comandi vocali. Nell’ottica di sviluppare uno smart environment, questo servizio assume un ruolo molto importante, dando la possibilità di utilizzare un assistente vocale per utilizzare le funzionalità peculiari della stanza in cui ci si trova. Nell’attuale implementazione, ad ogni Echo presente in una stanza, dotato ovviamente di un Id univoco, vengono associate delle informazioni sulla stanza in cui esso si trova. In un ottica futura, in cui l’assistente personale Alexa sia utilizzabile ed utilizzato anche su device personali, con opportune tecnologie di localizzazione indoor sarà possibile capire in che stanza si trova l’utilizzatore permettendo di interagire con l’assistente in modo coerente rispetto alle peculiarità della stanza.

4.4 Trauma Vocal Microservice

Questo microservizio è adibito esclusivamente alla gestione di tutte le funzionalità legate all’utilizzo di comandi vocali per interagire con l’applicazione trauma tracker. La Skill appositamente sviluppata per la gestione dei trauma si rivolgerà a questo servizio inviando i comandi appena ricevuti. Il microservizio una volta ricevuto il comando si occuperà di completare l’informazione ricevuta se parziale, per esempio aggiungendo le unità di misura standard dei farmaci somministrati se non specificata dal medico. Tutti i dati registrati verranno poi messi a disposizione dei consumatori interessati, o nella loro globalità, o filtrati in base alla stanza in cui sono stati acquisiti.

4.5 Gastro Vocal Microservice

Questo microservizio è invece adibito esclusivamente alla gestione di tutte le funzionalità legate all’utilizzo di comandi vocali nelle aree adibite all’esecuzione di esami gastroenterologici. La Skill appositamente sviluppata si rivolgerà a questo servizio inviando i comandi ricevuti. I comandi identificati e gestiti sono al momento quelli di inizio e fine di un esame. Anche in questo

caso tutti i dati registrati verranno poi messi a disposizione dei consumatori interessati secondo le stesse modalità viste per il Trauma Vocal Service.

4.6 Smart Screen Vocal Microservice

Microservizio adibito alla gestione di tutte le funzionalità legate all'utilizzo di comandi vocali finalizzati al pilotaggio di schermi e quindi alla visualizzazione di dati. Anche in questo caso tutti i dati registrati verranno poi messi a disposizione dei consumatori interessati.

4.7 Generic Vocal Microservice

Questo microservizio, come già anticipato nella parte introduttiva, non possiede nessun endpoint pubblicamente raggiungibile per l'inserimento di dati, ma ha lo scopo di canalizzare e registrare tutti i dati generati dagli altri microservizi. Risulta essere quindi un ottimo punto per il monitoraggio e l'auditing del sistema nella sua interezza. Tutti i dati registrati sono messi a disposizione dei consumatori interessati, o nella loro globalità, o filtrati in base al device da cui sono stati generati.

5 Interazione con Alexa

In questa sezione verranno mostrate le modalità di interazione con Alexa per registrare i vari elementi in Trauma Tracker.

5.1 Informazioni Stanza

Nell'ottica di realizzare uno smart environment si è sviluppata una skill tramite cui chiedere all'Alexa presente nella stanza quale funzionalità offre. Per aprire la skill l'utente può dire:

Apri informazioni stanza

Una volta aperta la skill l'utente può chiedere quali sono le abilità della stanza con queste frasi:

Cosa sai fare?

Cosa sa fare questa stanza?

Che servizi offre questa stanza?
Che funzionalità ha questa stanza?
Cosa puoi fare?

5.2 Iniziare o terminare caso su Trauma Tracker

Per avviare un caso su Trauma Tracker l'utente può dire:

Apri caso
Inizia registrazione trauma
Inizia trauma
Avvia caso
Incomincia caso

Per terminare un caso su Trauma Tracker l'utente può dire:

Concludi caso
Termina registrazione caso
Fine trauma
Chiudi caso
Termina caso

5.3 Farmaci

5.3.1 Farmaci con unità di misura

Per tutti i farmaci somministrati con dosi associate si può dire:

farmaco {nomeFarmaco}

Oppure:

farmaco {nomeFarmaco} dose {quantità}

Dove {nomefarmaco} può assumere uno dei valori nelle tabelle di seguito ed i valori fra parentesi quadre rappresentano il dosaggio di default, altrimenti si può specificare la quantità {quantità} e l'unità di misura verrà attribuita in automatico dal sistema.

Infusioni	Farmaci Generici
CRISTALLOIDI [500 ml]	KETAMINA [100mg]
SOLUZIONE IPERTONICA [250 ml]	SUCCINILCOLINA [100mg]
	MIDAZOLAM [5mg]
	TIOPENTONE [250mg]
	FENTANIL [100mcg]
	CURARO [10mg]
	ACIDO TRANEXAMICO [1g]
	MANNITOLO [300ml]

Farmaci ALS	Emoderivati
ATROPINA [1mg]	EMAZIE CONCENTRATE [1UI]
ADRENALINA [1mg]	PIASTRINE [1UI]
	PLASMA FRESCO CONCENTRATO [1UI]
	FIBRINOGENO [1g]
	COMPLESSO PROTROMBINICO [500UI]

5.3.2 Procedure farmacologiche

Per inserire le procedure farmacologiche “Profilassi ABT”, “P.T.M.” e “SHOCK ELETTRICO” si può dire:

procedura {nomeProcedura}

Dove {nomeProcedura} può assumere uno dei valori sopra elencati.

5.3.3 Infusioni Continue

Per ciascuna infusione continua è possibile registrare l’inizio (con uno specifico dosaggio), una o più modifiche al dosaggio con infusione già iniziata e la fine dell’infusione.

{fase} infusione {nomeFarmaco} dose {quantità}

Dove {fase} può assumere i valori INIZIO, FINE o MODIFICA e {nomeFarmaco} può assumere i valori:

- ADRENALINA
- NORADRENALINA

- PROPOFOL
- KETANEST
- (FARMACO GENERICO IN INFUSIONE)

5.4 Manovre

5.4.1 Manovre con parametri

Ad alcune manovre sono associati dei parametri i quali valori verranno richiesti da Alexa all'utente. Tutti i parametri di queste manovre sono di tipo booleano, cioè verrà chiesto di rispondere sì o no al parametro considerato. Un esempio è il seguente:

UTENTE: "Manovra drenaggio toracico"
 ALEXA: "Sinistro sì o no?"
 UTENTE: "No"
 ALEXA: "Destro sì o no?"
 UTENTE: "Sì"

Queste manovre sono registrabili dicendo:

- Manovra intubazione orotracheale
- Manovra drenaggio toracico
- Manovra decompressione pleurica

5.4.2 Manovre tempo-dipendenti

Alcune manovre (indicate dal simbolo *) sono tempo-dipendenti, ovvero viene registrato il tempo d'inizio e il tempo di fine, quindi con due registrazioni per ciascuna manovra: es. INIZIO manovra REBOA e FINE manovra REBOA. Per registrare questo tipo di manovre si può dire:

{fase} manovra {nomeManovra}

Dove {fase} può assumere i valori INIZIO o FINE e {nomeManovra} può assumere i valori: ALS, TORACOTOMIA, REBOA, TOURNIQUET

5.4.3 Resto delle manovre

Per registrare tutte le altre manovre non incluse nei casi precedenti si può dire:

manovra {nomeManovra}

oppure:

eseguita manovra {nomeManovra}

5.5 Diagnostica strumentale

Per registrare una diagnostica strumentale si può dire:

diagnostica {nomeDiagnostica}

Dove {nomeDiagnostica} può assumere uno dei seguenti valori: ECOGRAFIA (ECHOFAST), RX TORACE, RX BACINO, TC CEREBRALE-CERVICALE, TC TORACO-ADDOMINALE (MDC), WB-TC.

5.5.1 Diagnostica Laboratorio

Esistono due tipologie di diagnostiche di laboratorio: EGA e ROTEM, entrambe dotate di parametri aggiuntivi. Per la diagnostica EGA, bisognerà dire:

Diagnostica EGA {nomeParametro}

Dove {nomeParametro} può assumere i valori: **Lattati, BE, PH, HB**. A questo punto Alexa chiederà il valore del parametro specificato.

Per la diagnostica ROTEM, per iniziare la registrazione è sufficiente dire:

Diagnostica ROTEM

Alexa a questo punto chiederà i valori dei vari parametri associati alla diagnostica, l'utente dovrà ovviamente rispondere in modo coerente alle domande.

6 Implementazione

6.1 Microservizi

Per quanto riguarda l'implementazione dei servizi illustrati nella sezione precedente, è stato utilizzato il framework Vert.x. Pensato appositamente per la realizzazione di applicazioni reattive, questo toolkit offre tutti gli strumenti necessari alla creazione di sistemi scalabili e resilienti, quali il supporto nativo alla clusterizzazione. Infatti, sebbene non sia stata fatta nessuna prova in tal senso nell'implementazione attuale (ogni servizio è gestito da un solo nodo), risulterebbe semplice estendere il prototipo creato in modo che ogni modulo facesse affidamento su un cluster di macchine in modo da garantire scalabilità, grazie alla distribuzione del carico di lavoro, e resilienza (alla caduta di un nodo questo verrebbe immediatamente sostituito da un altro nodo del cluster).

Anche la realizzazione delle Web Api offerte dai vari servizi è stata semplificata dall'utilizzo di questo framework, che supporta nativamente la creazione di endpoint Rest, andando a gestire aspetti quali routing e sessione. Oltre tutto offre un'integrazione con la libreria SockJs, che è stata utilizzata per la distribuzione degli eventi di input vocale ai consumatori esterni.

Vert.x non offre di default un meccanismo di load balancing delle richieste alle interfacce Rest, quindi in caso di scaling orizzontale sarebbe necessario o implementare un componente che si occupi di questo aspetto (utilizzando Vert.x stesso), oppure utilizzando una soluzione esterna (Nginx per citarne uno).

Questo accorgimento andrebbe a coprire anche un altro aspetto non considerato nell'attuale implementazione ma che risulta essere di fondamentale importanza: la sicurezza. Infatti piuttosto che gestire la fase di autorizzazione all'interno dei microservizi stessi, questa potrebbe essere centralizzata in un unico modulo che funga in pratica da proxy per le chiamate.

Per quanto riguarda la comunicazione fra microservizi è stato utilizzato Apache Kafka, un Message Oriented Middleware ottimizzato per la gestione di enormi moli di dati con basse latenze. Anche per questa piattaforma Vert.x offre un connettore che ne ha permesso un'integrazione semplice ed efficace all'interno del sistema.

Ultimo aspetto degno di nota dell'implementazione del prototipo è la gestione dei dati all'interno dei servizi. Ognuno di questi possiede un proprio insieme di tabelle, specifiche per la parte di dominio gestita, in modo da favorire un completo incapsulamento delle informazioni e limitare le dipendenze fra essi. Per la realizzazione si è fatto affidamento sul servizio *Aws DynamoDB*, database NoSQL proprietario completamente gestito che supporta strutture chiave-valore e documentali.

6.2 Skill Alexa

Data la scelta iniziale di utilizzare Amazon Alexa come tecnologia abilitante per costruire un assistente vocale per ambienti ospedalieri si sono dovute creare delle Skill per ogni funzionalità identificata. Le Skill sono le applicazioni vocali di Alexa. Queste permettono di attivare nuove funzionalità e contenuti che arricchiscono l'esperienza fornita dai dispositivi Echo e da quelli con integrazione Alexa.

Nel creare una Skill lo sviluppatore dovrà formalizzare un modello d'interazione, grazie al quale l'utente potrà sfruttare le funzionalità del sistema. Ovviamente il modello d'interazione deve essere il più semplice ed intuitivo possibile, in modo da ridurre al minimo in tempo di formazione necessario al personale (o l'utenza) che ne dovrà usufruire.

La creazione di una Skill Alexa prevede due fasi, la prima, già citata, è quella della creazione del modello di interazione, la seconda consiste invece nella creazione di un Web Service *Https* di supporto alla Skill.

La necessità di questo web service nasce dallo schema di funzionamento di Alexa, infatti, come evidenziato in Figura 3, quando l'utente interagisce vocalmente con il device, l'audio viene mandato in Cloud, dove risiede tutto il modello di riconoscimento vocale, che in base al modello d'interazione specificato nella skill produrrà delle richieste post al Web Service *Https*. Il corpo della richiesta contiene parametri che il servizio può utilizzare per eseguire la logica e generare una risposta in formato JSON, risposta che sarà quindi ricondotta in formato vocale ed emessa dal dispositivo.

In questo caso, come fortemente consigliato dalle linee guida di Alexa, il Web Service *Https* è ospitato nel servizio *AWS Lambda Function*, dato che questo servizio permette di realizzare un web service gestendo in automatico aspetti come lo scaling, il deploy e la sicurezza. Le *lambda function* di supporto sono state scritte in Javascript basandosi sul runtime *NodeJS 8.10*.

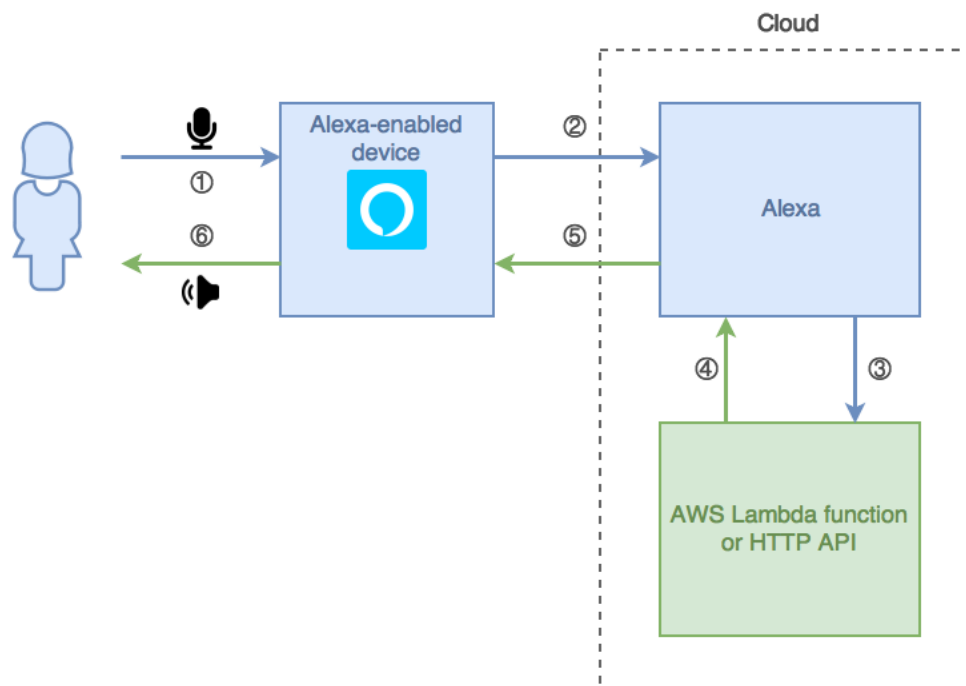


Figura 3: Schema di funzionamento Alexa Skill

6.2.1 Skill Trauma Tracker

La skill più importante realizzata è quella per registrare vocalmente le azioni intraprese durante la gestione di un caso in Trauma Room. Essa si basa interamente sul modello descritto nella Sezione 5, ed è stata sviluppata utilizzando l'ambiente fornito da Amazon denominato [Alexa Skills Kit Developer Console](#).

Per quanto riguarda l'endpoint Lambda ove è contenuta la logica applicativa per gestire e rispondere ai comandi utente, oltre a gestire tutti i comandi impartibili dall'utente, si è provveduto ad implementare delle funzionalità che controllassero l'effettiva abilitazione all'uso della skill da parte del device utilizzato. Nell'ottica di uno Smart Hospital si è infatti deciso di descrivere per ogni stanza quali sono le funzionalità, e quindi le skill, abilitate.

La lambda è responsabile della comunicazione con tutta l'architettura a microservizi già descritta, e nello specifico di questa Skill, di utilizzare il Trauma Vocal Service per registrare le varie azioni eseguite, fornendo una conferma

una volta correttamente memorizzate.

6.2.2 Skill Smart Hospital

Questa skill, che offre i semplici comandi descritti nella sezione 5, si premu- ra di richiedere allo Smart Hospital Microservice quali funzionalità espone la stanza in cui si trova il dispositivo attraverso il quale l'utente ha impartito dei comandi vocali, riportando quindi all'utente quali siano queste funzionalità e come si possono utilizzare.

6.2.3 Skill Gastroenterologia

Questa Skill dovrebbe coadiuvare il medico durante un esame gastroenterolo- gico, per ora però si è deciso di farne solo un Mock, per mostrare il potenziale utilizzo di funzionalità diverse in base alla stanza in cui ci si trova.

6.2.4 Skill Smart Screen

Questa Skill è adibita al pilotaggio degli Smart Screen tramite comandi vo- cali, ovviamente è utilizzabile in quelle stanze dove è presente uno smart screen.

7 Considerazioni Finali

7.1 Fault tolerance/resilienza

Per quanto riguarda la parte gestita da Aws, questa è completamente re- siliente grazie alla gestione completamente serverless del sistema dato dalle funzioni Lambda. La parte di microservizi, sebbene non siano stati presi al momento accorgimenti specifici per garantire fault tolerance, è stata proget- tata e realizzata in modo da permetterne una facile gestione futura, come già spiegato nella parte di implementazione.

7.2 Scalabilità

Di questo punto si è già parlato nella sezione implementativa per quanto concerne le possibilità di scalabilità orizzontale offerta dalla gestione a mi-

crosservizi. La parte Aws, invece, grazie alle potenzialità del Cloud, offre una scalabilità potenzialmente infinita.

7.3 Sicurezza

Per quanto riguarda la sicurezza della comunicazione della trafilata che porta l'informazione dal dispositivo Alexa fino alla lambda di supporto, essa è garantita da cifratura utilizzando TLS. Le lambda sono poi programmate per ricevere richieste soltanto dalla Skill collegata ad essa.

7.4 Portabilità/applicabilità generale

Il sistema sviluppato risulta completamente dinamico e teoricamente adatto a qualsiasi frangente in cui si renda necessario fornire un supporto agli input vocali. L'incapsulamento fra la parte applicativa che espone i dati raccolti e quella di raccolta, permette la modifica o la sostituzione di una delle due componenti senza che l'altra ne venga intaccata. Per esempio sarebbe semplice, se in un futuro delle tecnologie di riconoscimento vocale si rivelassero più promettenti di Alexa, andarlo a sostituire con uno dei competitor. Inoltre, la suddivisione orizzontale del sistema in base alle aree funzionali lo rende particolarmente adatto all'estensione, in modo da fornire supporto a nuove tipologie di comandi.

7.5 Manutenibilità ed estendibilità

Al fine di garantire estendibilità e manutenibilità al sistema vocale di ausilio a Trauma Tracker si è deciso di fornire alcuni meccanismi per aggiungere nuovi farmaci, manovre, diagnostiche o infusioni. Come doverosa promessa è bene ricordare che il modello di riconoscimento vocale specifico della skill viene addestrato in maniera automatica dall'ASK a partire dal modello d'interazione definito, quando però si sottomette la Skill creata per la certificazione ed ottenendo quindi la possibilità di distribuirla, non si può più modificare a patto di non volere affrontare nuovamente il processo di certificazione. Questo di per sé limiterebbe molto la possibilità di aggiungere nuovi elementi da riconoscere, fortunatamente però esiste la possibilità di aggiungere dinamicamente valori attraverso alcune direttive specifiche impartite tramite la lambda di supporto alla Skill.

Il Web Service Lambda ogni volta che la skill viene lanciata controlla se

esistono nuovi valori da riconoscere, chiedendoli al Trauma Vocal Service facendo un'apposita richiesta. Il Trauma vocal service controllerà quindi su una apposita tabella di storage quali valori aggiuntivi devono essere riconosciuti, fornendo inoltre delle apposite API per aggiungere questi valori attraverso chiamate REST.

L'unico difetto di questo approccio è che l'accuratezza di riconoscimento di questi nuovi valori è minore rispetto a quella degli elementi presenti nel modello appositamente addestrato prima della certificazione.

8 Validazione del sistema

Una volta terminato il sistema, si è proceduto alla sua validazione con il committente, coincidente con l'utente finale, in base ai requisiti con lui concordati. Il team ha incontrato il trauma leader Emiliano Gamberini per raccogliere eventuali feedback sulla modalità di interazione e sull'usabilità generale del sistema. Dopo aver proceduto alle modifiche ritenutosi necessarie, è stato presentato l'applicativo nella sua versione finale, sulla base del quale il committente ha espresso le seguenti considerazioni:

- il sistema ha un'accuratezza superiore alle aspettative per quanto concerne il riconoscimento dei comandi;
- il protocollo di interazione si è dimostrato sufficientemente intuitivo e calzante con il contesto di utilizzo;
- si è notato come il feedback dato dall'assistente possa essere reso più stringato in modo da rendere più reattivo il sistema;
- per quanto riguarda il caso specifico della diagnostica EGA, funzionalità Trauma Tracker, potrebbe essere migliorata l'interazione adottando una strategia multi-turno;
- sebbene il riconoscimento in ambienti poco rumorosi sia molto efficace, potrebbe non esserlo in ambienti rumorosi quali la trauma room. In questo contesto si potrebbe pensare di adottare un microfono personale per migliorare la qualità del riconoscimento.

In definitiva il trauma leader si è mostrato soddisfatto dei risultati ottenuti, dimostrandosi disponibile ad effettuare ulteriori test in ambiente reale.

Appendices

A Integrazione

In questa sezione verrà brevemente mostrato come poter integrare questo progetto con Trauma Tracker o con qualunque altro sistema. Come già citato in precedenza sono stati predisposti meccanismi standard per la comunicazione, ed in particolare due possibili modalità di interazione.

- La prima possibilità prevede l'utilizzo WebSocket per realizzare una comunicazione reattiva. Un client in qualsiasi momento può decidere di creare una connessione con un servizio, specificando il topic a cui è interessato, e da quel momento riceverà tutti i dati in tempo reale.

Più nel dettaglio viene utilizzata una funzionalità di Vert.x, ovvero SockJS Service Proxy, la quale permette di creare client che possono essere utilizzati in applicazioni esterne (ad esempio Node.js). Questi si basano sul bridge di SockJS, il quale media la comunicazione tra l'event-bus di Vert.x e SockJS. A livello pratico perciò tutto ciò che occorre fare è creare un client dell'event-bus, specificando l'indirizzo del servizio e il topic e si registra un handler per gestire la ricezione di informazioni. Nel nostro caso ogni servizio corrisponde ad una funzionalità, mentre tramite il topic viene specificata la stanza di interesse.

- Nel secondo caso invece la comunicazione avviene mediante una REST API, mediante la quale possono essere ottenuti comandi vocali a piacimento. Ogni micro-servizio infatti permette di effettuare una GET, dove occorre specificare la stanza a cui ci si riferisce, il timestamp a partire dal quale si intende ricevere i comandi e il timestamp di fine. Questo approccio è estremamente generale e inoltre copre anche il caso in cui, per qualsiasi motivo, vengano perduti eventi (ad esempio in assenza di connessione). Tuttavia in questo modo occorre gestire la logica di recupero dei comandi persi a livello del client, utilizzando il timestamp per capire quali siano quelli mancanti. In un'implementazione futura perciò si potrebbe pensare di utilizzare un Message Oriented Middleware in modo tale che queste problematiche vengano gestite in automaticamente dal middleware.

B Demo

B.1 Scenario A

2019-06-02 19:44 PRE-H Ingresso luogo: PRE-H
2019-06-02 19:50 PRE-H unknown: patient-accepted
2019-06-02 19:50 Shock-Room Ingresso luogo: Shock-Room

— Entra Alexa 1 —

2019-05-09 03:19 Shock-Room Inizio Caso
2019-05-09 03:21 Shock-Room Farmaco Cristalloidi dose 500
2019-05-09 03:21 Shock-Room Farmaco Fentanil dose 100
2019-05-09 03:21 Shock-Room Farmaco Fentanil dose 100
2019-05-09 03:21 Shock-Room Farmaco Ketamina dose 20
2019-05-09 03:21 Shock-Room Procedura Profilassi ABT
2019-05-09 03:21 Shock-Room Diagnostica Ecografia
2019-05-09 03:21 Shock-Room Diagnostica RX Torace
2019-05-09 03:21 Shock-Room Diagnostica RX Bacino

— Esce Alexa 1 —

2019-05-09 03:23 Shock-Room Uscita luogo: Shock-Room
2019-05-09 03:23 TAC-PS Ingresso luogo: TAC-PS

— Entra Alexa 2 —

2019-05-09 03:29 TAC-PS EGA lattati
Valore lattati?
2.68
2019-05-09 03:29 TAC-PS EGA be
Valore be?
-3
2019-05-09 03:29 TAC-PS EGA ph
Valore ph?
7.41
2019-05-09 03:29 TAC-PS EGA emoglobina
Valore emoglobina?

13.6

2019-05-09 03:29 TAC-PS Diagnostica TC Cerebrale-Cervicale
2019-05-09 03:35 TAC-PS Diagnostica TC Toraco-Addominale
2019-05-09 03:37 TAC-PS Fine Caso

— Esce Alexa 2 —

2019-05-09 04:10 TAC-PS Uscita luogo: TAC-PS

B.2 Scenario B

2019-05-11 19:46 PRE-H Ingresso luogo: PRE-H
2019-05-11 19:51 PRE-H Cambio trauma leader: Giuseppe Sabia
2019-05-11 20:08 PRE-H unknown: patient-accepted
2019-05-11 20:08 Shock-Room Ingresso luogo: Shock-Room

— Entra Alexa 1 —

2019-05-11 20:10 Shock-Room Inizio Caso
2019-05-11 20:14 Shock-Room Farmaco Curaro 12
2019-05-11 20:16 Shock-Room Diagnostica Ecografia
2019-05-11 20:16 Shock-Room Manovra Catetere Arterioso
2019-05-11 20:16 Shock-Room Manovra Foley Vescicale
2019-05-11 20:16 Shock-Room Infusione Adrenalina dose 50
2019-05-11 20:16 Shock-Room Manovra Sonda Gastrica
2019-05-11 20:20 Shock-Room EGA lattati
Valore lattati?
4.24
2019-05-11 20:20 Shock-Room EGA be
Valore be?
-4.7
2019-05-11 20:20 Shock-Room EGA ph
Valore ph?
7.471
2019-05-11 20:20 Shock-Room EGA emoglobina
Valore emoglobina?

10.9

— Esce Alexa 1 —

2019-05-11 20:31 Shock-Room Uscita luogo: Shock-Room

2019-05-11 20:31 TAC-PS Ingresso luogo: TAC-PS

— Entra Alexa 2 —

2019-05-11 20:31 TAC-PS Diagnostica TC Cerebrale-Cervicale

2019-05-11 20:45 TAC-PS Diagnostica TC Toraco-Addominale

2019-05-11 21:03 TAC-PS Farmaco Cristalloidi 500

2019-05-11 21:03 TAC-PS Manovra Reboa

2019-05-11 21:03 TAC-PS Farmaco Cristalloidi 500

2019-05-11 21:03 TAC-PS Farmaco Soluzione Ipertonica 250

2019-05-11 21:05 TAC-PS Fine Caso

— Esce Alexa 2 —

2019-05-11 21:09 TAC-PS Uscita luogo: TAC-PS